Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2006)
M.P. Cani, J. O'Brien (Editors)

Detail-Preserving Fluid Control

N. Thürey[1,2], R. Keiser[1], M. Pauly[1] and U. Rüde[2]

[1] Applied Geometry Group, ETH Zurich, Switzerland
[2] System Simulation Insitude (LSS), University Erlangen-Nuremberg, Germany

Abstract

We propose a new fluid control technique that uses scale-dependent force control to preserve small-scale fluid detail. Control particles define local force fields and can be generated automatically from either a physical simulation or a sequence of target shapes. We use a multi-scale decomposition of the velocity field and apply control forces only to the coarse-scale components of the flow. Small-scale detail is thus preserved in a natural way avoiding the artificial viscosity often introduced by force-based control methods. We demonstrate the effectiveness of our method for both Lagrangian and Eulerian fluid simulation environments.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling, I.3.7 [Computer Graphics]: Animation, I.6.8 [Simulation and Modeling]: Animation

1. Introduction & Related Work

Physics-based fluid simulation has recently achieved stunning results that are increasingly indistinguishable from their real-world counterparts (see e.g. [CMT04, LGF04, KLLR05, LIGF06]). While realism is one important aspect, the use of fluid simulation for animation is also greatly determined by the ability to efficiently control the behavior of the fluid. Since fluid motion is typically very hard to predict, a high burden is placed on an animator who tries to achieve a specific flow animation by changing simulation or scene parameters only. In many cases, accurate physical behavior is not even desired, e.g., when animating a 'fluid character', as long as the characteristic properties of a fluid are conveyed in a plausible manner. High-level control is thus crucial in a production environment, where animators are mostly interested in modifying the large-scale motion of the fluid, while the simulation should take care of fine-scale detail such as small eddies or drops.

Our control paradigm is based on the concept of control particles, similar to [FF01]. Since control particles are independent of the underlying fluid model they can be integrated easily in different flow simulation environments. In addition, many different control scenarios can be implemented, such as scripting, keyframing, or coarse-to-fine simulations. We show how control particles can be automatically generated from pre-computed functions, an animated target shape, or an existing flow simulation.

Figure 1: *A controlled fluid character.*

In previous methods, control particles directly influence the fluid velocity field, which can cause noticeable smoothing effects, e.g., when fluid velocities are forced to align with control velocities that are specified on a much coarser scale. To avoid this artificial viscosity, we decompose the velocity field into coarse- and fine-scale components and only apply control forces to the low-frequency part. High-frequency components are largely unaffected, thus small-scale detail and turbulence are significantly better preserved (see Fig-

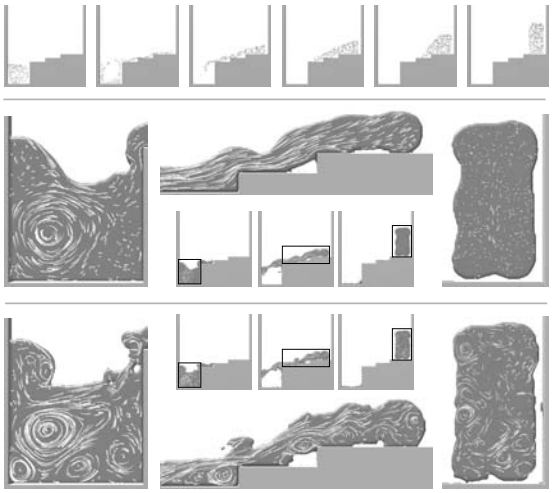

Figure 2: *Comparison of direct velocity control (middle) and scale-separated velocity control (bottom). Both simulations are controlled by the same 250 control particles (shown in the top row), that were generated by reversing an uncontrolled simulation of fluid flowing down the stairs.*

ure 2). Similar techniques are used in e.g. geometry processing or motion capture editing. We achieve this decomposition by smoothing the velocity field using a low-pass filter that is adapted to the influence kernels of the control particles. Velocity control forces are then computed with respect to the smoothed velocity field. Our results show that with this scale-separated fluid control, fluid detail is much better preserved, which results in more dynamic and realistic looking simulations. We demonstrate the versatility of the approach on a grid-based (lattice Boltzmann method) and a particle-based (Smoothed Particle Hydrodynamics) simulation.

Foster and Metaxas were among the first who dealt with fluid control. In [FM97] they propose to embed controllers for pressure and velocity to control the flow as well as the fluid surface. This concept is further extended in [FF01] by sampling 3D parametric space curves with oriented points to locally alter the velocity of the fluid. Lamorlette and Foster [LF02] use parametric space curves to model flames. These curves evolve according to physics-based, procedural, and hand-defined wind fields. [FOA03] already demonstrates the capabilities of particle based fluid control for animating explosions. Rasmussen et al. [REN*04] introduce viscosity, velocity divergence and level set particles for melting, expansion and contraction of the liquid. In [TMPS03] Treuille et al. presented an optimization technique to solve for the control parameters such that simulated smoke matches the user's density and velocity keyframes. The efficiency is greatly improved by adopting the adjoint method for solving the nonlinear optimization problem in [MTPS04]. [PCS04] demonstrates an approach that makes use of radial basis functions to control flow simulations. Fattal and Lischinski [FL04] proposed the idea of driving smoke towards target smoke density states by introducing a force term and counteract diffusion of smoke by adding a gathering term to the Euler equations. This simple technique is much faster than solving an optimization problem, while the results are comparable to the solution found using an optimization technique. Hong and Kim [HK04] derive potential fields from the initial distribution of smoke and a target shape. The force field is then defined as the gradient of this potential field. Shi and Yu control both smoke [SY05a] and liquids [SY05b] by matching the level set surface of the fluid with static or moving target shapes. Velocity constraints at the boundary drive the fluid to the desired shape. While for smoke a compressible fluid model is used [SY05a], the velocity needs to be divergence-free for liquids to guarantee volume preservation [SY05b]. This is achieved by changing the boundary forces such that the flux at the boundary is zero, yielding a constraint minimization problem.

2. Fluid Simulation Models

We briefly discuss the two fluid simulation models, smoothed particle hydrodynamics and the lattice-Boltzmann method, that we use to demonstrate our control method.

Smoothed Particle Hydrodynamics (SPH) is an approximation method that can be used to numerically solve the Navier-Stokes (NS) equations. The fluid is sampled with particles that serve as interpolation points. A fluid property $A(\mathbf{x})$ at an arbitrary position $\mathbf{x}$ in space is computed from the fluid properties A_j of the neighboring particles p_j as

$$A(\mathbf{x}) = \sum_j A_j V_j W(\mathbf{x} - \mathbf{x_j}, h), \tag{1}$$

where V_j and $\mathbf{x_j}$ are the volume and position of p_j, respectively. The normalized kernel function W depends on the distance of a particle to $\mathbf{x}$ and a length scale h. Usually, kernels similar to a Gaussian but with compact support are chosen for computational efficiency reasons. For detailed descriptions of the SPH method we refer to [Mon05, LL03]. For our SPH simulations we used the framework presented by Müller et al. [MCG03].

The Lattice-Boltzmann Method (LBM) is a grid-based technique that was derived from discrete gas molecule simulations. Each grid cell stores a set of *distribution functions* (DFs), which represent an amount of fluid moving with a fixed velocity. We use the common three-dimensional LBM model D3Q19 with nineteen grid velocities $\mathbf{e}_i, i = 1...19$. The macroscopic fluid properties, such as density ρ and velocity $\mathbf{v}$ can be calculated by summation of the DFs, and are needed to calculate the equilibrium DFs for a cell:

$$f_i^{eq}(\rho, \mathbf{v}) = w_i \left[\rho + 3\mathbf{e_i} \cdot \mathbf{v} - \frac{3}{2}\mathbf{v}^2 + \frac{9}{2}(\mathbf{e_i} \cdot \mathbf{v})^2 \right]. \tag{2}$$

For the D3Q19 model, the weights w are given as $w_1 = 1/3$, $w_{2..7} = 1/18$, and $w_{8..19} = 1/36$. The LBM algorithm proceeds by first handling the movement of the DFs, which is

equivalent to copying them to their adjacent neighbors, and then computes the molecular collisions that would have occurred during this movement. The collisions are handled by a relaxation towards the equilibrium with a relaxation factor ω that is given by the physical fluid viscosity:

$$f_i(\mathbf{x}, t+\Delta t) = (1-\omega) f_i(\mathbf{x}-\mathbf{e}_i, t) + \omega f_i^{eq}. \qquad (3)$$

More details of the algorithm and a derivation of the model can be found in [HL97, YMLS03]. To simulate a fluid with a free surface we use a method similar to the *Volume-of-Fluid* approach for NS solvers [KPR*05, TRK05].

3. Fluid Control

Our control framework uses a set of particles that locally exert forces on the fluid. Control particles are generated using an implicit function, a sequence of target shapes, or another fluid simulation. They directly induce forces to attract the fluid or influence the velocity field. We will demonstrate that these two forces can be used for a wide variety of effects. Furthermore, we will show how the small-scale detail can be retained by applying the control forces only on the coarse flow of the fluid.

3.1. Generating Control Particles

We employ three methods to generate control particles. The easiest way to create control particles is with a given precomputed function as described in [FM97, FF01]. To perform fluid simulations with a given target shape, we first regularly sample the initial control shape mesh. The control particles are then displaced for each mesh of the animation sequence using mean values coordinates [JSW05]. We can also generate control particles from another fluid simulation. For Lagrangian methods such as SPH, a subset of the fluid particles can be used as control particles. For grid based algorithms, such as LBM, massless particles can be traced in the fluid velocity field. This control simulation can usually be very coarse, and may even run in realtime to give instant feedback to an animator. It can likewise be controlled to yield the desired result. Control particle sets generated by a fluid simulation can be used to easily control larger volumes of fluid, and by changing or reversing the timing of the control particles interesting effects can be achieved. Both Figure 2 and Figure 4 make use of this approach.

3.2. Control Forces

A control particle p_i is given by its position $\mathbf{p}_i$, velocity $\mathbf{v}_i$, and influence radius h_i. We choose a constant radius h as 2.5 times the average sample distance of the control particles.

Attraction Force. Fluid attraction is controlled using a force that pulls fluid towards the control particles. In order to preserve as much of the natural fluid behavior as possible, this force is scaled down when the influence region of the control particle is already sufficiently covered with fluid. Let V_e denote the volume of a fluid element e (a filled grid cell for LBM, a mollified particle for SPH). We define a scale factor for the attraction force as

$$\alpha_i = 1 - \min\left(1, \sum_e V_e W(d_{i,e}, h)\right), \qquad (4)$$

where $d_{i,e} = \|\mathbf{p}_i - \mathbf{x}_e\|$ is the distance between p_i and the center $\mathbf{x}_e$ of fluid element e, and W is the normalized spline kernel with support h [MCG03]:

$$W(d,h) = \begin{cases} \frac{315}{64\pi h^9}(h^2-d^2)^3 & : \quad d < h \\ 0 & : \quad d \geq h. \end{cases} \qquad (5)$$

Summing up the attraction forces exerted by control particles p_i on a fluid element e then yields

$$\mathbf{f}_a(e) = w_a \sum_i \alpha_i \frac{\mathbf{p}_i - \mathbf{x}_e}{\|\mathbf{p}_i - \mathbf{x}_e\|} W(d_{i,e}, h), \qquad (6)$$

where w_a is a global constant that defines the strength of the attraction force. If w_a is negative, Equation (6) will result in a repulsive force.

Velocity Force. While the attraction force pushes fluid towards control particles that are not covered with fluid yet, we use a second force to modify the velocity of the fluid according to the flow determined by the control particles. We define a velocity force per volume $\mathbf{f}_v(\mathbf{x})$ similar to the attraction force

$$\mathbf{f}_v(e) = w_v \sum_i (\mathbf{v}_i - \mathbf{v}(e)) W(d_{i,e}, h), \qquad (7)$$

where $\mathbf{v}(e)$ is the velocity of the fluid element e, and w_v a constant that defines the influence of the velocity force.

Total Force. The total control force per volume is then simply the sum of the attraction and velocity force, i.e., $\mathbf{f}_c(e) = \mathbf{f}_a(e) + \mathbf{f}_v(e)$. Finally, we get the new total force per volume $\mathbf{f}(e)$ acting on a fluid element e as

$$\mathbf{f}(e) = \mathbf{f}_c(e) + \mathbf{f}_f(e), \qquad (8)$$

where $\mathbf{f}_f(e)$ is the fluid force from the physical fluid simulation. Integrating $\mathbf{f}(e)$ gives the new velocity $\mathbf{v}'(e)$ of a fluid element that is then used for the next SPH step, or the calculation of the equilibrium distribution functions for LBM. Thus, $f_i^{eq}(\rho, \mathbf{v}'(e))$ is used instead of $f_i^{eq}(\rho, \mathbf{v})$ during the LBM collision step in Equation (3). In order to apply the method to a level set based solver it will be necessary to ensure that the velocity field is divergence free, e.g., as in [SY05b].

3.3. Detail-Preserving Control

The velocity force of Equation 7 effectively leads to an averaging of the fluid velocities with the control velocities, which introduces undesirable artificial viscosity. This effect is illustrated in Figure 2, where 250 control particles force the fluid to flow up the stairs. The middle row of pictures is calculated

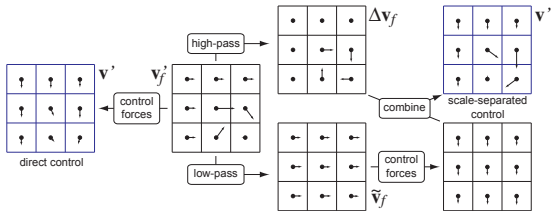

Figure 3: *Idealized illustration of velocity control. With direct control (left) the velocity field is forced to conform with a downward pointing control field. Scale-separated control (right) only applies control forces to the low-frequency components thus preserving small-scale vortices.*

with control forces as described above. Within the influence of the control particles, the fluid is unable to develop small-scale vortices and turbulent behavior.

We want to ensure that the fluid velocity matches that of the control particles without unnecessarily disturbing the natural small-scale fluid motion. To achieve this goal we separate the overall fluid motion from fine-scale detail using a low-pass filter on the current velocity field. Velocity forces are then computed with respect to the smoothed fluid velocities. We obtain the smooth velocity field $\tilde{\mathbf{v}}_f$ using an approximation of discrete convolution with the kernel W of the control particles:

$$\tilde{\mathbf{v}}_f(e) = \frac{\sum_i \tilde{\mathbf{v}}_i W(d_{i,e},h)}{\sum_i W(d_{i,e},h)} \quad \text{with} \quad \tilde{\mathbf{v}}_i = \frac{\sum_e \mathbf{v}'_f(e) W(d_{i,e},h)}{\sum_e W(d_{i,e},h)} \,,$$

where the filtered velocity for each control particle $\tilde{\mathbf{v}}_i$ is computed with the current fluid velocities $\mathbf{v}'_f(e)$ given by the simulation. This smoothed version of the fluid velocity then replaces $\mathbf{v}(e)$ in Equation 7.

To show that this new control force only modifies the low-frequency part, while retaining the high-frequencies, we integrate the control force $\tilde{\mathbf{f}}_c(e)$ separately yielding $\mathbf{v}'_c$. Note that $\mathbf{v}'_c = k(\mathbf{v}_p - \tilde{\mathbf{v}}_f)$, where $\mathbf{v}_p$ is the interpolated velocity of the control particles at a fluid element e and k is a constant depending on the user parameter w_v (see Equation (7)). By decomposing $\mathbf{v}'_f$ into the low-pass filtered velocity $\tilde{\mathbf{v}}_f$ and the high-frequency part $\Delta\mathbf{v}_f$, i.e. $\mathbf{v}'_f = \tilde{\mathbf{v}}_f + \Delta\mathbf{v}_f$, we get for the new fluid velocity

$$\begin{aligned} \mathbf{v}' &= \mathbf{v}'_f + \mathbf{v}'_c \\ &= \tilde{\mathbf{v}}_f + \Delta\mathbf{v}_f + k(\mathbf{v}_p - \tilde{\mathbf{v}}_f) \\ &= (1-k)\tilde{\mathbf{v}}_f + k\mathbf{v}_p + \Delta\mathbf{v}_f . \end{aligned} \tag{9}$$

This equation shows that the low frequency part of the fluid velocity is blended with the velocity of the control particles, while the high-frequency part is retained, as sketched in Figure 3. The bottom row of Figure 2 shows the effect of the scale-separated force control that significantly better preserves fine-scale fluid motion.

4. Results and Discussion

We have implemented our control algorithm for both an SPH and an LBM fluid solver. Within the SPH solver, the existing acceleration structures can be used to efficiently query fluid particles in the neighborhood of a control particle. For the LBM solver, control particles are rasterized to the grid, using early reject tests to prevent unnecessary evaluations of the influence forces. Due to the small changes of the flow state during a single LBM step it, is sufficient to update the control force array in intervals. For the simulations presented here we update the forces every 32 LBM steps.

The effect of our detail-preserving control approach is shown in Figure 5. A column of fluid splashes against a wall at a T-junction. Without control, the water symmetrically distributes on both sides (Figure 5 upper row). A coarse set of 216 control particles forces the fluid to flow towards the left (Figure 5 left and right columns). As can be seen on the left column, the control introduces artificial viscosity which smoothes out all turbulence and vortices of the flow. The pictures on the right column show the resulting fluid motion using the detail-preserving approach described in Section 3.3. The controlled flow with detail-preservation retains small-scale fluid features and therefore yields similar splashes as the original flow, resulting in a more natural looking fluid behavior. The simulation was done using LBM with a $240 \cdot 120 \cdot 120$ grid resolution which took 38s per frame on average (without rendering) on a standard Pentium IV 3 GHz PC. The computation of the control forces took $2-4\%$ of the total computation time.

In Figure 6 we demonstrate target matching on a running horse example, similar to [SY05b]. Note that unlike in [SY05b], the mesh is only used to generate a sequence of control particles as described in Section 3.1. During the simulation, the attraction forces cause the fluid to flow into the given shape sampled with control particles, while the fluid follows the motion of the mesh due to the velocity force. We sampled the initial horse mesh regularly with 69k control particles, and used 266k particles for the SPH simulation which took 102s per frame, including the computation of the control forces which took 14s per frame. In Figure 4 the fluid is forced to flow up the stairs and form a human figure. For the first part of the animation, we use 500 control particles generated from a time-reversed coarse simulation of fluid flowing down the stairs. As the simulation progresses, these control particles are blended with 5k control particles sampled from the 3D model of the human figure. Finally, all control is released and the fluid is pulled back down by gravity. The simulation using LBM with a 300^3 grid resolution took 142s per frame, including 4s for computing the control forces.

As shown in the examples, our detail-preserving approach clearly reduces the artificial viscosity introduced by the control forces. The separation of scales is automatically adapted to the density of control particles by coupling the width of

the low-pass filter with influence radius h of the control particles. This allows iterative refinement of controlled flow simulations. Starting with a coarse simulation with few control particles, the user can interactively adjust the parameters until the desired coarse-scale behavior of the fluid is obtained. She can then add another layer of control particles of higher resolution to also control the fluid dynamics at finer scales.

There are number of limitations and possible improvements for the future. Since control is force-based, it is difficult to define hard constraints on the flow, e.g., to guarantee that fluid does not enter a certain region of the simulation environment. When matching the shape of a dynamically moving object (Figure 6) the optimal weighting between attraction and velocity force depends on the dynamics of the object, which currently requires the user to experiment with different parameter settings for w_a and w_v. So far our control particles have a radial symmetric kernel. Anisotropic kernels would enable better directed control of the simulation. Our framework could also be used to control the deformation of elastic bodies, similar to [KKA05]. Finally, for grid-based methods, Fourier techniques could be an interesting alternative to explicit smoothing of the velocity field.

5. Conclusions

We presented a detail-preserving approach for controlling fluids based on control particles. The fluid flow is modified by attraction and velocity forces exerted from the control particles. Several possibilities for generating control particles have been used. This simple control scheme showed to be very flexible and efficient, and allows to simulate various effects such as filling and following moving target objects or time-reversed flows. We solve the problem of artificial viscosity introduced by the control forces by applying these forces on the low-pass filtered velocity field. Therefore, only the coarse scale flow of the fluid is modified while the natural small-scale detail is preserved, resulting in more natural looking controlled simulations.

6. Ackknowledgements

We would like to thank Doug Epps and Mark Carlson for discussions on the subject of fluid control. We furthermore thank Markus Gross and Marc Stamminger for their support.

References

[CMT04] CARLSON M., MUCHA P. J., TURK G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. 23*, 3 (2004). 1

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proc. of ACM SIGGRPAH* (2001). 1, 2, 3

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. *ACM Trans. Graph. 23*, 3 (2004), 441–448. 2

[FM97] FOSTER N., METAXAS D.: Controlling fluid animation. In *Proc. of CGI* (1997). 2, 3

[FOA03] FELDMAN B. E., O'BRIEN J. F., ARIKAN O.: Animating suspended particle explosions. In *Proceedings of ACM SIGGRAPH 2003* (Aug. 2003), pp. 708–715. 2

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential: Research articles. *Comput. Animat. Virtual Worlds 15*, 3-4 (2004), 147–157. 2

[HL97] HE X., LUO L.-S.: Lattice boltzmann model for the incompressible navier-stokes equations. *J. Stat. Phys. 88* (1997). 3

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (2005), 561–566. 3

[KKA05] KONDO R., KANAI T., ANJYO K.: Directable animation of elastic objects. In *Proc. of Symposium on Computer Animation* (2005), ACM Press. 5

[KLLR05] KIM B., LIU Y., LLAMAS I., ROSSIGNAC J.: Flowfixer: Using bfecc for fluid simulation. *EG Workshop on Natural Phenomena* (2005), 51–56. 1

[KPR*05] KÖRNER C., POHL T., RÜDE U., THÜREY N., ZEISER T.: Parallel Lattice Boltzmann Methods for CFD Applications. In *Nume. Sol. of Partial Differential Equations on Parallel Computers*, vol. 51. 2005. 3

[LF02] LAMORLETTE A., FOSTER N.: Structural modeling of flames for a production environment. In *Proc. of ACM SIGGRAPH* (2002). 2

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics 23*, 3 (2004), 457–462. 1

[LIGF06] LOSASSO F., IRVING G., GUENDELMAN E., FEDKIW R.: Melting and burning solids into liquids and gases. *IEEE TVCG 12*, 3 (2006), 343–352. 1

[LL03] LIU G. R., LIU M. B.: *Smoothed Particle Hydrodynamics, A Meshfree Particle Method*. World Scientific Publishing, 2003. 2

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proc. of Symposium on Computer Animation* (2003). 2, 3

[Mon05] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Rep. Prog. Phys. 68* (2005), 1703–1758. 2

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIC Z., STAM J.: Fluid control using the adjoint method. *ACM Trans. Graph. 23*, 3 (2004), 449–456. 2

[PCS04] PIGHIN F., COHEN J. M., SHAH M.: Modeling and editing flows using advected radial basis functions. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), ACM Press, pp. 223–232. 2

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Proc. of Symposium on Computer Animation* (2004). 2

[SY05a] SHI L., YU Y.: Controllable smoke animation with guiding objects. *ACM Trans. Graph. 24*, 1 (2005). 2

[SY05b] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *Proc. of Symposium on Computer Animation* (2005). 2, 3, 4

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIC Z., STAM J.: Keyframe control of smoke simulations. *ACM Trans. Graph. 22*, 3 (2003), 716–723. 2

[TRK05] THÜREY N., RÜDE U., KÖRNER C.: *Interactive Free Surface Fluids with the Lattice Boltzmann Method*. Tech. Rep. 05-4, 2005. 3

[YMLS03] YU D., MEI R., LUO L.-S., SHYY W.: Viscous flow computations with the method of lattice boltzmann equation. *Porgress in Aeospace Sciences 39* (2003). 3

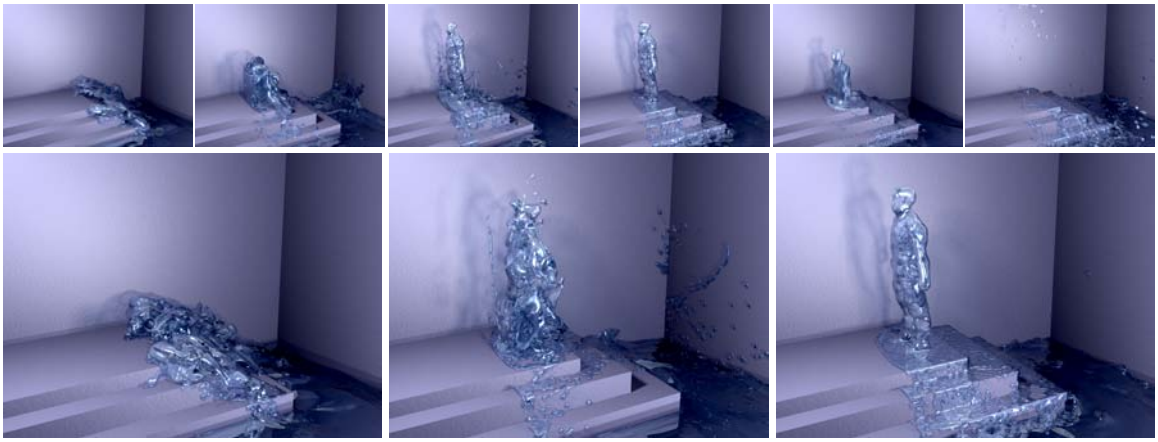

Figure 4: *A fluid simulation with LBM is controlled to flow up the stairs and form a human figure.*

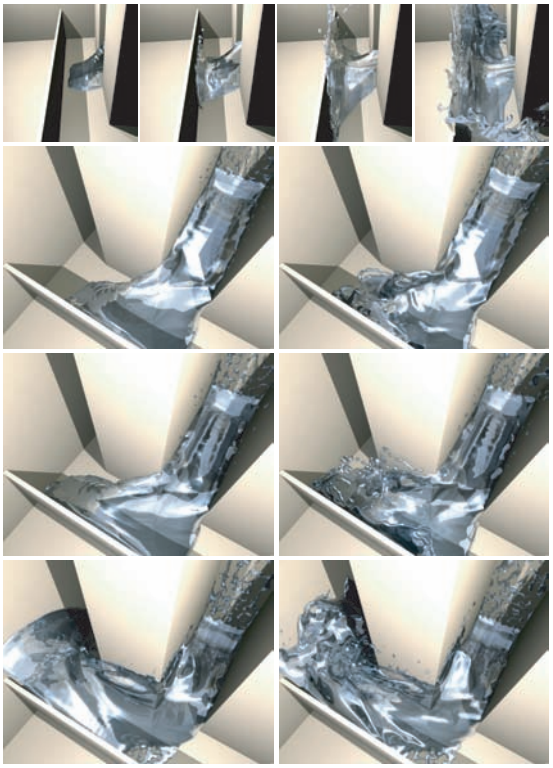

Figure 5: *Comparison of direct velocity control (left column) and detail-preserving control (right column) for LBM. The top row shows the uncontrolled simulation.*

Figure 6: *An SPH fluid following the shape of an animated model. Control particles are created by sampling the interior of the horse.*