# A Survey On Modelling & Rendering Fluids

UNIVERSITY OF BATH

DEPARTMENT OF COMPUTER SCIENCE: CS50244
Jacob Haynes (EngD Digital Entertainment)
Unit Leader: Yongliang Yang
18/12/2017

**A vital part of realism in computer graphics and animation is an accurate representation of physical, and often natural, phenomena. One particularly complex example of this is that of fluids. This paper aims to survey the literature on modeling and rendering of fluids for use in computer animation and graphics. The focus will be on the realistic modeling and rendering of fluids but the majority of examples will be of water models and renders.**

## 1 Introduction

Natural phenomena are often the most complex to realistically model and render due to the inherent complexity of such systems as well as their omnipresence in our physical lives, meaning any errors and discrepancies are easily noticeable. Fluids are particularly challenging natural objects in this regard.

Fluids move in complex ways and interact with each other very dynamically. Physically modeling many of these interactions leads to time being included explicitly in equations for motion. In addition to the mathematical complexities involved in fluid dynamics, there is also the issue of rendering these objects, many of which particle based and have complex optical effects such as transparency, reflection, and occlusion effects.

This paper will provide a general overview of the technologies and their developments, primarily focusing on water modeling and rendering. Water is, arguably, the most common fluid that is used in animation due to is omnipresence in everyday life. In addition, many other fluids can be modeled as water with variable physical or optical properties. Many of the papers reviewed will be in relation to water modeling and rendering, however, it is important to note that many of the techniques and technologies used for water can be applied to other fluids. Other fluids will also be directly considered but in a much less focused sense.

It is also important to identify that this paper focuses on technologies for the "realistic" modeling and rendering of fluids. In the case of computer graphics and animation; realistic means nothing more than does it look like the real phenomena [1]. This means no physically "correct" answer is expected from the model beyond that of the visual result. As such, part of this survey will include illumination models which would be irrelevant for scientific simulations but crucial for film and game applications.

The paper will be structured in chronological order, tackling first the early work on fluids through to contemporary cutting-edge topics. Broken down into three sections; early period, mid-period, and contemporary work with a focus on real-time methods. This will then conclude with a summary of topics covered.

## 2 Early Period

The early work in computer modeling and rendering of fluids, in particular water, was focused on bump mapping and ray tracing techniques. These methods focus on representing large masses of water with no physical boundaries, this was due to water being seen as a compact fluid rather than a particle model of individual water droplets.

Early attempts to render waves and water surfaces were based on a bump mapping technique developed by Blinn in 1978 [2]. Blinn's technique used a texturing function which performs a small perturbation to the direction of the surface normal before illuminating the surface. This yields realistic looking surface wrinkles which can be applied to solid bodies with a water texture to imitate waves. Alternative perturbation functions can also be used to a similar effect. Schacter developed a model involving a look-up table of precalculated waveforms [3]. This was a lightweight solution intended for real-time applications.

These very early implementations, although creating effective shapes, lacked important realistic effects such as reflections. Some of the first attempts to combine optical effects with these perturbation modeling techniques focused on the use of ray-tracing [4]. Ray tracing involves generating rays from the camera, finding the nearest intersection of the ray with the scene, and shading pixels appropriately. Ray tracing can be used to animate realistic reflections from ripples in liquids by using a model such as Whitted's [5]:

$$I = I_a + k_d \sum_{j=1}^{j-ls} (\bar{N} \cdot \bar{L}_j) + k_s S + k_t T \tag{1}$$

where $I$ is the reflected intensity, $I_a$ is the reflection due to ambient light, $k_d$ is the diffuse reflection constant, $\bar{N}$ is the surface normal, $\bar{L}_j$ is the vector in the direction of ray $j$ source, $k_s$ is the specular reflection coefficient, $S$ is the intensity of light incident from the angle of reflection, $k_t$ is the transmission coefficient, and $T$ is the intensity of light from the direction of transmission. All of these values can be calculated simply from Snell's law and material properties.

Max used a ray tracing approach to render wave surfaces [6]. The model used for the wave surface was simple and based on calculating wave velocity, $v$, from the wavelength, $\lambda$, as

$$v = \sqrt{\frac{g\lambda}{2\pi}} \tag{2}$$

where $g$ is acceleration due to gravity. Many assumptions on the fluid dynamics were also made, such as constant fluid density and uniform irrotational fluid flow. There were also issues with the waves passing through each other without modification due to linear Stokes approximations being made. Despite the model's limitations, impressive images were produced in Max's "Carla's Island" film as can be seen in figure 1. Although creating visually stunning images, especially when considering the technology at the time, Max identifies an additional problem with the render in that refraction of light on the water was not addressed.

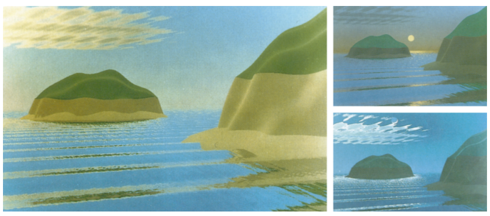

Figure 1: A collection of images from the film "Carla's Island" showing islands, ocean and clouds at various times of day [7].

Most other early attempts at modeling and rendering water use bump mapping or ray tracing to some degree.One example is Perlin's, which combined a bump map with a rich texture map to generate what the author called a "solid texture" [8]. This method drastically reduced the computational requirements for such renders by allowing the texture and shape to be independent of each other. Figure 2 shows an example of an ocean scene generated this way.

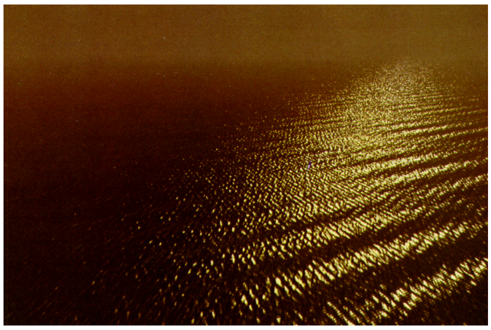

Figure 2: An image of an ocean produced by Perlin's solid texture bump map method [8].

These early methods are far from suitable for realistic rendering and animation. Bump mapping, although computationally inexpensive, the technique leaves the actual surface flat so silhouetting and intersections with other faces do not occur. In addition, any surface effects such as waves do not cast shadows. These flaws with bump mapping mean it is inappropriate for modeling the edges of bodies of water or views at more horizontal angles. Ray tracing also exhibits some significant disadvantages, including aliasing effects from points far from the focal point. Ray tracing is also incredibly computationally demanding with long calculation times making it unsuitable for animation.

# 3 Mid Period

This period saw substantial improvements in rendering via the adoption of particle models in fluid simulations. Major steps were also taken in fluid-solid interaction as well as wave effects, including wave refraction and collisions. These changes allowed interaction of waves with objects and also enabled correct interaction of waves with a change of depth.

Particle systems were first developed by Reeves in reference to natural objects and phenomenon such as fire, smoke, clouds and water [9]. They function by having particle objects be born, change form, move and die over time. Particle systems are ideal for describing natural phenomena which are not well described by polygons or surfaces. Particle systems complex, irregular, and variable shapes allow them to effectivly represent fluid like behaviour of gasses and liquids such as wind [10].

Wave realism was increased by addition of various other observed physical effects to existing models. Peachey [11] added wave refraction to a version of Max's model (see section 2). Peachey also modeled the breaking of waves and the subsequent spray using a particle system. Rendering of this model was carried out via a scanline algorithm, the particle systems were then generated seperately as small spheres of all the same colour with a small random colour component added.

A more physically complete version of an ocean shore model and render was demonstrated by Fournier and Reeves [12], examples of their results can be seen in figure 3. Their model is based on an oceanography model that describes partivles of water acting on elliptical stationary orbits. A more accurate rule for spray generation on breaking waves was also implemented which generates spray when there the difference between particles and the surface speed projected in the direction of the normal to the surface exceeds a threshold. This was used to distinguish between two different particle effects; one for spray (particles in direction of the normal) and foam (particles sliding along the surface of the fluid). The surface itself was modeled as a parametric surface allowing ray tracing [4] and adaptive subdivision [13] to be applied. To handle reflections, an envrionment map was used to map an image of the surrounding environment onto a sphere around the object to be rendered, the colour at a point was then drawn from this sphere via a ray from the object to the camera reflected accross the normal to the sphere.

Figure 3: An scene of waves with particle system formed spray and foam, generated by Fournier and Reeves [12]. Waves can be seen to interact with the depth as well as the shoreline.

## 3.1 The Introduction of Fluid Dynamics

A key physical phenomenon remains unexplored in all models up to this point, that is one of accurate fluid dynamics. Models still lack descriptions of state changes, wetting and drying, the behaviour of small bodies and droplets, and interactions with static or dynamic objects. In terms of rendering many light interactions with fluids had not been tackled, including reflections and other light-surface interactions. Fluid dynamics started to be included in models and two fields of thought emerged; fluid dynamics modeled by large particle systems, and fluid dynamics modeled by partial differential equations.

### 3.1.1 Large Particle Systems

In this method, the attractive and repulsive forces between particles are studied to simulate viscosity and state changes. These models also started to model various fluids such as Miller and Pearce's model which could model various fluids as well as powders and gelatinous solids [14]. This model described an object titled a "globule" which was used a particle system where the particles were connected by forces that vary with distance. This allows globules to flow over each other while they interact via attractive and repulsive forces as well as a drag term. Miller and Pearce also propose an efficient rendering technique for the globules using isosurface approximations on the globules.

Tonnesen [15] took this model one step further by implementing temperatures as a variable, allowing the model to describe changes in fluids geometry, movement, and volumes due to temperature change. This is achieved by simulating particle interactions according to pairwise potential energy functions, similar to that of a Lennard-Jones potential well (see [16] for reference). The Lennard-Jones style potential leads to repulsive forces at small separations and attractive forces and larger separations. The model also includes terms for heat transfer between particles and the surroundings. This model allows melting and freezing as well as changing fluid behavior based on temperature; the colder the temperature the more solid-like the fluid.

A third example of a large particle system scheme was implemented by Sims [17] (see figure 4). The model functioned by applying various accelerations to particles to simulate effects such as gravity, bouncing, and orbiting. Each particle could have variable shapes, sizes, colours, and transparencies as well as effects such as anti-aliasing and motion blur applied. This was achieved by using a parallel processing rendering technique.

These particle models are still fairly nondescript when it comes to particle dynamics and collisions. Sims' model, for example, requires more efficient collision detection to accurately describe the motion for more complex systems. Also, the particles themselves are not deformed in these models and act as fairly static objects apart from their motion.

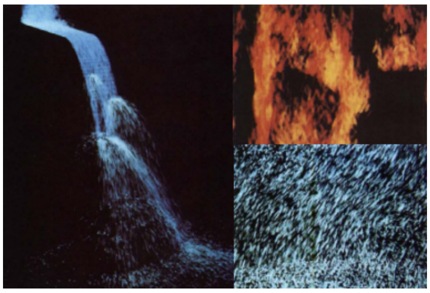

Figure 4: An image of a waterfall, burning letters and a snowstorm simulated via Sims' parallel computation technique [17].

The early versions of particle-based systems assumed the fluid particles moved in elliptical orbits around their initial positions, this is an inadequate assumption for small bodies of water where these particle models no longer realistically reproduce fluid dynamics. Large particle systems require the storage of both the position and velocity of each particle is required. Mallinder suggests a "string texture" method in which particle information is stored implicitly, similarly to solid textures [18]. This is identified as a key memory saving technique for particle systems.

#### 3.1.2 Partial Differential Equations

The more physically accurate approach is to solve partial differential equations that describe the fluid dynamics of the system. For a truly physical simulation, this requires calculation of the motion of the fluid for the entire volume. The calculations required for each iteration or time step is proportional to the volume of the fluid (and its resolution) leading to a very computationally demanding technique. The obvious solution to this issue is to sacrifice physical accuracy while maintaining visual accuracy.

Kass and Miller [19] model water via partial differential equations by simplifying the surface of the water as a hight field and ignoring the vertical component of the water particles velocity. The model implemented can generate the effects of wave refraction with depth and handles reflections and boundary conditions behavior. Due to the limitations of using a hight field modeling breaking waves is beyond the standalone model, however, the authors state that this could be achieved in combination with particle-based fluid models as previously described. To further increase the visual fidelity the work included a wetness map to compute wetting and drying of terrain and was rendered via caustic shading to simulate the refraction of rays at the surface of the fluid.

The Kass and Miller model, as already discussed, has some rather large assumptions in its implementation. As such it ignores the effects of pressure and rotational movements found in fluids, also as no internal pressures are calculated the model cannot handle buoyant objects. An alternative model can be found using Navier-Stokes equations which provide a complete description of fluid motions at any point within the fluid at an instant of time.

An example of which is a simplified two-dimensional case of the Navier-Stokes equations for incompressible flow as in [20]. This is achieved by removing vertical dependence and reducing the problem to a two-dimensional system. However, this leaves the fluid as 2D and as such obstacles must also be modeled as 2D so submerged objects are ignored. Despite these flaws, surface interactions can be effectively modeled as seen in figure 5.

Foster and Metaxas, on the other hand, tackles the 3D Navier-Stokes equations to accurately simulate fluid dynamics [21] [22]. This removes the zero depth assumption allowing for calculations involving convective wave effects, mass transport, and submerged obstacles. Foster provides realism by using finite difference approximations on the Navier-Stokes equations to allow for a complete velocity and pressure profile of the fluid. The model allows for effects such as wave refraction, reflection, and diffraction, as well as rotational effects including eddies and vortices. In the particular model in [21][22]; liquid sources and sinks can be included to allow liquid to flow, in addition, time dependent pressure fields can be applied to the surface to simulate wind effects. An example of the output of this model can be seen in figure 6.

One major drawback to using 3D Navier-Stokes equations is they can be unstable in large time-steps, thus limiting their speed and restricting their use in real-time animation. Stam [23] proposes a more stable model of fluids by, rather than solving explicitly like Forster, solves implicitly. Stam's model handles both motion and propagation of fluids, as seen in figure 7, but does not address the problem with free boundaries meaning the system works better for gasses than liquids.

Although highly accurate in describing fluid systems, 3D Navier-Stokes equations are computationally demanding at higher resolutions so impractical for interactive animation. O'Brien and Hodgins [24] return to some of the 2D assumptions made by Kass and Miller but adapt it to include a wider range of behaviors such as splashes. This is achieved by dividing the system into a three-part system, each subsystem models the main volume, the surface, and the spray. The volume is modeled by vertical tubes which allow flow between columns due to pressures, boundary conditions, and sources/sinks. The subsystem that tackles the surface functions via a mesh of control points, the vertical positions are found by averaging the surrounding column heights. The spray is modeled by a particle system similar to those described by Sims [17]. The model cannot deal with submerged objects but was used to demonstrate effects such as waves, impacts, splashes and buoyant objects.

Mould extended on this by including underwater terrain and removing vertical isotropy [25]. Removal of the vertical isotropy requirement allows interactions between the fluid below the surface and particles. The model does not accurately simulate physical effects such as friction and many parameters exist within the model that holds no direct physical meaning but help to establish the visual accuracy of the model. Wave propagation, wave patterns, depth of water, water droplets, floating and submerged objects, and splashing are all accurately modeled. One shortcoming, however, is that forces from incoming objects only apply in the vertical direction regardless of the angle of incidence.

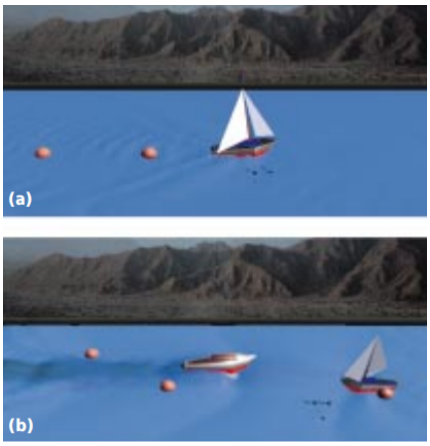


Figure 5: *a* shows ripples formed by a slow-moving boundary condition and *b* shows surface ripples formed by a fast-moving boundary condition calculated using 2D Navier-Stokes methods [20].

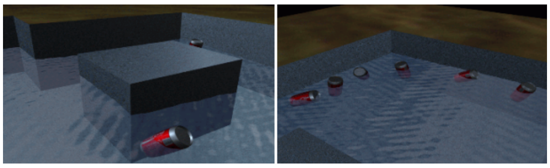

Figure 6: A 3D Navier-Stokes simulation of floating soda cans [21]. This demonstrates dynamic objects and colliding with obstacles. The left-hand image also shows objects getting stuck in local eddies.

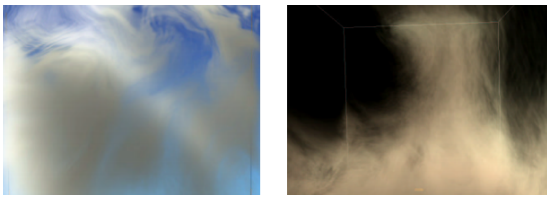

Figure 7: Images showing gas simulations via the Stam method [23].

#### 3.1.3 Annimating Droplets

Modeling and animating individual droplets can almost be treated as an entirely separate problem. A geometric approach to calculate droplet motion on a smooth surface was presented by Hégron [26]. Hégron assumed a single contact point between the fluid drop and the surface that was maintained throughout the animation. Discrete displacement steps on the tangent to the surface was then calculated and projected onto the surface to form the motion of the droplet. Much of the physics of the problem is ignored in this approach but surveys an animation purpose.

A more physically correct approach to droplets running down a surface was presented in [27]. Here the interactions between the fluid and solid are considered to change the shape and motion of the droplets depending on surface tensions, interfacial tensions, and gravitational forces. The main premise behind the physics model is that droplets will run down an inclined plane if their masses are greater than a static critical weight defined by the system properties at that point. The surface itself was modeled as a discretized mesh with the droplets moving from one mesh point to an adjacent one. An alternative approach to droplet animation is to use a combination of particle-based methods and implicit surfaces [28] [29] (see figures 8 & 9 for examples of this method).

In high quality renders persuing photorealism of waterdrop effects, a technique known as the metaball technique was commonly used. Originally introduced by Blinn [30] where they were entitled "blobs", the method is used to represent soft objects like liquids. The metaball method allows for free-form deformation as well as droplet merging, which leads to its popularity in this application.

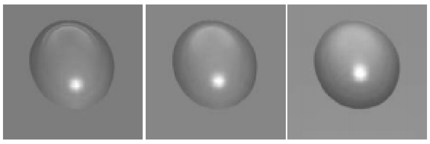

Figure 8: Illumination of droplets formed by a combined particle and implicit surface methods [28].

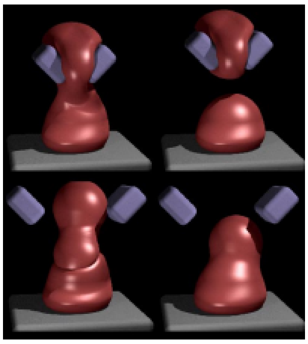

Figure 9: Images showing deformation of a fluid defined by an implicit surface and a 9 particle system[29].

# 4 Contemporary Work

The computational demands of accurate fluid dynamic models make many unsuitable for animation purposes, however, some recent approaches in fluid dynamic models have had significant advances. Forster and Fedkiw present a successful method of modeling liquids which again derives from the 3D Navier-Stokes equations[31]. They solve the equations using a semi-Lagrangian method and then adapt the solution (which is usually used for gas modeling) by combining it with a particle and implicit surface model. The implicit surface prevents mass dissipation while the particle system still allows for splashing which can be seen in figure 10.

Modern advances in illumination and lighting have drastically improved the visual fidelity of fluid animation. Fluids interact with light in complex ways, such as scattering and refraction, many of which are often ignored in earlier renders. Premože and Ashikhmin present an approach describing how light interacts with a body of water, entitled light transport [32]. Their method takes into account many optical properties of fluids including; diffusion, attenuation, scatter, and absorbance with the results being highly convincing of a realistic scene (figure 11).

## 4.1 Real Time Fluid Models

The major focus of most modern fluid animation techniques is interactivity and speed. Driven by developments in games and other real-time applications such as virtual reality, real-time rendering of complex fluid behavior is a challenging avenue of development. The most promising paths appear to be combining various methods to strike a balance between suitable physical realism and simulation speeds.

Bagar et. al. presents a real-time water simulation and rendering method based on a layered particle model [33]. The layered approach allows the body of water to be divided into sections which behave differently, such as a foam layer and a surface layer. However, being a full particle based system, computational limitations do exist in regard to how many particles can be simulated.

Returning to partial differential equation approaches, a group led by Auer used the closest point method for numerically solving these equations [34]. This method achieves real-time simulation of surface fluid effects. The speed is primarily achieved by exploiting modern graphics processing unit's (GPU's) architecture to rapidly compute solutions using the parallel processing power of the GPUs. Exploiting GPU hardware is a trend of many approaches to real-time fluid simulation. The work presented in [35] highlights the successes of parallel computing on CPU and GPU architecture for the simulation of complex fluid dynamic systems. They present a true real-time approach suitable for virtual reality and games that focus on Navier-Stokes equations applied to a particle system of fluid. Vuyst et. al. also propose a parallel computing approach that exploits GPU architecture for real-time fluid dynamics [36].

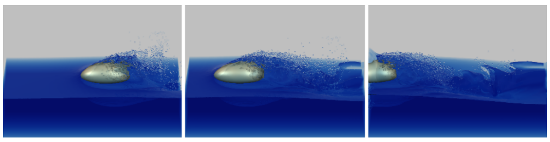

Figure 10: An ellipsoid creating a splash along the surface of a fluid body generated via Forster and Fedkiw's method [31].

Figure 11: The left-hand image is a photograph and the right a rendered scene generated using light transport techniques [32].

Applications of smoothed particle hydrodynamics (SPH), originally developed for astrophysical problems [37], has since been applied readily to modern fluid dynamic animation methods [38][39][40]. This is due to three main reasons; firstly SPH guaranteeing mass conservation (as the particles themselves have a mass), secondly SPH calculates weighted contributions from particles rather than linear combinations, and finally SPH can be used to simulate a two-phase interacting fluid boundary such as water and air.

Position based fluid (PBF) methods, an example of which can be seen in figure 12, are an alternative approach to SPH methods [41]. Due to SPH's sensitivity to density fluctuations and the computational cost associated with enforcing incompressibility in SPH, PBF was developed as an alternative. The method does have its limitations, these include problems with the artificial pressure term that is used. The author identifies deformable objects as a particularly promising application of the method.

Optimising performance real-time fluid simulation is a feature of many modern publications, such as Köster and Krüger's adaptive position based fluids method [42]. Based on the PBF model, the method uses a lightweight, easy-to-integrate extension to the standard iterative solver used by PBF to drastically improve efficiency and speed.

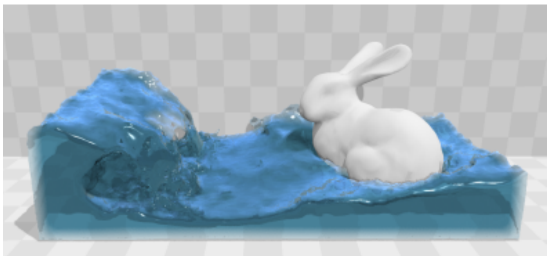

Figure 12: A real-time fluid simulation produced by PBF techniques [41].

# 5 Conclusion

This document offers a historical survey of literature regarding computer graphic and animation techniques for modeling and rendering fluids. Many examples offered are in reference to water simulations however many of the methods are also applicable to various other viscous fluids and gaseous fluids.

The initial portion of the paper details the foundations of fluid modeling, investigating bump map techniques and early ray tracing rendering methods. This then evolves into particle-based models and partial differential equation models for realistic fluid dynamics. The development of these methods is followed as well as animation techniques surrounding droplet motion.

Finally, contemporary work in the field is considered, with the focus being on real-time applications of fluid dynamic models. It is identified that hardware developments, particularly in GPU parallel computing, are a major contribution to the advances made in real time fluid simulation. New methods of modeling are also explored such as SPH and PBF, both of which are in common use in virtual reality and game applications.

With a current drive towards computational efficiency, technique optimisation, and constant developments in hardware; more and more realistic fluid dynamic systems will be able to be incorporated in real-time systems.

## References


[1] Alain Fournier and William T. Reeves. A simple model of ocean waves. *SIGGRAPH Comput. Graph.*, 20(4):75–84, August 1986.

[2] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978.

[3] Bruce Schachter. Long crested wave models. *Computer Graphics and Image Processing*, 12(2):187 – 201, 1980.

[4] Shirley P & Ashikhmin M & Marschner S. *Fundamentals of Computer Graphics*. CRC Press, Natick, 2009.

[5] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980.

[6] Nelson L. Max. Vectorized procedural models for natural terrain: Waves and islands in the sunset. *SIGGRAPH Comput. Graph.*, 15(3):317–324, August 1981.

[7] Nelson L. Max. Carla's island revisited. *The Visual Computer*, 2(3):171–173, Jul 1986.

[8] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, July 1985.

[9] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans. Graph.*, 2(2):91–108, April 1983.

[10] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *SIGGRAPH Comput. Graph.*, 19(3):313–322, July 1985.

[11] Darwyn R. Peachey. Modeling waves and surf. *SIGGRAPH Comput. Graph.*, 20(4):65–74, August 1986.

[12] Alain Fournier and William T. Reeves. A simple model of ocean waves. *SIGGRAPH Comput. Graph.*, 20(4):75–84, August 1986.

[13] Mark Dippé and John Swensen. An adaptive subdivision algorithm and parallel architecture for realistic image synthesis. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 149–158, New York, NY, USA, 1984. ACM.

[14] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. 13:305–309, 12 1989.

[15] David Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface*, volume 91, pages 255–262, 1991.

[16] John Daintith. Oxford University Press, 2008.

[17] Karl Sims. Particle animation and rendering using data parallel computation. *SIGGRAPH Comput. Graph.*, 24(4):405–413, September 1990.

[18] Hugh Mallinder. The modelling of large waterfalls using string texture. *The Journal of Visualization and Computer Animation*, 6(1):3–10, 1995.

[19] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, pages 49–57, New York, NY, USA, 1990. ACM.

[20] J. X. Chen, N. d. V. Lobo, C. E. Hughes, and J. M. Moshell. Real-time fluid simulation in a dynamic virtual environment. *IEEE Computer Graphics and Applications*, 17(3):52–61, May 1997.

[21] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471 – 483, 1996.

[22] N. Foster and D. Metaxas. Controlling fluid animation. In *Proceedings Computer Graphics International*, pages 178–188, Jun 1997.

[23] Jos Stam. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[24] 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.

[25] David Mould and Yee-Hong Yang. Modeling water for computer graphics. *Computers & Graphics*, 21(6):801 – 814, 1997. Graphics in Electronic Printing and Publishing.

[26] Gérard Hégron. Rolling on a smooth biparametric surface. 4:25 – 32, 01 1993.

[27] Kazufumi Kaneda, Takushi Kagawa, and Hideo Yamashita. *Animation of Water Droplets on a Glass Plate*, pages 177–189. Springer Japan, Tokyo, 1993.

[28] Patrick Fournier, Arash Habibi, and Pierre Poulin. Simulating the flow of liquid droplets, 1998.

[29] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 287–290, New York, NY, USA, 1995. ACM.

[30] Jim Blinn. A generalization of algebraic surface drawing. Association for Computing Machinery, Inc., July 1982.

[31] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 23–30, New York, NY, USA, 2001. ACM.

[32] S. Premoze and M. Ashikhmin. Rendering natural waters. In *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pages 23–434, 2000.

[33] Florian Bagar, Daniel Scherzer, and Michael Wimmer. A layered particle-based fluid model for real-time rendering of water. *Computer Graphics Forum*, 29(4):1383 – 1389, 2010.

[34] S. Auer, C. B. Macdonald, M. Treib, J. Schneider, and R. Westermann. Real-time fluid effects on surfaces using the closest point method. *Computer Graphics Forum*, 31(6):1909–1923, 2012.

[35] José Ricardo da Silva Junior, W Gonzalez Clua, Anselmo Montenegro, Marcos Lage, Andrade Dreux, Mark Joselli, A Pagliosa, and Lucille Kuryla. A heterogeneous system based on gpu and multi-core cpu for real-time fluid and rigid body simulation. *International Journal of Computational Fluid Dynamics*, 26(3):193–204, Mar 2012. dummy note.

[36] Florian De Vuyst, Christophe Labourdette, and Christian Rey. Gpu-accelerated real-time visualization and interaction for coupled fluid dynamics. *21ème Congrès Français de Mécanique, 26 au 30 août 2013, Bordeaux, France (FR)*, 2013.

[37] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.

[38] Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and Shi-Min Hu. Multiple-fluid sph simulation using a mixture model. *ACM Trans. Graph.*, 33(5):171:1–171:11, September 2014.

[39] Chen Huang, Jian Zhu, Hanqiu Sun, and Enhua Wu. Parallel-optimizing sph fluid simulation for realistic vr environments. *Computer Animation and Virtual Worlds*, 26(1):43–54, 2015. CAVW-13-0097.

[40] Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R. Martin, and Shi-Min Hu. Multiphase sph simulation for interactive fluids and solids. *ACM Trans. Graph.*, 35(4):79:1–79:11, July 2016.

[41] Miles Macklin and Matthias Müller. Position based fluids. *ACM Trans. Graph.*, 32(4):104:1–104:12, July 2013.

[42] Marcel Köster and Antonio Krüger. Adaptive position-based fluids: Improving performance of fluid simulations for real-time applications. *arXiv preprint arXiv:1608.04721*, 2016.