An Overview of Surface Tracking and Representation in Fluid Simulation

Listy Stephen
Master of Technology, Computer Science and Engineering
VJEC, Kannur University, India

Anoop Jose
Asst. Professor, Computer Science and Engineering
VJEC, Kannur University, India

***Abstract*—Realism in fluid animation can be achieve with physics based techniques and is the best among other approaches. Now, this area constitutes hot researches. There are number of mechanisms evolved with the advent of both hardware and software technologies. Most of the fluid simulation methods described with or without a clear surface representation. This paper focused on a quantitative survey of various fluid surface tracking and representation techniques. Suitable tracking schemes with the hybrid fluid simulation approach may give mind blowing visual effects for various applications.**

Keywords*—*Fluid simulation; Physics based animation; Realism; Surface representation; Tracking

I. Introduction

Physics based fluid simulation is not merely a new idea in the field 3D animation. Creation of fluid animation on the basis of physics laws normally called fluid simulation. Most of the techniques in the respective area, are the numerical approximation of fundamental Naiver-Stroke equation, which has some relation with the Computational Fluid Dynamics (CFD). But the applied idea is quite different from CFD. Here, it is appropriate to avoid the light weighted forces acting on the fluid body, in order to reduce the overhead of the system. Surface tracking and its representation is an attractive and enthusiastic charge field in fluid simulation. Many researching fields including simulation and rendering of free surfaces still need to solve this problem efficiently.

The advected surface by a general velocity field frequently changes its topology. Due to this reason, the moving surfaces are typically defined implicitly with the zero set of a scalar field rather than explicit representation of a mesh for instance. Deformable and rigid object surfaces are often represented by using an explicit triangle mesh. They are comparatively simple and efficient data structures and able to sent directly to Graphic processor units for rendering instances. Furthermore, the explicit representation makes direct Lagrangian simulation of soft bodies, are simply modifies the vertex positions. With high deformations, local re-sampling might be required. In contrast from the solid surface tracking methods, the free surface of a liquid constantly changes its topology by splitting and merging with itself. Processing with the explicit representations of the surface is a tricky one under this situation. Modification of the triangle mesh would need whenever a self intersection occurs.

Given, a surface representation and a velocity field at time t and building a representation of the surface at time t + Δt can be phrased as a surface tracking problem. Varied fluid phenomena such as fire, smoke or clouds etc do not posses a clear interface. But liquids have some hope here. It seems, for the intricacy of fluid motion, including merging, diverging, foam and bubble generation, the triangle meshes are not well suited for the effective surface representation.

II. Surface Tracking Methods

Over the past few years various surface tracking methods were raised and moved on hand in hand with different fluid simulation problems. Tracking will provide compelling advantage for the simulation methods, like detailed and visually pleasing result. Herewith mentioned an overview of such existing techniques.

A. Classical Metaballs

Visualization of deformable implicit surfaces is a challenging topic, as it is aimed at representing a whole nonrigid objects, ranging from soft bodies to water and gaseous phenomena. Metaballs are widely used type of implicit surface, invented by Blinn in the early period, [16] used for achieving fluid-like appearances.

The concept of metaballs is closely related to the idea of smoothed particle hydrodynamics (SPH) [17], a convention used for simulating fluids as clouds of particles. Both methods apply smooth scalar functions, that map points in space to a mass density. These scalar functions, considered as smoothing kernels, basically represent point masses that are smoothed out over a tiny volume of space, like a Gaussian blur in 2D image processing. Moreover, SPH-simulated fluids are visualized quite naturally as metaballs. The proposed techniques for visualizing metaballs rely heavily on the SPH method and referred as fluid atoms. They are animated on the CPU either by free-form animation techniques or by physics-based simulation. Moreover, the dynamics of the fluid atoms are interactively determined, hence preprocessing of the fluid animation sequence not possible in this case. Metaballs produces very blobby appearance.

B. Front Tracking Method

Apart from the idea of Classical Metaballs, the Front-Tracking method by Tryggvason, Bunner and Esmaeeli

[13] explicitly discretize the free surface using particles and stored a connectivity list between them [14]. It is highly complex to hold the connectivity list when the free surface experiences extensive deformations or topological changes.

Generally a front structure consists of points connected by elements. The linked list structure is used to store both the points and the elements (front objects), that contains pointers to the previous object and the next object in the list. The order of the list is completely arbitrary and has no connection to the actual order on the interfaces. Usage of a linked list makes the addition and removal of objects particularly simple. Only one information is stored in each point, that is the point coordinates. On the other side, the elements contain most of the front information. Each element knows about - the elements that are connected to the same endpoints; the points that it is connected to; the jump in density across the element; the surface tension; and any other quantities, that are required for a particular simulation. The elements provides direction and for a given front, all elements must have the same direction.

Deformation and stretches occurred, when the front moves. That causes the resolution of some parts of the front become insufficient, while other parts crowded with front elements. Additional elements must either be added to maintain accuracy when the separation of points becomes extra large or the points must be redistributed to maintain adequate resolution. Generally it is favorable to remove small elements. In addition to reducing the total number of elements used to represent the front, element removal usually preventing the formation of " twists and jerks " much smaller than the grid size. While the restructuring of the front makes codes, that uses more complex explicit tracking than front capturing codes. Usage of a suitable data structure made many of the crucial operations relatively straight forward. In early two-dimensional computations, the position of all interface points were adjusted either at every time step or at every few time step to preserving nearly uniform distance. However this is not necessary and able to remove points where needed. Restructuring is relatively simple in two-dimensions. Here large element were split up by inserting a point and delete an element through removing a point. Sometimes it is needed to put a new point at the mid-point between the old end-points of an element using linear interpolation. This is usually account for the curvature of the front by using a higher order interpolation, particularly significant where the surface tension is large. Non-smooth parts of the front leads to a large pressure variations. A simple Legendre interpolation usually works well in this case. For three-dimensional flows, the restructuring is more complicated. Not only are there several different ways to add and delete points, but other manners of the front, such as the connectivity and the shape of the elements. Adding and deleting elements can be done in a number of ways in 3D. Large elements and longest edges need to split into two and replace both this elements and the one sharing the long edge by four new elements. Likewise, elements are deleted, two at a time by merging the shortest edge into a point. Sometimes it is necessary to reconnect the points by swapping edges to make the elements better shaped. In many simulations this step is not following, since it is possible to show that a combination of element insertion and deletion will have approximately the same result.

Often cited shortcomings of Front Tracking are its algorithmic complexity in tracking surfaces of 3D flows, hardness in robustly handling interface merging and breakup particularly in 3D. The reason for these difficulties are the necessity to logically connect interface elements and record changes in connectivity during interface operations ie element addition, deletion and reconnection. In two dimensions these difficulties are actually minor and the implementation of a robust connectivity algorithm is fairly laid on the line. Nonetheless moving to three dimensions, the algorithmic complexity of connectivity increases fiercely and particularly in interface reconnection during topology changes.

C. Point Set Method

Surface tracking methods compelling to take care, when the free surface undergoes large deformations or topological changes. Torres and Brack-Bill [12] have suggested a method to avert this adversity, known as the point-set method. This mechanism varied from the standard front-tracking process, where the indicator function is calculated from the surface normals. Here the information about a surface of unconnected interfacial points extracted by, first constructing an indicator function $I(x)$ and the interfacial points embedded within a computational grid. With a given grid and the location of the interfacial points, an indicator function were constructed, which labels points $I(x)$ lying inside and outside of the interface. This is to distinguish phase or material from another. Zero and one would be the default values to an indicator function for an ideally thin interface. Admitting this approach expel the front tracking method from its dependence on point connectivity. The point regeneration algorithm is one of the complex and computationally expensive process so that, it may create extra burden for the system.

D. Marching Cube Algorithm

The primitive and most popular idea used in solving the contouring problem is marching cubes. This method have a tendency to create ill-shaped triangles which could fixed to some extend with a manner called dual contouring [8], also preserving sharp boundaries and provides adaptive contouring. Normal estimates at edge crossings are important in dual contouring and is sensitive to inaccuracies in it. Boissonnat and Oudot [7] presented a contouring technique which uses Delaunay triangulation to create possibly good triangulations. Additionally, this approach appears to be extraordinarily expensive for something which must run at every time step. Another alternative provided by Hilton [10], called marching triangles which takes a surface-based rather than volume-based approach of contouring. Marching triangles requires incomparably less computation time and fewer triangles. So it can produce good-quality triangles than marching cubes. This would be considered as, one of the bold advantage. The method is not ensured to create closed and manifold meshes in the presence of sharp or thin features, which proves a negative outlook.

Marching cubes algorithm used in Kunimatsu et al's work [9], which maintaining the inherent capability of dealing with topological changes. There are two primary steps in this approach for surface construction problem. Locating the surface corresponding to a user specified value and create triangles constitutes in the first step and then ensure a quality image of the surface by calculate the normals to the surface at each vertex of each triangle. Marching cubes uses a divide-and-conquer algorithmic approach to locate the surface in a logical cube generated from eight pixels. The algorithm determines how the surface intersects this cube, then moves or marchs to the next cube. Finding the surface intersection in a cube by assign one to a cube's vertex, if the data value at that vertex greater than or equals the value of the surface construction. The vertices being inside or on the surface. Cube vertices with values below the surface receive a zero and are outside the surface. The surface intersects those cube edges where one vertex is outside the surface and the other is inside the surface. These concepts helps to determine the topology of the surface within a cube and finding the location of the intersection later. Since there are eight vertices in each cube and two slates inside and outside, only $2^8 = 256$ ways a surface can intersect the cube. These 256 cases enumerated to create a table to look up surface-edge intersections, provide the labeling of a cube vertices.

E. Screen Space Mesh Approach

Muller et al. [15] introduced a powerful approach for the generation of surfaces defined by the boundary of a three-dimensional point cloud. A depth map with internal and external silhouettes was first generated in screen space. Then it is used to construct a 2D screen space triangle mesh with a technique derived from Marching Squares. The algorithm only generates surface where visible, such that view-dependent level of detail came for free and interesting visual effects, possible by filtering in screen space.

The advantages of the screen space mesh are like this. Resolving the parts of surface which are close to the camera with more triangles than distant parts, yielding camera-dependent level of detail. Since it operates in two dimensions, a method derived from Marching Squares can be employed. Which is substantially faster than the 3D Marching Cubes algorithm. In contrast to other screen space approaches such as ray tracing or point splatting, fast standard triangle shading hardware can be used for state-of-the-art forward shading of the surface and occlusion culling, since the mesh can easily be transformed back into world space.

F. VOF and SOLA

Topological changes of the fluid surface were efficiently handled in VOF method [11], which usually walk-in with the marching cubes algorithm. Fundamental technique only uses one scalar value, that is the volume of fluid for one cell. The metric estimate the exact location of the liquids and calculate the total volume of fluid inside the simulation domain. A light weight version of the basic solution algorithm (SOLA) used in the MAC method. SOLA failed to treat free surfaces. An extended version, SOLA-SURF is also available, that uses surface height function method. The significant qualities like simplicity and flexibility of the SOLA codes make them best foundations for the development of more sophisticated implementations. An alternate mesh version of the SOLA code named as SOLA-VM, was chosen as a fundamental for representing the VOF technique. The new experimental version called SOLA-VOF. Many developments have been made and the basic technique has grown through applications to a broad class of problems. Several other attempts combined the SOLA-SURF code with a different interface tracking techniques based on VOF concept.

G. Level Contour Reconstruction Method

The consolidation of the VOF method and the marching cubes algorithm used in Kunimatsu et als attempt [6], is matched with the level contour reconstruction [5], which carrying the inherent potential of handling with topological changes.

The proposed concepts of researchers and their implementation of the interface reconstructions appeared to be quite simple and plain-dealing. Due to the interface stretches and deforms greatly in the simulations, it is prime to add and remove the interface elements during the solving period. Topology change is an inherent feature of both boiling flows and splashing liquids. The Level contour reconstruction of interfaces must be allowed to recombine when either parts of the similar interface or parts of two separate interfaces came together. Interface reconstruction procedure replaces additional burden of all of these particle addition, deletion, and topology changes.

H. Level Set Method

Liquid surfaces can be efficiently represented using level set method and dynamic implicit surface, proposed by Osher and Fedkiw [1]. A signed distance field $\Phi(x)$ explicitly point out the liquid surface and that could defined as the shortest distance from position x to the liquid surface. So that surface of the liquid featured by the zero contour of signed distance field. Positive $\Phi(x)$ symbolizes x is outside the fluid region while negative $\Phi(x)$ identify the inside. Velocity field u(x), which derived from the Navier-Stokes equations used for simulation purpose. Solving the level set equation using the upwind schemes generates the implicit liquid surface and the victim equations is :

$$\frac{\partial u}{\partial t} + (u.\nabla\Phi) = 0$$

Several useful properties are constitutes on signed distance function such as surface normal and curvature, that are resolved by :

$$N = \nabla\Phi$$
$$k = \nabla^2\Phi$$

Dynamic level set is used to track the fluid surfaces. Evolution of liquid surface volume tracked by attaching the particles directly to the surface at initial position and then just move them around in the velocity field. Requirement of adding extra particles is when the surface becomes over sparsely resolved, and removing the additional ones as the surface folds, or splashes back over itself. A flipside method which is possibly similar, exception in not using the particles. This method was developed by Osher and Sethian and is called the level set method. Super accurate fifth order WENO scheme is applied in solving the level set equations, but it still deteriorates from high smoothing artifacts and volume loss. This problem soothed by coupling particles with level sets. Particles assigning at both sides of the interface would eventually improved the level set based method. Applied particles able to revise the level set values of each frame, that would manage the detailed surface features as well as the total volume of fluid. This approach was named as particle level set (PLS) due to the presence of the auxiliary particles. Another mechanism is Marker level set method (MLS) which help to track the dynamic liquid surface. Here only marker particles were seeds at the interface, that obtaining more power full and highly accurate results than PLS. Moreover, the surface markers helps the Marker level set method to supervise non-diffusive surface texture advection.

I. Semi-Lagrangian Contouring

Stam introduced Semi-Lagrangian methods [3] that have been extensively used in computer graphics community to solve the nonlinear advection term of Navier-Stokes equations. Surface tracking method is build on the top this approach. The Semi-Lagrangian Contouring (SLC) [2] proposed by Bargteil, is different from the PLS. The SLC modifies its signed distance field by Semi-Lagrangian fashion and hence the triangle meshes extracted by Marching Cube algorithm in every time step. Additionally, it benefits from implicit surface and explicit polygon meshes with volume conservation, adaptive resolution, also an easy surface property convection.

Here explicit surface representation begins with each time-step. Fluid simulator is able to build up both distance function and a velocity field. The zero set of the field creates new surface. Backward tracing through the flow field helps to obtain the value of the field function. A special structure called distance tree (balanced octree subdivision of the spatial domain), which is widely used in the implementations. Underlined benefits of this structures includes, fast spatial index and fast approximate signed-distance function, which are cumulatively simplified the whole mesh related procedures by assisting the contouring algorithm.

The greatness of the SLC is the ability to track surface properties, such as texture coordinates, color, simulation variables etc accurately at trivial additional cost. It provides a mapping between surfaces at adjacent time-steps. If vertex v in the current mesh maps to point p in the old mesh and some surface property was stored at p, this property can be copied to v. Tracking of surface properties done on this way.

J. Surface Tracking With Error Compensation

When the free surface is physically valid (means that when the normal is anti-parallel with the pressure gradient), the energy is minimized. The surface normal obtained from the surface tracker and the pressure gradient from the fluid simulation, so that any deviation from the minimum energy state causes an unphysical conflict between the fluid simulation and surface tracker. This error can be scale down by following the direction of steepest descent of the energy function [26]. The energy gradient is the derivative of energy function with respect to its free variables. Here the surface tracker is over detailed and under constrained, hence only need of surface tracker is to adjust the control variables. Approximation of divergence free motion done through constraining the surface tracker adjustments to be local rotations. Thus, degrees of freedom made as the orientations of the surface tracker's normal n.

III. Other Methods And Comparative Analysis

Several other methods adapted the high-resolution surface conform to the low-resolution physics, instead of adapting the fluid simulation to the surface tracker. According to Wojtan and Turk [18], and Kim et al. [19], high-frequency visual artifacts can be removed using smoothing algorithms. Another attempt was by Yu and Turk [20]. They tried to use anisotropic smoothing kernels to bias the loss of surface detail. Same effects also provided by Williams [21] and Thurey et al. [22] for volume-loss artifacts with bi-Laplacian smoothing. Such smoothing approaches provide better result in small doses. But they quashed up many surface details by applying too much enthusiasm. Results physically incorrect surface motions. Small-scale viscous flows may support smoothing with over-damped surface tension and it may not be appropriate for inviscid liquid phenomena. To handle the lack of detailed surface motion when combining a low-resolution simulation and a detailed surface, Thurey et al. [25] Bojsen-Hansen [23] and Yu et al. [24] were proposed a high-resolution dynamic surface waves. These methods masked the high resolution surface artifacts with rippling motions, but limited with unnecessary restrictions like height-field, shallow water, and constant wave speed assumptions. Other invention by Wojtan et al. [18] removes topological inconsistencies in the surface tracker by re-sampling the surface, but suffers from the problem of surface noise.

To visualize the implicit surfaces, normally surface particles are not used. Most of the methodologies are following marching cubes algorithm or ray tracing. Discretization of 3D volume into a grid of cells possible with the former method and later one shoots rays from the viewer at the surface to calculate the color and depth of the surface at every pixel. The third, Point-based method is not a common mechanism for visualizing implicit surfaces. Because of the high computational cost and need of large numbers of particles for visualization. Ray tracing is expensive as well but are often effortless to implement on CPU based hardware. Therefore it is more obvious option for offline rendering. Particle systems are good for exploiting temporal coherence. Once the positions of surface particles on a fluid surface are established at a specific time, they have to be moved only a small range to represent the fluid surface a fraction of time later. Marching cubes and ray tracing cannot take care of this property.

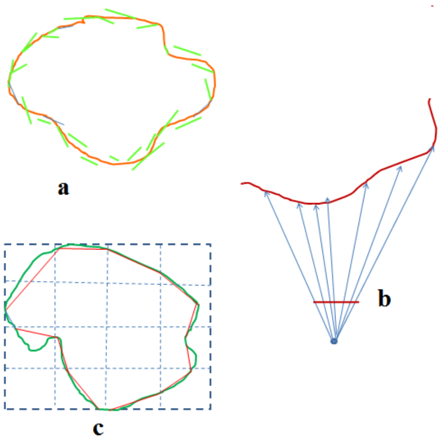

Fig. 1: *Implicit surface representations : a. Marching cube, b. Ray tracing, c. Point based method*

Surface visualization seems like offloaded to graphics hardware with the advancement in the growth of GPU processing power. Furthermore, the applied parallelism of current graphic processor units allows, much faster processing over time. That dominates GPU-run methods over CPU-run methods. Parallelism can also be applied in CPU in some extent with special tools, such as Intel TBB. Doesn't mean all surface visualization techniques are easily modified to work in a parallel environment. For instance, an iterative optimization that walks over the surface by visiting adjacent grid cells is not suitable for parallelization. Its complexity is therefore worser than the point-based method, which has to frequently modify all surface particle positions and the number of particles linearly related to the fluid surface area.

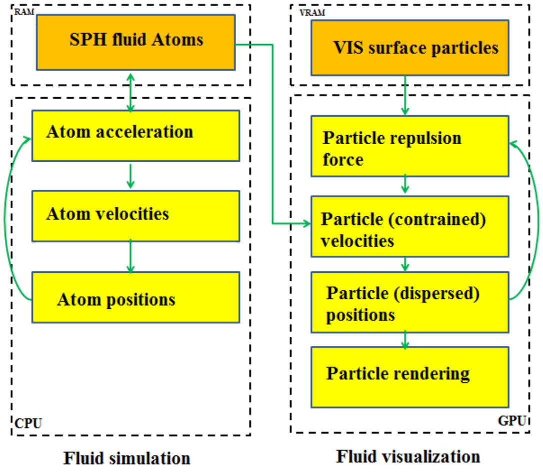

Fig. 2: *Surface visualization on GPU*

Figure 2 illustrate the point based surface visualization by nvidia developers [27]. This method offload the procedures from CPU to GPU and enhances the efficiency.

IV. Conclusion And Future Work

Novel methods for tracking the surface of liquids were surveyed in this paper. It is difficult to propose a best method, that is suitable for all applications ranging from entertainment field to scientific field. The computational cost, appearance and speed of processing are different with each mechanisms. Due to this fact, it is hard to suggest an excellent classification based on standard metrics. Efficient fluid simulation with clear surface representation might be capable of lighten the realism in animation.

Fluid simulation and surface tracking are two different type of problems. Simulation of fluid body can be done without tracking, but in order to add minute details, it will help a lot. In the other sense it can be used as a deformation tool. This survey gives a solid idea of how to carefully choose a particular surface tracking method to complete the hybrid fluid simulation approaches. A fruitful classification is a possible extensions of this survey, that can be considering as future work.

Acknowledgment

The authors would like to thank the reviewers for their useful suggestions to improve the earlier drafts of this manuscript and also thank to Jasmine Hydrose, National Institute of technology Manama, Bahrain.

References

[1] Stanley Osher, Ronald Fedkiw *Level set methods and dynamic implicit surfaces* Springer-Verlag New York, Inc., 2003.

[2] Adam W. Bargteil, Tolga G. Goktekin, James F. O Brien, and John A. Strain *A semi-Lagrangian contouring method for fluid simulation*, 25(1), 2006.

[3] Jos Stam *Stable fluids* ACM Press, Addison-Wesley Publishing Co. New York, NY, USA, 121-128. DOI=10.1145/311535.311548 1999

[4] Foster N, and Fedkiw R. *Practical animation of liquids*, In SIGGRAPH 01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, 23C30, 2001.

[5] Shin S, and Juric D, *Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity*, J. Comput. Phys., 180, 427C470 (2002).

[6] Kunimatsu A, Watanabe Y, Fujii H, et al, *Fast simulation and rendering techniques for fluid objects*, Computer Graphics Forum (Eurographics 2001 Proc.), 20(3), 357C367 (2001).

[7] BOISSONNAT, J. D. AND OUDOT. *Provably good surface sampling and approximation*, In SGP-03: Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. Eurographics Association, Aire-la-Ville, Switzerland,2003 918.

[8] Tao Ju, Frank Losasso ,Scott Schaefer ,Joe Warren *Dual contouring of hermite data* ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2002 TOG Homepage Volume 21 Issue 3, July 2002 Pages 339-346

[9] Kunimatsu A, Watanabe Y, Fujii H, et al, *Fast simulation and rendering techniques for fluid objects* , Computer Graphics Forum (Eurographics 2001 Proc.), 20(3), 357C367 (2001).

[10] Hilton and Illingworth and Delaunay, *Marching Triangles: Delaunay Implicit Surface Triangulation*, CiteSeer publications 1997.

[11] Hirt C W, and Nichols B D, *Volume of Fluid (VOF) method for the dynamics of free boundaries* , J. Comput. Phys., 39, 201C255 (1981).

[12] Torres D J, and Brackbill J U, *The point-set method: front-tracking without connectivity*, J. Comput. Phys., 165, 620-644 (2000).

[13] Tryggvason G, Bunner B, Esmaeeli A, et al. *A front tracking method for the Computations of multiphase flow*, J. Comput. Phys., 169, 708-759 (2001).

[14] Unverdi S, and Tryggvason G, *A front tracking method for viscous, incompressible, multifluid flows*, J. Comput. Phys., 100 25 (1992).

[15] Muller M, Schirm S, and Duthaler S. 2007. *Screen space meshes.* In SCA 07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, Eurographics Association, Airela Ville, Switzer-land, Switzerland, 9C15.

[16] Blinn, James F. 1982. *A Generalization of Algebraic Surface Drawing.*, In ACM Transactions on Graphics 1(3), pp. 235256.

[17] Matthias Mller and David Charypar and Markus Gross, *Particle-Based Fluid Simulation for Interactive Applications*, 2003

[18] WOJTAN, C., AND TURK, G. 2008. *Fast viscoelastic behavior with thin features.*, ACM Transactions on Graphics (SIGGRAPH) 27, 3, 47:147:8

[19] KIM, D., SONG, O.-Y., AND KO, H.-S. 2009. *Stretching and wiggling liquids.*, ACM Transactions on Graphics (SIGGRAPH Asia) 28, 5, 120:1120:7.

[20] YU, J., AND TURK, G. 2010. *Reconstructing surfaces of particle based fluids using anisotropic kernels.*, In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), 217225.

[21] WILLIAMS, B. 2008. *Fluid surface reconstruction from particles.*, Masters thesis, The University Of British Columbia.

[22] THUREY , N., WOJTAN, C., GROSS, M., AND TURK, G. 2010. *A multiscale approach to mesh-based surface tension flows.*, ACM Transactions on Graphics (SIGGRAPH) 29, 4, 48:148:10

[23] BOJSEN-HANSEN, M. 2011. *A Hybrid Mesh-Grid Approach for Efficient Large Body Water Simulation.*, Masters thesis, Aarhus University

[24] YU, J., WOJTAN, C., TURK, G., AND YAP, C. 2012. *Explicit mesh surfaces for particle based fluids.*, Computer Graphics Forum (Eurographics) 31, 2, 815824.

[25] WOJTAN, C., THUREY , N., GROSS, M., AND TURK, G. 2010. *Physics-inspired topology changes for thin fluid features.*, ACM Transactions on Graphics (SIGGRAPH) 29, 4, 50:150:8.

[26] Morten Bojsen-Hansen and Chris Wojtan. 2013. *Liquid surface tracking with error compensation.*, ACM Trans. Graph. 32, 4, Article 68 (July 2013), 13 pages. DOI=10.1145/2461912.2461991 http://doi.acm.org/10.1145/2461912.2461991

[27] Kooten,Gino van den Bergen Alex Telea *Chapter 7 :Point-Based Visualization of Metaballs on a GPU*, Eindhoven link: http://http.developer.nvidia.com/GPUGems3/gpugems3/ch07.html