Compact Iso-Surface Representation and Compression for Fluid Phenomena

Todd Keeler
University of British Columbia
Tolk Consulting

Robert Bridson
University of British Columbia
Autodesk

ABSTRACT

We propose a novel method of compressing a fluid effect for real-time playback by using a compact mathematical representation of the spatio-temporal fluid surface. To create the surface representation we use as input a set of fluid meshes from standard techniques along with the simulation's surface velocity to construct a spatially adaptive and temporally coherent Lagrangian least-squares representation of the surface. We then compress the Lagrangian point data using a technique called Fourier extensions for further compression gains. The resulting surface is easily decompressed and amenable to being evaluated in parallel. We demonstrate real-time and interactive decompression and meshing of surfaces using a dual-contouring method that efficiently uses the decompressed particle data and least-squares representation to create a view dependent triangulation.

CCS CONCEPTS

• **Computing methodologies** → *Physical simulation*; *Parametric curve and surface models*; Shape analysis;

KEYWORDS

Fluid Animation, Compression, Level-Of-Detail

ACM Reference format:
Todd Keeler and Robert Bridson. 2017. Compact Iso-Surface Representation and Compression for Fluid Phenomena. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/10.1145/3084363.3085080

1 INTRODUCTION

The output meshes of modern visual effects fluid simulations are highly detailed but temporally incoherent due to standard surfacing techniques for volumetric methods or mesh merging and refinement for ones employing explicit mesh tracking. The combination of large data footprint with incoherent triangulation makes these meshes prohibitive for real-time use cases such as games or virtual reality due to storage and bandwidth issues; naively, connectivity and point data must be retained for every frame.

While storage and bandwidth are of concern to real-time applications, adaptive reconstruction or data with varying levels of detail is also a primary concern for real-time and offline rendering schemes. Varying the level of detail of triangulated mesh data requires either creating new lower resolution meshes which create even more data, or unwieldy runtime decimation schemes.

In terms of fluid animation, static per-frame triangulation is inefficient since the fluid itself is a product of continuous physical forces and exhibits significant temporal coherency. In addition, the temporal incoherence of the output connectivity indicates that the triangle connectivity itself is of little importance to the utility of the surface, the object of primary importance is actually the geometry of the implicit surface – the mesh is just a discrete representation necessary for rendering.

We propose a novel method of compressing temporally coherent time dependent iso-surfaces. We create a compact mathematical representation of the spatio-temporal surface using simply a Lagrangian set of surface incident points and their normals which we refer to as LSQ points. These points are adaptively scattered on the fluid surface and then advected with the simulation's velocity. The surface representation is created by calculating a second order polynomial surface approximation at each point by explicitly minimizing the polynomial surface at neighbouring points. These per point approximations are then locally blended using a simplified least-squares approximation to create a smooth global surface.

We construct this surface representation using triangulated mesh output from standard fluid simulation techniques, with the addition of the values of the simulation's per-vertex surface velocity. The representation is spatially adaptive and temporally coherent. The spatial least squares approximation gives a super-quadratic form of lossy spatial compression while the temporal coherence allows us to apply additional compression by representing the time dependent data with a high-order interpolation called Fourier extensions [2014].

Creating the triangular mesh at run-time offers significant control over the level-of-detail being rendered. The first and simplest way is to simply control the detail level of the triangulation based on camera proximity. We demonstrate this using a simple dual-contouring technique which efficiently uses the decrypted point positions as the initial vertices in a marched triangulation of the surface.

In addition to this level-of-detail mechanism, we could achieve further run-time gains by dynamically controlling the point distribution, simply by dynamically choosing a subset of the initial LSQ point distribution. Our implicit surface construction is amenable to adaptive triangulation, as in Ju et al. [2002] and also amenable to screen space or frustrum conforming techniques, for example, that described by Müller et al. [2007].

Our unoptimized research implementation attains adaptive interactive playback by meshing tens of thousands of triangles per frame. The decompression and reconstruction of the surface and

SIGGRAPH '17 Talks, July 30 - August 03, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5008-2/17/07…$15.00
https://doi.org/10.1145/3084363.3085080

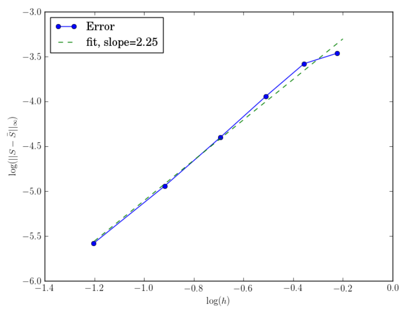

Figure 1: Convergence of the least squares representation on the surface of a torus. The log plot shows the error convergence with a linear fit for the approximate asymptotic order.

subsequent mesh generation is purely local and thus amenable to parallelization. We expect a gpu-compute implementation to make the playback and storage of standard visual effects fluid simulations feasible for interactive purposes such as games and virtual reality.

2 SURFACE REPRESENTATION

For each point-normal pair we define a local polynomial approximation Γ to the implicit surface,

$$\Gamma_i(\vec{x}) = d + \vec{n}_i \cdot \vec{r}_i + a\tau_{i1}^2 + b\tau_{i1}\tau_{i2} + c\tau_{i2}^2 \quad (1)$$

where a, b, c, d are computed geometric parameters, $\vec{r}_i$ is $\vec{x} - \vec{x}_i$ and τ_1 and τ_2 are the projection of $\vec{r}$ on unit tangent vectors $\vec{\tau}_{i1,2}$ that are orthogonal to each other and the normal at $\vec{x}_i$.

$$\tau_{i1} = \vec{r}_i \cdot \vec{\tau}_{i1}$$
$$\tau_{i2} = \vec{r}_i \cdot \vec{\tau}_{i2}$$

The first two terms of our reconstruction (1) constitute a planar approximation to the surface; in order for the zero contour to be at the point's position, d is zero.

Given a neighbouring set of LSQ points $\vec{x}_j$, we construct a (usually overdetermined) linear system for a, b, c

$$\Gamma_i(\vec{x}_j) = 0 = \vec{n}_i \cdot \vec{r}_{ij} + a\tau_{ij1}^2 + b\tau_{ij1}\tau_{ij2} + c\tau_{ij2}^2 \quad (2)$$

We solve the normal form of (2) and get the minimal norm solution. These local surface approximations are then blended to create a global representation of the surface. Figure 1 shows the super-geometric convergence of a Poisson distribution of LSQ points on an analytic torus.

3 COMPRESSION

Table 1 lists the compression rates for 240 frames of a smoothed cube undergoing rotation and translation. The amount of temporal compression is determined by setting the required temporal error. In our implementation, the Fourier extensions will reduce the number of interpolating basis functions until this error is surpassed for compressing the time dependent values of positions and normals of the LSQ points. The average error and maximum error show the normal distance from the original mesh vertices to the

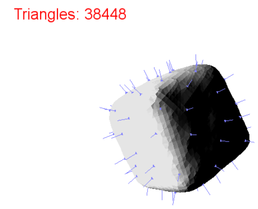

Figure 2: LSQ points representation of a moving, rotating smoothed box.

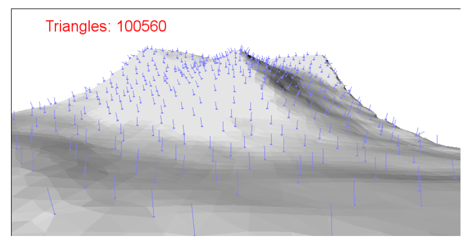

Figure 3: LSQ points representation of a fluid simulation created from commercial software.

approximating surface, the smoothed cube's length dimensions were approximately 2 units. Total, Spatial and No Compression (TC,SC,NC) list the number of words; ie the number of floats and integers, required to represent the mesh in its totally compressed form, only spatial compression using per Frame LSQ points , and using the original per frame vertex and face data directly.

RTE	avg err	max err	TC	SC	NC
.001	.0073	.071	54k	132k	3429k
.01	.0076	.070	38k	132k	3429k
.1	.014	.12	27k	132k	3429k

Table 1: Spatial and Temporal compression for a Moving Smoothed Cube:

4 RESULTS

Figure 2 shows a test case of a moving rotating smoothed box. The play-back applies an adaptive level of detail by marching with a smaller spatial step sizee when the box nears the camera position in real-time. Figure 3 shows an interactive replay of a compressed mesh created by a commercial visual effects software.

REFERENCES

Ben Adcock, Daan Huybrechs, and Jesús Martín-Vaquero. 2014. On the numerical stability of Fourier extensions. *Foundations of Computational Mathematics* 14, 4 (2014), 635–687.

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. 2002. Dual contouring of hermite data. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 339–346.

Matthias Müller, Simon Schirm, and Stephan Duthaler. 2007. Screen space meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 9–15.