Time-Reversed Art Directable Smoke Simulation

J. Oborn[†1] and S. Flynn[‡1] and P. Egbert[§1] and S. Holladay[¶1]

Abstract

Physics-based fluid simulation often produces unpredictable behavior that is difficult for artists to control. We present a new method for art directing smoke animation using time-reversed simulation. Given a final fluid configuration, our method steps backward in time generating a sequence that, when played forward, is visually similar to traditional forward simulations. This allows artists to create simulations with fast turnaround times that match an exact art-directed shape at any timestep of the simulation. We address a number of challenges associated with time-reversal including the problem of decreasing entropy.

CCS Concepts

•***Computing methodologies*** → ***Physical simulation;*** *Simulation by animation;* •***Applied computing*** → *Media arts;*

1. Introduction

Fluid simulation has become a well-established tool in visual effects and animation pipelines. It gives artists a reliable way to create complex, photorealistic fluid motion for film and other media without having to hand-craft every frame of the animation. While fluid simulation has allowed artists to achieve tremendous amounts of detail on enormous scales, it also restricts how much influence they have on the shape and behavior of the final result.

Art direction of fluid simulation often demands that the simulated fluid fill a very specific shape at some timestep in the middle of the simulation in order to create an appealing composition. It is often critical to meet a director's vision. However, this is a very expensive, time-consuming process involving many iterations of careful adjustments to the initial state.

Previous methods have attempted to address this problem by manually placing control forces calculated from a set of user-defined guides. These methods tend to either be very slow, or they don't exactly match the target shape.

Our work allows artists to achieve an exact, art-directed shape at any desired timestep, as opposed to being restricted to the very beginning. Our system can then solve both forward and backward from that point. The most important visual moment of any fluid simulation is usually towards the middle or end of the sequence and not at the very beginning. This will allow artists to achieve the exact desired shape at the most meaningful point in the simulation, without compromising natural fluid motion or simulation speed.

2. Related Work

The practical use of 3-dimensional fluid simulation for animation and visual effects [Sta99] [FSJ01] has been tremendous. However, their inherent unpredictability has motivated years of research in efficient control techniques.

Much research has been done with the concept of smoke density targets [TMPS03] [MTPS04] [PM17]. These approaches are computationally expensive, requiring a small number of control forces and coarse simulation grids. Fattal and Lischinski [FL04] introduced a much faster method by using a driving force term to hit target shapes, and a smoke gathering term to prevent diffusion. The downside of this method is that it does not exactly match the target shape.

Other control techniques are built around the idea of using simpler simulation methods to generate control particles, which are then used to influence the fluid simulation [REN*04] [SS17].

The time-reversibility of the Euler equations as an accuracy benchmark for fluid solvers has been explored with an energy-preserving integration scheme [DOW] [MCP*09]. However, we did not use these methods as we were able to achieve plausible results with the modified MacCormack method described in Selle et al. [SFK*08] with second order accuracy.

3. Time-Reversed Simulation

3.1. Overview

Our time-reversible simulator is an extension of the popular semi-Lagrangian smoke simulator introduced by Stam [Sta99]. The tra-

† egbert@cs.byu.edu
‡ ecthelisean@gmail.com
§ jeremy.oborn@gmail.com
¶ seth_holladay@byu.edu

DOI: 10.2312/egs.20181030

ditional forward simulator requires an initial state at time t_0 and advances each timestep, t_i, according to:

$$t_{i+1} = t_i + \Delta t \tag{1}$$

Where i is any given timestep between 0 and n. Our reverse simulator requires an initial state at time t_n and advances a given timestep, t_i, in the reverse direction by:

$$t_{i-1} = t_i - \Delta t \tag{2}$$

The result of this time-reversed process is intended to be presented advancing forward in time. The fluid state at time t_n can also be passed into a forward simulator and the results of the separate simulations can be concatenated in sequence as a single, continuous flow. The result will be presented as follows:

$$t_0 \ldots t_n \ldots t_f$$

Where t_0 is the initial timestep, t_n is the art directed timestep, and t_f is the final timestep of the sequence. The sequence from t_0 to t_n is generated by our reverse simulator.

Algorithm 1 Time-Reversed Fluid Simulation Algorithm

1: **for** $i = n$ to 0 **do**
2: External forces
3: Self-attraction force (section 3.4)
4: Backwards dissipation (section 3.5)
5: Modified divergence projection (section 3.4.1)
6: Advection
7: **end for**

3.2. "Initial" End State

Our simulator will accept a polygonal mesh as input for the fluid state at time t_n. This state is chosen by the artist and can be defined at any timestep, t_n, of the simulation, although from here on we will refer to it as the "end" state. Twigg and James [TJ08] discuss a number of limitations for time-reversed simulation that are caused by an end state that is not the result of forward simulation. Their solution is to jitter the end state, which we do for fluids by inputting the target shape as the initial state to the forward simulator and running it for a small number of timesteps. The goal is to let it run long enough to develop texture features, but not so long that it significantly distorts the target shape. The result of this short simulation is then used as the end state for the reverse simulation.

3.3. Reversibility Paradox

The incompressible Euler equations are time-reversible:

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p \tag{4}$$

Dupontcheel et al. [DOW] showed that this property is preserved in fluid solvers that have an energy-preserving integration scheme and accurate time-stepping. Such a solver can be made to step backwards in time by simply reversing the velocity field and running the simulator as usual. Reversing a velocity field that is the result of several timesteps of a forward simulation and running the accurate solver for the same number of timesteps will produce visually plausible behavior, and, if it is accurate enough, can even recover the initial state. However, this requires some assumptions about the initial conditions of the simulator—namely, an end state that is the result of a forward simulation. In our case, we will have no such end state but define one based on the user's input target shape as described in section 3.2. Without such a state, the reverse simulator does not follow the second law of thermodynamics, which states that entropy must always increase. This becomes obvious when the reverse simulation is played forward—disorder grows into order and the motion feels unnatural. Thus, there is an incompatibility between reversible dynamics and irreversible processes. We found that the most useful explanation of the nature of this conflict is due to "an explicitly asymmetric assumption about the way in which real systems were formed in the first place" [Dav77]. Because we are essentially reversing this assumption by starting from a user-defined end state, we will need to introduce a constraint outside of the dynamics to achieve visually plausible reversed behavior.

3.4. Self-Attraction Force

In order to mimic decreasing entropy, we force the fluid to become less spread out over time. We use a self-attraction force similar to Newtonian gravity.

$$F_e = \frac{Em_1m_2}{r^2} \tag{5}$$

We replace the gravitational constant G with a (generally much larger) user-defined constant E that describes how marked the self-attraction will be. Accurately calculating this force for each cell requires $O(h^2)$ calculations where h is the number of fluid cells.

$$\mathbf{f}_e^i = \sum_{j=1}^{h} \frac{Em_im_j(\mathbf{r}_j - \mathbf{r}_i)}{||\mathbf{r}_j - \mathbf{r}_i||^3} \tag{6}$$

We can reduce it to $O(h * log(h))$ by approximating it using a Barnes-Hut tree [BH86]. However, even with the approximation, this step is still a computational bottleneck (see Table 1). We experimented with setting a max traversal depth for the tree and found that in most cases a depth of 1 is all that is needed. In this case, the self-attraction force is reduced to a center of mass calculation $\mathbf{R}$:

$$\mathbf{f}_e^i = \frac{Em_im_{total}(\mathbf{R} - \mathbf{r}_i)}{||\mathbf{R} - \mathbf{r}_i||^3} \tag{7}$$

Adding limited amounts of energy back into the system can actually improve the believability of the fluid flow. This is because, in the forward direction, energy dissipates due to viscosity. In the backward direction, introducing small amounts of energy can have the effect of approximating viscosity.

This force alone will cause the fluid to contract as the reverse simulation progresses. However, there is a limit to how much it can compress due to the incompressibility constraint.

	Barnes-Hut Tree	Center of Mass
Bunny (Fig 1)	5.217 s/f	0.889 s/f
Eurographics (Fig 2)	16.412 s/f	8.017 s/f
Hand (Fig 3)	15.865 s/f	2.719 s/f

Table 1: *Computing the gravitational force with a Barnes-Hut Tree compared with using the center of mass of the object.*

3.4.1. Modified Divergence Projection

Allowing the gas to compress has the positive side effect of reversing gaseous free expansion, which is the process by which gas expands to fill its container. During free expansion, the density of the gas decreases as it takes up more space. In reverse, we want the density to increase as it takes up less space. We accomplish this with the method introduced by Feldman et al. [FOA03] of enforcing non zero divergence to modify fluid behavior. In their case they enforce a positive divergence in order to cause rapid expansion for explosions. In our case, we enforce a negative divergence which will cause the fluid to contract. For traditional incompressible fluids we force the velocity to be divergence free.

$$\nabla \cdot \mathbf{u} = 0 \tag{8}$$

This is done by solving a Poisson equation for a scalar pressure field.

$$\nabla^2 p = \nabla \cdot \mathbf{u}_i \tag{9}$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \nabla p \tag{10}$$

In order to allow the fluid to compress, we force the divergence of the fluid velocity to be equal to the divergence of the self-attraction force F_e (equation 7).

$$\nabla \cdot \mathbf{u} = \nabla \cdot F_e \tag{11}$$

We solve by making the following change to equation 9.

$$\nabla^2 p = \nabla \cdot \mathbf{u} - \nabla \cdot F_e \tag{12}$$

Modifying the divergence in this way allows the fluid to compress at most to the extent defined by the magnitude of the self-attraction force. This has the potential to hurt believability, as a very large self-attraction force will cause the smoke to diminish rapidly and compress into a singularity. However, we found that the force was effective at small magnitudes, generally between 0 and 1. Figure 1 shows such a case.

3.5. Dissipation

In fluid simulation we often want to simulate dissipation, especially when we are modeling steam or mist. We describe a process for creating this effect in a time-reversed simulation. We simulate dissipation in the forward process using the decay model:

$$\rho_{t+1} = max(0, \rho_t - k_d \Delta t) \tag{13}$$

Where k_d is the dissipation constant. For cells with nonzero density, the reversal is trivial:

$$\rho_{t-1} = \rho_t + k_d \Delta t \tag{14}$$

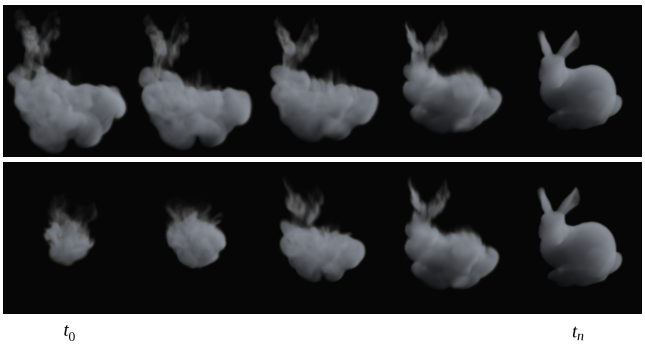

Figure 1: *An example of a time-reversed simulation without (top) and with (bottom) entropy reduction. Both were simulated backwards in time, but here we present them forward in time (as an audience will view them).*

	Reverse	Forward
Bunny (Fig 1)	0.889 s/f	0.717 s/f
Eurographics (Fig 2)	8.017 s/f	6.98 s/f
Hand (Fig 3)	15.865 s/f	2.239 s/f

Table 2: *Comparison of average simulation times for each of our results in seconds per frame. The corresponding forward simulations were set up with the final state of the reverse simulation as the initial state.*

However for cells with zero density there are an infinite number of possible results. Given an empty cell at time t_i, it could have any density value in the range $[0, k_d \Delta t]$ at time t_{i-1}. We need to be able to choose a reasonable value based on context.

We set the initial density values outside the target shape based on the following equation

$$\rho_0 = -\beta \phi_i + \alpha \psi(i) \tag{15}$$

where $\psi(i)$ is a noise function, ϕ_i is the level set value for cell i and β and α are user-adjustable parameters. α defines how much noise to add while β determines how quickly the shape of the fluid shrinks due to dissipation. With $\alpha = 0$, there will be no random variation and density will dissipate uniformly in all directions. High values of β cause the fluid to shrink rapidly while low values cause it to shrink slowly.

4. Results and Conclusion

Figure 1 shows the results of a time-reversed simulation (presented forward in time) of smoke forming the shape of a bunny with and without the self-attraction force. As can be seen, entropy appears to be decreasing in the wrong direction of time in the top sequence, as the shape shrinks over forward time instead of diffusing outward. Our self-attraction force improves this significantly. The bottom sequence was simulated with the max traversal depth of the Barnes-Hut tree set to 0, essentially only adding a center of mass calculation (see timing data in Table 2).

Figure 2 shows an application of time-reversed dissipation. A

Figure 2: *An example of time-reversed dissipation. The smoke gradually dissipates to form words.*

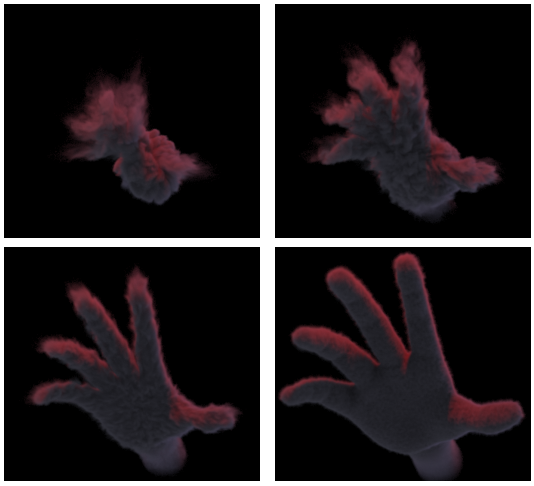

Figure 3: *A large scale example of a time-reversed simulation using our self-attraction force with the full Barnes-Hut Tree.*

cloud of smoke gradually dissipates to form words. This example was generated using the self-attraction force. The reverse dissipation parameters were $\alpha = 0.6$ and $\beta = 0.125$.

Figure 3 shows a large scale example using our self-attraction force. In this case, the force was calculated using the full Barnes-Hut tree, which increased computation time noticeably (see Table 2). Using the full tree improved the result for this broad, complex shape.

We have described a time-reversed simulation method that produces visually plausible behavior. Our method has a number of advantages over existing techniques. With simulation times very close to those of a contemporary forward simulator, it has much better performance than optimal controllers such as [PM17]. It also has the enormous benefit of being able to guarantee an exact match to the target shape, which is a notable drawback of proportional-derivative controllers such as [FL04]. However, this comes with the trade off of being limited to exactly one keyframe. An interesting area of future research would be to investigate how time reversed simulation could be combined with these control methods to minimize this drawback.

References

[BH86] BARNES J., HUT P.: A hierarchical o (n log n) force-calculation algorithm. *Nature 324*, 6096 (1986), 446–449. 2

[Dav77] DAVIES P. C. W.: *The physics of time asymmetry*. Univ of California Press, 1977. 2

[DOW] DUPONCHEEL M., ORLANDI P., WINCKELMANS G.: Time-reversibility of the euler equations as a benchmark for energy conserving schemes. *Journal of Computational Physics 227*, 19. 1, 2

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, pp. 441–448. 1, 4

[FOA03] FELDMAN B. E., O'BRIEN J. F., ARIKAN O.: Animating suspended particle explosions. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 708–715. 3

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 15–22. 1

[MCP*09] MULLEN P., CRANE K., PAVLOV D., TONG Y., DESBRUN M.: Energy-preserving integrators for fluid animation. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 38. 1

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. In *ACM Transactions On Graphics (TOG)* (2004), vol. 23, ACM, pp. 449–456. 1

[PM17] PAN Z., MANOCHA D.: Efficient solver for spacetime control of smoke. *ACM Transaction on Graphics (TOG) 36*, 162 (2017). 1, 4

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 193–202. 1

[SFK*08] SELLE A., FEDKIW R., KIM B., LIU Y., ROSSIGNAC J.: An unconditionally stable maccormack method. *Journal of Scientific Computing 35*, 2 (2008), 350–371. 1

[SS17] STOMAKHIN A., SELLE A.: Fluxed animated boundary method. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 68. 1

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 121–128. 1

[TJ08] TWIGG C. D., JAMES D. L.: Backward steps in rigid body simulation. *ACM Transactions on Graphics (TOG) 27*, 3 (2008), 25. 2

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 716–723. 1