RENDERING WATER AND LAND INTERACTION USING A SPRING SYSTEM

Yifan Sui
Andrew Davison
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University
Hat Yai, Songkla, Thailand
E-mail: suisuige@yahoo.com.cn
E-mail: ad@fivedots.coe.psu.ac.th

KEYWORDS

Spring system, wave curve, collision detection, mesh, height value.

ABSTRACT

This paper describes a spring-based model for the interaction of water and land, which reconciles realism and fast rendering. The system controls the motion and interdependences of water vertices utilizing two kinds of springs and collision detection. As a consequence, a wave's movement affects the waves around it, and a wave 'hits' the land, rebounding with a suitably changed height and velocity.

1. INTRODUCTION

The rendering of large areas of water is well understood (Tessendorf 1999; Johanson 2004), and has become common in games. However, there is little physics-based interaction with the shoreline as waves move up and down, and generate spray and foam.

Most 3D systems employ Perlin noise functions (Johanson 2004), although some ocean effects (such as refraction and obstacle collision) have been utilized (Iglesias 2004). For instance, Peachey (Peachey 1986) and Fournier and Reeves (Fournier and Reeves 1986) model waves that approach and break on a sloping beach. However, there is no real force connection between the water and the land, since the wave profile is changed according to wave steepness and water depth.

Foster and Fedkiw (Foster and Fedkiw 2001), and Enright, Marschner and Fedkiw (Enright et al. 2002) simulate breaking waves using a combination of textures and particles. The computational cost of the former is several minutes per frame on a PentiumII 500MHz.

Mass-spring systems are arguably the simplest and most intuitive of all deformable models for simulating fluid (Nealen et al. 2006). Using a spring system to simulate the motion of water over a coastline is compulationally feasible, as this paper illustrates.

2. WAVE CURVES AND THE COASTLINE

The land is a 128*128 textured mesh contoured with a height map. The waves in the water mesh are modeled using Peachey's method (Peachey 1986), so the height of the water vertices varies as a sum of water level and wave height. The color of each vertex is based on its current height. Our prototype was created using JOGL (a Java binding for OpenGL (JOGL, 2009)). Figure 1 shows a screenshot of the model with its elements.

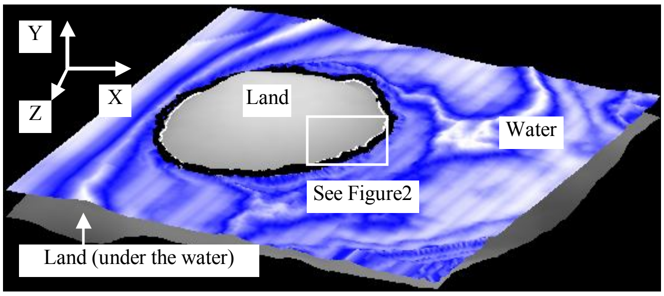

Figure 1: An overview of the model.

The coastline is the series of water vertices that are closest to the land, (see Figure 2). Wave curve 1 is the line of vertices one mesh interval away from the coastline, wave curve 2 is the line of vertices one mesh interval away from wave curve 1. In this way, we define a coastline and four wave curves, which are linked with springs as explained in section 3.

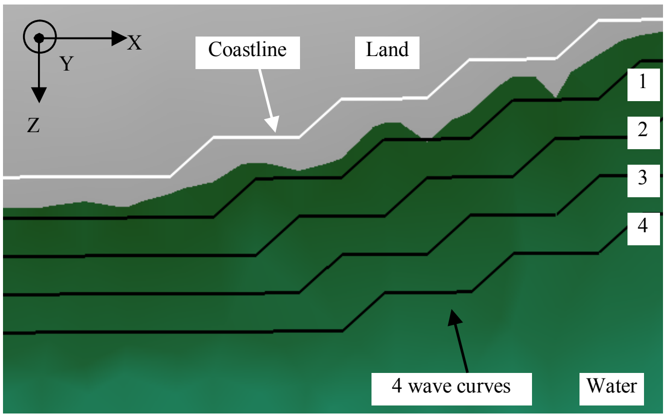

Figure 2: A view of the model from overhead, showing the coastline and four wave curves.

2.1 Vertical Water Mesh Movement

In shallow water, the water vertices move up and down by employing a summed combination of four versions of the height function (1):

$$Y_i = 8*(\frac{\sqrt{x_i^2 + z_i^2}}{L_i} - \frac{t}{T} - \frac{1}{2})^2 - 1 \qquad (1)$$

where i is the index of a vertex, Y is the wave height of the vertex, x and z are the (X, Z) coordinate values of the vertex, t is the time which increases by 0.1 in each frame. T is the period of the function, equal to 80 frames to look realistic. L is the wavelength at the vertex position.

When a wave enters the shallows, where the depth is less than one-twentieth of the wavelength, the wave length L is determined by Equation (2):

$$L = T\sqrt{gd} \qquad (2)$$

where d is the depth of water, and g the gravity (Alonso and Finn 1992; Sverdrup 2006).

Figure 3 illustrates how these equations affect the height of the water mesh as it approaches the shallows.

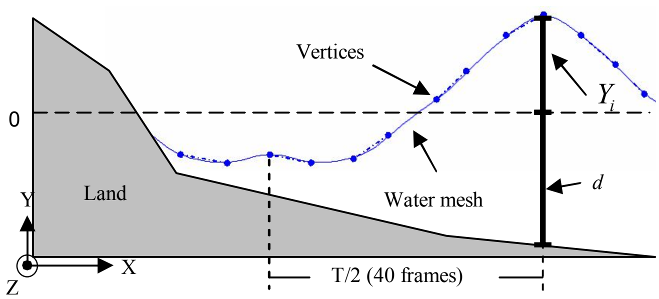

Figure 3: The height of water vertices in shallow water.

2.2 The Coastline

The coastline is the line of water mesh vertices closest to the land mesh, as shown in Figure 2 and Figure 4.

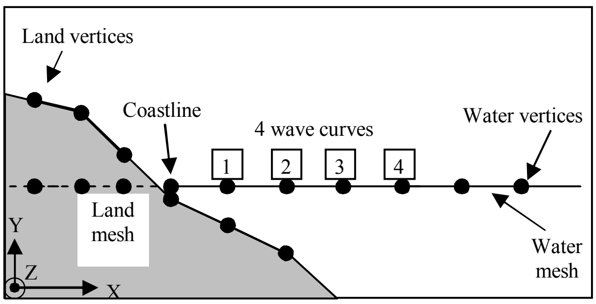

Figure 4: The position of the coastline.

The coastline boundary moves up and down due to waves, but doesn't shift in the X-Z plane.

2.3 Wave Curves

Wave curve 1 is the line of water mesh vertices adjacent to the coastline, but one mesh interval away from the land. Wave curve 2 is the line of water mesh vertices adjacent to wave curve 1, but one mesh interval further away. Wave curve 3 and 4 are calculated in a similar way. Our model is limited to four wave curves as a balance between interaction realism and computational efficiency.

Wave curves 3 and 4 can move in the X-Z plane, towards and away from the coastline. Each vertex in the curves has a movement direction pointing from its original position toward the nearest coastline vertex. The vertices of wave curve 3 can move up to the coastline, while the vertices of wave curve 4 can move up to wave curve 1 (see Figure 5). Wave curve 4 can not easily pass through wave curve 3. These restrictions on curve interaction produce realistic wave behavior, and are implemented using our spring system and collision detection, as detailed in sections 3 and 4.

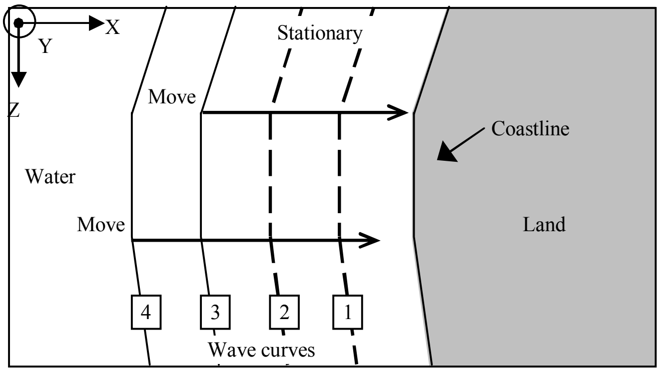

Figure 5: The movement of wave curves.

Wave curves 1 and 2 can not move in the X-Z plane, which means that the ebb and flow of the water against the coastline is driven by wave curves 3 and 4.

3. INTERACTION BETWEEN WAVE CURVES

The interaction between the water and land use *position springs* and *wave springs* to modify the X- and Z- velocities of the vertices in wave curves 3 and 4.

3.1 Position Springs

A position spring ensures that a vertex is pulled back to its original position after moving towards the land. Every vertex in wave curve 3 and 4 has its own position spring.

Figure 6 shows a vertex N_s. At time 0, it is at its rest position, labeled as $N_{s,0}$. At time t, it has moved to be at position $N_{s,t}$. The position spring P extends from the $N_{s,0}$ rest position and will pull N_s back from its $N_{s,t}$ position.

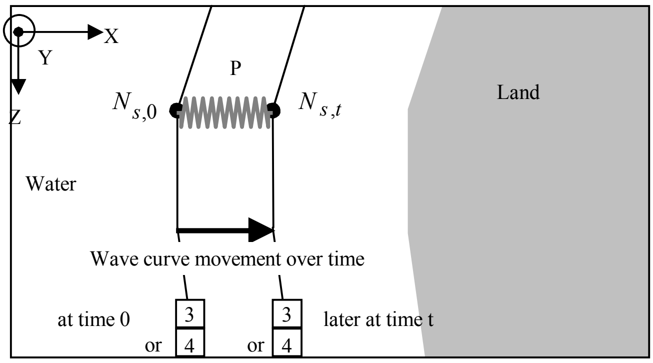

Figure 6: A position spring P for vertex N_s .

3.2 Wave Springs

Every neighboring pair of vertices in wave curves 3 and 4 are linked by a wave spring. For example, Figure 7 shows a wave spring W linking the vertices N_s and N_r of wave curves 3 and 4.

Wave springs help to deal with crossover behavior when vertices in wave curve 4 are moving faster than those in wave curve 3, and attempt to pass through it. Wave springs slow down wave curve 4 vertices as they approach wave curve 3.

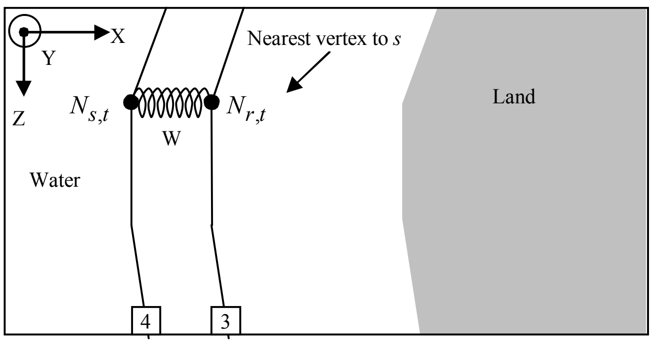

Figure 7: A Wave spring W between vertices N_s and N_r in wave curves 3 and 4.

4. COLLISION DETECTION

Our model deals with two kinds of collision:

1) between the water and the land, as represented by wave curve 3 and the coastline;

2) between waves, as represented by wave curves 3 and 4.

Our approach builds upon real time collision detection (Ericson 2005) by applying it in the context of our spring system.

4.1 Water and Land Collision

The collision detection algorithm is simplified by utilizing the coastline to represents the land, and wave curve 3 as the leading edge of the water.

Each coastline vertex is surrounded by a bounding sphere, whose diameter is equal to the initial inter-mesh spacing.

If a wave curve 3 vertex moves inside the bounding sphere of a coastline vertex, a collision has happened. The velocity of the wave curve 3 vertex is reversed, to make it head back towards its rest position.

Figure 8 shows a vertex N_s in wave curve 3. At time 0, it is at position $N_{s,0}$, then moves towards the coastline and 'hits' the coastline vertex C_p at time t. The velocity of N_s , $V_{s,t}$, is reversed to be $-V_{s,t+1}$ at the next time interval t+1. A scaling factor also reduces the velocity, to take account of the way a wave loses energy when rebounding.

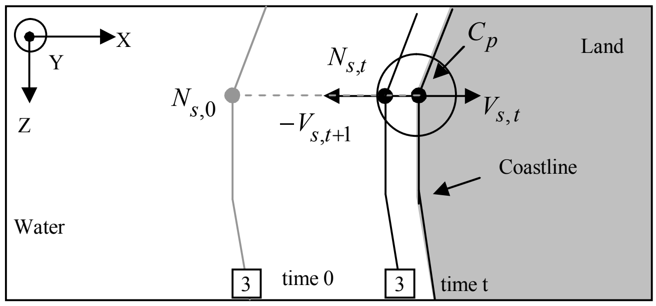

Figure 8: Water and land collision.

4.2 Wave Collision

As explained in section 3, wave springs implement crossover slowdown, but if the velocity of wave curve 4 is much higher than wave curve 3 then crossover could still occur. This is prevented by collision detection between the vertices of wave curves 3 and 4. When a vertex in wave curve 4 hits wave curve 3, their velocities are equalized, so the two wave curve segments will move together, or perhaps separate. This is implemented by updating the velocity of the vertex in wave curve 4 and its nearest neighbor in wave curve 3.

Figure 9 shows the case when vertex N_s is about to hit the wave curve segment V1-V2. A collision is detected between N_s and the segment, and the velocities of N_s and V2 are modified.

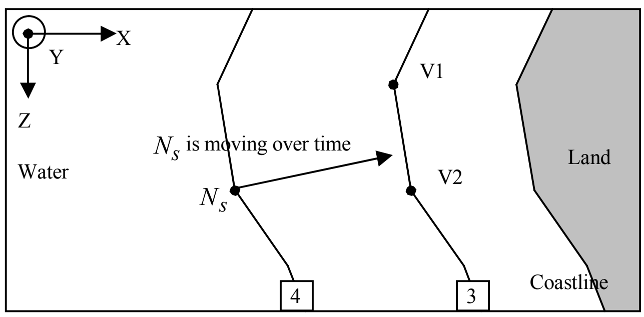

Figure 9: Wave Collision.

The overall behavior of N_s will be more complicated than this (and more realistic) by also being affected by a wave spring linking it to V2 (its nearest neighbor in wave curve 3), which is not shown in Figure 9.

5. TESTING

On a single core 1.73 GHz 2GB DDRII-533 RAM Intel GMA 950, the model executes at about 70 FPS; on a two-core 1.86 GHz 1GB DDRII-533 RAM X300 graphics card, about 140 FPS are achieved, and our OS both are Windows XP SP3 Professional. When we extend the mesh size to 256*256, the model executes on two machines at about 54 FPS and 26 FPS.

Figure 10 is a cross-sectional view of the model showing water moving towards the land.

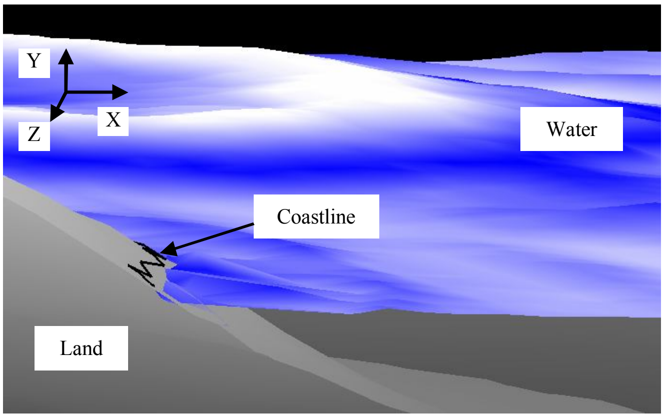

Figure 10: Water moving towards the land.

Figure 11 shows the land, coastline, and wave curves of Figure 10 from overhead.

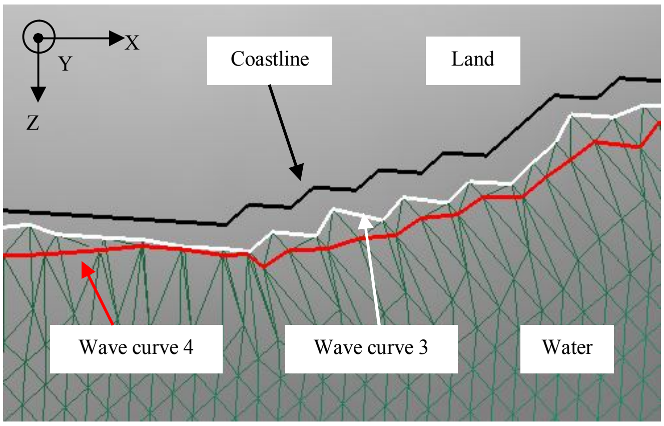

Figure 11: Wave curves moving towards the land.

Figure 12 shows the scene later after the water has rebounded from the land.

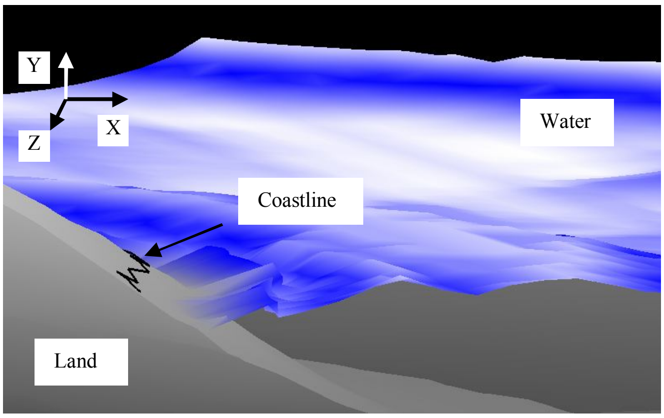

Figure 12: Water retreating from the land.

Figure 13 is a view of Figure 12 from overhead. It shows that the position springs in wave curves 3 and 4 are drawing their vertices back to their rest positions. The interaction between wave curves 3 and 4, as controlled by wave springs and collision detection, is also visible.

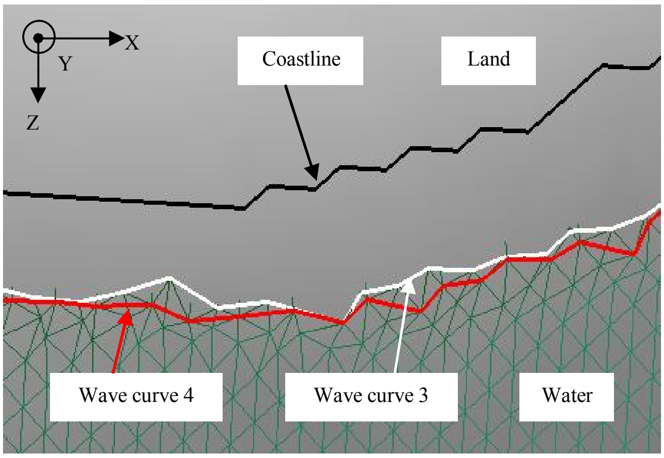

Figure 13: Wave curves rebounding from the land.

Figure 13 illustrates that crossover still occurs, as it does in real waves, but is a rare event, and is soon followed by the waves either moving in unison or pulling apart.

6. CONCLUSIONS

Our system models the interaction of water and land using a novel combination of two types of springs (position and wave springs) and two forms of collision detection. The simulation exhibits realistic behavior between waves and the coastline, and between the waves themselves, while rendering at very acceptable speeds. The spring system is relatively simple to understand and fine-tune, and is based on the physical characteristics of real waves.

We plan to improve the visualization by adding particle-based foam and spray. It will appear on wave crests, the coastline, and wherever collisions occur.

When the water recedes from the land, the exposed areas should look wet. We intend to color these areas accordingly, and let the color gradually fade over time.

Our long term goal is to use this approach to model Tsunami-land interaction. The spring system will need to be modified to deal with large waves (over 30m in height) moving at very high speeds (more than 800 km/h) (Kaitoku 2008). The coastline interaction will need to be more complicated to deal with the way a tsunami can wash over a large body of land.

REFERENCES

A. Iglesias. 2004. "Computer Graphics for Water Modeling and Rendering: A Survey". *Future Generation Computer Systems*, Volume 20, Issue 8, (Nov), 1355-1374. http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6V06-4CVX0RT-2-3&_cdi=5638&_user=267327&_orig=se

arch&_coverDate=11%2F01%2F2004&_sk=999799991&view=c&wchp=dGLbVzz-zSkzV&md5=c4eea2bb7547c6e532e217a9ec196151&ie=/sdarticle.pdf (last accessed Oct 30, 2009)

Alain Fournier and William T. Reeves. 1986. "A Simple Model of Ocean Waves". *ACM SIGGRAPH Computer Graphics*, Volume 20, Issue 4, (Aug), 75-84. http://portal.acm.org/citation.cfm?id=15894 (last accessed Oct 30, 2009)

Andrew Nealen; Matthias Müller; Richard Keiser; Eddy Boxerman and Mark Carlson. 2006. "Physically Based Deformable Models in Computer Graphics". *Computer Graphics Forum*, Volume 25, Number 4, (Dec), 809-836. http://www.matthiasmueller.info/publications/egstar2005.pdf (last accessed Oct 30, 2009)

Christer Ericson. 2005. Real-Time Collision Detection. Morgan Kaufmann.

Claes Johanson. 2004. "Real-time Water Rendering- Introducing the Projected Grid Concept". Master's thesis. Department of Computer Science, Lund University, (Mar). http://fileadmin.cs.lth.se/graphics/theses/projects/projgrid/ (last accessed Oct 30, 2009)

Darwyn R. Peachey. 1986. "Modeling Waves and Surf". *ACM SIGGRAPH Computer Graphics*, Volume 20, Issue 4, (Aug), 65-74. http://portal.acm.org/citation.cfm?id=15893 (last accessed Oct 30, 2009)

Douglas Enright; Stephem Marschner and Ronald Fedkiw. 2002. "Animation and Rendering of Complex Water Surfaces". *ACM Transaction on Graphics*, Volume 21, Issue 3, (Jul), 736-744. http://physbam.stanford.edu/~fedkiw/papers/stanford2002-03.pdf (last accessed Oct 30, 2009)

Jerry Tessendorf. 1999. "Simulating Ocean Water". *SIGGRAPH Course Notes*. http://graphics.ucsd.edu/courses/rendering/2005/jdewall/tessendorf.pdf (last accessed Oct 30, 2009)

JOGL. 2009. http://kenai.com/projects/jogl/pages/Home (last accessed Oct 30, 2009)

Keith A. Sverdrup; Alison B. Duxbury and Alyn C. Duxbury. 2006. "Waves and Tides". In *Fundamentals of Oceanography*, 5th edition, McGraw-Hill, 180-192

Marcelo Alonso and Edward J. Finn. 1992. "Wave Motion". In *Physics*, Addison-Wesley, 747-766

Nick Foster and Ronald Fedkiw. 2001. "Practical Animation of Liquids". *SIGGRAPH 2001*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 23-30. http://physbam.stanford.edu/~fedkiw/papers/stanford2001-02.pdf (last accessed Oct 30, 2009)

Tammy Kaitoku. 2008. *Tsunami, the Great Waves*. 5th edition, International Tsunami Information Center. http://ioc3.unesco.org/itic/contents.php?id=169 (last accessed Oct 30, 2009)