Computers & Graphics ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section

Simulation and control of breaking waves using an external force model

Mathias Brousset [a], Emmanuelle Darles [a], Daniel Meneveaux [a], Pierre Poulin [b], Benoît Crespin [c]

[a] XLIM, UMR CNRS 7252, Université de Poitiers, France
[b] Dept. I.R.O., Université de Montréal, Canada
[c] XLIM, UMR CNRS 7252, Université de Limoges, France

ARTICLE INFO

Article history:
Received 18 December 2015
Received in revised form
6 March 2016
Accepted 7 March 2016

Keywords:
Physics-based animation
Fluids
SPH
User control

ABSTRACT

This paper presents an extension of a paper initially presented at the conference VRIPHYS 2015 [1]. It describes a new method for intuitively managing swells and breaking waves within fluid solvers, based on an external force controlled with only a few parameters. Generating waves with conventional approaches requires pushing particles with oscillating planes. As the resulting waves are only handled by the fluid simulator, they cannot be controlled easily; breaking waves are also difficult to produce in practice. Instead, we propose to use a new wave model that physically describes the behavior of "wave forces" with parameters that explicitly affect wave speed, height, and width. We also propose to map each parameter with user-defined curves. As shown in the results, these forces applied to a smoothed-particle hydrodynamics (SPH) system provide a wide range of effects, such as swells with varying speed and height, breaking waves, curved wave fronts, overtaking waves, and curved wave paths. We show how it is possible to handle crossing waves with our model. Each effect demonstrated in our results only requires a few easy-to-implement operations.

1. Introduction

Over the past decade, fluid simulation has spawned major interest in the computer graphics community. Motions of many fluid types (e.g., water, fire, smoke) have been simulated successfully with physically based methods, mostly derived from the Navier-Stokes equations. Whether based on Eulerian or Lagrangian solutions, these methods can deliver accuracy and realism, but they often suffer from long computation times, large memory requirements, and very little control during computations.

Waves over a large body of water capture an essential part of our vision of any coastline, but initiating conditions in existing fluid solvers in order to generate the expected waves is a daunting task. However, creating realistic water simulations without these solvers is without doubt as difficult. We seek to provide controls to more easily create different waves in a physically based fluid solver, thus benefiting from its power, yet providing more artistic freedom.

Waves are examples of turbulent fluid phenomena that have often been used as illustrative demonstrations. They are usually produced with an oscillating panel for both real fluid demonstrations and physically based simulation methods [2–4]. In such a configuration, a fine control of waves remains difficult to achieve in practice.

Only a few papers address how to simulate such physically complex natural phenomena, while still offering artificial or artistic control with intuitive user interactions or parameterizations. One rare exception for waves comes from Mihalef et al. [5], who controls waves using a library of 2D breaking waves. In their system, a user selects a set of 2D slices to produce a 3D animation of a wave in an Eulerian solver. Since modeling and controlling the fluid depend on a library of 2D slices and their physical properties, the manipulation cannot be accomplished during the simulation, which can strongly interfere with an artistic expression. Moreover, as the slices store velocity vector fields, the artist also loses much design freedom.

In another work worth mentioning, Radovitzky and Ortiz [6] develop a 2D finite Laitone solution that represents a breaking wave using hyperbolic periodic functions, following the shape of plunging waves. Unfortunately, their model requires several parameters, and controlling the generated waves during simulation is impossible.

From the game industry perspective, we should point out that some solutions offer high-level controls for the shape of a single wave, in the form of Bezier curve attractors. They can produce the general shape of a wave, but do not handle breaking waves, nor do they integrate well in a physical fluid simulator for important secondary fluid effects.

http://dx.doi.org/10.1016/j.cag.2016.03.003
0097-8493/

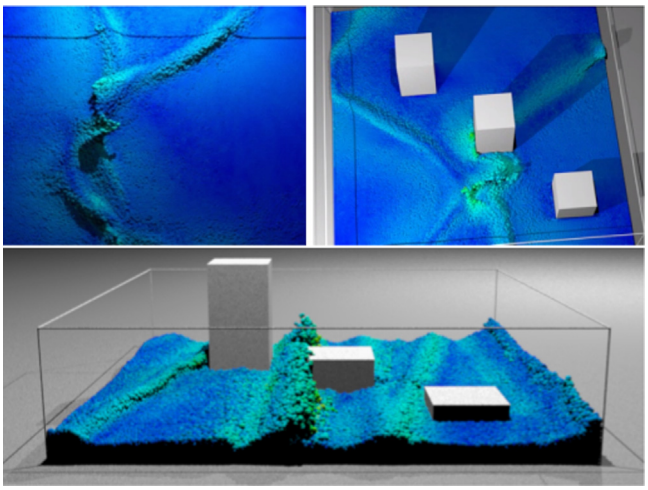

Fig. 1. Examples of user-defined wave fronts and their interactions: (top) two curved wave fronts crossing each other at different moments; (bottom) multiple waves and blocks.

This paper proposes a representation based on an external "wave force" that can be controlled by the user, relying on intuitive parameters such as wave speed, height, width, and orientation. In this extended version of our original paper [1], we develop a 2D control of waves for creating curved wave fronts and wave paths. We have applied our external force model to a smoothed-particle hydrodynamics (SPH) solver. Some results obtained with our approach are shown in Fig. 1. More specifically, our main contributions are:

- a new wave model corresponding to an external force applied to fluid particles, based on physical observations; it enables a user to control choppy, linear, or breaking waves, with ease of implementation;
- a 2D *vertical* control that can be used to interactively modulate the wave profile, speed, height, and width during its propagation;
- a 2D *horizontal* control that extends our paper [1] for controlling wave fronts and wave paths on the water surface while creating crossing and overtaking waves.

The paper is organized as follows. Section 2 reviews existing methods for wave simulation in computer graphics. Section 3 introduces our wave model, together with its parameters to control swells and breaking waves, as well as curves affecting the propagation of wave fronts. Section 4 describes our implementation and presents various results. Finally, Section 5 gives our conclusions and future work.

2. Related work

Over the past two decades, computer generated ocean waves have drawn the attention of several authors [7,8,5,9,10]. They can be classified in two main families: on the one hand, procedural or spectral methods that characterize water height according to time; on the other hand, simulation methods that aim at solving Navier-Stokes (NS) equations in 2D or 3D.

2.1. Procedural and spectral wave simulation

Procedural or spectral methods are often used to model waves for oceans of infinite depth, where the water surface is mostly represented as a heightfield. These methods are common because they are very fast to compute and visually appealing, although they do not correspond to physically based models. Some models [7,11] use parametric equations to derive a surface to approximate swells and breaking waves.

Tessendorf [8] animates a heightfield using fast Fourier transforms (FFTs), to which Bruneton et al. [12] add levels of detail to more realistically render the ocean surface in real time. Because tuning parameters for these methods is usually more difficult, Thon and Ghazanfarpour [13] propose to use real-world measurements as heightfields, and to add Perlin noise to reduce repetitive visual artifacts. The main limitation of these methods is that they cannot represent breaking waves, since a heightfield has only one height value z for each horizontal position (x,y). In addition, these methods are difficult to tune for handling interactions with solid objects.

2.2. Navier–Stokes-based Wave Simulation

A large body of work in computer graphics has focused on computational fluid dynamics (CFD) for liquids, based on Navier-Stokes partial differential equations (NSE). Numerical solutions usually make use of an Eulerian description with finite differences to approximate a solution [14,15], with particles and a level set to track the fluid surface and reduce compressibility [16–19]. A Lagrangian approach considers the fluid as a set of particles and computes interaction between them to approximate a solution [20,2,3,21–24].

From a library of 2D simulated wave slices, Mihalef et al. [5] generate 3D waves. Their method consists in combining a series of 2D slices to model the new wave at given times; velocities between slices are linearly interpolated. In their framework, each slice corresponds to a snapshot of an Eulerian 2D simulation with its associated vector field. Starting from an initial 3D geometry, which can be shaped as a swell or a breaking wave, the targeted wave is eventually obtained during the simulation.

With shallow water equations, based on a simplified version of NSE, Thurey et al. [25] simulate breaking waves at interactive framerates with a 2D plane. The wave height and propagation speed can be controlled, and breaking waves are completed using an additional mesh. However, interactions between water volumes and solid objects cannot be physically handled.

Based on oceanography studies, the velocity field of a 2D NSE simulation can be initialized by a combination of hyperbolic functions that represent a solitary breaking wave [6]. The idea is to combine horizontal and vertical descriptions of a nonlinear wave [26], using hyperbolic secant and hyperbolic tangent functions, i.e., $y = H\mathrm{sech}(x-\omega t)\tanh(x-\omega t)$, where H is the wave height, and ω denotes the pulsation, encoding the propagation speed in the x direction. Unfortunately, wave control remains subtle, and the resulting waves are only valid in shallow water. In addition, the resulting waves only break when the ground rises, which limits their applicability to the relief of an ocean floor. Moreover, these waves propagate only along the 2D plane corresponding to the propagation direction. Finally, because their interaction requires to solve the Korteweg-de Vries (KdV) partial differential equation [27], they remain too costly for interactive applications.

Darles et al. [28] extend this model for 3D breaking waves by adding a procedural force to a multiscale SPH solver. This method produces various types of plunging and surging breaking waves of varying height, but it depends highly on the fluid depth, and breaking is controlled by the ground geometry only.

This paper proposes a new method that reduces many of these limitations, allowing an artist to create several configurations of breaking waves with few intuitive parameters. Our model can be employed to produce propagating and interacting waves in various directions. It can also generate and control swells, independent of the depth. It corresponds to an extension and a simplification of

Please cite this article as: Brousset M, et al. Simulation and control of breaking waves using an external force model. Comput Graph (2016), http://dx.doi.org/10.1016/j.cag.2016.03.003