Statistical Fluid Flow Synthesis

Mauricio Vines*　　　Won-Sook Lee†

University of Ottawa

ABSTRACT

We present our ongoing work on developing techniques for real-time simulation and visualization of fluids. We propose synthesizing fluid motion from simulation examples obtained with a full numerical solver. We segment our simulation data in order to capture the properties of local fluid behavior and apply statistical techniques to obtain the main modes of variation of the flow interacting with solid obstacles. Most research in fluid simulation has been focused on techniques for solving the Navier-Stokes equations for fluid motion, obtaining physically accurate simulations. We aim to obtain visually plausible flows by combining the results of physically accurate simulations preserving the general flow behavior. Our technique is suitable to be applied for visual effects design in movies as well as real-time applications such as videogames and virtual environments with haptic interaction, where efficient computations are required.

KEYWORDS: Fluid simulation, principal component analysis, visualization.

INDEX TERMS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1 INTRODUCTION

Simulating fluids remains one of the most challenging problems in computer graphics. Recent advances in computer applications demand more realism and interactivity. Several techniques have been developed to provide accurate and realistic fluid simulations, however, at a great computational cost. This is due to the interplay of different complex phenomena involved, such as advection, diffusion and turbulence, which are difficult to simulate. Fluid behavior is characterized by the well known Navier-Stokes equations which are a system of partial differential equations for which no analytical solution is known.

In computer graphics, the main goal is to achieve visual plausibility, even at the cost of physical accuracy. Several approximations to fluid behavior have been developed, including particle-based simulations [4] and height field approximations for liquid surfaces [3]. Applications of fluid simulations include visual effects in movies and videogames, as well as real-time interaction with virtual environments [1][7].

We propose a different method to efficiently obtain visually plausible simulations using fluid motion examples. These examples are obtained by running a standard fluid simulation under different flow conditions and the resulting motion data is spatially partitioned into different fragments. We apply principal component analysis (PCA) on the fragments and we employ this information to synthesize new fluid motion. Fluid motion databases have been used in the past for computing fluid forces on immersed solid objects [1], however we focus on visualizing the fluid behavior in real-time. Composing fluid behavior from a set of simulation components or *tiles* has been applied with model reduction [6]. Here each component contains a base of possible fluid velocities. In our scenario, we do not use model reduction for our simulations, but synthesize motion from examples obtained by numerically solving the Navier-Stokes equations.

* e-mail: mvine059@uottawa.ca
† e-mail: wslee@uottawa.ca

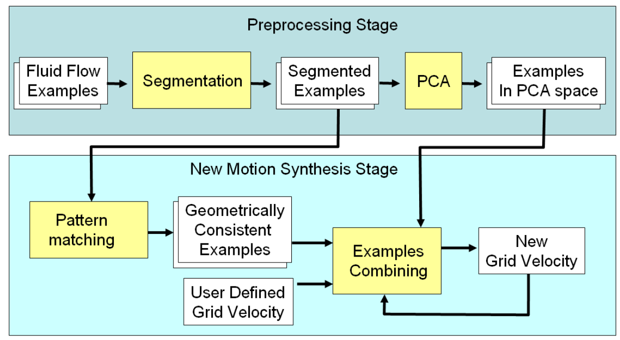

Figure 1. Schema of our approach

2 METHODOLOGY

In this section we discuss the two stages of our simulation framework: the construction of the examples database and the synthesis of new fluid motion from these examples. We represent the fluid using an Eulerian approach, both in the preprocessing and motion synthesis stages. The simulation space is divided into a regular grid of cells. In each cell the fluid velocity and pressure are stored. Solid objects are assumed to be static and they are voxelized as in [2], conforming to the simulation space configuration. In the preprocessing stage simulation examples are produced and analyzed. In the synthesis stage we find velocity values for each grid cell to approximate fluid behavior by combining the flow information from the simulation examples. A schema of our approach is shown in Figure 1.

2.1 Database construction

Fluid flow examples are obtained from simulating the fluid on the grid in the pre-processing stage. For simulating the fluid we employ a finite difference method based on operator splitting and semi-Lagrangian advection [5]. We run several simulation rounds for different solid configurations in order to have a rich set of fluid motion examples. In a simulation round the obstacle geometry is defined, and the fluid is simulated under different flow conditions, varying the direction and magnitude of the fluid velocity. Velocity values for each time step of simulation are stored for segmenting and statistical analysis.

Once the examples have been obtained, the results of each time step are segmented in fragments of equal size. A fragment consists on several grid cells, each storing velocity and pressure values. For each fragment we keep track of its neighboring fragments as well as the corresponding fragment in the next time step. As we consider only the case of static solids, the velocities in the grid cells occupied by a solid are zero. An example of the fluid examples generated and its fragments is shown in Figure 2.

We apply PCA on the fragments to capture the main modes of variation of the fluid velocity. We construct a covariance matrix from the fluid velocity data in each fragment and we obtain the eigenvectors and eigenvalues of this matrix. Then we transform our simulation data into the PCA space, this is, we represent the data in terms of the principal components, which are the eigenvectors with highest eigenvalues associated. The fragments spatial and temporal associations are stored, enabling to quickly determine the local evolution of the fluid for a specific fragment.

2.2 Flow synthesis

Synthesizing new fluid motion in the simulation grid is performed using the data from the examples and the current velocity in the grid. A specific velocity field may be defined by the user as initial conditions for the simulation, or as result of real-time interaction. The new simulation data is segmented into fragments the same way as the examples and the velocity data for each fragment is translated into the PCA space. For fragments that contain solid cells, or that share a boundary with a solid, a pattern matching step is performed in order to determine the best example fragments that resemble the specific local solid geometry.

Let f be a fragment in the new simulation and S the set of example fragments that are geometrically compatible with f. The fluid velocity in f is determined by finding a combination of examples such that it matches the behavior observed in f. Formally, let $e_1, e_2, \ldots, e_n$ be the first n principal components derived from the examples, and let $B=[b_1, b_2, \ldots b_n]^T$ be the vector of coefficients corresponding to f in the PCA space. For each example $s^i \in S$, let $C^i=[c^i_1, \ldots, c^i_n]^T$ be the vector of coefficients in the PCA space. Then synthesizing new motion reduces to solving the following system:

$$\begin{bmatrix} C^1 & \cdots & C^k \end{bmatrix} \cdot x = B \tag{1}$$

The coefficients of vector x indicate how the samples must be combined to match the current motion in the current time step. As the matrix in equation (1) may not be square, the system may have to be solved in the least squares sense. If no external force is added by the user, then solving the fluid velocity in f for the next time step consists on finding the corresponding fragments in S for the next time step and combining them using the coefficients in x. When an external force is added, it is necessary to solve the system in equation (1) as the fluid motion may have drastically changed. Notice that as we assume the solid objects are static, determining the set S of geometrically compatible fragment examples for each new fragment is performed just once at the beginning of the simulation. Using this simple technique we can simulate fluids interacting with static rigid solids.

3 Discussion

Our approach may be used to simulate fluids at different detail scales, depending on the size of the simulation fragments. Smaller fragments will provide more detailed information, at a higher computational cost. We must note that the flow results in general will not be mass-conserving in the strict sense. The mass conservation principle expresses that any change in fluid mass in a region depends only on the net flux of mass in that region. This is usually enforced as an additional constraint when solving the Navier-Stokes equations. Although the examples used to construct our database are mass-conserving, linear combinations of these examples may not. While this is not a big problem when simulating a gas, care must be taken when simulating liquids using our technique. Liquid simulation requires the proper definition of a free surface. A non-mass conserving fluid will appear as evaporating, hence losing mass. To properly simulate liquids using our technique, examples of the free surface have to be generated and statistically analyzed as well.

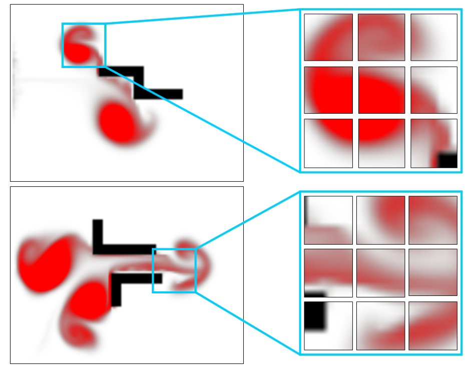

Figure 2. Frames of fluid simulation examples and a detail of fragments for each

We note as well that the different visual effects that can be obtained by this technique are limited to the fluid motion variations in the database. Extending the examples database would allow simulating more diverse solid-fluid interactions.

4 Conclusion

We present an approach to synthesize fluid motion from examples. Statistical analysis is performed on the examples and they are combined in order to approximate fluid behavior. This approach is suitable to be used for real-time applications such as videogames and interaction with virtual environments. In the future we aim to extend our method to properly simulate fluid interaction with moving solids. Also we aim to provide more animation control by allowing the user to specify the velocity in specific fragments in different locations and times, so fluid motion must be synthesized to match the user constraints.

References

[1] Y. Dobashi, M. Sato, S. Hasegawa, T. Yamamoto, M. Kato and T. Nishita. A fluid resistance map method for real-time haptic interaction with fluids. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology VRST'06* (Limassol, Cyuprus, November 1–3, 2006), pages 91–99, November 2006.

[2] N. Foster and D. Metaxas. Realistic animation of liquids. In *Graphical Models and Image Processing*, volume 58, issue 5, pages 471–483, 1996.

[3] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH Computer Graphics*, volume 24, issue 4, pages. 49-57, 1990.

[4] M. Müller, D. Charypar and M. Gross. Particle-Based Fluid Simulation for Interactive Applications. in *Proceedings of ACM SIGGRAPH Symposium on Computer Animation (SCA) 2003*, pages 154–159, 2003.

[5] J. Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques SIGGRAPH '99*, pages 121–128. ACM Press/Addison-Wesley Publishing Co. 1999.

[6] A. Treuille, A. Lewis and Z. Popović. Model reduction for real-time fluids. In *Transactions on Graphics*, volume 25, issue 3, pages 826–834. ACM Press, July 2006.

[7] M. Vines, J. Mora and W.-S. Lee. Haptic Display of 3D Liquids for Interactive Applications. In *IEEE Consumer Electronics Games Innovation Conference ICE-GIC 2009* (London, United Kingdom from August 25-28), 2009.