Haptic Interaction With Fluid Media

William Baxter Ming C. Lin

Department of Computer Science
University of North Carolina at Chapel Hill
{baxter,lin}@cs.unc.edu

Abstract

We present a method for integrating force feedback with interactive fluid simulation. We use the method described to generate haptic display of an incompressible Navier-Stokes fluid simulation. The force feedback calculation is based on the equations of fluid motion, and enables us to generate forces as well as torques for use with haptic devices capable of delivering torque. In addition, we adapt our fluid-haptic feedback method for use in a painting application that is based on fluid simulation, enabling the artist to feel the paint. Finally we describe a force filtering technique to reduce the artifacts that result from using 60Hz simulation data to drive the 1KHz haptic servo loop, a situation which often arises in practice.

Key words: Haptic Interaction. Fluid Simulation.

1 Introduction

The haptic display of information can add an extra sensory dimension to virtual reality applications. Much effort has been applied to the problem of haptic display of rigid body contact as well as compliant elastic contact, especially in the context of tissue modeling for surgical simulation [2]. Some progress has also been made in haptic rendering of textured surfaces [26, 31]. We propose perhaps for the first time to simulate the haptics due to interaction with fluid in a virtual environment.

We interact with fluids frequently in our everyday lives. Stirring a cup of coffee in the morning, taking a bath or going for a swim at the gym, or sticking one's hand out of the car window and feeling the air stream outside – all of these are examples of interaction with fluids, and in each case the force exerted by the fluid can be felt. We are motivated by these examples from everyday life to develop a computational technique which can be used in virtual reality applications to haptically display these kinds of scenarios to the user. Furthermore, though we focus on interactive fluid simulation, the force and torque computation technique we present is equally applicable to haptic exploration of data sets generated by offline CFD simulations. To compute forces and torques, all we require is the availability of pressure and velocity fields sampled on a grid.

We have been working on the simulation of oil- and acrylic-like viscous fluid paint media, using a variety of methods [3, 4, 5]. In our simulations we use a haptic stylus to interact with the fluid media. It has been an open question how to haptically display the paint to the user. In this paper we present one possible approach which is based directly on the simulation data.

Computers and numerical algorithms have reached the point where it is possible to simulate fluids, with certain restrictions, at interactive rates on modest-sized grids. Stam [32] was the first to demonstrate a Navier-Stokes fluid simulation at interactive rates by using a grid-based numerical method free from timestep restrictions. The method is still quite computationally expensive, so for interactive update rates of 30-100Hz, one must restrict the simulation to a 2D domain, with computational grids generally between 64×64 and 128×128 cells, or to a 3D grid with a similar number of total grid cells.

We propose to compute forces and torques from this type of simulation for use in haptic display so that users can *feel* the fluid, as well as see it, as they interactively affect the fluid. While the overall technique applies to either 2D or 3D, for explanatory purposes we describe only the 2D case in detail. The extension to 3D is fairly straightforward; the force computation just needs to be extended from a discretized line integral into a discretized surface integral. The additional cost of the haptic computation is small compared to the cost of the fluid simulation itself, since the force computation requires at most a $D - 1$ dimensional surface integral for a fluid simulation in D dimensions.

We focus on a simple, fixed-grid solver because more recent high-accuracy numerical methods that use adaptive grids, finite elements, or other techniques are too expensive for use in an interactive setting. If interactive modification of the fluid can be sacrificed, then any simulation method that can generate pressure and velocity on a grid is acceptable.

Finally, we demonstrate the use of our haptic feedback computation to simulate the feel of the viscous paint media in one of our interactive painting applications [5]. This application simulates the fluid in the vicinity of the brush using a small 3D grid.

2 Related Work

In this section, we survey previous work related to haptic rendering, applications of haptics, and force computation in simulated fluid-structure interaction.

2.1 Force Feedback

Several techniques have been proposed for integrating force feedback with real-time virtual environments to enhance the user's ability to perform interaction tasks [8, 22, 30].

Ruspini et al. [29] presented a haptic interface library "HL" that uses a virtual proxy and a multi-level control system to effectively display forces using 3-DOF haptic devices. Hollerbach et al. [15, 24] described a haptic display system for contact and manipulation in the CAD design of mechanical assemblies, and Thompson et al. [33] have presented a system for direct haptic rendering of sculptured models.

Durbeck et al. [9] have described a system for enhancing scientific visualization with haptic feedback. The force is computed by using a point-probe model and a simple vector-to-force magnitude mapping function. Unlike the method we describe, it does not compute actual aerodynamic or hydrodynamic forces, nor does it support volumetric probes. More recently, Lawrence et al. presented a haptic display technique using a 5-DOF force feedback device for shocks and vortices in hypersonic CFD datasets [21].

2.2 Volumetric Approaches

Gibson [11] proposed an algorithm for object manipulation including haptic interaction with volumetric objects and physically-realistic modeling of object interactions. The algorithms presented by Avila and Sobierajski [1] rely on interactive force feedback and rendering to allow a user to quickly explore and modify volumetric scenes.

Essentially, most present techniques for direct haptic rendering of volumetric data follow a general method, where the force display is defined as a vector valued function, or is transformed to one [1, 11, 16]. From this function, force feedback is generated by the data around the probe and from the velocity of the tip.

2.3 6-DOF Haptic Rendering

Iwata describes a 6-DOF haptic master and the concept of time-critical rendering at a lower update rate of hundreds of Hz [17]. Iwata and Noma also proposed methods for presenting volume data by force sensation using a 6-DOF force reflecting master manipulator with an update rate of a hundred Hz [16].

Recently [23] proposed "point-voxel sampling," a discretized approximation technique for contact queries that generates points on moving objects and voxels on static geometry. This approximation algorithm is the first to offer run-time performance independent of the environment's input size by basically sampling the object geometry at a resolution that the given processor can handle.

A recent approach proposed in [12] is limited to haptic display of object-object interaction for relatively simple models that can be easily represented as unions of convex pieces. [19] attempts to increase the stability of the force feedback using contact clustering, but their algorithm for contact queries suffers from the same computational complexity. [25] introduces a "*sensation preserving*" simplification algorithm for faster collision queries between two polyhedral objects in haptic rendering, thus achieving time-critical 6-DOF haptic rendering for highly complex models and contact configurations.

2.4 Fluid Force Computation

Foster and Metaxas [10] used a form of the hydrostatic force equations to create animations of rigid body objects in their offline Navier-Stokes simulation. Hydrostatics ignore the dynamic effects of fluid flow on the objects. Their simulation also did not account for the effect of the objects on the fluid.

Tu computed forces by using a boundary integral of the relative velocity between a fish's fin surface and surrounding fluid for simulating how swimming motions propel a fish[35]. Like [10] this method apparently does not take the influence of the immersed surface on the fluid, since at the actual surface of an immersed object, the physical boundary conditions demand that the velocity of the fluid relative to the surface is always zero, which would mean the force should always be zero as well with this method.

Ristow [27, 28] has created a number of offline simulations of spherical and elliptical particles falling through fluids using accurate force computations based on the fluid stress tensor. Our proposed method is based on the same equations but uses a different numerical procedure to compute the force. A further distinction is that our "particle" is an actively controlled haptic probe rather than a passively simulated object.

To the best of our knowledge, no one to date has used real-time fluid simulation to generate the force feedback to drive a haptic display, allowing the user to both feel and influence the fluid at the same time.

3 Preliminaries

The motion of an incompressible Newtonian fluid is described by the Navier-Stokes equations. The momentum equation describes the transport of momentum in the fluid:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{u} + \mathbf{F} \qquad (1)$$

where $\mathbf{u}$ is the fluid velocity field, p is the pressure field, ρ is the density, ν is the coefficient of kinematic viscosity, and $\mathbf{F}$ is an external force per unit mass such as supplied by gravity. We will also need the dynamic viscosity later, which is $\mu = \rho\nu$. Incompressibility adds an additional constraint, known as the continuity equation, which for a constant-density fluid can be written:

$$\nabla \cdot \mathbf{u} = 0 \qquad (2)$$

Appropriate boundary conditions are necessary as well. For a "no-slip" boundary condition on velocity and pure Neumann boundary condition on pressure, in a domain Ω with boundary $\partial\Omega$, these are

$$\begin{aligned} \mathbf{u} &= 0 \in \partial\Omega & (3)\\ \frac{\partial p}{\partial \mathbf{n}} &= 0 \in \partial\Omega & (4) \end{aligned}$$

In this section we present methods for computing a force for haptic feedback from a fluid simulation. Before presenting the accurate method, we first look at some common force approximations used in fluid dynamics.

3.1 Approximated Fluid Forces

For rough aerodynamic calculations of lift and drag forces, a few simple approximations are commonly used. One standard equation used to approximate the aerodynamic drag force on an object with relative airspeed v [6] is

$$F_{\text{AD}} = C_{\text{AD}}\ \rho v^2 A, \qquad (5)$$

where A is the projected cross-sectional area of the object, C_{AD} is an aerodynamic drag coefficient which depends upon the shape and material properties of the object, and ρ is the fluid density. this approximation is only applicable to low-viscosity, and low vorticity situations such as the laminar air stream in a wind tunnel.

For high-viscosity, or slow flow scenarios in which the Stokes approximation holds, one can obtain a similar expression for the viscous drag force[20]:

$$F_{\text{VD}} = C_{\text{VD}}\ \mu v A \qquad (6)$$

where C_{VD} is a coefficient of viscous drag. This type of simple linear relation between force and velocity has been used quite extensively in haptics to simulate viscous or dynamic friction, because it is simple, computationally inexpensive, and quite stable. In such uses, the source of the viscous force is generally taken to be at rest so that the probe's velocity is the overall relative velocity.

Both of these approximations make the assumption that the velocity field can be characterized by a single vector, so these methods can only generate forces, not torques. If we choose a single sample velocity from the flow field, the method becomes similar to that used by many haptic volume display systems, where a vector field is haptically rendered using a force proportional to the vector at the probe location.

One can use these equations for approximating the fluid force on a point probe, but difficulty arises if you wish to also interact with the fluid via the same point probe. If interaction is implemented by simply injecting velocity into the simulation at the probe location, as is common, then this injected velocity becomes a large part of what gets sampled and the resulting force is incorrect. A better, more physically correct approach is to cause the fluid to move via boundary conditions on a finite probe. This will be discussed more in Section 4.

3.2 Accurate Force Computation

The proper description of the internal forces in a viscous incompressible ideal fluid is given by the stress tensor, $\boldsymbol{\sigma}$, which in index notation is given by [20]

$$\sigma_{ik} = -p\delta_{ik} + \mu\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right). \qquad (7)$$

The force per unit area at a given point in the fluid, $\mathbf{x}$, on an infinitesimal area dA with normal $\mathbf{n}$ is given by:

$$\mathbf{P} = \boldsymbol{\sigma}(\mathbf{x})\cdot\mathbf{n}; \qquad (8)$$

The net force acting on a closed object submerged in the fluid can then be obtained by integrating this expression over the surface of the object:

$$\mathbf{F}_{\text{obj}} = \int_S \boldsymbol{\sigma}\cdot\mathbf{n}\,dA. \qquad (9)$$

In terms of vector components, $\mathbf{P}$ is given by

$$P_i = (\boldsymbol{\sigma}\cdot\mathbf{n})_i = -pn_i + \mu\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)n_k \qquad (10)$$

or, for a two dimensional viscous flow,

$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} -p_x n_x + \mu\left(2\frac{\partial u_x}{\partial x}n_x + (\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x})n_y\right) \\ -p_y n_y + \mu\left(2\frac{\partial u_y}{\partial y}n_y + (\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x})n_x\right) \end{pmatrix} \qquad (11)$$

Similarly, the torque on the object about the point $\mathbf{c}$ is given by integrating the cross product

$$\boldsymbol{\tau}_{\text{obj}} = \int_S \mathbf{r}\times\boldsymbol{\sigma}\cdot\mathbf{n}\,dA. \qquad (12)$$

where $\mathbf{r}$ is the vector from $\mathbf{c}$ to each infinitesimal surface element. For computing the haptic force, $\mathbf{c}$ acts as the point of attachment of the haptic stylus to whatever probe geometry is desired. In two dimensions where torque is a scalar, we have:

$$\tau_{\text{obj}} = \int_S (r_x P_y - r_y P_x)\, dA. \tag{13}$$

Equations 9 and 12 tell us everything we need to know to generate haptic feedback from a fluid solution. The next step is to determine an appropriate discretization of these equations in order to evaluate them numerically on a computational grid, which will be discussed in the next section.

4 Haptic Display of Fluid Flows

There are numerous ways to discretize the Navier-Stokes equations, with various strengths and weaknesses. For interactive purposes we desire a simulation method that generates results with the lowest computational cost, even if this means we must sacrifice some accuracy. For interactivity, it is also good to select a method free from timestep restrictions. Many solution methods require for stability that the simulation timestep be selected such that a CFL condition such as $\Delta t\ \mathbf{u}_{i,j} < \Delta \mathbf{x}$ holds for all (i, j) on the discretized grid. Even worse are timestep restrictions on viscosity which typically require $\Delta t < O(\Delta x^2 \nu^{-1} \|\mathbf{u}_{i,j}\|^{-1})$. Even if the basic time-stepping procedure is fast, the number of steps required to advance the simulation by a desired time increment ΔT can be prohibitive, and cause the simulation clock to fall behind the wall clock time.

A better approach for interactive fluid simulation is that presented by Stam et. al. [32]. By using an implicit time-stepping procedure to handle viscosity, and a semi-Lagrangian scheme for advection, all stability restrictions on the timestep are removed, and thus it is possible to prevent the simulation clock from falling behind the wall clock.

The approach is an operator splitting method with a Chorin projection [7] to solve for the pressure. The differential operator is approximated by splitting it in time so that the advection term, viscosity term, and pressure terms are handled sequentially instead of simultaneously. The incompressibility constraint is ignored for the first substeps, and then the resulting flow field is projected onto the space of divergence-free vector fields to satisfy the continuity equation. This requires the solution of a discretized Poisson problem. The overall procedure is first-order accurate using first-order discretized operators.

We have implemented the solver for the most part like Stam, except we use the conjugate gradient method with an incomplete Cholesky preconditioner to solve the sparse linear systems of equations that arise from the implicit viscosity step and the Poisson equation for pressure [18]. Also we use a staggered grid rather than a cell-centered grid. A typical grid cell is shown in Fig. 1.

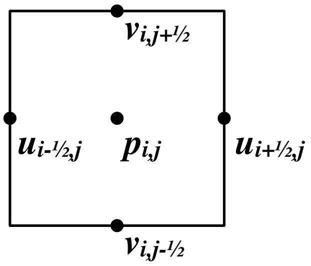

Figure 1: A typical grid cell, at grid location (i, j).

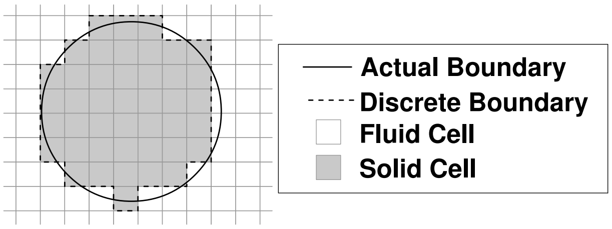

Figure 2: Boundary discretization used for incorporating the haptic probe geometry into the grid-based simulation.

To discretize the boundary of the haptic probe, we use the common first-order approach of treating each grid cell as either entirely fluid or entirely solid, thus snapping the actual boundary shape to the nearest grid cell edges as shown in Figure 2. This is the same approach as proposed in the original Marker-and-Cell method [14].

To incorporate the motion of the haptic probe into the simulation, the u and v velocity boundary values are set along the perimeter of the discretized probe boundary on the grid. For a north facing edge, for example, we enforce the no-slip boundary condition by setting

$$\begin{aligned} v_{i,j+\frac{1}{2}} &= v_{\text{probe}}\left(i\Delta x, (j+\tfrac{1}{2})\Delta y\right) && (14)\\ u_{i\pm\frac{1}{2},j} &= 2u_{\text{probe}}\left((i\pm\tfrac{1}{2})\Delta x, j\Delta y\right) - \\ &\quad\ u_{i\pm\frac{1}{2},j+1} && (15) \end{aligned}$$

where the velocity at index $j+1$ indicates an average of the values at $j-\frac{1}{2}$ and $j+\frac{1}{2}$. Similar expressions are used for the other cell edges. See [13] for more details on boundary conditions in this type of fluid simulation.

4.1 Force Computation

After an iteration of the numerical fluid solver, we have values for p, and and $\mathbf{u} = (u, v)$ on the grid. To compute the force and torque on the probe at this point we simply need to traverse the boundary and and approximate the boundary integrals Eqs. 9 and 12, using a discrete Riemann sum. For a north facing edge, for instance, the normal pointing in toward the probe is $\mathbf{n} = (0, -1)$ and the contributions to the total force and torque obtained by using this $\mathbf{n}$ in Eqs. 9-13 are

$$\begin{aligned} \begin{pmatrix} P_x \\ P_y \end{pmatrix} &= \begin{pmatrix} \mu \frac{v_{i,j+3/2} - v_{i,j+1/2}}{\Delta x} \\ -p + 2\mu \frac{v_{i,j+3/2} - v_{i,j+1/2}}{\Delta y} \end{pmatrix} && (16) \\ \mathbf{F}_i &= \Delta x \begin{pmatrix} P_x \\ P_y \end{pmatrix} && (17) \\ \tau_i &= \Delta x \left(r_x P_y - r_y P_x \right). && (18) \end{aligned}$$

Similar expressions are obtained for the remaining three edge orientations. After traversing the entire boundary of the probe, we have that $\mathbf{F}_{\text{obj}} = \sum_i \mathbf{F}_i$ and $\tau_{\text{obj}} = \sum_i \tau_i$.

4.2 Force Filtering

The above method does suffice for calculating the force from the flow, but generally in the simulations of interest–a 64×64 grid or larger–the fluid simulation cannot execute at the 1KHz rate desired for smooth haptic rendering. 40-70Hz is a more typical simulation rate for such a grid. For general haptics, involving the display of rigid contact with rigid surfaces, any sort of smoothing filters can lead to an undesirable softening of the surfaces by removing the high frequency components of the force displayed. However, for simulating the feel of a fluid, the high frequency components contain much less energy, and we have found some amount filtering to be both acceptable and beneficial in reducing the artificial stair-stepping and vibration that comes from updating the haptic output at less than the desired 1KHz rate.

In selecting a filter we desired a filter which would respond to a step function input by smoothly transitioning both away from the previous state and into the new state, to avoid introducing high frequency artifacts. Finite impulse response (FIR) window filters match these desired characteristics well. We selected a 10th order Bartlett-Hanning window using the Matlab signal toolbox. Our results indicate that filtering the force in this manner yields a notable improvement in the smoothness of the haptic feedback for our application. The filter's smoothing effect on an input signal can be seen in Fig. 3. Although the filtering does introduce some delay into the force response time, we did not find this to be noticeable in the case of fluid force display.

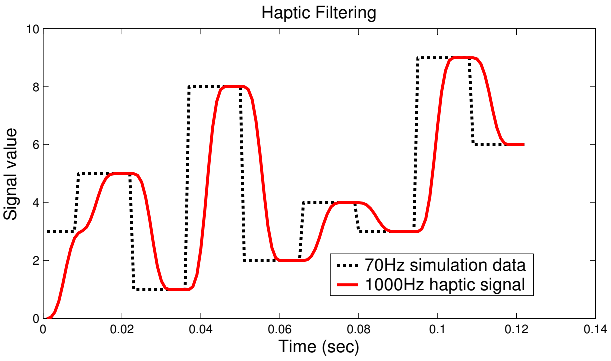

Figure 3: The results of filtering a random 70Hz input signal for haptic display at 1KHz.

5 Results and Discussion

We have implemented the force feedback method as described on a 2GHz Pentium IV computer with a Phantom haptic interface from SensAble Technologies, Inc. For force only, we used a Desktop Phantom model with 3DOF of force feedback. For force and torque, we used a Premium-A 6-DOF model Phantom.

5.1 Performance

In our implementation we control the final scaling of the force with a user-defined parameter. This parameter needs to be chosen carefully since too low a value will make the fluid forces too weak, but too high a value can add excessive energy back to the haptic probe which is then fed back to the fluid leading to unstable oscillations. The scale used in our simulation is typically in the range of 2-10. This is for a unit domain of $[0, 1] \times [0, 1]$ at 64×64 resolution with a Reynolds number of about 10^3.

Figure 6 shows a few frames from a fluid interaction sequence in which a square probe is swept along a path. In Fig. 4 we show the force as computed by our simulation method along that path. The square begins at rest, accelerates as it sweeps out an arc, then comes to rest. Then it reverses direction and traces the same path backwards. At the same time the square spins with an angular velocity that has a similar profile of acceleration. We have split the 2D force into two components, one the drag force, tangential to the motion of the probe, and the other component is perpendicular to the motion.

The force and torque profiles can be seen to contain some amount of noise. This jitter comes from the discretization of the boundary. Fig. 5 shows how the noise is reduced when we sweep along the same path repeatedly at different grid resolutions. Although the simulation does not run interactively at the higher resolutions, the reduction in noise indicates that boundary discretiza-

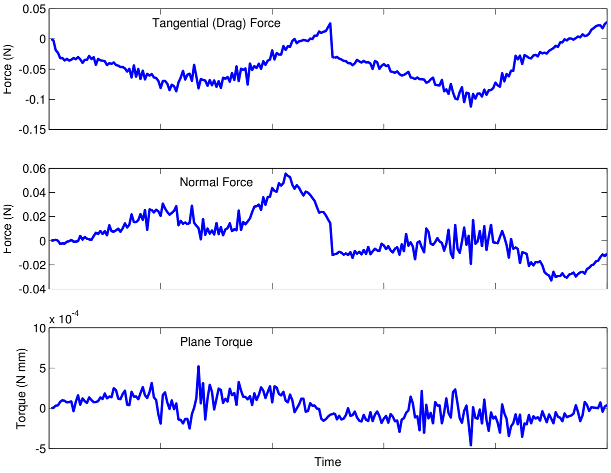

Figure 4: Unfiltered haptic feedback forces generated by our method for a square probe dragged and twisted along a path. Our haptic filtering method smooths these values for the final force display. The object first accelerates from rest, decelerates to a stop at the middle of the graph, and then returns to its starting point (see Fig. 6). The drag force graph (top) clearly shows the fluid force working against the direction of motion, as expected.

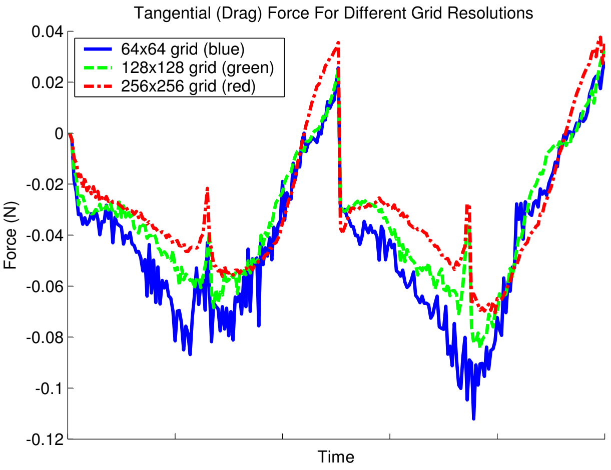

Figure 5: The effect of grid resolution on the force computed. The noise in the signal is reduced with each increase in grid resolution from the coarsest grid (blue) to the finest grid (red).

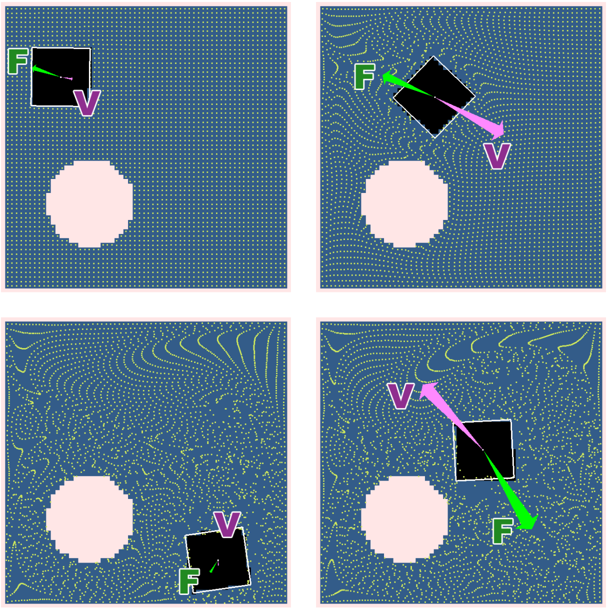

Figure 6: A sequence of images captured from our interactive fluid simulation with haptic rendering, showing both the instantaneous force (green arrows) and velocity vectors (magenta arrows) at several instants along the probe's path.

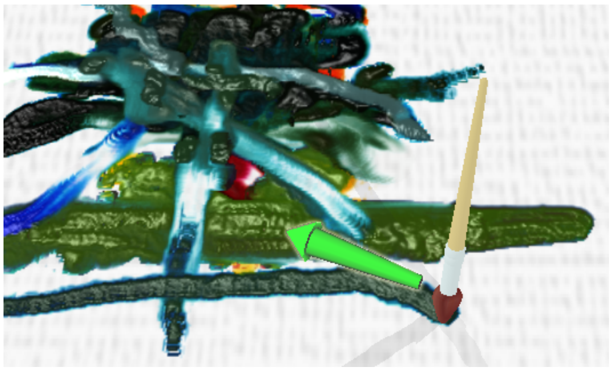

Figure 7: This is a screen capture from a modified painting application that uses our method to display the feel of the viscous paint to the user. The green arrow shows the force being displayed as the user paints a stroke from left to right.

tion is the major source of the noise. Higher order numerical treatments of the boundary have been developed, such as [34], and these would likely reduce the noise in the force signal.

To demonstrate the practical utility of our method in virtual applications we have also integrated the force computation into a painting simulation application that is based on viscous fluid flow equations [5]. Using our method, a user is able to feel the viscosity of the the paint as he or she makes brush strokes using the haptic stylus. See Figure 7.

5.2 Limitations and Observations

Given the hardware available today, there are few numerical techniques which can achieve the desired interactive fluid update rates of 30-60Hz. Most recent numerical techniques are designed for accuracy rather than interactivity, thus we use an older simple, fixed-grid method. Even with this fluid simulation technique, grid sizes are limited to approximately 128×128 cells in 2D, or an equivalent total number of cells in 3D. Such grid sizes do somewhat limit the applicability of interactive haptic fluid computation currently.

The computational cost of the fluid simulation places an especially restrictive limitation on 3D grid sizes; however, a full 3D simulation is not always necessary to generate convincing results in a 3D virtual environment. For some scenarios a 2D simulation can be used by treating the 2D fluid as a volume with the same velocity at every depth, similar to the depth-averaged Navier-Stokes method that leads to the shallow water equations [36].

Another issue with the technique presented in this paper is that the simple boundary discretization used leads to pressure artifacts around the boundary, which turn into oscillations in the force. The pressure artifacts come from the discontinuities when cells on the boundary instantly change from one state to another as the moving boundary passes through them. A higher-order numerical scheme for discretizing boundaries, such as the GENSMAC method [34], might eliminate this problem. We plan to investigate this issue further.

6 Summary Work

We have presented a technique for computing force feedback from Navier-Stokes fluid simulations. This technique can be used along with fluid simulation to add an extra dimension to virtual reality applications. As an example we incorporated the technique into an existing painting application allowing the user to feel the liquid paint.

In the future we are interested in experimenting with higher-order boundary treatments, and we are also interested in exploring alternate non-grid algorithms for the fluid simulation such as smoothed particle hydrodynamics (SPH). Finally, we are interested in further investigating the use of this haptic feedback technique in haptic visualizations of higher-accuracy offline CFD simulations.

Acknowledgments

This research is supported in part by NSF IIS-9821067, ONR N00014-01-1-0067, and Intel. William Baxter is also supported by fellowships from The Link Foundation and NVIDIA.

References

[1] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. *Proceedings of Visualization'96*, pages 197–204, 1996.

[2] N. Ayache and H. Delingette. *Surgery Simulation and Soft Tissue Modeling*. Springer, 2003.

[3] W. Baxter, V. Scheib, M. Lin, and D. Manocha. DAB: Haptic painting with 3D virtual brushes. *Proc. of ACM SIGGRAPH*, pages 461–468, 2001.

[4] William Baxter, Jeremy Wendt, and Ming Lin. IMPaSTo: A realistic, interactive model for paint. *The 3rd International Symposium on Non-Photorealistic Animation and Rendering*, 2004.

[5] William Baxter, Liu Yuanxin, and Ming Lin. A viscous paint model for interactive applications. Technical Report TR-04-06, University of North Carolina at Chapel Hill, 2004.

[6] John Bloomer. *Practical Fluid Mechanics for Engineering Applications*. Marcel Dekker, 2000.

[7] A. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[8] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. *IEEE Conference on Robotics and Automation*, pages 3205–3210, 1994.

[9] L. Durbeck, N. Macias, D. Weinstein, C. Johnson, and J. Hollerbach. SCIRun haptic display for scientific visualization. *Phantom Users Group Meetings*, 1998.

[10] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing: GMIP*, 58(5):471–483, 1996.

[11] S. Gibson. Beyond volume rendering: Visualization, haptic exploration, and physical modeling of element-based objects. In *Proc. Eurographics workshop on Visualization in Scientific Computing*, pages 10–24, 1995.

[12] A. Gregory, A. Mascarenhas, S. Ehmann, M. C. Lin, and D. Manocha. 6-DoF haptic display of polygonal

models. *Proc. of IEEE Visualization Conference*, 2000.

[13] Michael Griebel, Thomas Dornseifer, and Tilman Neunhoeffer. *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. SIAM Monographcs on Mathematical Modeling and Computation. SIAM, 1990.

[14] Francis. H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, December 1965.

[15] J. Hollerbach, E. Cohen, W. Thompson, R. Freier, D. Johnson, A. Nahvi, D. Nelson, T. Thompson, and S. Jacobsen. Haptic interfacing for virtual prototyping of mechanical cad designs. In *ASME Design for Manufacturing Symposium*, 1997.

[16] H. Iwata and N. Noma. Volume haptization. *Proc. of IEEE VRAIS*, pages 16–23, 1993.

[17] Hiroo Iwata. Artificial reality with force-feedback: Development of desktop virtual space with compact master manipulator. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 165–170, August 1990.

[18] David S. Kershaw. The incomplete cholesky–conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26:43–65, 1978.

[19] Y. Kim, M. Otaduy, M. Lin, and D. Manocha. 6-DOF haptic display using localized contact computations. *Proc. of Haptics Symposium*, pages 209–216, 2002.

[20] L.D. Landau and E.M. Lifschitz. *Fluid Mechanics*. Pergamon Press, 1975.

[21] D. A. Lawrence, C. D. Lee, L. Y. Pao, and R. Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. *Proc. of IEEE Visualization*, pages 131–137, 2000.

[22] T. M. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. *Proc. of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1:295–301, 1994.

[23] W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.

[24] A. Nahvi, D. Nelson, J. Hollerbach, and D. Johnson. Haptic manipulation of virtual mechanisms from mechanical CAD designs. In *Proc. of IEEE Conference on Robotics and Automation*, pages 375–380, 1998.

[25] M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. *Proc. of ACM SIGGRAPH*, 2003.

[26] Miguel A. Otaduy, Nitin Jain, Avneesh Sud, and Ming C. Lin. Haptic rendering of interaction between textured objects. Technical Report TR-04-07, University of North Carolina at Chapel Hill, 2004.

[27] Gerald H. Ristow. Particles moving in spatially bounded,viscous fluids. *Computer Physics Communications*, pages 43–52, 1999.

[28] Gerald H. Ristow. Tumbling motion of elliptical particles in viscous two-dimensional fluids. *International Journal of Modern Physics C*, 11(0), 2000.

[29] D.C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. *Proc. of ACM SIGGRAPH*, pages 345–352, 1997.

[30] K. Salisbury, D. Brock, T. Massie, N Swarup, and C. Zilles. Haptic rendering: Programming touch interaction with virtual objects. *Proc. of 1995 ACM Symposium on Interactive 3D Graphics*, pages 123–130, 1995.

[31] Juhani Siira and Dinesh K. Pai. Haptic texturing - a stochastic approach. *Proceedings of the Int'l Conf. on Robotics and Automation*, 1996.

[32] Jos Stam. Stable fluids. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 121–128, Los Angeles, 1999. Addison Wesley Longman.

[33] T.V. Thompson, D. Johnson, and E. Cohen. Direct haptic rendering of sculptured models. *Proc. of ACM Interactive 3D Graphics*, pages 167–176, 1997.

[34] Murilo F. Tome and Sean McKee. GENSMAC: A computational marker and cell method for free surface flows in general domains. *J. Comp. Phys.*, 110:171–186, 1994.

[35] Xiaoyuan Tu. *Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behavior*. PhD thesis, University of Toronto, January 1996.

[36] C. B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. Kluwer Academic Publishers, 1994.