Noname manuscript No.
(will be inserted by the editor)

A Density-Accurate Tracking Solution for Smoke Upresolution

Arnaud Schoentgen · Jonas Zehnder · Pierre Poulin · Bernhard Thomaszewski · Philippe Meseure · Emmanuelle Darles

Received: date / Accepted: date

Abstract Controlling smoke simulations is a notoriously challenging and tedious task, usually requiring many trial-and-error iterations that prevent using expensive computations at high resolutions. Unfortunately, naïvely going from a more efficient low-resolution simulation to a high-quality high-resolution simulation usually results in a different behavior of smoke animation. Moreover, the longer the animation, the more different the result. We propose a tracking procedure where we optimally modify the velocity field of the simulation in order to make the smoke density distribution closely follow the low-resolution density in both space and time. We demonstrate the benefits of our approach by accurately tracking various 2D and 3D simulations. The resulting animations are predictable, preserving the coarse density distribution of the low-resolution guides, while being enhanced with plausible high-frequency details.

Keywords Physics-based animation · Smoke simulation · Fluid control

1 Introduction

Animating natural phenomena is a difficult task with traditional 3D graphics tools. Simulation thus comes as an effective solution to handle complex intricate movements, key to realism. However, it is quite challenging to manipulate initial conditions, physical properties, and propagation in order to achieve a particular animation.

Controlling animations of smoke is a typical example that falls in this category. To achieve a specific targeted smoke simulation, an animator has to face the complexity of fluid dynamics and the lack of intuitive specialized 3D tools for manipulating density fields, not to mention the many art-directed iterations usually required. When performed at high resolutions, this process becomes a very complex and tedious task, aggravated by long simulation times. One potential solution is to perform simulation or art-direction at a lower resolution, before computing the final simulation at high resolution. However, changing the resolution of a simulation domain can affect even the coarse look of a physically based animation, leading to surprising and undesirable behaviors in the final animation.

This paper introduces a method that enables artists to use a low-resolution dense sequence of density distributions to *guide* an Eulerian smoke simulation at a higher resolution. In our method, the velocity field of the high-resolution simulation is optimally modified in order to increase the density matching with the low-resolution simulation. The corresponding high-resolution animation closely follows its low-resolution guide, while being enriched with simulated small-scale details, leading to more predictable and plausible results. For additional flexibility, an artist can also reduce tracking fidelity, giving more freedom to the simulation.

As principal contribution, this paper introduces a new way of tracking a high-resolution smoke using a low-resolution simulation, built on optimization-based density matching. Where previous methods failed to reproduce a density distribution similar to a guide, our method correctly animates the smoke in order to match

Arnaud Schoentgen
Université de Montréal and Université de Poitiers
E-mail: arnaudschoentgen@gmail.com

J. Zehnder, P. Poulin, B. Thomaszewski
Université de Montréal

P. Meseure, E. Darles
Université de Poitiers

the coarse density distribution of the guide, while giving birth to high-frequency simulated details.

We have evaluated our method on several scenarios, different in terms of domain size, animation duration, tracking parameters, and obstacles. We have also applied it to enrich data of real-world dynamic smoke with physically simulated details. Our method proved to faithfully and automatically reproduce guide simulations of smoke density distributions. We have also generated simulations from less realistic animations. As such, our method introduces a basis for tools for artists to generate plausible controlled smoke animations.

2 Previous Work

Fluid animation is a very wide domain of research. We concentrate our discussion on fluid control, which itself has been an active field in computer graphics for many years. The pioneering work of Foster and Metaxas [5] provides fluid control by modifying physical parameters, such as pressure or surface tension, in order to edit fluid behavior. By providing an indirect control over a simulation, their method is not suited when the objective is to accurately make the controlled simulation match a target in both space and time.

Optimal smoke control.

Several methods tackle fluid control by solving a space-time optimization problem over the space of possible control forces constrained by the Navier-Stokes equations. The work of Treuille et al. [26], and later, the computationally improved version from McNamara et al. [12], solve such an optimization problem to find forces used to make the simulation match a very small set of a few sketched keyframes. A gradient-based optimizer is used to solve this problem. Pan and Manocha [18] split this optimization into two subproblems, and apply the alternating direction method of multipliers (ADMM). In those methods, each optimization step requires a full re-simulation. Unfortunately, solving for all frames simultaneously (in parallel) does not scale well with the number of frames in an animation. In contrast, our method performs a re-simulation at higher resolution, i.e., a sequential process that scales better with the number of frames. Moreover, and to the best of our knowledge, this type of approach has not been applied to fluid upresolution yet.

Inglis et al. [8] solve smoke guiding by minimizing an objective function using a first-order Primal-Dual method, prescribing that the guided velocity field should be as close as possible to a target velocity field, both blurred. Their method generates turbulence effects when using large blur radii, but does not accurately follow the simulation target. When used to guide a smoke simulation with a target velocity field extracted from a lower-resolution simulation, this method does not maintain a similar density distribution.

Low- to high-resolution physical simulation.

Among the methods proposed to control physical simulations, several tackle the problem of generating a high-resolution simulation to make it similar to a lower-resolution simulation taken as an input. Bergou et al. [2] introduce TRACKS, a method for deformable surface animations where low- and high-resolution discretizations of a same surface are decomposed into patches with a pairing between low- and high-resolution components. The method constrains each high-resolution patch to make it follow closely the behavior of its low-resolution counterpart, while being enhanced with physically simulated details. An extension is proposed by Milliez et al. [14] for art-directable hair simulation, but nothing similar for smoke.

Several methods have been proposed to perform liquid upresolution. Mercier et al. [13] increase the apparent resolution of a particle-based liquid simulation, adding high-frequency turbulent surface details. Nielsen and Bridson [15] constrain a high-resolution liquid simulation to remain close to a potentially coarser guide simulation by restricting the solve to a thin outer shell of liquid around this guide shape. Thürey et al. [25] introduce a method for guiding Smoothed Particle Hydrodynamics (SPH) and Lattice Boltzmann (LBM) simulations of liquids by replacing the large-scale motions from the simulated velocity with a target velocity. When applied to an Eulerian-based smoke simulation, the guided simulations exhibit undesirable artifacts over time, as pointed out by previous work [8]. Zhang et al. [30] introduce a method to control a position-based liquid in order to match a target defined by a geometrical mesh. Control forces based on springs are used to match a rapidly changing target.

In order to simulate smoke driven by a low-resolution simulation, Nielsen et al. [17,16] solve a minimization problem prescribing that the low-frequency component of the simulated velocity field should be as close as possible to the upsampled guide velocity field. In practice, density variations between the simulation and the guide can be observed after a few advection steps and accumulate over time. This is mainly due to the fact that the amount of density diffusion and density dissipation depends on the resolution, as mentioned by the authors themselves. As this is the closest work related to our goals, we will compare their work to our method in Section 4. Gregson et al. [6] introduce a method for tracking a smoke simulation from real-world tomography data. Their approach, which does not focus on density matching, struggles to maintain a simi-

lar density distribution over time, similarly to Nielsen et al. [17,16]. In a different approach, after interpolating low-resolution data into a fine grid, Liu et al. [11] enhance the appearance of small-scale details through frequency-domain analysis using lifting wavelet decomposition. It is not clear how these details can be controlled to lead to natural behaviors of smoke.

High-resolution fluid synthesis.

A few methods have been proposed to synthesize a high-resolution version of fluid simulations using precomputed data. Sato et al. [20] introduce an efficient data-driven method for synthesizing high-resolution 3D fire from low-resolution fluid simulation data. Xiao et al. [27] present a learning-based flow-correction method for fast previewing based on low-resolution smoke data. Chu and Thürey [3] use deep learning to synthesize high-resolution flow simulations from low-resolution space-time flow data. Good results are achieved with their method. However, large amounts of data and long computation times are required for training. Moreover, the quality of the results depends highly on the dataset used for training. As a consequence, the system thus has to be re-trained in case of a modification of simulation parameters. Furthermore, divergence-free motions cannot be guaranteed. Deep learning has been used for super-resolution smoke with temporal coherence [28]. Methods have also been proposed to perform turbulence [19] or style [9] transfer to a smoke simulation.

Procedural techniques.

Kim et al. [10] use a procedural turbulence function to generate a high-resolution density field from a low-resolution velocity field taken as input. In addition to be divergence-free, the procedural perturbations generated on top of the interpolated velocity field are located in a spectral band guaranteeing the preservation of existing structures. Although their method is simple and elegant, visual results are not always plausible because of the procedural nature of the details added to the smoke simulation. This method has been applied as a "beautification pass" in several papers [6]. Due to its visual quality and popularity, we also compare our method to the one from Kim et al. [10] in Section 4.

Proportional-derivative controllers.

Several methods propose to locally generate empirical forces to reduce an error metric computed at each frame. Shi et al. [21] present a technique to control the density and dynamics of a smoke simulated with an empirical compressible fluid model, so that the synthetic appearance of the smoke resembles a static or a moving object. Taking a low-resolution smoke simulation as an input, Huang et al. [7] sample the simulation domain using match points, before using a correction force to reduce an error estimated at this position. Similarly, Yuan et al. [29] use a guiding force proportional to the velocity error between the upsampled low-resolution and high-resolution velocity fields. These methods are usually computationally inexpensive. Unfortunately, the amount of control generated is not optimal and can lead to oscillatory behaviors around the target. Moreover, to produce good results, the empirical forces used depend on user parameters that have to be modified according to the fluid properties and scene configuration.

To conclude this review of previous work, several methods [17,16,8] modify the velocity field of a smoke simulation to make it similar to a target velocity field. These methods usually lead to an inaccurate density distribution compared to the guide simulation. The density distribution being what we do visualize when rendering smoke, we believe that faithfully reproducing it is crucial when tracking a smoke simulation. In contrast to these methods, our method uses a density matching objective close to the one used in keyframing techniques [26,12,18]. Our method is robust and ensures a good density matching over time between the simulation and the lower-resolution guide, while allowing the simulation to generate small-scale simulated turbulence.

3 Our Method

3.1 Overview

Our goal is to use a lower-resolution simulation (called *guide* simulation) to guide a smoke simulation (called *tracked* simulation), respecting the same global movements and visual appearance while generating physically-plausible smaller-scale details. Unfortunately, a direct re-simulation performed at a higher resolution with the same set of parameters but without any control (called *free* simulation) can lead to unpredictable behaviors, as shown in several examples throughout Section 4. Our proposed method is designed to allow artists to ultimately more easily edit a coarse simulation in order to shorten the art-direction process, confident that a higher-resolution simulation adapted to the artistic vision could be generated.

In designing our controller used to perform smoke tracking, we are guided by three main considerations:

1. Once advected through the controlled velocity field, we want the tracked simulation to be as close as possible to the guide simulation in terms of density distribution.
2. The overall control applied to the tracked simulation should be minimized in order to penalize solutions leading to an over-controlled simulation.

3. The velocity field modification applied to the tracked simulation must be divergence-free in order for the simulation to conserve mass over time.

These requirements give rise to several challenges, addressed by casting the problem as an optimization

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}, \nabla\cdot\boldsymbol{u}=0}{\arg\min}\, \phi(\boldsymbol{u}), \tag{1}$$

solving for an optimal divergence-free velocity modification $\boldsymbol{u}^*$ with an appropriately chosen objective ϕ. The specifics of the objective matter much, and we discuss them in the following sections. Using a continuous optimization solver, we require a spatially-continuous and differentiable advection scheme in order to compute optimal velocity perturbations (Section 3.5.) Furthermore, to achieve reliable density tracking without over-constraining the high-resolution simulation, we must distinguish between fine- and coarse-scale components of the density field in a computationally efficient way (Section 3.3).

Throughout the paper, we refer to the density distribution $\boldsymbol{\rho}$ of simulation type k at time t as $\boldsymbol{\rho}_k^t$, with $k \in \{G, T\}$. Here, G and T refer respectively to the guide and the tracked simulation. Note that $\boldsymbol{\rho}_k^t$ is a fairly large vector that contains all densities of the grid stacked on top of each other. Similarly, we refer to the velocity field $\boldsymbol{v}$ of simulation type k at time t as $\boldsymbol{v}_k^t$. We refer to the blur operator as $\mathbf{B}$, to the projection operator as $\mathbf{P}$, and to the advection of field A through vector field $\boldsymbol{v}$ as $\mathrm{adv}(A, \boldsymbol{v})$.

3.2 Smoke Simulation

Our smoke tracking framework is based on Stable Fluids [22]. In a standard Eulerian smoke simulation, the density and velocity states are iteratively updated using a splitting scheme on the Navier-Stokes equations. We propose to augment the standard simulation loop by inserting an extra *tracking* step, after the projection step and before the advection step. For each simulation time step, our tracking step adds a velocity field modification $\boldsymbol{u}^*$ to the projected velocity field of the tracked simulation, such that its density field becomes as close as possible to the corresponding low-resolution field after advection. The resulting velocity field must be divergence-free, and so, we constrain the solution of the optimization to be divergence-free. Figure 1 gives a schematic overview of the augmented standard smoke simulation loop for our method.

3.3 Density Tracking and Blur

For each simulation step t, the introduced tracking step solves for the velocity field modification $\boldsymbol{u}^*$, minimizing the difference between the advected tracked density $\boldsymbol{\rho}_T^{t+1} = \mathrm{adv}(\boldsymbol{\rho}_T^t, \boldsymbol{v}_T^t + \boldsymbol{u}^*)$ and the next guide density $\boldsymbol{\rho}_G^{t+1}$. Unfortunately, simply minimizing density differences leads to several issues. Since the guide density field has a lower resolution than the tracked simulation, it obviously lacks information to fully determine the desired high-resolution density distribution. Using a frequency argument, we can estimate that the guide tells us what the low-frequency component of the final density distribution should be. To only track the low-frequency component, we apply a low-pass filter $\mathbf{B}$ on the density distributions before measuring the difference between the guide and tracked distributions as

$$\boldsymbol{\rho}_{err} = \mathbf{B}\,\mathrm{adv}(\boldsymbol{\rho}_T^t, \boldsymbol{v}_T^t + \boldsymbol{u}) - \mathbf{B}\,\boldsymbol{\rho}_G^{t+1}, \tag{2}$$

where $\boldsymbol{\rho}_G^{t+1}$ is upsampled to the resolution of the tracked simulation using nearest-neighbor interpolation. In practice, $\mathbf{B}$ is a separable normalized Gaussian kernel. We estimate the correct size of this kernel to be $1.5\times$ the size of the low-resolution grid cell. An illustration of the visual impact of the blur over the tracked simulation is found in Figure 2.

3.4 Objective Function and Derivative

In this section, we define the objective function to minimize as part of our smoke tracking problem, as well as its analytical derivative.

Our objective function is a weighted sum of a density tracking term ϕ_m, a regularization term ϕ_r, and a perturbation smoothness term ϕ_g, leading to an objective of the form:

$$\phi = k_m\phi_m + k_r\phi_r + k_g\phi_g. \tag{3}$$

As stated in Section 3.3, we minimize the blurred difference between the tracked and guide density distributions. We minimize the norm of this mismatch, giving the corresponding density matching objective

$$\phi_m = \frac{1}{2}\left\|\boldsymbol{\rho}_{err}\right\|^2 = \frac{1}{2}\left\|\mathbf{B}\,\mathrm{adv}(\boldsymbol{\rho}_T^t, \boldsymbol{v}_T^t + \boldsymbol{u}) - \mathbf{B}\,\boldsymbol{\rho}_G^{t+1}\right\|^2 \tag{4}$$

and the gradient

$$\frac{\partial\phi_m}{\partial\boldsymbol{u}}^T = \frac{\partial\mathrm{adv}(\boldsymbol{\rho}_T^t, \boldsymbol{v}_T^t + \boldsymbol{u})}{\partial\boldsymbol{u}}^T \mathbf{B}^T\,\boldsymbol{\rho}_{err}. \tag{5}$$

Since the use of a non-zero $\boldsymbol{u}$ makes the simulation deviate from a free, realistic smoke simulation, we want

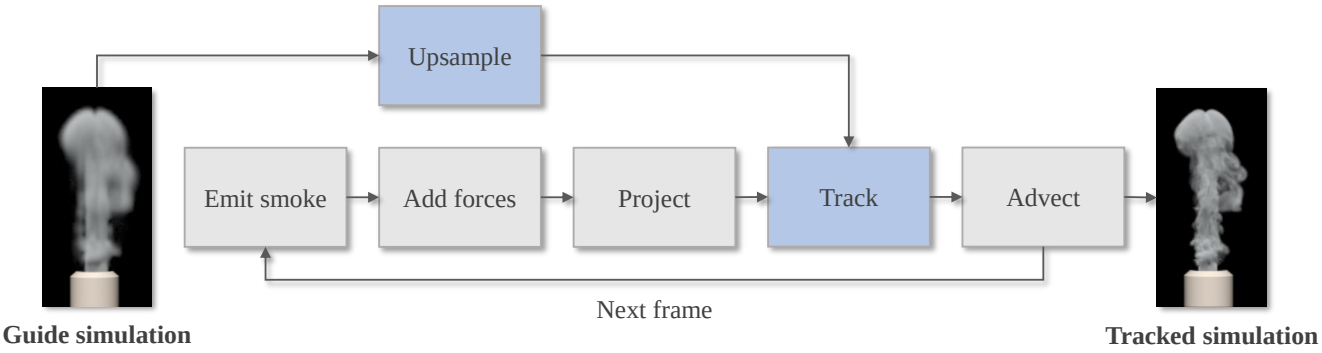

Fig. 1 Schematic overview of our tracking method. In blue are our added steps over a standard Eulerian smoke simulation.

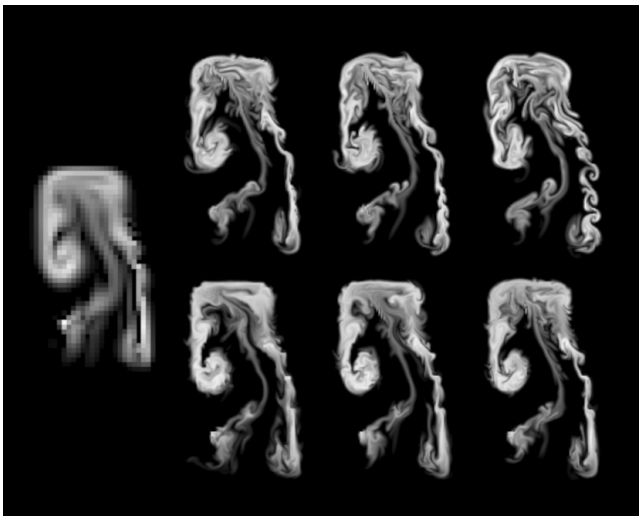

Fig. 2 Visual impact of the blur radius. Left: guide. Right: tracked with blur radius $r \in \{0, 2, 4, 6, 8, 12\}$, in the usual order. Note that r is given in terms of high-resolution grid cell size, with an upresolution factor of 4.

to foster solutions leading to a minimal modification of the velocity field. We thus add a regularization term to our objective, corresponding to the L^2 norm of $\boldsymbol{u}$:

$$\phi_r = \frac{1}{2} \|\boldsymbol{u}\|^2 \quad \text{with} \quad \frac{\partial \phi_r}{\partial \boldsymbol{u}}^T = \boldsymbol{u}. \tag{6}$$

Minimizing a linear combination of these two terms may lead to a highly spatially-varying solution. In order to increase the spatial smoothness of the vector field used to control the tracked simulation, we add a smoothness term to our objective, corresponding to the L^2 norm of the gradient of the velocity field modification, and given by

$$\phi_g = \frac{1}{2} \|\nabla \boldsymbol{u}\|^2 \quad \text{with} \quad \frac{\partial \phi_g}{\partial \boldsymbol{u}}^T = \nabla^T (\nabla \boldsymbol{u}). \tag{7}$$

A gradient-based optimization method is used to perform the solve. The gradient of the objective function can be trivially expressed in terms of the derivatives given by Equations (5), (6), and (7):

$$\frac{\partial \phi}{\partial \boldsymbol{u}} = k_m \frac{\partial \phi_m}{\partial \boldsymbol{u}} + k_r \frac{\partial \phi_r}{\partial \boldsymbol{u}} + k_g \frac{\partial \phi_g}{\partial \boldsymbol{u}}. \tag{8}$$

An illustration of the visual impact of the objective weights can be found in Figure 3.

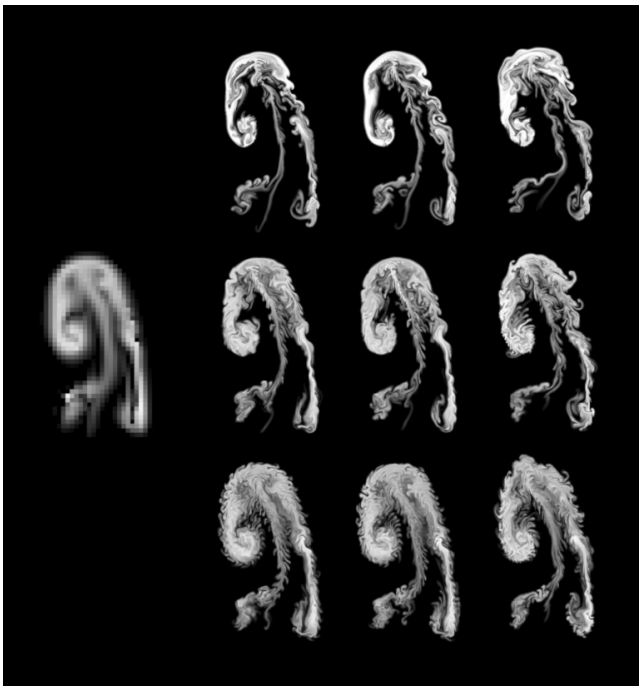

Fig. 3 Visual impact of the regularization weight k_r and the space-variation weight k_g. Left: guide. Right: tracked with $k_r \in \{0.0, 0.001, 0.03\}$ horizontally, and $k_g \in \{0.01, 0.1, 0.5\}$ vertically, in the usual order. Since the optimization is a proportion of three weights, we fixed arbitrarily one factor, here $k_m = 1.0$, in all our examples.

3.5 Advection Scheme Differentiability

To solve the optimization problem using a gradient-based method, we must compute the derivative of the advection operator $\text{adv}(A, \boldsymbol{v})$ with respect to the vector field $\boldsymbol{v}$, which implies that this operator has to be C^1-continuous in $\boldsymbol{v}$.

Although our method could be used with several types of advection schemes, we use a first-order semi-Lagrangian advection scheme. This choice is motivated by the simplicity to differentiate the first-order semi-Lagrangian advection scheme, its stability, and its low diffusivity compared to a purely Eulerian scheme. Linear interpolation has been used frequently when performing spatial density interpolation occurring in the semi-Lagrangian advection scheme. Unfortunately, the use of a linear interpolation does not lead to a differentiable advection operator. In order to enforce C^1-continuity of the advection operator, we use cubic Hermite splines to perform density interpolation. Tangents are enforced in each spatial dimension on both sides of a cell for the advected density field to be C^1-continuous. For each cell of the grid, an analytical expression of the density value obtained after advection can be expressed in terms of the position of the grid cell center (i, j), the velocity components $u[i, j]$, $u[i + 1, j]$, $v[i, j]$, $v[i, j + 1]$, the simulation time step dt, and the density values stored in the $4 \times 4 = 16$ cells surrounding the position of the traced-back position of the center of (i, j) using cubic Hermite splines. We then take the derivative of this expression with respect to each velocity component, in order to apply the Jacobian matrix of the advection operator. Although we explained above for the 2D case for the sake of clarity, we can trivially extend this method in 3D, considering a third component for both positions and vectors, and using the $4 \times 4 \times 4 = 64$ cells surrounding the position of the traced-back position. Note that we discretize the different equations using a standard staggered MAC grid, storing velocities on cell faces, and both density and pressure in cell centers.

3.6 Divergence-free Constraint Enforcement

To satisfy the divergence-free constraint, we solve for a general velocity field $\tilde{\boldsymbol{u}}$ that has non-zero divergence, but we remove the divergence before adding the velocity perturbation to the simulation, leading to the equivalent optimization problem

$$\tilde{\boldsymbol{u}}^* = \arg\min_{\tilde{\boldsymbol{u}}} \phi(\mathbf{P}\tilde{\boldsymbol{u}}), \qquad \boldsymbol{u}^* = \mathbf{P}\tilde{\boldsymbol{u}}^*. \tag{9}$$

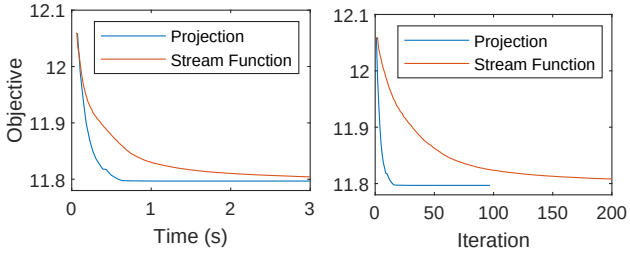

Fig. 4 Convergence of the optimization for one tracking step. We compare two different parameterizations of the divergence-free velocity field on the 100th time frame of the 2D example in Figure 7, starting with the same velocity and density fields.

The gradient is then given by

$$\frac{\partial \phi}{\partial \tilde{\boldsymbol{u}}} = \mathbf{P}^T \frac{\partial \phi}{\partial \boldsymbol{u}}. \tag{10}$$

An alternative would be to use a stream function [1] by defining $\boldsymbol{u} = \nabla \times \Psi$, with $\nabla\times$ being the curl operator; this velocity field is always divergence-free and thus does not require pressure projection. However, we found that this was not more efficient, as can be observed in Figure 4. Even though every iteration is cheaper, since no projection is necessary, many more iterations are required. We suspect that this is because changing exactly one velocity value u_i requires global changes in Ψ, as u_i is defined by the curl of Ψ, and thus discretized by differences of values of Ψ at different locations. Therefore the variables in the stream function parameterization are highly coupled, making it hard to optimize the objective.

3.7 Implementation

We solve the non-linear optimization problem using L-BFGS. It increases memory consumption, but has faster convergence than a pure gradient-based method, and does not require the computation of the Hessian matrix, unlike Newton's method. L-BFGS stores a set of vectors to approximate the Hessian matrix, making it well suited for high-dimensional problems such as our case. We have not experienced unstable behaviors. Our method has been implemented as a *Mantaflow* [24] plugin. We use OpenMP for parallelization.

4 Results

We applied our smoke tracking method on several scenarios. All the simulations were run on a desktop PC with an i9-9900K 8-core CPU at 3.60 GHz with 128 GB of memory. In these examples, identical time steps were used on the low- and high-resolution simulations. We

used $k_m = 1.0$, $k_r = 0.001$, $k_g = 0.1$, and a blur radius equivalent to 1.5× the size of the low-resolution grid cell. The set of weights has been found experimentally, corresponding to a good trade-off between the natural look of the high-resolution simulation and good density tracking with the guide. Those weights have been used to generate all the examples presented in this paper. All images in the paper are part of animations accessible in supplementary material.

4.1 Scenarios

2D Smoke Plume

As a first scenario, a plume in 2D, emitted from the bottom of the scene, spreads freely in the simulation domain. After computing a low-resolution guide simulation, we generated different simulations using our method with increasing upresolution factors, ranging from 2× to 16×. Figure 5 shows an image from each of these simulations, taken at an identical time. The high-resolution simulations generated using our method follow closely the guide regardless of the upresolution factor, while being enhanced with adapted high-frequency simulated details.

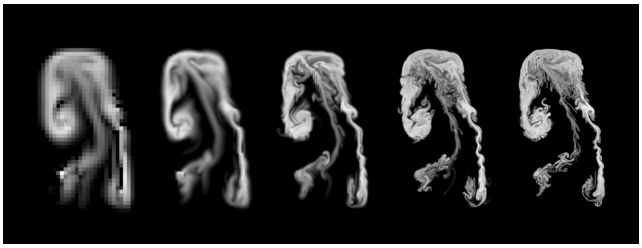

Fig. 5 2D rising plume. Left to right: guide (32×48), tracked 2×, 4×, 8×, and 16×. The larger the upresolution factor, the finer the added simulated details.

The blocky aspect of the density distribution of the guide is indicative of its coarse resolution. Recall that the low-resolution density is upsampled to the resolution of the tracked simulation when evaluating the density error.

It is important to note that we include 2D scenarios mainly to simplify the observation in a 2D image, and to better point out differences. 3D scenarios suffer from mixing smoke at different depths in a pixel, affected by absorption or occlusion. However, a 3D animation of smoke looks much more natural than its 2D counterpart, as can be appreciated in our images and animations.

3D Smoke Plume

Our method is trivially extended to 3D. In the next scenario, a smoke emitter placed in a pipe generates a rising plume in a 3D box domain. Buoyancy forces are applied. We performed three simulations. At first, a low-resolution ($32 \times 48 \times 32$) simulation, then two high-resolution ($128 \times 196 \times 128$) simulations: a free simulation and a simulation tracked with our method using the low-resolution simulation as a guide.

Figure 6 depicts an image from each of these simulations at an identical time. Even in this simple scenario, we can notice the different coarse appearances between the guide and free simulations. Indeed, the plume from the free simulation tends to rise faster, while the one from the guide appears slightly thicker. In contrast, the plume generated with our method in the tracked simulation exhibits the same coarse appearance than the one from the guide simulation, while being enhanced with high-frequency turbulent details.

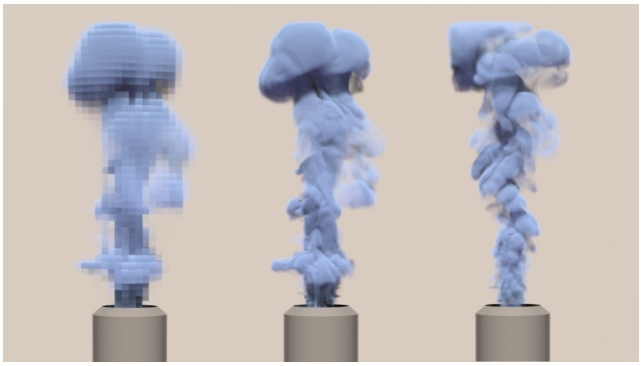

Fig. 6 3D rising plume. Left to right: guide, tracked, free.

Comparison with Previous Work

We compared our method with two previously discussed methods. First, we consider the tracking method of Nielsen et al. [17], which shares our general goals of control. As stated earlier, their method introduces a custom projection step, prescribing that a smoothed version of the simulated velocity field should be as close as possible to the input guide velocity field.

In a simple 2D smoke plume scenario, we observe that while both our method and the one from Nielsen et al. [17] correctly track the coarse behavior of the smoke, the simulation tracked using our method turns out to be much closer to the guide in terms of density distribution. Since the density distribution is the visible part of a smoke simulation, we found this error metric to be of interest. We support this observation by computing the RMS error between the guide and tracked blurred density fields. The blur operator is the same than the one used in our objective function, as it provides a reasonable approximation of a correct low-pass filter. Figure 7 shows an image from each of these simulations at an identical time frame, as well as the corresponding

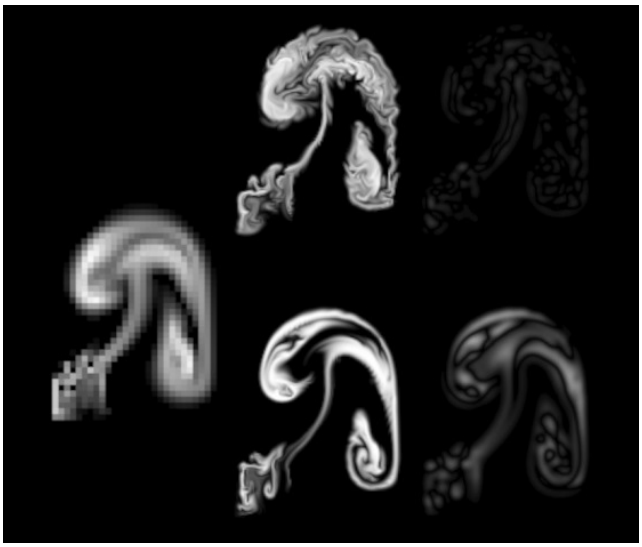

Fig. 7 2D comparison with Neilsen et al. [17]. Left: guide. Top center: tracked using our method. Bottom center: tracked using Nielsen et al. [17]. Right: RMS error for both methods.

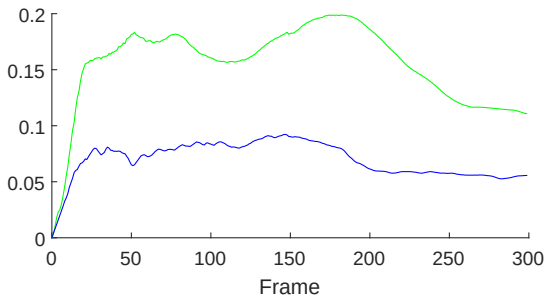

Fig. 8 RMS error per frame of the 2D simulations depicted in Figure 7. Blue: our method. Green: Nielsen et al. [17].

RMS error distribution. We also summed these errors and divided by the number of occupied pixels, before plotting them in the graph of Figure 8. It shows that the density distribution RMS error is roughly half for our method.

We performed a 3D comparison of our method with both Nielsen et al. [17] and Kim et al. [10] by tracking a 3D rising plume. We observe in Figure 9 that the density distribution of the tracked simulation generated using our method is visually closer to the guide. Again, note that 3D smoke density distribution appears more natural than their 2D counterparts. Faithfully representing density distribution of the guide density distribution is crucial in order to get a predictable simulation after the tracking step. In the scene displayed in Figure 10, we cast shadows from spotlights, of the previous smoke simulations. Compared to the simulation generated using Nielsen et al. [17], the emphasized shadows using our method are closer to the shadows generated by the guide. Working with density distributions from guide simulations could allow an artist to perform art-direction over a low-resolution density simulation in order to obtain targeted shadowing effects, being confident that the coarse aspect of the shadows generated by the tracked simulation would be similar.

Compared to the method of Nielsen et al. [17], we explain differences in terms of high-resolution density distribution as follows. In the spatial interpolation step of a semi-Lagrangian advection, we are taking a weighted average of values from the previous time step. Averaging tends to smooth out sharp features, diffusing or dissipating them. The accuracy of the semi-Lagrangian advection depends on the resolution of the grid used to perform spatial interpolation. Indeed, the higher the resolution of the grid, the lower the diffusion. As a limitation to their method, Nielsen et al.. [17] themselves stated that: “the amount of density-diffusion and -dissipation is significantly higher in low resolution. For this reason, features in the high-resolution guided simulation may deviate in intensity from the low-resolution simulation”. For obvious reasons, our approach consisting in directly tracking the potentially highly-diffused low-resolution density field will result in a high-resolution density field more similar to the one from the guide.

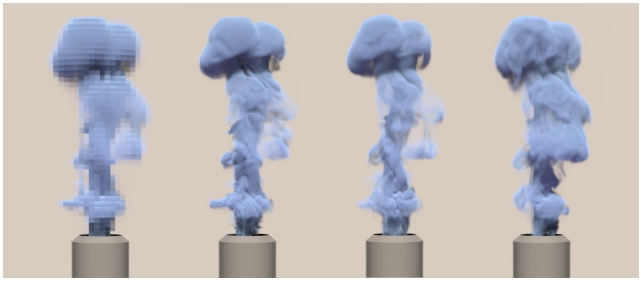

Fig. 9 Comparison with different methods. From left to right: guide, ours, Nielsen et al. [17], Kim et al. [10].

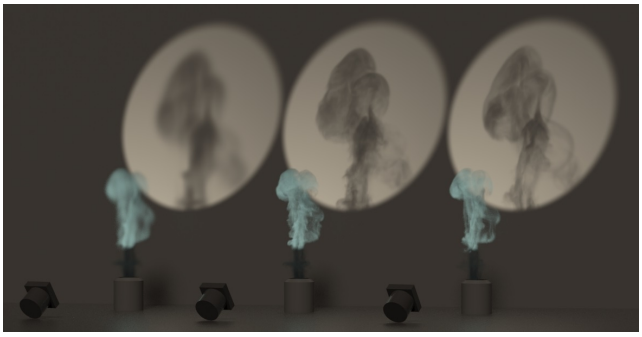

Fig. 10 3D comparison under a different lighting. From left to right: guide, ours, Nielsen et al. [17].

Smoke Interacting with Obstacles

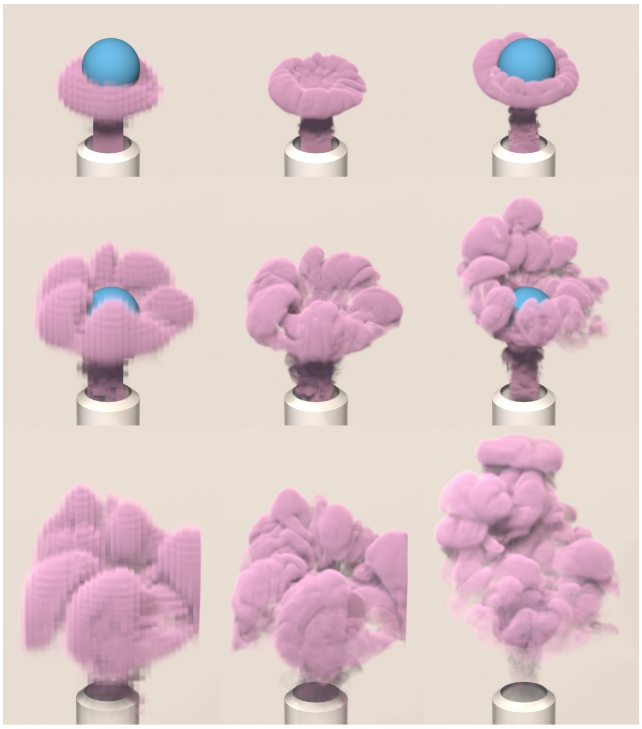

Fig. 11 Simulating with and without a sphere. Vertical: Images at three different times. Left: guide with the sphere. Center: tracked with our method without the sphere. Right: free with the sphere.

Our method can also be used to track a guide simulation in which smoke interacts with objects. In the scenario depicted in Figure 11, a rising plume interacts with a static sphere. The tracked simulation, generated using our method and simulated without any obstacle in the scene, follows closely the motion of the guide smoke interacting with the obstacle, while giving birth to high-frequency plausible motion details. We can observe again how much the free simulation deviates from the guide simulation.

The fidelity to the guide simulation near boundaries is only achieved thanks to the density matching term, which is one component of our objective. However, tracking the guide density usually does not lead to exact solid boundary conditions. In some scenarios, one could prefer to accurately represent the frontiers between smoke and obstacles. In order to increase the spectrum of possibilities in terms of art direction, the use of boundary condition enforcement in our objective is left as a choice to the artist. This is illustrated in the scenario displayed in Figure 12, where a rising plume interacts with a set of three horizontal cylinders of different radii at different heights. Two tracked simulations have been generated. A first one in which the smoke is only tracked using the guide, and a second one in which the cylinders are added as boundaries in the tracked simulation. The shapes of the cylinders in the former simulation are still visible, even though their silhouettes are not as sharply delimited as with the actual cylinders used as obstacles.

The obstacles do not have to remain static. In Figure 13, a dragon is moved horizontally to create a cushion of smoke underneath it. Our method follows well the guide, while a free animation at the same high resolution results in a much different distribution of the smoke.

Fig. 12 Simulating with and without three cylinders (not displayed). Top left: guide. Tracked simulations using our method without (top right) and with (bottom left) boundary conditions enforced. Bottom right: free with the cylinders, exhibiting a much different smoke distribution.

Tracking Captured Data

We demonstrate the interest and robustness of our method by enriching captured data with physically simulated details. We used real-world reconstructed density distributions from Eckert et al. [4]. Since the original resolution ($100 \times 178 \times 100$) of these simulations is currently impracticable (on our computer) to be used as an input to our method, we generated a guide simulation by downsampling by half the density distribution

Fig. 13 Smoke interacting with an animated mesh. Left: A low-resolution smoke plume interacts with a dragon mesh moving horizontally, forming a smoke cushion underneath the model. Center: A high-resolution smoke plume generated with our method. Right: The corresponding high-resolution free simulation behaves much differently, quickly surrounding the dragon.

of a database simulation. We then computed a tracked simulation at a $200 \times 356 \times 200$ resolution using the generated guide. The resulting animation, thus generated in a grid twice as fine than the original data, closely follows the guide in both space and time, while being enriched with physically simulated small-scale details. In this example, faithfully reproducing the density is crucial since a deviation from the guide can be seen as a failure to respect the captured data.

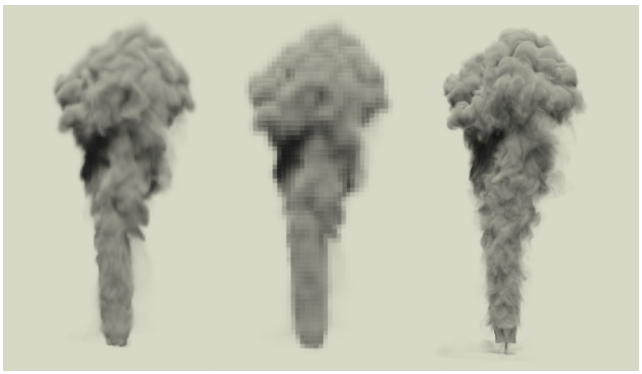

Fig. 14 Simulation guided by captured data from Eckert et al. [4]. From left to right: original captured data ($100 \times 178 \times 100$), guide downsampled from Eckert et al. ($50 \times 89 \times 50$), tracked ($200 \times 356 \times 200$) with our method.

Art-directed Guide

Since we only use a density distribution to guide a high-resolution simulation, our method is compatible with the use of non-physical and/or edited inputs. We demonstrated this concept in two scenarios.

First, we generated a non-physical scenario, depicted in Figure 15. In this scenario, we generated two plumes of smoke, modifying for each of them the orientation of the buoyancy forces to make them rise toward the ascending diagonal. We generated a guide by summing up the density values at each cell of the grid, before using the resulting data to guide a high-resolution simulation. Although the guide could not be physically simulated into a single simulation, our method is able to generate a plausible simulation, matching the guide. We believe that the composition of a low-resolution guide with multiple animations, followed by the generation of a high-resolution physically-based simulation thanks to our tracking method, is an interesting direction to control fluids. Indeed in image edition, artists are familiar with the generation of a targeted result by combining layers. For obvious reasons, this type of guiding could not have been achieved by tracking the sum of velocity fields using the method of Nielsen et al. [17].

In a second example, depicted in Figure 16, a thin plume of smoke is generated and distorted to follow a spiral when rising. The parametric function used to perform the distortion allows us to easily generate a targeted shape from a few parameters. A high-resolution smoke simulation is then generated using this non-physical simulation as a guide. Although the guide is highly aliased and non-physical, our method is able to generate a detailed smoke simulation that follows its general shape.

4.2 Statistics

Table 1 gives statistics about the simulations presented in this section. In these scenarios, high-resolution tracked and free simulations are $4\times$ larger in every dimension than their respective low-resolution guide. An equivalent free simulation (same initial conditions and dimensions) takes 0.06s per frame in our 2D examples, and between 1.3s and 1.7s in our 3D examples; the associated tracked simulation takes 1.3s per frame in 2D, and between 60s to 80s per frame in 3D. The reported memory usage is directly related to the size of the tracked simulation dimensions.

It is obvious that computing a free simulation, without any optimization, is much faster than a tracked simulation. Using an optimization scheme such as L-BFGS on large sets of variables comes with an important computational cost. From Table 1, we observe that our tracked simulations took between about 35–100$\times$ the computation times of equivalent free simulations in 3D. However, such free simulations do not follow any guide, and thus do not satisfy our goals.

We implemented the approach of Nielsen et al. [17] using the biconjugate gradient stabilized method (BiCGSTAB) with a diagonal preconditioner. A frame of the simulation depicted in Figure 9 took on average 123s to compute, compared to 76s for our method. Unfortunately, we did not implement the multi-grid solution described by Nielsen et al. [17], and so we cannot speculate about a fair full comparison.

Fig. 15 Merging two rising 3D plumes. Left: The two independent guide simulations. Center: The combined two guide simulations. Right: The tracked simulation generated using our method.

Example	Figure	Grid resolution		Avg. time per frame (s)		Comp. Factor	Memory (GB)
		Guide	Track	Track	Free	Track / Free	
2D plume	7	32×48	128×192	1.32	0.06	22.0×	0.05
3D plume	6	$32 \times 48 \times 32$	$128 \times 192 \times 128$	76.6	1.32	58.0×	6.88
Sphere	11	$32 \times 48 \times 32$	$128 \times 192 \times 128$	83.3	1.42	58.7×	6.88
Sphere (with object)	—			72.7		51.2×	
Cylinders	—	$32 \times 48 \times 32$	$128 \times 192 \times 128$	63.1	1.68	37.6×	6.88
Cylinders (with object)	12			61.6		36.7×	
Dragon (with object)	13	$32 \times 48 \times 32$	$128 \times 192 \times 128$	71.18	1.83	38.9×	6.88
Real-world data	14	$50 \times 89 \times 50$	$200 \times 356 \times 200$	679.6	6.84	99.4×	31.9
Combined plumes	15	$32 \times 48 \times 32$	$128 \times 192 \times 128$	80.3	—	—	6.88
Magic teapot	16	$32 \times 48 \times 32$	$128 \times 192 \times 128$	83.4	—	—	6.88

Table 1 Statistics.

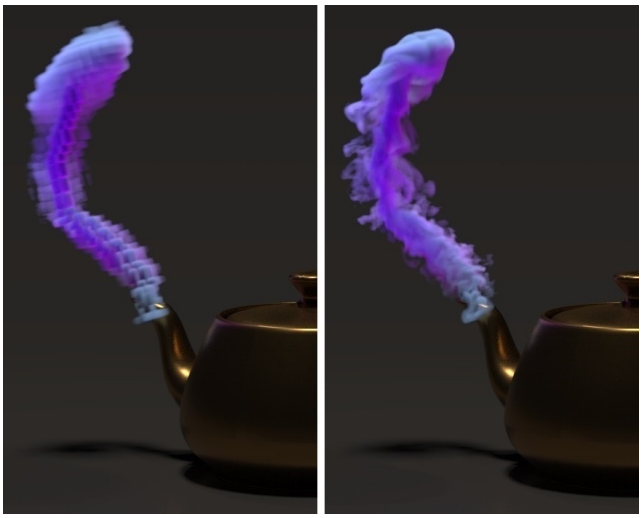

Fig. 16 Non-physical edited smoke, rising from a magic teapot. A low-resolution rising plume is distorted to follow a spiral using a parametric function. A high-resolution simulation, tracking the guide, is generated using our method.

While there is room to improve our code, or optimizing with ADMM [18] could offer better performance, the increased computation factor due to our method should remain important given the size of the problem. For instance, our largest optimization contains close to 43 million values to optimize for. Fortunately, in the context of developing smoke edition tools and/or choosing between a number of simulations with different conditions, such a range of computational factors seems acceptable because it requires no artist interventions to generate a final simulation which closely follows the guide.

In our tests and when the scenes where carefully designed (e.g., emitters located at the same position in space), we did not observe any stability issue with our method. However, we observed slower convergence when the guide is further than physically plausible. The fact is that larger velocity modifications are required to make the high-resolution simulation match the guide.

5 Conclusions and Future Work

We have presented a method that uses a low-resolution simulation of smoke in order to guide a high-resolution simulation. We have demonstrated in a number of scenarios that the simulations generated using our method follow the global movements of the low-resolution simulation, while allowing the generation of fine details necessary to increase realism. By optimizing the tracking

of density instead of the associated velocity field, we observe that our simulations are visually and numerically closer to the guide, and their appearances in 3D are satisfying. Our method can thus be used to automatically and robustly increase the resolution of simulated, non-physical, and captured smoke animations. In combination with edition tools for smoke animations, we believe that working on lower-resolution 3D data will be more affordable for artists. As such, our method offers great potential to shorten the iterative loop of artist-designed smoke animations, thanks to the simulation at a more efficient lower resolution.

We have used an Eulerian discretization as the basis for our fluid tracking framework. In principle, our formulation should apply directly to Lagrangian approaches such as SPH when using a grid as intermediate representation, similar in spirit to MPM [23]. However, some applications might not allow for this memory-intensive detour, making a particle-level reformulation of our method another exciting direction for future work.

Compliance with Ethical Standards

The authors declare that there is no known conflict of interests with respect to this work. Pierre Poulin acknowledges financial support from NSERC and the Université de Montréal.

References

1. Ando, R., Thuerey, N., Wojtan, C.: A stream function solver for liquid simulations. ACM Trans. Graph. **34**(4), Art. 53 (2015)
2. Bergou, M., Mathur, S., Wardetzky, M., Grinspun, E.: TRACKS: Toward directable thin shells. ACM Trans. Graph. **36**(3), Art. 50 (2007)
3. Chu, M., Thuerey, N.: Data-driven synthesis of smoke flows with CNN-based feature descriptors. ACM Trans. Graph. **36**(4), Art. 69 (2017)
4. Eckert, M.L., Um, K., Thuerey, N.: Scalarflow: A large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. ACM Trans. Graph. **38**(6), Art. 239 (2019)
5. Foster, N., Metaxas, D.: Controlling fluid animation. In: Proc. Computer Graphics International, pp. 178–188. IEEE (1997)
6. Gregson, J., Ihrke, I., Thuerey, N., Heidrich, W.: From capture to simulation: Connecting forward and inverse problems in fluids. ACM Trans. Graph. **33**(4), Art. 139 (2014)
7. Huang, R., Melek, Z., Keyser, J.: Preview-based sampling for controlling gaseous simulations. In: ACM SIGGRAPH/Eurographics Symp. on Computer Animation, pp. 177–186 (2011)
8. Inglis, T., Eckert, M.L., Gregson, J., Thuerey, N.: Primal-dual optimization for fluids. Comput. Graph. Forum **36**(8), 354–368 (2017)
9. Kim, B., Azevedo, V.C., Gross, M., Solenthaler, B.: Transport-based neural style transfer for smoke simulations. ACM Trans. Graph. **38**(6) (2019)
10. Kim, T., Thürey, N., James, D., Gross, M.: Wavelet turbulence for fluid simulation. ACM Trans. Graph. **27**(3), Art. 50 (2008)
11. Liu, S., Xu, Y., Noh, J., Tong, Y.: Visual fluid animation via lifting wavelet transform. Computer Animation and Virtual Worlds **25**(3-4), 473–483 (2014)
12. McNamara, A., Treuille, A., Popović, Z., Stam, J.: Fluid control using the adjoint method. ACM Trans. Graph. **23**(3), 449–456 (2004)
13. Mercier, O., Beauchemin, C., Thuerey, N., Kim, T., Nowrouzezahrai, D.: Surface turbulence for particle-based liquid simulations. ACM Trans. Graph. **34**(6), Art. 202 (2015)
14. Milliez, A., Sumner, R., Gross, M., Thomaszewski, B.: HairControl: A tracking solution for directable hair simulation. Comput. Graph. Forum **37**(8), 115–123 (2018)
15. Nielsen, M.B., Bridson, R.: Guide shapes for high resolution naturalistic liquid simulation. ACM Trans. Graph. **30**(4), Art. 83 (2011)
16. Nielsen, M.B., Christensen, B.B.: Improved Variational Guiding of Smoke Animations. Comput. Graph. Forum **29**, 705–712 (2010)
17. Nielsen, M.B., Christensen, B.B., Bin Zafar, N., Roble, D., Museth, K.: Guiding of smoke animations through variational coupling of simulations at different resolution. In: ACM SIGGRAPH/Eurographics Symp. on Computer Animation, pp. 217–226 (2009)
18. Pan, Z., Manocha, D.: Efficient solver for spacetime control of smoke. ACM Trans. Graph. **36**(5), Art. 162 (2017)
19. Sato, S., Dobashi, Y., Kim, T., Nishita, T.: Example-based turbulence style transfer. ACM Trans. Graph. **37**(4), Art. 84 (2018)
20. Sato, S., Morita, T., Dobashi, Y., Yamamoto, T.: A data-driven approach for synthesizing high-resolution animation of fire. In: Proc. Digital Production Symp., p. 37–42 (2012)
21. Shi, L., Yu, Y.: Controllable smoke animation with guiding objects. ACM Trans. Graph. **24**(1), 140–164 (2005)
22. Stam, J.: Stable fluids. In: Proc. ACM SIGGRAPH, pp. 121–128 (1999)
23. Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A.: A material point method for snow simulation. ACM Trans. Graph. **32**(4), Art. 102 (2013)
24. Thuerey, N., Pfaff, T.: Mantaflow (2016). URL http://mantaflow.com
25. Thürey, N., Keiser, R., Pauly, M., Rüde, U.: Detail-preserving fluid control. In: Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation, pp. 7–12 (2006)
26. Treuille, A., McNamara, A., Popović, Z., Stam, J.: Keyframe control of smoke simulations. ACM Trans. Graph. **22**(3), 716–723 (2003)
27. Xiao, X., Wang, H., Yang, X.: A CNN-based flow correction method for fast preview. Comput. Graph. Forum **38**(2), 431–440 (2019)
28. Xie, Y., Franz, E., Chu, M., Thuerey, N.: tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. ACM Trans. Graph. **37**(4), Art. 95 (2018)
29. Yuan, Z., Chen, F., Zhao, Y.: Pattern-guided smoke animation with Lagrangian coherent structure. ACM Trans. Graph. **30**(6), Art. 136 (2011)
30. Zhang, S., Yang, X., Wu, Z., Liu, H.: Position-based fluid control. In: Proc. Symp. on Interactive 3D Graphics and Games, pp. 61–68 (2015)