ROSE HULMAN INSTITUTE OF TECHNOLOGY

Splashes and Water Wave Packets

by

Xingfang Yuan

A thesis submitted in partial fulfillment for the
degree of Computer Science

in the
Micah Taylor
Computer Science and Software Engineering

May 2018

Chapter 1

Introduction

The simulation of fluid is essential for many modern applications such as aerodynamics. These simulations are grounded by Navier-Stokes equations for great precisions but very time consuming. Yet in the film industry and video game industry, there are many use cases that the time to compute the simulation is more important than the precision and the motion of the fluid that is underneath the water. Thus, how to produce a realistic and detailed water surface in a very short time frame is widely researched. A traditional way to attack the problem is to model the water surface by two-phase incompressible Navier-Stokes equations. Due to the fact that this method is still too computationally expensive for real time simulation, many researchers introduced various ways to simplify the computation. They make assumptions like shallow water [1], infinite depth [2], omitting solid boundaries [3] to handle general scenarios in real time. Yet there are still problems with stability, energy conservation and artistic control.

Jeschke and Wojtan's [4] brought up a model to approximate waves by packing an entire spectrum of wavelengths and wave trains into a particle, which they called it 'wave packet'. With the calculation of the movement and deformation of the wave packets, they can simulate a very detailed water surface with interaction with boundaries and objects in real time with good energy conservation. Yet detailed fluid effects like splashes are not covered in their work. O'Brien and Hodgins [5] introduced a model that supports splashes by simulating the spray separately from the main volume. When the upward velocity of a portion of the surface exceeds a threshold, a volume of water is disconnected from the main volume. Droplets are then created from the disconnected volume to simulate splashes. Based on these models, we'd like to figure out whether: **The wave packet model can be extended to support the simulation of splashes by combining O'Brien's model that supports dynamic simulation of splashing fluids.**

Chapter 2

Previous Work

The simulation of water surface has been a topic discussed for long. Since it takes too much computation to accurately simulate water particles based on Navier-Stokes equation in real time, many assumptions are made to simulate approximate result within a fair amount of time [2] [3]. Kass and Miller [1] modeled the surface with a height field and tracked the flow between adjacent columns of fluid. The computational cost is proportional to the size of the height field, which is a square rather than a cube. O'Brien and Hodgins further extended the model to support dynamic detailed effects upon interaction with rigid bodies by having a particle system that comes from the excess volumn of the height field.

Besides Eulerian simulation, which focuses on specific locations in the space through which the fluid flows as time passes, Lagrangian simulation, where the observer follows an individual fluid parcel as it moves through space and time, has gained atention in the recent the decade. Yuksel's [6] 'Wave Particles' represents each wave crest as particles. These wave particles are straightforward to implement, parallelize, and control to provide wave reflections and interactions with dynamic objects. However, the method has several limitations: long wave trains and high-frequency waves can enlarged the number of wave particles; wave dispersion is also hard to simulate since a dimension, wavelength, other than crest needs to be included. Jeschke's [4] 'Wave Packets' extended the method. By representing a group of sinusoidal waves with equal or similiar wavelengths traveling in the same direction as a particle, Jeschke et al. supported some of the limitations in Yuksel's work but still no implementation for dynamic detailed fluid effect like splashes.

Some efforts are also made towards art industry to help artists work better on scenes of ocean. Many reseachers made efforts to match input simulations/animations in the area of fluid control [7] [8]. Others focused on the forward Fourier transform for estimating and synthesizing oceanwaves from an input sequence of some form. Frechot [9] sampled ocean spectrums based on a quadtree representation of wave vectors and amplitudes.

Chapter 3

Splashes and Water wave packets

To extend the wave packet model to support the simulation of splashes, we combine it with O'Brien simulation model, which provides the method to simulate splashes. However, the volume, one of the essential part of O'Brien model, is not a defined attribute in wave packet model. One of the focuses of our method is to overcome this problem. In this chapter, we will discuss related contents from both models along with our solution, which is divided into the detection of splashes, the generation of the splashes particles and the simulation and recycle of the particles.

3.1 Water wave packets [4]

Water wave packet simulates real time scenes for water waves. With the assumption that the amplitude of the wave is relatively small, the shape of water waves can be approximated using sinusoidal waves. Each wave packet represents a series of sinusoidal waves that have the same location, similar wavelength and similar traveling speed and direction. The surface of water can be approximated by a sum of wave packets, as given by the expression

$$\eta(x,t) \approx \Sigma_{j=1}^{N} a_j \phi(x - c_g t) cos(k_j(x - c_p t)) \tag{3.1}$$

where x is the location vector of the packet, t for time, j for packet j, a_j for the amplitude, $\phi(k)$ for the kernel function, which is Gaussian function, k_j for the representative wavelength of packet j, c_g for the group speed and c_p for the phase speed.

The energy of each packet is given by the expression

$$E(k) = \int_A \frac{1}{2}(\rho g + \sigma k^2)(a(x))^2 dA \tag{3.2}$$

and it corresponds to the amplitude and the wavenumber. The energy of the model is conservative. When the wavenumber increases as the packet stretching out, the amplitude of the packet needs to be decreased.

The derivation of the equations is in Jeschke's paper and we will not discuss it here.

3.2 Dynamic simulation of Splashing Fluids [5]

O'Brien simulation model consists of three components: the main volume, the free surface of the fluid and the disconnected component of the fluid. In the main volume, the water body is divided into a grid of connected vertical columns. Fluid motion is simulated as the interchange of volume between adjacent columns. The surface component is responsible for interaction with external objects. As we have not explored the potential of rigid body interaction for our solution, we will not discuss the surface component here.

Splash particles are generated in the disconnected component. Particles are created when the vertical velocity of the surface is greater than a certain threshold. The vertical velocity of the surface is given by the rate of change in the volume of the column. To generate splashes, particles are uniformly distributed above the area of the surface. The initial velocity follows the surface velocity. After the creation of the particles, gravity is applied to the particles and particles do not interact with each other. The volume of each particle is subtracted from the column from which it was created to keep the total volume in the system constant. After particles fall back onto the surface, they are removed and the volume of each particle is added to the column that it falls into.

3.3 Splash detection

We first need to find the locations to generate splashes. O'Brien model approaches this by looking at the rate of change of the volume of each columns while the wave packet model does not have the main volume component. We determine that the final shape of the water from the rendering step of the wave packet model can serve the same purpose as the main volume component in O'Brien model. Thus, we create a heightmap, which is a two-dimensional array of the height of the water body at each location as the representation of the final shape of the water. Equation (3.1) is used to calculate the values in the heightmap.

The vertical speed of the surface, which is the rate of change in the volume, is then given by the rate of change in the heightmap, as shown in the expression

$$V_{i,j} = H_{i,j}(t + \delta t) - H_{i,j}(t) \tag{3.3}$$

where $H_{i,j}(t)$ represents the height of the water at location (i, j) at time t, $V_{i,j}$ represents the vertical speed at location (i, j). When $V_{i,j}$ is larger than a threshold, we will generate splash particles at location (i, j)

3.4 Particle Generation

When generating the particles, a certain amount of volume that is carried by the particles needs to be taken out from the columns in O'Brien model. Thus, we need to change the wave packets so that the corresponding heightmap also decreases the height of the water at the locations that will generate splashes. In order to find all the packets that need to be changed, a brute-force method can be used: For each location (i, j), we examine all the packets and determine that if the area of any of them spans over (i, j). The time complexity is $O(mn)$, where m is the number of locations and n is the number of packets. Since we are aiming for real time simulation and the number of packets n can be relatively large, optimization for time efficiency is needed. We use an R-tree for this purpose. We choose R-tree because it is one of the easiest way for us to integrate into our program as the implementation is provided by Boost. The decision of what spacial tree to use can be taken into further consideration though. The bounding box of each packet is derived from the boundary of each packet. The four corners of the boundary is given by

$$\begin{cases} C - sD - sT \\ C - sD + sT \\ C + sD - sT \\ C + sD + sT \end{cases} \tag{3.4}$$

where C is the center of the packet, s is the size of the packet, T is the traveling direction of the packet and D is perpendicular to T. The lower-left corner of the bounding box has the minimal value of x and y from the corners of the boundary while the upper-right corner has the maximal value of x and y from the corners of the boundary. Then the R-tree is built with all the bounding boxes. After the tree is done, we query the R-tree to determine what packets have their area spans over each location (i, j) where generates splashes. Each query takes $O(logn)$, and overall time complexity becomes $O(mlogn)$, which is acceptable for real time simulation.

To change the height of the water at the locations where generate splashes, we take a portion of energy out of each packet. As shown in (3.2), the energy is quadratic in amplitude. When the energy decreases, so does the amplitude of the wave. It results in the decrease of the height of the water. The regions near the locations where generate splashes are also effected, but we consider it is acceptable since this is an approximation of wave simulation.

Particles are then distributed uniformly above the area of the locations that we detected. For each location (i, j), the vertical speed of each particle is given by the vertical speed at (i, j) and the horizontal speed is given by the average speed of the packets that have their area over (i, j).

3.5 Particle simulation and recycle

While the splash particles are floating in the air, they are affected by gravity but are not affected by each other, as in O'Brien model. When each particle falls back onto the water surface, the energy it carries is injected into the water and along with that it creates a circular wave, which consists of a circle of wave packets at its location to preserve the energy in the whole system.

Chapter 4

Results and Future Work

4.1 Results

We implement our method all on the CPU. We parallel the simulation for splash particles and multi-threaded the generation of the heightmap. The particles are rendered as small dots. We test the implementation with a large energy injection, creating a circular wave front as well as splashes. The results are shown in the images 4.1.

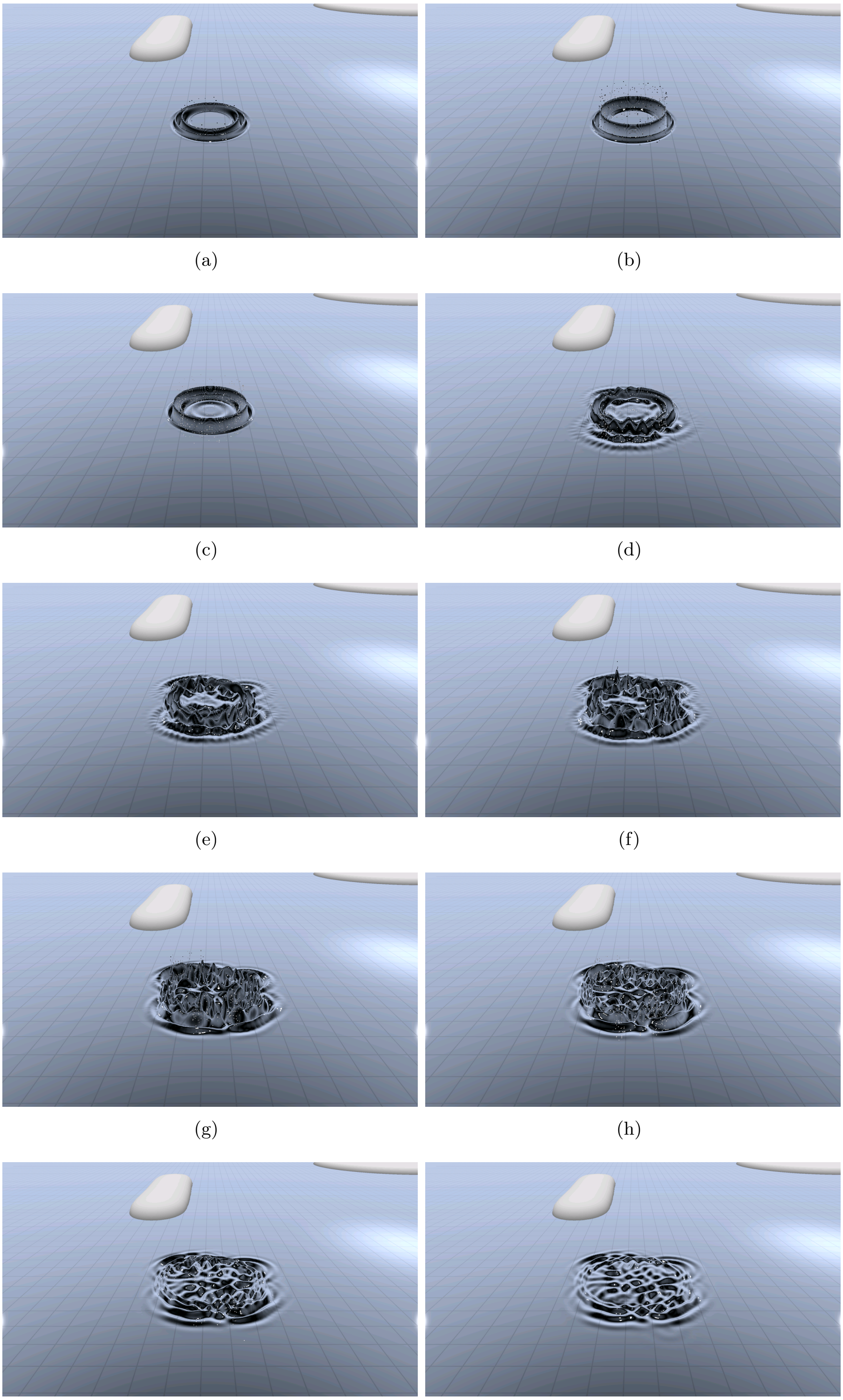
We implemented our solution on Dell Precision 5510, which is equipped with a NVIDIA Quadro M1000M with 2GB GDDR5 dedicated memory and a Intel i7-6820HQ Quad Core. We compared the performance of our solution to the simulation without splashes as a baseline.

	Base		Splash	
Packet	FPS	pkt/ms	FPS	pkt/ms
1k	21.55	4548	21.97	26
50k	10.97	4178	1.18	61
100k	6.70	2678	.62	64
150k	1.43	1214	.38	54

4.2 Future Work

Currently our implementation is all on the CPU side. Though we have some optimization like multi-threading, the most efficient way to run the simulation is to run it on the Graphics processing Unit(GPU). We can expect a great performance improvement once we utilize the parallel computation resource offered by GPU.

As we mentioned earlier, the choice of spacial tree needs further consideration. We may want to investigate with different trees to see which one performs the best in our usage. Additionally, since currently we are rebuilding the tree every frame, it is helpful that the tree can support the upgrade operation so we do not have to rebuild the tree.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

FIGURE 4.1: Results

We would also like to test our method with rigid bodies interactions to investigate the possibility of our model. We may can make use of O'Brien model's surface model to achieve this.

Bibliography

[1] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. *SIGGRAPH Comput. Graph.*, 24(4):49–57, September 1990. ISSN 0097-8930. doi: 10.1145/97880.97884. URL http://doi.acm.org/10.1145/97880.97884.

[2] Gary A. Mastin, Peter A. Watterberg, and John F. Mareda. Fourier synthesis of ocean scenes. *IEEE Comput. Graph. Appl.*, 7(3):16–23, March 1987. ISSN 0272-1716. doi: 10.1109/MCG.1987.276961. URL http://dx.doi.org/10.1109/MCG.1987.276961.

[3] Alain Fournier and William T. Reeves. A simple model of ocean waves. *SIGGRAPH Comput. Graph.*, 20(4):75–84, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15894. URL http://doi.acm.org/10.1145/15886.15894.

[4] Stefan Jeschke and Chris Wojtan. Water wave packets. *ACM Trans. Graph.*, 36(4):103:1–103:12, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073678. URL http://doi.acm.org/10.1145/3072959.3073678.

[5] J. F. O'Brien and J. K. Hodgins. Dynamic simulation of splashing fluids. In *Proceedings of the Computer Animation*, CA '95, pages 198–, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7062-2. URL http://dl.acm.org/citation.cfm?id=791214.791474.

[6] Cem Yuksel. Wave particles. In *ACM SIGGRAPH 2007 Computer Animation Festival*, SIGGRAPH '07, pages 148–, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1829-7. doi: 10.1145/1281740.1281871. URL http://doi.acm.org/10.1145/1281740.1281871.

[7] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint method. *ACM Trans. Graph.*, 23(3):449–456, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015744. URL http://doi.acm.org/10.1145/1015706.1015744.

[8] Michael B. Nielsen and Robert Bridson. Guide shapes for high resolution naturalistic liquid simulation. *ACM Trans. Graph.*, 30(4):83:1–83:8, July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964978. URL http://doi.acm.org/10.1145/2010324.1964978.

[9] Jocelyn Frechot. Realistic simulation of ocean surface using wave spectra. 01 2006.