Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005)
K. Anjyo, P. Faloutsos (Editors)

Taming Liquids for Rapidly Changing Targets

Lin Shi and Yizhou Yu

University of Illinois at Urbana-Champaign

Abstract

Following rapidly changing target objects is a challenging problem in fluid control, especially when the natural fluid motion should be preserved. The fluid should be responsive to the changing configuration of the target and, at the same time, its motion should not be overconstrained. In this paper, we introduce an efficient and effective solution by applying two different external force fields. The first one is a feedback force field which compensates for discrepancies in both shape and velocity. Its shape component is designed to be divergence free so that it can survive the velocity projection step. The second one is the gradient field of a potential function defined by the shape and skeletion of the target object. Our experiments indicate a mixture of these two force fields can achieve desirable and pleasing effects.

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

Recent progress on physics-based fluid simulation in the graphics community has produced stunning dynamics comparable to their counterparts in the real world. Nevertheless, animators not only need realism, but also need to achieve certain design goals when producing animations. In the past, liquid actors and animals, such as those in "The Abyss" and "Terminator 2: Judgment Day," have been created with realistic rendering, but without the desirable liquid dynamics. Recent techniques on liquid control [MTPS04, REN*04] have incorporated physics-based fluid dynamics. But they are not particularly convenient to use. Fluid control methods should be evaluated with the following criteria:

- **Control capability.** When given a set of user-specified constraints, such as key frames or target shapes, a control method should be able to force the movement of the fluid to approximately satisfy the constraints.
- **Ease to use.** As an authoring tool, the method should be able to produce desirable fluid animations without too much computational cost or user intervention.
- **Fluid-like motion.** The control method should not overconstrain the fluid. The natural movement of the fluid should be preserved as much as possible.
- **Stability.** The controlled movement of the fluid should be stable without obvious oscillations.

Figure 1: *A controlled simulation with a water horse emerging, running and collapsing.*

In this paper, we focus on controlling liquids to match rapidly changing target shapes which represent regular non-fluid objects. In other words, we assume the animator prepares a continuous sequence of frames instead of sparse keyframes. This is a somewhat different goal from previous work, and is particularly useful when the whole sequence of

target shapes is essential to represent the target motion such as a running water man. The physical properties of liquids and the spatially varying large velocities and accelerations over the target shape pose challenges on the control method. First, the large magnitude of the velocities and accelerations of the target shape demands a stable and well-balanced fluid control method to track the fast-changing global shape while maintaining natural fluid motion. Second, when compared to gases, liquids have a clear boundary at their interface with the air and their volume is strictly incompressible. These properties imply that undesirable artifacts introduced by a control method can be easily noticed.

With the unique challenges posed by the aforementioned problem, we present a simple but effective control technique by applying two different external force fields.

- The first one is a feedback force field which compensates for discrepancies in both shape and velocity (Section 3). Its shape component is designed to be divergence free so that it can survive the velocity projection step. Its velocity component can effectively prevent overshooting and improve stability.
- The second one is the gradient field of a potential function defined by the shape and skeletion of the target object (Section 4). We choose to use a nonlinear function of the signed distance transform of the target shape as the potential so that large deviations will be reacted to more strongly.

Our fluid control method was inspired by the optimal control theory in [Bul00] which is based on both potential shaping and feedback control.

Our control method meets all of the aforementioned criteria. It is capable of tracking rapidly changing targets while maintaining fluid-like motion. Given a target animation, computing the spatially varying feedback and potential forces are automatic and fast. There are only three tunable parameters representing the overall strengths of the force fields. Thus, our control technique does not incur much extra computation and user intervention beyond a regular fluid simulation. As a result, it typically only takes a few hours to author a fluid animation with controlled behavior.

2. Related Work

This work has been made possible by much of the previous work on fluid simulation [Sta99, FSJ01, FF01, EMF02, LGF04, GBO04, CMT04], and the level set method [Set99, OF01]. In particular, our liquid simulation solves the Navier-Stokes equations and follows the particle level set implementation in [Enr02]. In the following, we are going to focus on fluid control methods which are most relevant to this paper.

Foster and collaborators initiated fluid control in graphics. Foster and Metaxas [FM97] proposed embedded controllers that allow animators to specify control parameters in a fluid animation. In [FF01, LF02], animator-designed space curves and surfaces can be applied to control the motion and structures of fluids. The tangents of the curves or the normals of the surfaces indicate the directions of motion. In light of this approach, a control method based on relatively dense particles (one particle per grid cell) [SHG*03, REN*04] has produced impressive results. The control particles can not only control the velocity of the fluid in their local neighborhoods, but also control viscosity, the level set of the liquid surface, and the velocity divergence. Although the results of production quality, the keyframes for all the particles need to be generated from an animation package, which involves a significant amount of work from an animator. These techniques do not allow the user to directly specify higher-level objectives, such as matching the density or shape of a target object.

Recently, a systematic optimization framework for controlling both smoke and liquid simulations through user-specified keyframes was introduced in [TMPS03, MTPS04]. By viewing fluid simulation as a composition of functions, simulation derivatives can be surprisingly obtained from the fluid solver itself. However, since the derivatives with respect to each control parameter need to be computed throughout an entire simulation, this approach is very expensive. The adjoint method was adopted in [MTPS04] to significantly improve the efficiency of these derivative evaluations for each iteration during the nonlinear optimization. Nevertheless, a number of iterations are still necessary to obtain a good solution. As a result, this framework is not particularly animator friendly since the optimization takes a long time and can only be performed on relatively coarse grids. Furthermore, the objective function of the optimization forces the fluid to tightly match the target density, sacrificing its natural appearances. The method introduced in this paper is much more efficient and can handle grids with two orders of magnitude more voxels. In addition, our method can maintain much of the natural motion of fluids.

An efficient and novel technique to match smoke density against user-specified distributions was reported in [FL04], which involves a custom-designed driving force term and a smoke gathering term. It has demonstrated much shorter computing time than nonlinear optimization. However, the driving force term requires a nonuniform density distribution for its gradient estimation. Meanwhile, another efficient method based on the level sets of the smoke was introduced in [SY05]. It can effectively control the smoke shape without the forcing and gathering terms in [FL04]. However, it relies on a compressible fluid model to eliminate visual artifacts caused by the constraints on the smoke boundary. Such a compressible model becomes inappropriate for liquids which are strictly incompressible. The method in this paper applies a divergence-free feedback force field for shape differences to avoid this problem. It also applies additional velocity feedback forces to achieve stability.

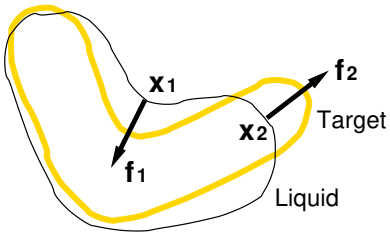


Figure 2: *The initialization of the shape feedback force on the liquid interface. The initial force* $\mathbf{f_1}$ *at* $\mathbf{x_1}$ *outside the target object is defined according to (3);* $\mathbf{f_2}$ *at* $\mathbf{x_2}$ *inside the target is defined according to (4).*

Yet another simple and clever control scheme for smoke simulation was presented in [HK04], which exploits a potential function based on the shape of the target object. The negative gradient of the potential function serves as an extra force field which tries to reduce the overall potential of the smoke region. Upon its convergence, the shape of the smoke region coincides with the target shape, which represents the lowest energy configuration of the potential function. The original paper [HK04] focuses on smoke and only has results for static target shapes. When the target shape has complex and rapidly changing motion, it is unclear how to keep the liquid in pace with the target. In addition, when the configuration of the liquid is inconsistent with the target shape, undesirable oscillations around the target shape become obvious because of the inertia of the liquid. The method in this paper overcomes these problems.

3. Feedback Control Forces

During a controlled liquid animation, our algorithm applies a feedback control force field to reduce the shape discrepancy between the liquid and the target. Since the target surface may have large accelerations, control forces only based on shape differences become insufficient to guarantee both convergence and natural motion. Inspired by control theories, such as [Hau04, Bul00], we decided to apply feedback forces to compensate for velocity differences as well. As a result, the complete feedback force has two terms,

$$\mathbf{f}_{feedback} = \mathbf{f}_{shape} + \mathbf{f}_{velocity}, \tag{1}$$

where $\mathbf{f}_{shape}$ compensates for shape differences and $\mathbf{f}_{velocity}$ compensates for velocity differences. In control terminology, $\mathbf{f}_{shape}$ corresponds to proportional control and $\mathbf{f}_{velocity}$ corresponds to derivative control. There is no obvious definition of a force field corresponding to integral control in the context of shape matching. Therefore, we replace the role of integral control with a geometric potential field which will be discussed in Section 4.

At any moment t, the liquid boundary and target shape are represented by two signed distance functions, $d_L(\mathbf{x},t)$ and $d_T(\mathbf{x},t)$, respectively. $d_T(\mathbf{x},t)$ is positive when $\mathbf{x}$ is outside the target shape, and $d_L(\mathbf{x},t)$ is positive when $\mathbf{x}$ is outside the liquid volume.

3.1. Velocity Feedback

To effectively apply this feedback force, we design a force field throughout the liquid volume. At any point $\mathbf{x}$ either inside the liquid volume or on the liquid boundary, the velocity feedback force is simply

$$\mathbf{f}_{velocity}(\mathbf{x}) = -\beta\,(\mathbf{v}_L - \mathbf{v}_T), \tag{2}$$

where β is the gain for derivative control, $\mathbf{v}_L$ represents the velocity of the liquid at $\mathbf{x}$ and $\mathbf{v}_T$ represents some target velocity at the same point. We use $\beta = 25$ in our experiments.

Ideally, the target velocity, $\mathbf{v}_T$, should be the velocity of the target shape. When a point on the liquid boundary is outside the target shape, the velocity of the target shape at that point is not well defined. Nevertheless, the signed distance function of the target shape is computed everywhere, and it moves with the target shape. We use the signed distance function to propagate the velocities of the target shape to all the points in the work space [AS99]. Thus, a point outside the target shape is assigned the velocity of the closest point on its surface. In our experiments, such a propagation produced desirable target velocities. For interior points of the target shape, usually this velocity can be uniquely determined. For example, the pointwise velocity of a rigid body can be obtained from its angular velocity and the velocity of its center of mass. In our experiments, articulated bodies are modeled as a collection of rigid segments. Given an interior point, we first determine which rigid segment it belongs to and then compute its velocity. For objects with a deformable surface, we assume the vertices of the surface correspond from frame to frame and apply the same technique for determining external velocities.

3.2. Shape Feedback

This feedback force only concerns the differences between the target boundary and the liquid boundary. However, the effect of such a force field can be easily diminished by the projection step [Sta99, FF01] which forces the entire velocity field to be divergence-free. That means the force field needs to be divergence-free to survive the projection step. To obtain such a force field, we take the following steps. We first initialize the forces on the boundary. Further optimization is performed to guarantee that the flux of the forces on the boundary is zero, which is a necessary condition for the liquid to be volume-preserving. At the end, a divergence-free force field at the interior of the liquid can be solved by considering the forces on the boundary as the boundary condition.

3.2.1. Force Initialization on the Liquid Boundary.

Let us focus on points on the liquid boundary. As shown in Fig. 2, when such a point $\mathbf{x}$ is outside the target shape, the initial force at this point is formulated as

$$\tilde{\mathbf{f}}(\mathbf{x}) = -\alpha\, d_T \frac{\nabla d_T}{\|\nabla d_T\|}; \tag{3}$$

otherwise, the initial force is formulated as

$$\tilde{\mathbf{f}}(\mathbf{x}) = \alpha\, d_L \frac{\nabla d_L}{\|\nabla d_L\|}, \tag{4}$$

where α is the gain for proportional control. Note that the force follows the opposite direction of ∇d_T when the point is outside the target so that the liquid can be pulled toward the target faster. For the portion of the liquid surface that is inside the target shape, faster expansion is achieved by using the normal direction of the liquid surface. In our experiments, we usually set $\alpha = 625$.

3.2.2. Force Optimization on the Liquid Boundary.

On a discrete grid, let $\{\mathbf{n}_i\}_1^m$ be the normals of the m surface faces on the liquid boundary, and $\{\mathbf{f}_i\}_1^m$ be the finalized feedback forces on the boundary. To ensure that the total flux of these forces is zero, we enforce the following compatibility condition,

$$\Phi_{\mathbf{f}} = \sum_i \mathbf{f}_i \cdot \mathbf{n}_i = \sum_i \sum_{j=1}^{3} f_i^j\, n_i^j = 0, \tag{5}$$

where $\mathbf{f}_i = [f_i^1 \;\; f_i^2 \;\; f_i^3]^T$ and $\mathbf{n}_i = [n_i^1 \;\; n_i^2 \;\; n_i^3]^T$. The area of the faces has been eliminated from (5) since it is the same for all the grid cells.

Let $\{\tilde{\mathbf{f}}_i\}_1^m$ represent the initial forces on the liquid boundary. We would like to adjust such an initialization by minimizing the following objective function while satisfying (5),

$$\sum_i \|\mathbf{f}_i - \tilde{\mathbf{f}}_i\|^2, \tag{6}$$

where the squared differences measure the deviation from the initialization. Such a constrained minimization problem can be solved by introducing a Lagrange multiplier for (5). Suppose $\Phi_{\tilde{\mathbf{f}}} = \sum_i \tilde{\mathbf{f}}_i \cdot \mathbf{n}_i$. The final solution of the boundary forces is simply as follows [PT85].

$$\mathbf{f}_i = \tilde{\mathbf{f}}_i - \frac{\Phi_{\tilde{\mathbf{f}}}}{m}\, \mathbf{n}_i, i = 1, \ldots, m. \tag{7}$$

3.2.3. The Complete Shape Feedback Force Field.

Once the flux of the forces on the liquid boundary is zero, it becomes feasible to solve a divergence-free force field throughout the liquid volume. We formulate the complete shape feedback force field as the gradient field of a scalar function, H. Thus, $\mathbf{f}_{shape} = \nabla H$.

To ensure zero divergence everywhere, $\nabla \cdot \nabla H$ has to be zero everywhere in the liquid. Thus, H is a harmonic function which is the solution to the following Laplace equation

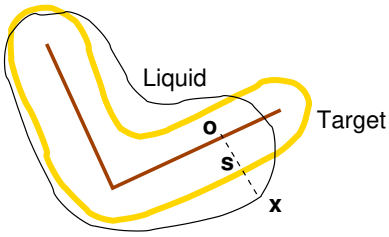


Figure 3: *A potential field is defined with respect to the target object's shape and skeleton. The skeleton is assigned a distance value d_0 while the surface of the target is assigned a distance value d_1 with $d_0 < d_1$. The potential at any point $\mathbf{x}$ within the liquid volume is defined according to (10) and (9).*

with a boundary condition specified by the gradients on the boundary [PT85],

$$\nabla^2 H = 0, \quad \nabla H|_{\partial\Omega} = f^*|_{\partial\Omega}, \tag{8}$$

where $\partial\Omega$ denotes the liquid boundary, and f^* represents the boundary control forces obtained from the previous section. The discretization of (8) on a volume grid gives rise to a sparse linear system which can be efficiently solved by a preconditioned conjugate gradient method [GL96].

4. Adaptive Geometric Potential

Controlling smoke simulation with the help of a potential field defined by the geometric shape of a target object is first investigated in [HK04]. In their method, the interior of the target object is assigned with a uniform minimal potential value, and points outside the target have higher potential values. The negative gradient of the potential field, which is zero inside and nonzero outside the target, served as the driving force. The smoke converges to the target shape when the equilibrium with minimal potential energy has been reached.

In this paper, we generalize this method to liquids and design an effective potential field for them. The general form of the potential field we adopt is simply a monotonically increasing function of the signed distance of the target shape,

$$\phi(\mathbf{x}) = C\, \mathrm{sgn}(d_T(\mathbf{x}))\, |d_T(\mathbf{x})|^\gamma \tag{9}$$

where C is a constant factor representing the overall strength of the potential field. $C = 1300$ and $\gamma = 2$ in most of our experiments. The reason for this formulation is that the gradient of this new potential field has increasing magnitude when the distance to the target surface becomes larger. That means the potential field generates a larger force to pull the liquid back when it is further away, but the force remains small around the target surface to produce more natural fluid motion. This is in contrast to the original signed distance which only has a constant gradient. This new potential still has the same equilibrium as before. When the target object

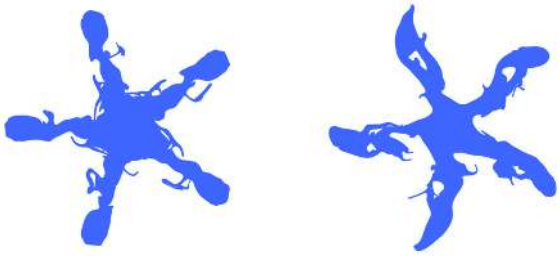


Figure 4: *Two frames from a two-dimensional controlled fluid simulation with a rotating star shape as the target motion. The 2D grid resolution is* 1000^2*, and the computational time is 2.5 minutes per frame.*

is moving, we simply calculate its potential field at every frame.

The negative gradient of the above potential field points towards the central skeleton of the target shape. However, using a function of the signed distance as the potential field creates problems when the target has a complex shape, such as the shape of a human character whose limbs and torso have very different thickness. Thick regions can tolerate more deviation in shape than thin regions which thus demand larger gradient forces to produce faster response. For this purpose, we define an adaptive signed distance based on two isosurfaces assuming that a connected skeleton of the target shape is known. In this definition, all the points on the surface of the target shape are required to have the same distance value, d_1, and all the points on the skeleton of the target shape are required to have another distance value, d_0 ($< d_1$). As shown in Fig. 3, the signed distance at a point $\mathbf{x}$ inside the liquid volume is defined to be

$$d_T^a(\mathbf{x}) = d_0 + (d_1 - d_0)\frac{\|\mathbf{ox}\|}{\|\mathbf{os}\|}, \qquad (10)$$

where $\mathbf{o}$ is the closest point of $\mathbf{x}$ on the skeleton, and $\mathbf{s}$ is the intersection between the boundary of the target shape and the line defined by $\mathbf{ox}$. Thus, the potential field in (9) should be updated by replacing $d_T(\mathbf{x})$ with $d_T^a(\mathbf{x})$. With such an adaptive potential field, thin regions have a larger gradient force because the Euclidean distance between the skeleton and the surface is smaller. Another advantage of considering the skeleton as an isosurface is that it is easier for the liquid to move around and less likely to be trapped in a local region. In practice, we always set $d_0 = -1$ and $d_1 = 0$. In general, such a gradient force field does not guarantee zero divergence.

5. Implementation and Analysis

The input to our system is a continuous animation of single or multiple target shapes, which can be created from key-

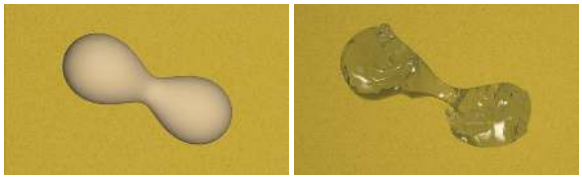


Figure 5: *Left: The target shape and orientation at one of the frames of the target animation. Right: A rendered image from our liquid control method. The resolution is* 300^3*, and the computational time is 3.2 minutes per frame.*

frames, physics-based simulations, or motion capture. The sequence of animated target shapes should be automatically converted to a sequence of implicit functions because the liquid surface is represented as an implicit function as well. The output from our system is a liquid simulation that approximately follows the target animation while still maintaining natural liquid dynamics.

Our overall control scheme exploits both types of forces introduced in the previous two sections. The feedback control force field is very powerful in making the liquid follow the target shape and motion. However, such a strong control is not desirable because the liquid motion would not look natural anymore. On the other hand, the force generated by the geometric potential field resembles the gravity in the real world. If we look at the earth from the space, the gravity actually points roughly towards the center of the earth. Due to this similarity, when the target does not have large accelerations, the controlled fluid motion under the geometric potential is expected to be similar to that under the gravity, and therefore, appears natural. However, the control capability of the geometric potential weakens and produces undesirable high frequency oscillations when it is changing quickly from frame to frame. The velocity feedback force can effectively alleviate such oscillations. Therefore, to achieve both desirable control and natural fluid-like motion, we apply both force fields simultaneously, but use a relatively small feedback force to avoid artifacts. The typical overall strengths of these force fields have been given in the previous sections.

According to Section 3 and 4, our control technique computes spatially varying force fields automatically. Such computations are fast, and do not incur much extra cost beyond a regular fluid simulation. As a result, we can adopt any grid resolution that a regular fluid simulation can use. There are only three parameters that an animator needs to interactively adjust to achieve desirable controlled simulation. Namely, the gain of the shape feedback, the gain of the velocity feedback and the overall strength of the geometric potential field. It should be noted that a small amount of user intervention is necessary for an authoring tool since an animator should be able to choose preferable results by tuning a small number of "control knobs". In our experiments, the control parameters do not need to be changed drastically for different target

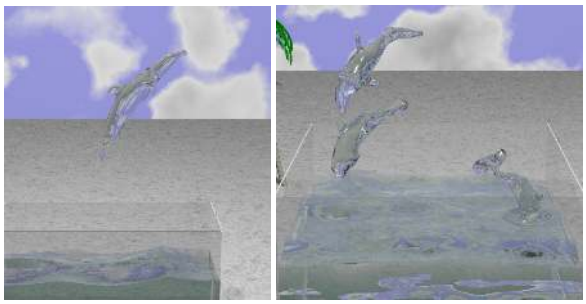


Figure 6: *Left: One liquid dolphin in the air. The resolution is 234x180x108, and computational time is around 4.8 minutes per frame. Right: Three liquid dolphins jumping back to the water. The resolution is 180x78x180, and computational time is 4.2 minutes per frame.*

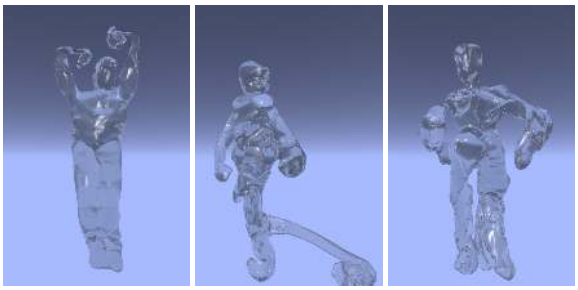


Figure 7: *A liquid simulation following a dancing sequence. The simulation grid resolution is* 300^3*, and the computational time is 7 minutes per frame.*

sequences. An animator should be able to author new animations by incorporating minor changes to existing parameters. One can also start with relatively strong control parameters, and relax them until the results become desirable. Experiments also indicate that our force fields are insensitive to the resolution of the simulation grid. Similar effects can be obtained at various different resolutions except for very coarse ones. Thus, a coarser grid can be used during the tuning stage to let the animator quickly see intermediate results. Therefore, it typically only takes a few hours to author a reasonable fluid animation with controlled behavior.

6. Experimental Results

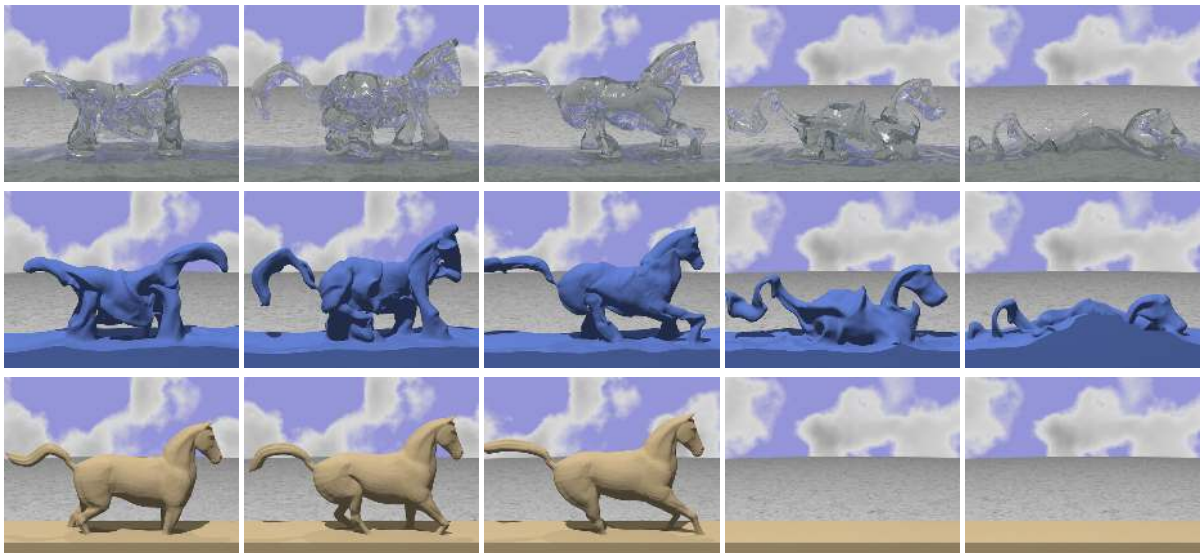
We have successfully applied our liquid control method to a few examples. Fig. 4 shows two frames from a two-dimensional controlled fluid simulation which uses a rotating star shape as the target animation. The overall shape of the fluid boundary resembles the target object. However, there are also many intricate details indicating this is a fluid. Fig. 5 shows a three-dimensional controlled simulation which uses the rigid body motion of a dumbbell shape as the target animation. The liquid not only follows the target motion, but also has small waves on the surface.

The left image in Fig. 6 shows one frame from a controlled fluid simulation with a dolphin jumping sequence as the target animation. When the target jumps out of the water, it also brings a portion of the water with it. The isolated portion of the water evolves into a dolphin shape. The right image in Fig. 6 shows another related simulation with three dolphin-shaped targets.

Fig. 1 and 8 show a liquid horse emerging from a water, running on the water and then collapsing back into the water. Both the simulated liquid surface (blue meshes) and the underlying target shape (yellow meshes) are shown as well. The target shape has two components, the horse shape and a horizontal plane under the horse. The horizontal plane defines the target shape for the free moving water which is supposed to have a flat surface when it stabilizes. The horse mesh sequence is from [SP04]. Fig. 8(a)-(c) show interesting liquid dynamics when the shape of the horse is being formed. Fig. 8(d) shows a frame where the liquid has stabilized and matched the target shape. Most small-scale details on the target shape are preserved by the matching liquid interface. To generate the collapse in Fig. 8(e)-(f), we simply remove the horse shape and only leave the horizontal plane. Interestingly, we obtained more natural results in this example by setting $\gamma = 1$ in (9).

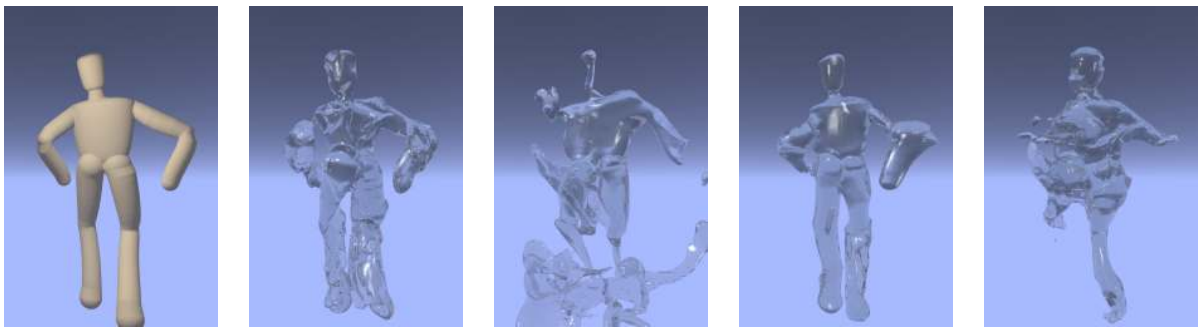
Fig. 7 shows another controlled liquid simulation using a MOCAP sequence as the target animation. The MOCAP data have very large, rapidly changing accelerations. Nevertheless, our control method was able to keep the liquid in pace with the target. Meanwhile, the simulated liquid surface exhibits natural waves and other fluid motion. A comparison is given in Fig. 9. We tested our method on this sequence against the original potential-based method in [HK04] and the smoke control technique in [SY05]. To preserve fluid appearance, a weak potential field is used in Fig. 9(b), but the liquid does not follow the rapidly changing target shape. Once a relatively strong potential field is used to control the liquid as in Fig. 9(c), the liquid does not have natural motion but many high frequency oscillations. In Fig. 9(d) we show the method from [SY05] cannot control the liquid well either even with an extremely strong force field. All the animations can be found in the accompanying video.

In our experiments, we use an effective grid resolution up to 300^3 using a grid windowing and resizing technique similar to that in [REN*04]. The computational time is almost the same as a regular liquid simulation which costs 5-10 minutes a frame on a high-resolution grid and only around 30 seconds a frame on a coarse grid ($< 100^3$) using an AMD 3200+ processor. Computing the control force fields only costs less than 10% of the total simulation time. It should be noted that the finest grid resolution used in [MTPS04] is 50x50x50 which only has 125,000 elements. Thus, the grid



(a) (b) (c) (d) (e)

Figure 8: *First row: a liquid simulation representing a water horse emerging (a & b), running (c) and collapsing (d & e). Second row: the simulated liquid surfaces rendered as meshes. Third row: the underlying target animation. The resolution is 275x250x75, and the computational time is 4.4 minutes per frame.*



(a) (b) (c) (d) (e)

Figure 9: *(a) The target shape at one of the frames of the dancing sequence. (b) A rendered image from our liquid control method. (c) A rendered image from the method in [HK04], with a weak potential field. (d) Same method as in (c) with a relatively strong potential field. (e) A rendered image from the method in [SY05].*

in their simulations has two orders of magnitude less elements than ours.

7. Conclusions and Future Work

Following rapidly changing target objects is a challenging problem in fluid control. We introduced a simple but very effective solution by applying two external force fields: a carefully designed feedback force field and the negative gradient field of a geometric potential function. Experiments indicate that our method can achieve pleasing results.

In future, we would like to extend this work to sparse keyframes and include interactions between the controlled liquid and other more rigid objects. During such interaction, interesting phenomena such as splashing should be simulated. A hybrid system based on both target shapes and control particles might be necessary for such simulations. The technique in [CMT04] might also be helpful.

Acknowledgments

We wish to thank David Forsyth, John Hart and Steve LaValle for their helpful suggestions during the project, Alan Perez-Rathke for his assistance with rendering, the authors of [SP04] for sharing their mesh sequences, and the anony-

mous reviewers for their valuable comments. This work was partially supported by National Science Foundation (CCR-0132970).

References

[AS99] ADALSTEINSSON D., SETHIAN J. A.: The fast construction of extension velocities in level set methods. *J. Comput. Phys. 148*, 1 (1999), 2–22. 3

[Bul00] BULLO F.: Stabilization of relative equilibria for underactuated systems on riemannian manifolds. *Automatica 36*, 12 (2000), 1819–1834. 2, 3

[CMT04] CARLSON M., MUCHA P., TURK G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graphics 23*, 3 (2004), 375–382. 2, 7

[EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Transactions on Graphics 21*, 3 (2002), 736–744. 2

[Enr02] ENRIGHT D.: *Use Of The Particle Level Set Method For Enhanced Resolution of Free Surface Flows*. PhD thesis, Stanford University, 2002. 2

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *SIGGRAPH 2001 Conference Proceedings* (2001), pp. 23–30. 2, 3

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. *ACM Transactions on Graphics 23*, 3 (2004), 439–446. 2

[FM97] FOSTER N., METAXAS D.: Controlling fluid animation. In *Proceedings of Computer Graphics International* (1997), pp. 178–188. 2

[FSJ01] FEDKIW R., STAM J., JENSEN H.: Visual simulation of smoke. In *SIGGRAPH 01 Conference Proceedings* (2001), pp. 15–22. 2

[GBO04] GOKTEKIN T., BARGTEIL A. W., O'BRIEN J.: A method for animating viscoelastic fluids. *ACM Trans. Graphics 23*, 3 (2004), 461–466. 2

[GL96] GOLUB G., LOAN C. V.: *Matrix Computations*. The Johns Hopkins Unversity Press, Baltimore, third edition, 1996. 4

[Hau04] HAUGEN F.: *PID Control of Dynamic Systems*. Tapir Forlag, 2004. 3

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds 15*, 3-4 (2004), 147–157. 3, 4, 6, 7

[LF02] LAMORLETTE A., FOSTER N.: Structural modeling of flames for a production environment. In *SIGGRAPH 02 Conference Proceedings* (2002), pp. 729–735. 2

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics 23*, 3 (2004), 455–460. 2

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. *ACM Trans. Graphics 23*, 3 (2004), 447–454. 1, 2, 6

[OF01] OSHER S., FEDKIW R.: Level set methods: an overview and some recent results. *Computational Physics 169*, 2 (2001). 2

[PT85] PEYRET R., TAYLOR T. D.: *Computational methods for fluid flow*. Springer series in computational physics. Springer, 1985. 4

[REN*04] RASMUSSEN N., ENRIGHT E., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2004). 1, 2, 6

[Set99] SETHIAN J.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999. 2

[SHG*03] SUMNER N., HOON S., GEIGER W., MARINO S., RASMUSSEN N., FEDKIW R.: Melting and terminatrix. In *SIGGRAPH 2003 Sketches & Applications, Conference Abstracts and Applications* (2003). 2

[SP04] SUMNER R., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics 23*, 3 (2004), 397–403. 6, 7

[Sta99] STAM J.: Stable fluids. In *SIGGRAPH 99 Conference Proceedings* (1999), pp. 121–128. 2, 3

[SY05] SHI L., YU Y.: Controllable smoke animation with guiding objects. *ACM Transactions on Graphics 24*, 1 (2005), 140–164. 2, 6, 7

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