MODELLING AND RENDERING LIQUIDS IN MOTION

Alan Murta and James Miller
Advanced Interfaces Group
Department of Computer Science
University of Manchester
Manchester M13 9PL, UK.
amurta@cs.man.ac.uk

ABSTRACT

An experimental method for modelling and rendering liquids in motion is presented. A particle system is used to model dynamic liquid behaviour, while implicit surfaces serve as a wrapper to support realistic rendering methods. Aspects of particle-environment interaction and the use of dynamically sized particle primitives are discussed. Techniques used in the generation of implicit surfaces are also described, including environmental distortions of surface profiles and the management of surface extents to maintain constant volume. Preliminary work on multi-fluid interactions is also outlined. A custom ray tracer is used for final image generation, and several examples of the liquid modelling system are provided as an illustration of the method.

CR Categories: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.5 [**Computer Graphics**]: Object Modeling; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism.

Keywords: Natural phenomena, ray tracing, particle systems, implicit surfaces.

1. INTRODUCTION

The presentation of liquid behaviours within synthetic environments is a challenging computer graphics problem. Liquids conform neither to the Euclidean robustness of solid bodies, nor to the random diffusibility of gaseous phenomena. They exist in an intermediate state: dynamic, amorphous, incompressible, and exhibit a readiness to coalesce or separate in arbitrary measures. Liquids engaged in pouring and splashing actions possess a form which is intimately linked with the geometric and physical character of their environment, with a topology which may change dramatically over time.

Any complete approach to the graphical depiction of such behaviours must unite physical modelling techniques with rendering methods suited to the representation of amorphous forms. This paper describes a 'proof of concept' investigation into the use of a particle-based modeller and implicit surface renderer for portraying the macroscopic characteristics of liquids in motion. A number of customisations to the modelling and rendering processes are described to produce a more authentic interpretation of liquid behaviours.

1.1. Organisation

The paper begins with a brief survey of work related to the modelling and rendering of liquids. This is followed in section 3 by a description of a new particle model developed to approximate dynamic liquid behaviours. The generation, management and rendering of implicit surfaces is described in section 4. Section 5 presents a series of images from the liquid modeller, and finally, conclusions and future extensions are outlined.

2. RELATED WORK

Early investigations into the graphical modelling of dynamic liquid behaviour focussed upon the reproduction of surface wave effects. A common approach is to use Fourier synthesis techniques to apply animated ripples to a surface [Musgr93], together with a renderer which uses bump or displacement mapping to convey the surface detail. The method may be extended to incorporate effects due to underwater topography, diffraction and reflection around obstacles and the formation of surf in breaking waves [Peach85, FouRe86].

Wcilso et al. [WciKi98] have developed a method to model displacement behaviours in a settled body of liquid. The simulation space is decomposed uniformly into a number of cubic cells which record the location of the settled liquid within the scene. Dropping a solid object into the liquid body invokes a set of rules which model flow between cells, and can reproduce the effects of Archimedian displacement. Although the liquid surface is perturbed as a result of the object impact, true wave-like behaviour is not established. Also, there is no direct means of extending this approach to mimic realistic splashing phenomena.

Yu et al. [YuJun98] describe a technique for rendering liquid droplets in contact with a flat surface. Droplet shapes are represented using spherical implicit primitives (metaballs), which are subjected to geometric distortions in order to mimic the effects of gravity. Shape irregularities may be introduced through the manual manipulation of control points which apply further distortions to the droplet. A ray tracer is used to provide high-quality output. The overall approach is somewhat limited by its static nature – all droplets are considered to be at rest, and there is no direct support for pouring, splashing or surface flow effects.

Desbrun and Gascuel [DesGa95] have refined earlier work of Miller and Pearce [MilPe89] and Tonnensen [Tonne91] to model viscous fluids using particle systems. They define a physical model of particle interaction, and use implicit surfaces to render the resulting forms. Their simulated materials may be cut, torn, poured and fused.

An ongoing area of research is the development of efficient methods to reproduce the *optical* effects associated with bodies of liquid. Caustics, for example may be approximated using animated projected textures [Stam96], or may employ computationally expensive 'forward' ray tracing from the light source [WatPo98].

3. PARTICLE BEHAVIOUR

Particle systems have long been used to model a variety of amorphous or 'fuzzy' objects [Reeve83]. The simplest systems consist of a collection of *independently* acting particles (point masses), each of which may act in response to global influences such as gravity and may be subject to environment collision rules. Such systems exhibit little in terms of structural behaviour, and are best suited to the modelling of ethereal entities such as fire, smoke, explosions and the movement of fine powders. The simulation of more coherent materials such as liquids and deformable solids typically requires the application of interaction behaviours between the particles themselves.

3.1. Particle-particle interactions

We use a particle system to define the distribution and movement of a simulated liquid within a scene. Our modelling system approximates macroscopic liquid behaviours using an inter-particle force profile which exhibits a net repulsion over short distances, a net attraction within the medium range and zero effect over long distances. The force profile is defined using a simple tabular approach: force values are specified at a series of key inter-particle distances, and linear interpolation is used to compute force values within the intervals. The attractive force supports localised coalescing of liquid droplets, while the repulsive component prevents the collapse of the fluid to a singularity [DesGa96]. The basic force model may be extended through the introduction of simple friction and mechanical damping rules to mimic more closely the properties of real liquids. We use simple Euler integration of forces with small time steps to help reduce the dynamic instabilities which are characteristic of multiple body problems of this kind. Computational efficiency is promoted through the use of a uniform voxel-based spatial decomposition to prevent the unnecessary evaluation of negligible forces between remote particles.

3.2. Families

A novel feature of our modelling system is its support for the creation of distinct particle *families*, thereby allowing the specification of different inter-particle force profiles within and between family groups. This discrimination affords the modelling of multi-fluid interactions, such as the separation of oil and water, or the passage of gas bubbles through a body of liquid. Figure 1 shows the differentiation of two colour-coded particle families under zero gravity. Particles within each group have a greater affinity for their own family members than those belonging to the opposing family, and this results in an overall separation of fluid groups.

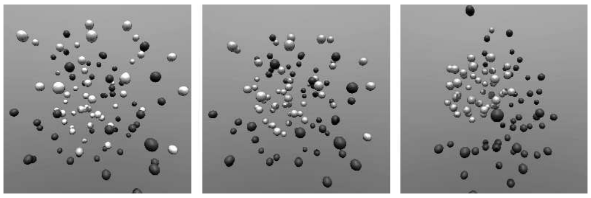

Figure 1: Separation of particle families

3.3. Particle-solid interactions

Our system implements a variety of particle-solid interaction behaviours. Particle-solid collisions are detected using a method similar to the ray-surface intersection tests found in ray tracing [Glass89]. If a particle impacts a surface at high velocity, reflection rules are applied to model splashing phenomena. A Gaussian random deflection is applied to the reflection vector to encourage a more natural non-deterministic behaviour. Low velocity surface impacts result in the application of *sliding* rules. In real liquids, droplets which encounter surfaces at low speeds are unlikely to possess sufficient kinetic energy to offset adhesion forces, and remain in a state of surface contact. The sliding rules of our model move slow particles in a direction parallel to the local surface. Frictional forces are also applied. Particles may leave contact with a surface if their flow velocity moves them beyond the surface's extent, or as a result of external forces (gravity, for example, may liberate surface particles where local surface normal points in a downward direction).

3.4. Particle splitting and merging

A problem with the modelling approach described so far is its fixed granularity. Whereas sizeable agglomerations of liquid may be represented by a collection of nearby particles, there is a fixed limit – due to each particle's finite size – to how finely such bodies may be subdivided. Therefore this simple liquid model differs from macroscopic real world behaviour in that it features a minimum droplet size, and all liquid bodies are comprised of integer multiples of these unit primitives. One way to obscure these artefacts is to reduce the scale of the model primitives by some factor n. However, such an approach rapidly becomes unwieldy due to the $O(n^3)$ increase in the number of particles used to represent a given volume of liquid.

Instead, our modelling approach allows individual particles to be split on demand into several smaller components, and also implements the inverse process which fuses small particles into larger ones. We thus define a maximal (default) particle size and permit the fission and fusion of these unit components. In the current model splitting occurs in situations of dynamic stress, such as particle-solid collisions. When a small, single particle droplet strikes a solid surface it is instantaneously replaced by a small number of separate child particles (typically 4–6), each possessing a random fraction of the parent's mass. These smaller primitives have modified inter-particle force profiles to maintain a space-filling behaviour which is consistent with the parent particle. The child particles are then subject to scattering or sliding rules in a manner appropriate to the initial impact conditions. Recursive subdivision of child particles is permitted; we have found a recursive depth limit of 2–3 to give good results.

A second situation in which subdivision may occur is when individual or small groups of particles are isolated from a larger body of liquid. The splitting decision is based upon an evaluation of the local packing density for each individual particle. Subdivision in sparse regions is used to disguise the discrete nature of the underlying particle model.

Particle merging occurs when several close particles share similar velocity characteristics. Only 'small' particles are considered as merging candidates, thereby preventing the formation of overlarge particles in sizeable bodies of liquid.

4. IMPLICIT SURFACE GENERATION

An implicit surface is the shape formed by all three-dimensional points $\overline{s}$ which satisfy $f(\overline{s}) = c$ for some function f and scalar isosurface constant c [Bloom97]. We use particles as our implicit surface primitives: each generates a spherical potential field, the influence of which is set within some radial distance R. The value of c is chosen so that particles in close proximity (maintaining the natural packing density of the particle system) exhibit a single merged surface. The sum of these fields define the function f in 3-space. The combined field at any sample point $\overline{s}$ is given by:

$$f(\overline{s}) = \sum_i \begin{cases} \frac{2d_i^3}{R^3} - \frac{3d_i^2}{R^2} + 1 & \text{if } d_i < R \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where d_i represents the distance between $\overline{s}$ and the ith particle at $\overline{p_i}$ weighted by the particle's size m_i:

$$d_i = \frac{|\overline{s} - \overline{p_i}|}{\sqrt[3]{m_i}} \quad (2)$$

The size weighting causes larger particles to generate a more extensive field in their locality, with respect to smaller ones. Our choice of field function is not unique, and other alternatives may be used – see [Wyvil86] for example. The implicit surface is rendered using a ray tracer, since we are interested in reproducing the refractive characteristics of our simulated liquid. The field function is calculated at intervals along the ray path, and its value is carefully monitored with respect to the chosen isosurface constant c. The interval spacing is reduced in regions where the field value approaches c, and interpolation methods are used to compute any isosurface intersection points to a high degree of accuracy. The normal of the isosurface at the ray intersection point P may be determined by jittering the ray to produce two nearby intersections on the surface. The direction vectors of these secondary intersections with respect to P are calculated, allowing the local surface normal to be approximated using a cross product.

4.1. Isosurface management

The isosurfaces which are produced by the above method may form several distinct droplet agglomerations or 'clusters'. At each time step of the particle simulation we automatically label each particle according to which cluster it belongs. Cluster membership is determined by monitoring the isosurface field function along a network of lines which join all nearby particle locations. If it is possible to traverse this network from particle P_A to particle P_B without the field strength falling below the isosurface constant or encountering solid objects then these particles belong to the same droplet cluster, otherwise they lie within disjoint clusters (Figure 2).

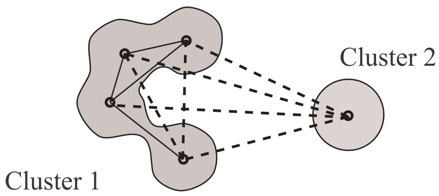

Figure 2: Determining cluster membership

We use cluster membership information to locally modify the implicit generator function so as to maintain a constant volume of liquid throughout the simulation process. Clusters may also be used to prevent unwanted blending between nearby droplets (by constructing a blending graph of the kind described in [DesGa95]), although this modelling refinement has not been implemented in the current system.

4.2. Volume adjustment

A fundamental property of liquids is their incompressibility. A naïve particle-based simulation of macroscopic liquid behaviours which uses a fixed implicit surface wrapper for rendering purposes will not obey the constant volume constraint of real liquids. Ideally, we would like the volume of liquid exhibited by an n-particle cluster to be the same as that contained within n isolated single particle droplets.

For a given isosurface constant it is relatively straightforward to calculate the volume contained within an isolated single particle droplet, due to its spherical shape. Scaling this amount by the number of particles present in any cluster gives an accurate measure of the cluster's ideal volume V_i. The *actual* volume V_a produced by the merged potential fields of the cluster particles will differ from this value. Determining the magnitude of the error will allow us to make adjustments to the cluster's implicit volume function to enforce the constant volume constraint.

Analytically computing V_a is difficult, so we generate an estimate using an octree-based subdivision of space around the cluster (Figure 3). Octree cells whose eight vertices all lie within the isosurface are classed as droplet internal volumes, whereas those cells with all vertices outside are deemed external. The contained volume is given by the sum of the internal volumes and a fractional component of the mixture cell volumes (the fractional contribution of a cell being approximated by counting the number of cell vertices which lie within the isosurface and dividing this value by eight). Setting minimum and maximum recursion depths allows errors to be kept within user-defined limits.

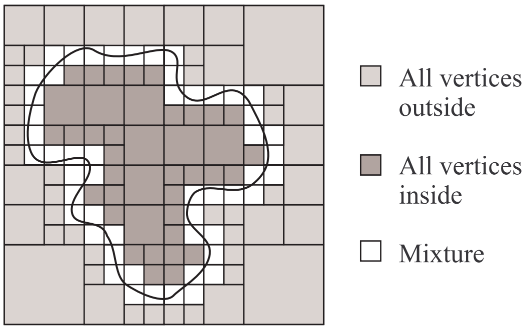

Figure 3: Estimating actual cluster volume

The error V_e in the cluster volume, given by $|V_i - V_a|$, may be minimised by adjusting the potential field function for the given cluster. We extend equation 1 to include a per-particle scale factor k_i which locally inflates or deflates the field profile:

$$f(\overline{s}) = \sum_i \begin{cases} k_i(\frac{2d_i^3}{R^3} - \frac{3d_i^2}{R^2} + 1) & \text{if } d_i < R \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Scale factor adjustments are made simultaneously to all particles within a cluster. The value of k_i is determined by trial and error: a binary chop refines its value until V_e is less than a user specified tolerance. Individual particles maintain their k_i state throughout the simulation, and as a result only small adjustments are required on each time-step.

An alternative method of implicit surface volume adjustment may be found in [Desga95].

4.3. Distortion of potential fields

The potential field generated by an individual particle is spherical in profile. Whereas the resulting isosurface shape may be acceptable for liquid droplets in free-fall, it does not give convincing results for droplets lying on a solid surface. We therefore identify particles which are in contact with solid bodies and subject their potential fields to scale and shear distortions to produce more realistic droplet profiles.

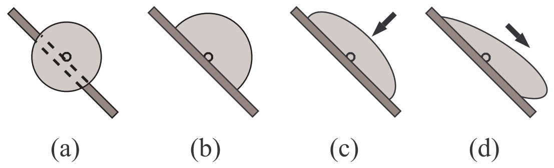

Figure 4: Isosurface distortions

Figure 4 illustrates the transformations which are applied to a droplet in contact a solid surface. In this example the droplet contains a single particle, however the method extends trivially to larger clusters since the field distortions operate on a per-particle basis. Part (a) of the figure shows the raw spherical state of the droplet isosurface. In (b), the portion of the droplet which notionally penetrates the surface is first culled, and the isosurface profile is scaled to compensate for the loss of volume (the required scale factor being determined analytically). Images (c) and (d) show the distortions which result from gravitational components oriented normal and tangential to the surface, as indicated by the arrows. The normal component flattens the profile whilst a radial expansion parallel to the surface accounts for droplet spread. Finally, a non-linear shear is applied along the tangential component, resulting in a distinctive droplet profile.

4.4. Multiple particle families

The surface model which has been implemented in our current system does not yet cater for multi-fluid simulations of the kind outlined in section 3.2. In such cases, each particle family creates its own potential field, and the interaction between fields must be accounted for in the surface generation process. A way to achieve this would be to modify the isosurface generator functions to accommodate the overlapping fields of several family groups. Equation 4 presents a surface formulation for a family A whose potential field competes with a second family B:

$$f_A(\overline{s}) = \begin{cases} c & \text{if } f_B(\overline{s}) < c \\ f_B(\overline{s}) & \text{otherwise} \end{cases} \quad (4)$$

where c is an isosurface constant shared by both family groups. Family B has an equivalent isosurface specification. Figure 5 illustrates the concept for two non-mixing fluids sited in close proximity. The condition of equation 4 prevents the mutual penetration of isosurface regions (dotted lines) by diverting the shared fluid boundary along the equipotential field path of both families.

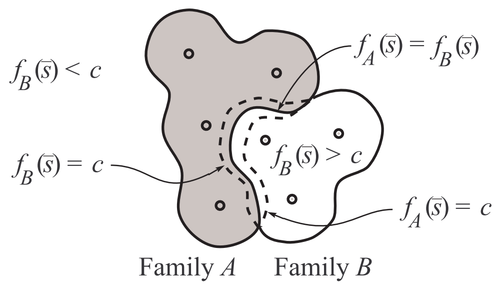

Figure 5: Interaction of multi-fluid surfaces

5. EVALUATION

A series of images are presented here to illustrate various properties of the modelling and rendering system.

Figure 6 shows the distortions which are applied to the potential fields of particles in contact with a solid surface. The contact state of any particle in the system is determined using a particle-surface proximity rule. The first image features a single-particle droplet in free-fall, while subsequent images show surface contact profiles for various inclinations.

Figure 7 shows the merging behaviour of two single-particle droplets. As the profiles combine on close approach, the algorithm modifies the po-

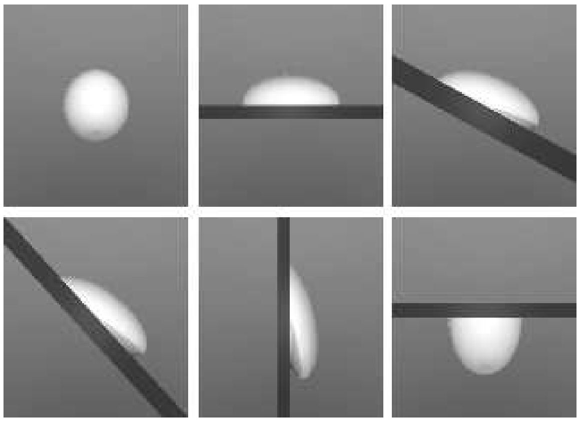

Figure 6: Droplet profiles

tential field strength of particles in the cluster to conserve the total liquid volume.

Figure 7: Two droplets merging

A droplet impact sequence is illustrated in Figure 8. The particles contained within the falling droplet are split into smaller primitives upon hitting the surface (as described in section 3.4), and are scattered 'randomly' in a Gaussian distribution. Some of the spawned particles settle near the point of impact and quickly re-merge into larger particles. Others are projected clear of the impact site and eventually come to rest as independent droplets.

The first image of Figure 9 shows a quantity of liquid flowing along and spilling from an open-ended gutter. The free-falling liquid initially remains coalesced, but eventually separates into individual droplets as gravitational acceleration increases the average particle spacing. A similar effect is observed in real liquid flows. The second image shows a combined splashing and sliding behaviour, as a moderate liquid flow strikes a horizontal surface.

5.1. Performance

The modelling techniques described are relatively costly their processing requirements. Many aspects of the implementation rely upon iterative refinement (cluster volume estimation and adjustment, ray-isosurface intersection testing) and contribute significantly to the overall complexity

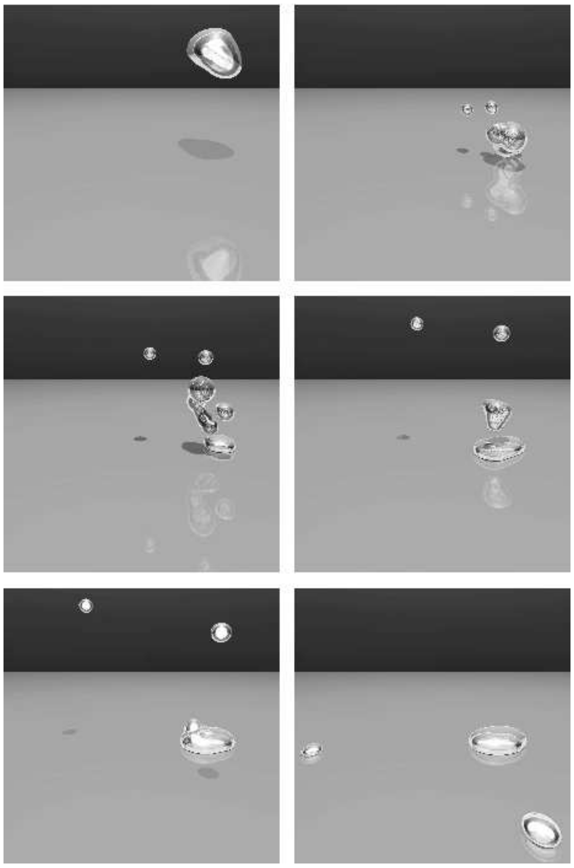

Figure 8: Impact sequence

of the algorithm. The greater use of analytical methods should help to reduce processing costs. The images presented earlier in this section typically required several hours to render on a Silicon Graphics O2 workstation, with cluster volume management and ray tracing a 512×512 image taking roughly the same proportion of the time. Fortunately several aspects of the algorithm parallelise well. For example, volume management may be independently computed on a per-cluster basis, and ray tracing parallelises trivially in image space.

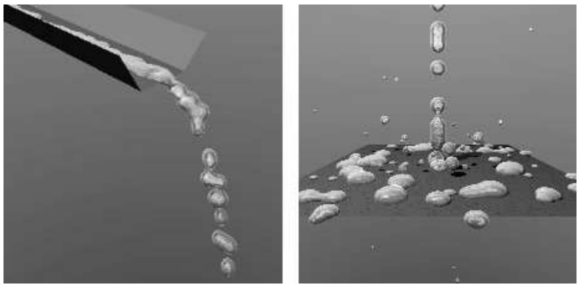

Figure 9: Pouring, splashing and sliding

6. CONCLUSIONS

Our aim has been to establish a means of portraying dynamic liquid behaviours such as pouring and splashing within synthetic scenes. For this we have developed several new approaches including: particle splitting and merging behaviours to dynamically adjust the model granularity, particle families to support multi-fluid interactions, and a variety of potential field modification rules to conserve fluid volume and increase the realism of isosurface profiles.

Static images do not provide an ideal means of illustrating the theme of this paper – liquids in motion, however the results do bear a visual resemblance to dynamic liquid behaviours recorded using high speed photography.

A more detailed presentation of this work can be found in [Mille98].

6.1. Further work

The work presented is currently at an early stage of development, and several modelling refinements and algorithmic optimisations still need to be addressed.

The inter-particle force profiles and implicit surface field functions we use at present are rather *ad hoc*. The modelling system is likely to benefit from a more rigorous approach of the kind described in [DesGa96]. It is envisaged that this would also assist in the development of more efficient analytical solutions to the problem of cluster volume management.

Whereas the particle system already supports the modelling of multiple-fluid interactions, the isosurface rendering extensions of section 4.4 remain to be implemented. This development will permit phenomena such as the separation of oil-water mixtures and the passage of gas bubbles through a liquid to be simulated. An assessment may then be made regarding the effectiveness of the model in such situations.

Particle-isosurface approaches of the kind described do not work well for scenes which feature settled bodies of liquid, since completely flat liquid surfaces cannot be represented easily. Realistically combining dynamic and settled liquid behaviours in a unified modelling framework remains an unsolved problem.

Finally, it would be interesting to incorporate temporal effects into the simulation. Maintaining a history of the isosurface extent would allow the presentation of damp areas in the wake of liquid flows.

ACKNOWLEDGEMENTS

We wish to thank the University of Manchester Advanced Interfaces Group members for their support, the UK Engineering and Physical Sciences Research Council for postgraduate funding, and the anonymous WSCG reviewers for their constructive comments.

REFERENCES

[Bloom97] Bloomenthal, J. (ed.), Introduction to Implicit Surfaces, *Morgan Kaufmann*, San Francisco, 1997.

[DesGa95] Desbrun, M., Gascuel, M.P., Animating Soft Substances with Implicit Surfaces, *ACM Computer Graphics (SIGGRAPH '95 Proceedings)*, 1995.

[DesGa96] Desbrun, M., Gascuel, M., Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies, *6th Eurographics Workshop on Animation and Simulation*, 1996.

[FouRe86] Fournier, A., Reeves, W.T., A Simple Model of Ocean Waves, *ACM Computer Graphics (SIGGRAPH '86 Proceedings)*, pp.75–84, 1986.

[Glass89] Glassner, A.S., An Introduction to Ray Tracing, *Academic Press*, London, 1989.

[MilPe89] Miller, G., Pearce, A., Globular Dynamics: A Connected Particle System for Animating Viscous Fluids, *SIGGRAPH '89 Course Notes 30*, pp.305–309, 1989.

[Mille98] Miller, J.S., Modelling Liquids using Particle Systems and Implicit Surfaces, Ph.D. Thesis, *University of Manchester Department of Computer Science*, 1998.

[Musgr93] Musgrave, F.K., Methods for Realistic Landscape Imaging, Yale University Department of Computer Science, TR-1038, May 1993.

[Peach86] Peachey, D.R., Modeling Waves and Surf, *ACM Computer Graphics (SIGGRAPH '86 Proceedings)*, pp.65–74, 1985.

[Reeve83] Reeves, W.T., Particle Systems – A Technique for Modeling a class of Fuzzy Objects, *ACM Computer Graphics (SIGGRAPH '83 Proceedings)*, pp.359–367, 1983.

[Stam96] Stam, J., Random Caustics: Natural Textures and Wave Theory Revisited, *SIGGRAPH '96 Technical Sketch (Illumination)*, 7 August 1996.

[Tonne91] Tonnensen, D., Modeling Liquids and Solids using Thermal Particles, *Graphics Interface '91*, Calgary, pp.255–262, 1991.

[WatPo98] Watt, A., Policarpo, F., The Computer Image, *Addison-Wesley*, pp.149–152, 1998.

[WciKi98] Wcislo, R., Kitowski, J., Mościński, J., Cellular Automaton as a Fast Tool for Animation of Liquid in Multi-Object Scenes, *6th International Conference in Central Europe on Computer Graphics and Visualization (WSCG '98)*, pp.417–423, 1998.

[Wyvil86] Wyvill, G., McPheeters, C., Wyvill, B., Data Structure for Soft Objects, *The Visual Computer*, 2(4), pp.227–234, 1986.

[YuJun98] Yu, Y.J., Jung, H.Y., Cho, H.G., A New Rendering Technique for Water Droplet using Metaball in the Gravitation Force, *6th International Conference in Central Europe on Computer Graphics and Visualization (WSCG '98)*, pp.448–454, 1998.