From Capture to Simulation - Connecting Forward and Inverse Problems in Fluids

James Gregson, Ivo Ihrke, Nils Thuerey, Wolfgang Heidrich

From Capture to Simulation – Connecting Forward and Inverse Problems in Fluids

James Gregson[1] Ivo Ihrke[2] Nils Thuerey[3] Wolfgang Heidrich[1,4]

[1] University of British Columbia [2] INRIA Bordeaux / LP2N
[3] TU München [4] King Abdullah University of Science and Technology

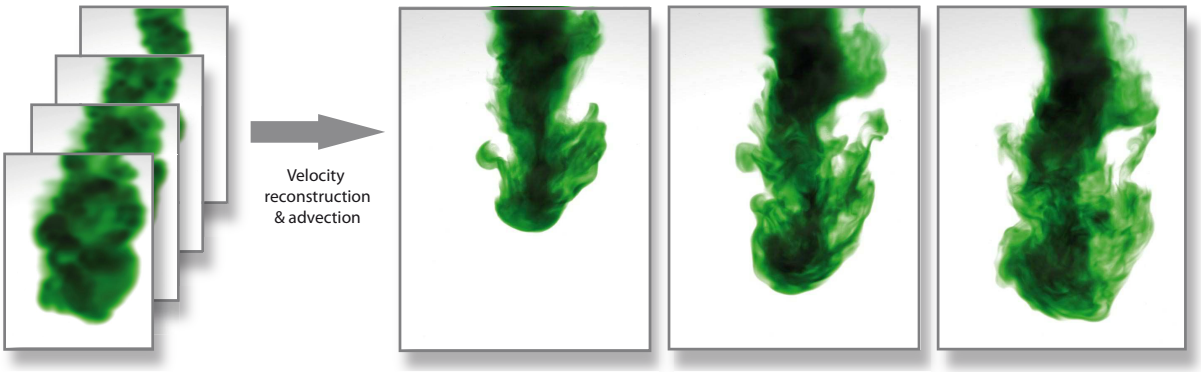


Figure 1: *Low-resolution captures obtained by tomographic scanning (left) are used as inputs to our method which estimates physically plausible dense velocity fields. Such velocity fields fully determine the fluid state and can be applied in a variety of applications including fluid super-resolution (right) allowing capture to be integrated into pipelines for visual effects simulation.*

Abstract

We explore the connection between fluid capture, simulation and proximal methods, a class of algorithms commonly used for inverse problems in image processing and computer vision. Our key finding is that the proximal operator constraining fluid velocities to be divergence-free is directly equivalent to the pressure-projection methods commonly used in incompressible flow solvers. This observation lets us treat the inverse problem of fluid tracking as a constrained flow problem all while working in an efficient, modular framework. In addition it lets us tightly couple fluid simulation into flow tracking, providing a global prior that significantly increases tracking accuracy and temporal coherence as compared to previous techniques. We demonstrate how we can use these improved results for a variety of applications, such as re-simulation, detail enhancement, and domain modification. We furthermore give an outlook of the applications beyond fluid tracking that our proximal operator framework could enable by exploring the connection of deblurring and fluid guiding.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and Scanning

Keywords: fluid-capture, velocity estimation, optimization

Links: DL PDF

1 Introduction

Fluid simulation and fluid capture are active research areas in computer graphics. Due to difficulties in acquiring full flow descriptions, the use of captured data in simulations has been limited [Wang et al. 2009; Li et al. 2013].

We aim at decreasing this gap by showing a connection between proximal methods and fluid tracking as well as simulation. Proximal methods are a class of recent, successful numerical methods from the image processing/computer vision community that split difficult multi-objective optimization schemes into simpler sub-problems. These sub-problems are expressed and accessed only by application-specific solvers known as proximal operators. This approach encourages modularity and the development of highly optimized solvers for each sub-problem.

Our key finding is that the pressure solve employed in many fluid simulators can be interpreted as a proximal operator. This enables its incorporation as a physical constraint into proximal methods. We demonstrate the flexibility of this approach by applying it to the problem of fluid tracking where it allows for an elegant inclusion of prior knowledge into the velocity estimation procedure. In particular, we couple the Navier-Stokes equations into the estimation procedure, which enables significant improvements in tracking accuracy as compared to state-of-the-art techniques.

Having an accurate velocity field of the flow phenomena enables flexible new applications, such as applying fluid super-resolution techniques to the captured flows, as well as re-simulations in modified domains. In this way we demonstrate a meaningful integration of captured data into fluid simulation pipelines. The specific contributions of this paper are

- establishing the relationship between fluid pressure solves and proximal operators,
- a physically constrained, multi-scale tracking method for fluid densities to reconstruct temporally coherent velocity fields,

- re-simulation of the captured fluids for adding detail or changing the domain, and
- initial results for solving other mixed forward and inverse problems using the same proximal operator framework.

We believe many more such applications can be developed based on the same methodology. As such, we are confident that recent convex optimization methods will prove to be a very powerful tool in the field of computational fluid dynamics.

2 Related Work

Fluid Capture aims at the measurement of the full fluid properties of a flow existing in the real world. The flow is typically characterized by its physical properties such as advected densities and fluid velocities. Capture work is made difficult by the fact that optically observable properties are further influenced by other factors than those governing the flow. These phenomena include e.g. emission, scattering, absorption, fluorescence, and refraction.

For this reason, capture technologies in computer graphics and computer vision have first concentrated on phenomenological descriptions. Hasinoff and Kutulakos [2007], as well as Ihrke and Magnor [2004] reconstruct emission volumes of flames. The captured information relates to the chemical composition, density, and burn rate of the combustible but not to physical flow properties. Smoke scanning [Hawkins et al. 2005; Gu et al. 2013] recovers scattering properties or smoke densities, but again, no information relevant to flow. Atcheson et al. [2008] recover dynamic refractive index volumes, a quantity related to fluid density, while Gregson et al. [2012] investigate fluid mixing by observing dye concentrations. The transport of both observed quantities is directly governed by the underlying fluid velocities, which are therefore suitable data for velocity tracking. However, all references above only recover per-frame information without explicit temporal correlation. Fluid velocities are not computed.

Methods described in the physics literature often involve sophisticated hardware setups such as scanning with the help of laser and electron beams, or complex optical arrangements. Examples include Particle Image Velocimetry (PIV), (Planar) Laser Induced Fluorescent Imaging (PLIF), Electron Beam Fluorescence (EBF), or Schlieren-based systems. In some cases, such as PIV, the flow has to be physically seeded with particles to enable its visualization. An extensive overview is given by Tropea et al. [2006]. For Schlieren systems, both classical and computational, the reader is referred to [Settles 2001; Panigrahi and Muralidhar 2012]. A relatively recent addition to the fluid capture toolbox is TomoPIV [Elsinga et al. 2006], a method to reconstruct instantaneous 3D PIV data. None of the capture methods mentioned above, except for EBF, is able to capture fluid velocity directly.

Velocity Estimation for 3D flows has only received sparse attention due to its inherent difficulties. In graphics, 3D velocity reconstruction have been used to guide fluid simulations of water surfaces. Wang et al. [2009] capture free-flowing water columns, whereas Li et al. [2013] aim at generating dynamic height field water surfaces from video.

The fluid mechanics literature has more extensive coverage, even though full 3D tracking is rare and temporal coherence has not been addressed. Early work aimed at compensating PIV plane scanning artifacts by performing an operation similar to rolling shutter compensation on acquired volume data sets [Van Vliet et al. 2004]. However, only a coarse mean flow was computed by cross-correlation.

The equivalence between *optical flow* and fluid flow was realized early, but its formal conditions have not been established until recently [Liu and Shen 2008]. The authors put particular emphasis on the image space projection of 3D transport mechanisms and associated boundary effects. The full 3D transport equation was inverted by Su and Dahm [1996] in the context of scanned PLIF data.

The acquisition of fluid velocities using optical flow techniques rests on physical priors, such as divergence-free constraints for incompressible flows [Yuan et al. 2007; Vlasenko and Schnörr 2009]. These methods typically require complex discretization steps due to their higher order regularizers. Moreover, the discretized operators must be compatible, i.e. they must preserve the continuous case vector identities in the discrete setting. As an alternative to these complex numerical techniques, basis function expansions that only cover the appropriate sub-spaces have been used, e.g. divergence-free wavelets [Kadri Harouna et al. 2013]. This method allows for an elegant incorporation of the constraint and a natural multi-scale representation, but inter-connecting with other parts of a fluid simulation pipeline requires conversion. All examples discussed in this paragraph have been developed for (quasi-)2D flows.

In contrast to the fluid mechanics literature, we show that pressure projection can be naturally incorporated into a velocity estimation scheme as a convex regularizer. This allows for a simpler handling of physical constraints in the estimation, while maintaining compatibility and modularity of the implementation. We tightly couple the velocity estimation and fluid simulation steps to predict good initial guesses for the flow velocities in subsequent time steps, effectively coupling a fluid simulation with global history into the estimation procedure. Most similar to this approach, Barbu et al. [2013] have recently coupled two frame estimates in a discrete TomoPIV setting, however, without incorporating physical constraints as we do.

Fluid simulation has the goal of calculating the motion of a flow based on a set of boundary conditions and a physical model. Such simulations have a long history in research. One of the influential papers in the area, e.g., introduced the *staggered grid* discretization [Harlow and Welch 1965], which is the basis for many works in computer graphics. While Foster and Metaxas [1996] were the first to use fluid simulations in graphics, the introduction of the stable fluids approach [Stam 1999] has lead to the wide-spread use of fluid simulation for visual effects. These simulations have since become a crucial tool in digital movie productions, especially the FLIP method [Zhu and Bridson 2005] and the purely particle based SPH approaches [Müller et al. 2003] are widely used. Recent works make it possible to simulate advanced rigid-body interactions with particle-based flows [Akinci et al. 2012], propose adaptive discretizations for gains in efficiency [Ando et al. 2013], and show how to handle complex incompressible materials [Stomakhin et al. 2013]. The single phase fluid solver we use is in line with those commonly used in graphics. A good overview is, e.g., given in the book by R. Bridson [2008]. Our particular choice of algorithms, such as the use of BFECC [Kim et al. 2005] is described in more detail below.

In a flow modification context, previous work has demonstrated the generation of flow detail based on physical models [Kim et al. 2008; Narain et al. 2008], which we show to be applicable to captured data using our estimation framework.

Our method can be considered as a simulation that is guided by observed data. Previous work has employed non-linear optimizations with the adjoint method to guide fluids [McNamara et al. 2004], while other works have used forces-based approaches targeting liquid surfaces [Shi and Yu 2005; Thuerey et al. 2006]. The guiding of high resolution simulations based on a coarse input flow has, e.g., been investigated by Nielsen et al. [2009], and we will explore the connection between guiding and our ADMM framework in Sec-

tion 6.

3 Background and Notation

Incompressible Fluid Flow can be modeled by the incompressible Navier-Stokes equations. These equations describe the time-evolution of the fluid velocity $\vec{u} = (u, v, w)$ as a function of itself, pressure P, density ρ, viscous forces induced by the viscous stress tensor τ as well as external body forces $\vec{f}$, subject to the condition that the flow be mass-conserving (divergence-free):

$$\nabla \cdot \vec{u} = 0 \quad (1)$$

$$\vec{u}_t + \vec{u} \cdot \nabla \vec{u} + \nabla P/\rho = (\nabla \cdot \tau)/\rho + \vec{f} \quad (2)$$

Throughout this paper we will use symbols with arrows (e.g. $\vec{u}$) to denote continuous vector fields, and bold symbols (e.g. $\mathbf{u}$) to denote the corresponding discretized field, with all components rearranged into a single column vector.

In a general flow, the advection of a scalar field ϕ based on a velocity field $\vec{u}$ is given by

$$\phi_t + \nabla \cdot (\vec{u}\phi) - \overline{\alpha}\nabla^2\phi = s,$$

where ϕ refers to an observable concentration of a marker fluid such as a dye or smoke, $\overline{\alpha}$ is the diffusivity and s is a source term. The marker fluid and the ambient fluid are assumed to have approximately the same mass density. In the case of $s = 0$, negligible diffusivity ($\overline{\alpha} = 0$), and incompressibility ($\nabla \cdot \vec{u} = 0$), the equation reduces to

$$\phi_t + \vec{u} \cdot \nabla \phi = 0, \quad (3)$$

which corresponds to a pure advection. We note that Eq. 3 is equivalent to the *brightness constancy constraint* of optical flow as introduced by Horn and Schunck [1981].

Eqs. 2 and 3, in conjunction with the constraint in Eq. 1, describe the time-evolution of a non-diffusive incompressible fluid flow carrying a marker density. A difficulty in fluid simulation is that there is no time-evolution equation for fluid pressure P. When evolving the fluid velocities, the gradient of the scalar pressure field can *only* contribute to divergent portions of the updated velocity field. Modern incompressible fluid simulations advance the velocity forward in time, omitting the ∇P term. This is followed by a *pressure-projection* step in which a pressure field is computed that exactly counteracts divergent components of velocity introduced during time stepping.

This pressure projection step is the central component for the new proximal operator that we introduce below.

Convex Optimization is an area that has seen rapid progress in recent years, and has found many applications in areas such as imaging and inverse methods. A common form of convex optimization problem is given as

$$\min_{\mathbf{u}} F(\mathbf{u}) + G(\mathbf{u}), \quad (4)$$

where F is often an ill-posed linear least squares data term $F(\mathbf{u}) = \|\mathbf{b} - \mathbf{A}\mathbf{u}\|_2^2$, while G is a convex regularizer or prior.

To solve this kind of problem, we can turn the unconstrained optimization problem of Equation 4 into a constrained one by introducing a *slack variable* $\mathbf{v}$:

$$\min_{\mathbf{u},\mathbf{v}} F(\mathbf{u}) + G(\mathbf{v}) \quad \text{subject to } \mathbf{u} = \mathbf{v}. \quad (5)$$

This *splitting* approach is advantageous because it separates the two component terms F and G in Eq. 4, and allows us to use an array of recent optimization methods to solve the problem. Many recent non-linear solvers are *proximal methods* [Parikh and Boyd 2013], which are based on the so-called *proximal operators* for F and G:

$$\mathbf{prox}_{\lambda F}(\mathbf{z}) := \quad \arg\min_{\mathbf{u}} F(\mathbf{u}) + \tfrac{1}{2\lambda}\|\mathbf{z} - \mathbf{u}\|_2^2 \quad (6)$$

$$\text{and} \quad \mathbf{prox}_{\lambda G}(\mathbf{z}) := \quad \arg\min_{\mathbf{v}} G(\mathbf{v}) + \tfrac{1}{2\lambda}\|\mathbf{z} - \mathbf{v}\|_2^2. \quad (7)$$

These two operators are individual minimizers for F and G respectively, which remain close to some reference point $\mathbf{z}$. In this way the proximal operator can be interpreted as a trust-region method with weight $\frac{1}{\lambda}$ and has several benefits, among them

- the proximal operators for F and G can be independently derived and implemented, making it possible to easily combine high performance implementations of different F and G.
- $\mathbf{prox}_{\lambda F}$ is strongly convex even when F is weakly convex (e.g. under-determined for least-squares problems)
- generally improves conditioning of quadratic F by adding a small multiple ($\frac{1}{2\lambda}$) of identity.

Given the proximal operators for a given problem, we can choose between many different algorithms to solve Equation 5. One of the most popular and easy to implement choices is the Alternating Direction Method of Multipliers (ADMM, [Parikh and Boyd 2013]), which can be expressed with proximal operators as follows:

Algorithm 1 Alternating Direction Method of Multipliers

procedure ADMM($\mathbf{prox}_{\lambda F}, \mathbf{prox}_{\lambda G}$)
  while $k < maxIters$ **do**
    // Update primary variable $\mathbf{u}$
    $\mathbf{u}^{k+1} \leftarrow \mathbf{prox}_{\lambda F}(\mathbf{v}^k - \mathbf{q}^k)$

    // Update slack variable $\mathbf{v}$
    $\mathbf{v}^{k+1} \leftarrow \mathbf{prox}_{\lambda G}(\mathbf{u}^{k+1} + \mathbf{q}^k)$

    // Update Lagrange multipliers
    $\mathbf{q}^{k+1} \leftarrow \mathbf{q}^k + \mathbf{u}^{k+1} - \mathbf{v}^{k+1}$
  end while
  return $\mathbf{u}^k$
end procedure

Here k is an iteration index, $\mathbf{v}$ is a vector of dual (slack) variables, and $\mathbf{q}$ is a vector of Lagrange multipliers that is initialized to zero. For convex F and G, and reasonable choices of λ, ADMM causes the value of $\mathbf{u}^k$ to converge to a minimizer of the original optimization problem of Equation 4 [Parikh and Boyd 2013].

4 Pressure Projection as Proximal Operator

We now show that pressure projection as used in many incompressible flow solvers is equivalent to a proximal operator for the indicator function of divergence-free velocity fields $\mathcal{C}_{DIV}$. Intuitively, this operator corresponds to a projection $\Pi_{\mathcal{C}_{DIV}}(\cdot)$ onto the set of divergence-free vector fields. This follows from the Helmholtz decomposition, which states that every vector field $\vec{u}^*$, subject to appropriate boundary conditions, can be uniquely decomposed into a curl-free component, ∇P, defined by the gradient of a scalar function (pressure in our case) and a divergence-free component, $\nabla \times \vec{a}$, defined by the curl of a vector potential $\vec{a}$.

The vector identities $\nabla \times (\nabla P) = 0$ and $\nabla \cdot (\nabla \times \vec{a}) = 0$ further show that these fields are orthogonal. Incompressible flow solvers commonly exploit this fact by advancing the incompressible Navier-Stokes equations forward in time using all terms from Equation 2 except for the pressure gradient $\nabla P/\rho$ to obtain an intermediate velocity field $\vec{u}^*$. Since the pressure gradient term *can only* contribute to the divergent components of the velocity field, $\vec{u}^*$ can be made divergence-free by computing a pressure field that exactly counteracts the divergent component introduced by time integration, i.e.:

$$P = \left(\nabla^2\right)^{-1}(-\nabla \cdot \vec{u}^*) \quad (8)$$

$$\vec{u} = \vec{u}^* - \nabla P/\rho, \quad (9)$$

where $\left(\nabla^2\right)^{-1}$ indicates Poisson integration for the pressure. Solving Equations 8-9 in sequence results in a velocity field $\vec{u}$ that is divergence-free. From the orthogonality of the divergence-free and curl-free component and from the uniqueness of the decomposition it is apparent that the update is the minimum norm change to the velocity field required to make it divergence-free, and therefore, a projection onto the subspace of divergence-free vector fields $\mathcal{C}_{DIV}$.

This implies that the pressure projection step commonly employed in fluid simulators can alternatively be stated as a simple minimization problem. In terms of the proximal operator framework, we may write:

$$\mathbf{prox}_{\lambda DIV}(\mathbf{u}^*) = \underset{\mathbf{u}\in\mathcal{C}_{DIV}}{\mathbf{arg\,min}} \|\mathbf{u} - \mathbf{u}^*\|_2^2 = \Pi_{\mathcal{C}_{DIV}}(\mathbf{u}^*). \quad (10)$$

Equation 10 shows that the pressure projection step is a convex operation and thus may be used in convex optimization algorithms. Moreover, the projection onto the sub-space of divergence-free vector fields is a proximal operator, enabling its application as a regularizer in proximal optimization algorithms. We note that Equation 10 may practically be implemented either through a pressure solve as in Equations 8-9 or as a projection operation as in Eq. 10. The implementation using the pressure solve step benefits from the availability of optimized solvers in the fluid simulation community. Moreover, since a scalar update function is computed, only one third of the variables are needed in the solve as compared to implementing Eq. 10 directly.

5 Fluid Tracking

As mentioned in Sect. 3, the optical flow brightness constancy formulation and the transport equation for inviscid, incompressible flow are equivalent. In order to perform fluid tracking, we can therefore use the proximal operator derived in the previous section with the 3D equivalent of an optical flow data term (Eq. 3) to formulate incompressible fluid tracking as a minimization problem

$$\min_{\mathbf{u}} \underbrace{E_{PC}(\mathbf{u}) + \alpha E_{SM}(\mathbf{u}) + \beta E_{KE}(\mathbf{u})}_{E_{FT}(\mathbf{u})=F(\mathbf{u})} + \underbrace{\gamma E_{DIV}(\mathbf{u})}_{G(\mathbf{u})}. \quad (11)$$

Here, E_{PC} is the optical-flow photoconsistency (or fluid transport), term, E_{SM} is a smoothness term, E_{KE} is a kinetic energy penalty, and E_{DIV} is a divergence-free term as in the previous section. We have already indicated how this minimization problem maps to the one from Equation 4. In order to apply ADMM to this problem, we have to define the individual terms, and derive the proximal operators for F and G. As we show in the following, these are a velocity estimation problem $E_{FT}(\mathbf{u}) = F(\mathbf{u})$ and a pressure projection $E_{DIV}(\mathbf{u}) = G(\mathbf{u})$, respectively.

Technically, we propose several changes to established optical flow algorithms. These modifications significantly enhance the velocity reconstructions compared to previous work, and we evaluate the resulting gain in quality in Section 6.1. The main differences between fluid tracking and standard optical flow can be summarized as:

- the suppression of spurious velocities, Sect. 5.1,
- a divergence-free multi-scale scheme incorporating advection of fluid velocities, Sect. 5.1.1 and 5.2, and
- a sequential reconstruction pipeline that ensures temporal coherence of the flow by coupling the Navier-Stokes equations into the estimation procedure, Sect. 5.3.

5.1 Velocity Estimation

First, we derive the velocity estimation subproblem. We use a standard ℓ_2 objective function with a second-order smoothness term and a kinetic energy constraint. The data term is chosen according to Eq. 3:

$$E_{PC}(\vec{u}) = \frac{1}{2}\int_\Omega (\phi_t - \vec{u}\cdot\nabla\phi)^2\, d\Omega, \quad (12)$$

where ϕ_t is the time-derivative of the volume density and $\nabla\phi$ is its spatial gradient.

We use a voxel basis to represent the continuous space fields, and discretize the image gradient $\nabla\phi$ with the first-order finite difference matrix $[\boldsymbol{\nabla\Phi}]$ to obtain

$$E_{PC}(\mathbf{u}) = \frac{1}{2}\left(\Phi_t - [\boldsymbol{\nabla\Phi}]\mathbf{u}\right)^2 \quad (13)$$

$$\triangleq \frac{1}{2}\mathbf{u}^T[\boldsymbol{\nabla\Phi}]^T[\boldsymbol{\nabla\Phi}]\mathbf{u} - \Phi_t^T[\boldsymbol{\nabla\Phi}]\mathbf{u}, \quad (14)$$

where the second line omits terms that are constant with respect to $\mathbf{u}$, and therefore not relevant to the minimization problem.

In flow tracking applications we typically encounter regions with very little variation in density (analogous to untextured image regions in optical flow applications). In order to extrapolate velocities to these regions we introduce a smoothness prior as a penalty. This can be expressed in continuous space as a minimization over the spatial gradients of the components of the flow field, i.e.

$$E_{SM}(\vec{u}) = \sum_j \int_\Omega \|\nabla u_j\|_2^2 d\Omega, \quad (15)$$

which after discretization becomes

$$E_{SM}(\mathbf{u}) = \frac{1}{2}\sum_j (\mathbf{D}_j\mathbf{u})^2 = \frac{1}{2}\sum_j \mathbf{u}^T\mathbf{L}_j\mathbf{u}, \quad (16)$$

where $\mathbf{D_j}$ is a discrete derivative matrix for the j^{th} velocity component, and $\mathbf{L}_j = \mathbf{D}_j^T\mathbf{D}_j$ is the corresponding discrete Laplacian.

Finally we introduce a Tikhonov regularization term that is physically motivated as a kinetic energy penalty to favor minimally energetic flows:

$$E_{KE}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{u} \quad (17)$$

The overall objective function $F(\mathbf{u})$ for the velocity estimation subproblem is then the weighted sum of Equations 13, 16 and 17 as indicated in Equation 11. Since all terms of the fluid tracking problem

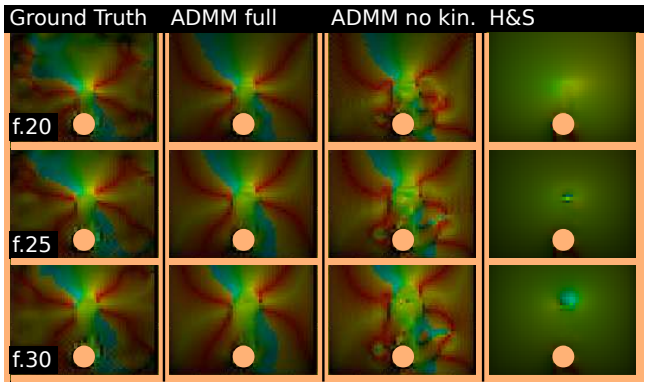


Figure 2: *Comparison of 3D reconstructed velocity fields to ground truth data (left) for the full ADMM method that includes kinetic and divergence-free penalties (middle left), the same method without kinetic energy penalty ($\beta = 0$, middle right), and multi-scale Horn-Schunck flow ($\beta = 0, \gamma = 0$, no ADMM, far right). The figure shows three different time steps (20, 25 and 30). The full method clearly reproduces the features of the ground truth velocity pattern, whereas the method without kinetic energy penalty invents new structures (lower right corner). Standard Horn-Schunck flow produces smooth flow fields that only on average correspond to the ground truth. For visualization, the 3D velocity vectors are converted to spherical coordinates which are then mapped to HSV color space. The figure shows a central slice of the 3D domain.*

$E_{FT}(\mathbf{u}) = F(\mathbf{u})$ are quadratic functions, the corresponding proximal operator turns out to be a simple linear least-squares problem:

$$\begin{aligned} \mathbf{prox}_{\lambda F}(\mathbf{v}) &= (\mathbf{I} + \lambda \mathbf{A}_{FT})^{-1} (\mathbf{v} - \lambda \mathbf{b}_{FT}) \qquad (18) \\ \mathbf{A}_{FT} &= [\boldsymbol{\nabla}\boldsymbol{\Phi}]^T [\boldsymbol{\nabla}\boldsymbol{\Phi}] + \alpha \sum\nolimits_j \mathbf{L}_j + \beta \mathbf{I} \\ \mathbf{b}_{FT} &= [\boldsymbol{\nabla}\boldsymbol{\Phi}]^T \Phi_t \end{aligned}$$

Note that $(\mathbf{I} + \lambda \mathbf{A}_{FT})$ is a positive definite matrix since it is encoding a proximal operator. Thus, Conjugate Gradient is a feasible solver for this linear system. We use it together with the ILDLTT preconditioner (incomplete Cholesky factorization), caching the preconditioner and warm-starting the solver every time the proximal operator is called by ADMM.

The velocity estimation method derived in this section can be interpreted as a variant on the well known Horn-Schunck [1981] optical flow algorithm. Instead of using a photo-consistency argument, we have exploited the link to the transport equation (Eq. 3) and incorporated physical constraints on the divergence of the velocity field and the kinetic energy. This last prior can be very important for fluids: without it, Horn-Schunck optical flow will find the smoothest possible solution. Although this is desirable for optical flow which generally has large simple motions, for fluids it is preferable to look for low-energy solutions which helps to reduce spurious velocities and allows lower smoothness penalties to be used. For a visual comparison of the effect, see Fig. 2.

Scale Splitting of Velocity In preparation for introducing a multi-scale algorithm for the fluid tracking problem, we revisit the derivation of the velocity estimation step, and rewrite it in terms of a low frequency estimate $\vec{u}_{in}$ and a high frequency perturbation $\Delta\vec{u}$: $\vec{u} = \vec{u}_{in} + \Delta\vec{u}$. In the multi-scale algorithm below, we obtain $\vec{u}_{in}$ by upsampling a low-resolution solution, and optimize for $E_{FT}(\Delta\mathbf{u})$ at each scale. Following the same derivation as above, we can derive a new proximal operator, which is identical to the one from Equation 18, except for the right hand side vector $\mathbf{b}_{FT}$, which is now:

$$\mathbf{b}_{FT} = [\boldsymbol{\nabla}\boldsymbol{\Phi}]^T \Phi_t + \alpha \sum_j \mathbf{L}_j^T \mathbf{u_{in}} + \beta \mathbf{u_{in}}$$

5.1.1 Pressure Projection

To apply the proximal operator from Equation 10 in the context of our splitting of the fluid velocity into an initial estimate and a perturbation, $\vec{u} = \vec{u}_{in} + \Delta\vec{u}$, we modify the proximal operator slightly to account for the fact that the velocity estimate $\vec{u}_{in}$ may not be divergence-free:

Algorithm 2 Pressure projection proximal operator, $\mathbf{prox}_{\lambda G}$

procedure $\mathbf{prox}_{\lambda G}(\Delta\mathbf{u}^*)$
  // $\mathbf{u_{in}}$ is coarse scale velocity estimate
  $\mathbf{u}^* = \mathbf{u_{in}} + \Delta\mathbf{u}^*$
  $\mathbf{u} = \mathbf{prox}_{\lambda DIV}(\mathbf{u}^*)$ ▷ Project $\mathbf{u}$ via Equations 8 and 9
  return $\mathbf{u} - \mathbf{u_{in}}$
end procedure

We base our implementation on the staggered discretization with co-located velocities from [Ando et al. 2013]. We use it to discretize anisotropic, rectangular grids, but our algorithm is oblivious to the different types of pressure solvers, as long as pressure projection and optical flow energy are discretized consistently.

5.2 Extension to Multiscale

We now describe a multi-scale extension that allows the method to recover large motions. Each scale of the method computes an update $\Delta\vec{u}$ to the velocity field computed by coarser scales that improves the fit to the input data, returning the sum as output. The multi scale algorithm makes use of three auxiliary functions:

- SmoothAndCubicDownsample(ϕ, η, σ) blurs ϕ by a Gaussian of width σ and cubicly downsamples by a factor of η
- CubicUpsample(ϕ, η) cubicly upsamples by a factor of η
- Advect$(\phi, \mathbf{u}, \Delta t)$ advects ϕ through the velocity field $\mathbf{u}$ by Δt units of time, using BFECC path tracing with 3rd order spatial interpolation and min-/max-limiting

Algorithm 3 Multiscale fluid tracking, $0 < \eta < 1$ is downsampling factor, σ is blur for downsampling, $l_{\max}$ is the number of levels

1: **procedure** FLUIDTRACKINGMS$(\phi_1, \phi_2, \Delta t, \eta, \sigma, l)$
2: **if** $l < l_{\max}$ **then**
3: $\phi_1^{\mathcal{C}} \leftarrow$ SmoothAndCubicDownsample(ϕ_1, η, σ)
4: $\phi_2^{\mathcal{C}} \leftarrow$ SmoothAndCubicDownsample(ϕ_2, η, σ)
5: $\mathbf{u}^{\mathcal{C}} \leftarrow$ FluidTrackingMS$(\phi_1^{\mathcal{C}}, \phi_2^{\mathcal{C}}, \Delta t, \eta, \sigma, l+1)$
6: $\mathbf{u} \leftarrow \frac{1}{\eta} *$ CubicUpsample$(\mathbf{u}^{\mathcal{C}}, \frac{1}{\eta})$
7: **end if**
8: $\phi_2^* \leftarrow$ Advect$(\phi_2, \mathbf{u}, -\Delta t)$
9: // Solve for $\Delta\mathbf{u}$ using ADMM
10: // using $\mathbf{u_{in}} = \mathbf{u}$ in Algorithm 2.
11: $\Delta\mathbf{u} \leftarrow ADMM(\mathbf{prox}_{\lambda F}, \mathbf{prox}_{\lambda G}, \phi_1, \phi_2^*)$
12: **return** $\mathbf{u} + \Delta\mathbf{u}$
13: **end procedure**

The multi-scale algorithm conceptually works by warping the second density backwards in time along the estimated velocity field $\mathbf{u}$ computed at the next coarser scale, so that the Taylor expansion used to discretize the fluid tracking time step is valid. Note that ADMM in line 11 of Algorithm 3 only computes an update to

the velocities $\mathbf{u}$. Note that $\mathbf{u}$ is only used as a constant offset for $\mathbf{prox}_{\lambda G}$ to enforce a divergence free total velocity field in Algorithm 2 and is otherwise unused by ADMM.

Our algorithm largely follows the procedure used by Meinhardt-Llopis et al. [2013], except we use ADMM to additionally enforce the divergence-free constraint and use higher-order path tracing for the advection step to improve the results. However, a crucial topic for accurately tracking fluids is temporal coherence of the flow, which is closely related to conservation of momentum. We outline our modifications to embed multi-scale tracking into the fluid simulation context in the following section.

5.3 Temporal Coherence

In order to improve tracking quality we first compute an estimate of the next velocity, and then use the fluid tracking algorithm of the previous section to compute an update of these estimated velocities. Based on a velocity from the previous time step, we estimate the next velocity by warping it with itself, and projecting it to make it divergence free (essentially a forward fluid simulation step with operator splitting). This gives us an estimate for the current time step, for which we only have the captured densities ϕ^t. As the constraints we are solving for are convex, using the velocity estimate as initial guess has little influence on the solution. Instead, we warp ϕ^t by $-\Delta t$ with the velocity estimate to align it with ϕ^{t-1} as much as possible, and use our fluid tracker to compute the residual change of the velocities. The final velocity at time t is thus given by the tracking solution plus the velocity estimate. This algorithm is summarized in Algorithm 4.

This tight coupling of fluid simulation estimates and tracking has several implications: most importantly, it significantly simplifies the problem the fluid tracking step has to solve. Instead of relying on it to solve the complex, non-linear Navier-Stokes dynamics from Eq. 2, we give it a much simpler problem, namely to only estimate changes in initial conditions, and similar effects that are not covered by the simulation step.

This coupling also serves as an implicit temporal coherence prior since the method searches for low-energy updates to the velocities predicted by forward simulation. Even though there is no direct expression of temporal smoothness, velocity fields that are temporally smooth imply lower-energy updates and so are favored over those that 'pop'. This complements the spatial prior, which extrapolates the motions computed in high-texture regions to areas without detail.

Unfortunately the coupling of tracking and simulation also implies that tracking is now a sequential process, in contrast to algorithms that only take two density images as input. In fact, the complete simulation history is contributing to the current time estimate.

Algorithm 4 Fluid tracking, compute next velocity field given previous state ϕ^{t-1}, $\mathbf{u}^{t-1}$ and next density ϕ^t

procedure COMPUTEVELOCITY($\phi^{t-1}, \phi^t, \mathbf{u}^{t-1}, \Delta t$)
  $\tilde{\mathbf{u}}^t \leftarrow \text{Advect}(\mathbf{u}^{t-1}, \mathbf{u}^{t-1}, \Delta t)$
  // project to be div. free with $\mathbf{u_{in}} = 0$ in Alg. 2
  $\tilde{\mathbf{u}}^t \leftarrow \mathbf{prox}_{\lambda DIV}(\tilde{\mathbf{u}}^t)$
  $\phi_{tmp} \leftarrow \text{Advect}(\phi^t, \tilde{\mathbf{u}}^t, -\Delta t)$
  $\hat{\mathbf{u}} \leftarrow FluidTrackingMS(\phi^{t-1}, \phi_{tmp}, \Delta t)$
  $\mathbf{u}^t \leftarrow \tilde{\mathbf{u}}^t + \hat{\mathbf{u}}$
  return $\mathbf{u}^t$
end procedure

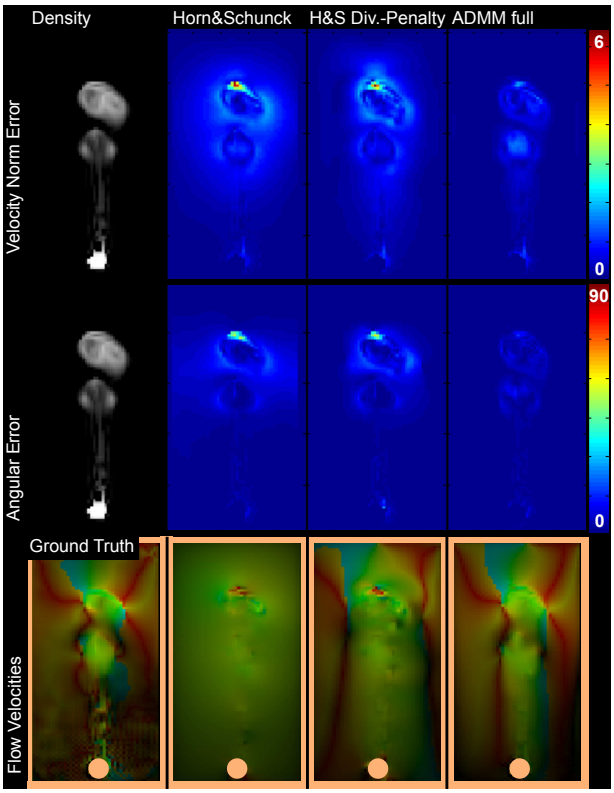


Figure 3: *A visual comparison between Horn-Schunck (H&S), divergence-penalized H&S, and our proposed method. From top to bottom: magnitude of vector difference w.r.t. ground truth, angular error, absolute HSV color-encoded direction plots. The figure shows the central slice of our simulation volume.*

6 Results and Discussion

In this section we provide quantitative evaluations of the fluid tracking performance, and show several applications of using, resimulating, and modifying the resulting flow fields. All results are based on real captured data of mixing fluids from Gregson et al. [2012]. These datasets are based on tomographic reconstructions of dye concentrations in mixing fluids. These are very challenging datasets for a number of reasons. First, they were acquired with a very small number of cameras, which limits the amount of interior texture detail that our flow tracking could rely on. Second, as outlined by Gregson et al. [2012], their method suffers from self-shadowing and other photometric calibration problems, which means that the brightness constancy assumption holds only approximately and on short time scales. Finally, each frame is reconstructed individually, so that the data suffers from small amount of temporal flicker. However, despite these challenges, the data is representative of the best captures of fluid densities described in the literature so far, which motivates our decision to use these datasets in our experiments. Note that single-phase phenomena in air or water can be regarded as equivalent if they are correctly matched in terms of their parametrization and Reynolds number. As such we will treat the flows captured in water as rising smoke in some of the examples below.

6.1 Tracking Validation

We compare our proposed method with two common approaches for estimating the motion between two still frames. The first is a

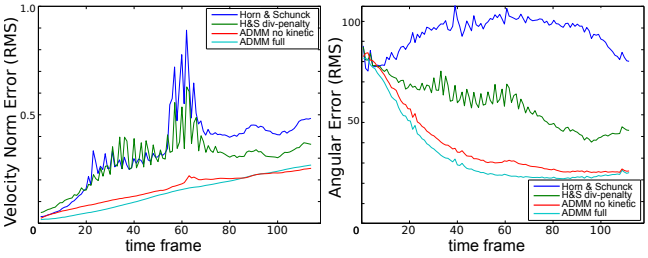


Figure 4: *Ground truth comparison for a 3D simulation for different methods. We show velocity norm error (left) and norm-weighted angular error (right) plots. The Horn-Schunck-based two-frame methods suffer from temporal artifacts. The direction estimates of the flow vectors are unsatisfactory. In contrast, our proposed method shows very good temporal coherence and significantly improved directional estimates.*

multi-scale extension of the well known Horn-Schunck ("H&S") technique that is designed to recover large motions. This choice is motivated by its continued competitiveness when implemented in a modern way [Sun et al. 2014]. The second is an extension to the multi-scale Horn-Schunck method "H&S div. penalty" that incorporates a divergence penalty. This is to mimic the performance of divergence-free methods such as [Kadri Harouna et al. 2013]. Our proposed method is referred to as "ADMM full" and we include a version without kinetic energy constraint "ADMM no kin.". The reconstruction parameters are provided in Table 1.

Our validation experiment consists in generating a simulated ground truth sequence of a 3D incompressible flow (resolution $50 \times 100 \times 50$) transporting a density through a simple domain. The sequence consists of 118 frames. The central slice 25 through the simulated density volume is shown in Fig. 3 (upper left).

To evaluate the different methods, we investigate two error measures that are adapted from the optical flow community. The *velocity norm error* measures the norm of the vector difference between the estimated and the ground truth vectors. The *angular error* measures the directional accuracy of the motion. We weight the angular error with the norm of the velocities to help distinguish large-scale trends from finer features such as a slight shifting of eddies that produce high angular errors even though the overall flow behavior is consistent.

Figure 3 shows a visual comparison of the angular error, velocity norm error, and a color-coded visualization of the velocity vectors. All plots show the same volume slice 25 and have a common scale. We see that our proposed method using tightly coupled simulation and velocity estimation significantly outperforms the other two methods. The error is significantly lower, and our method is the only one that successfully captures the overall structure of the velocity field. This is even more apparent in the supplemental video footage, where the Horn-Schunck and divergence-penalized methods show severe temporal artifacts.

The temporal behavior of the different algorithms is also shown in Fig. 4. The Horn-Schunck derived techniques are susceptible to incoherent estimates whereas our proposed method shows a smooth error evolution that is due to cumulative tracking error (left). The large angular errors at the beginning of the sequence stem from a random velocity initialization of the ground truth simulation (right): Initially, there is no density in the simulation domain which could be used by the algorithm do identify these "hidden" velocities. Around frame 60, the simulation has reached the full domain and the optical flow obtains its highest accuracy. Whereas the directional data recovered by the Horn-Schunck derived methods are

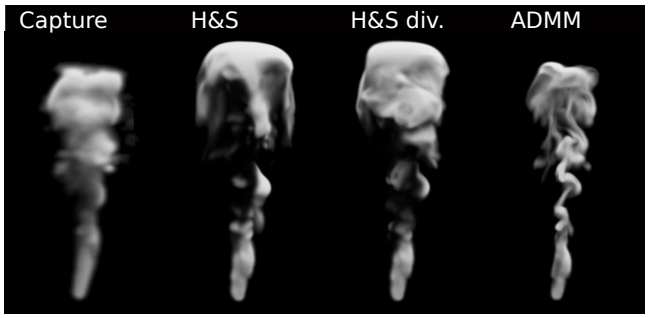


Figure 5: *Passive advection of densities in a reconstructed time-varying velocity field. From left to right: captured densities, result based on Horn-Schunck reconstruction, H&S with divergence penalty, and our proposed algorithm.*

	H&S	H&S div.	ADMM full	ADMM no kin.
α	0.2	0.2	1e-4	1e-4
β	0.0	0.0	1.0	0.0
γ	0.0	100.0	h.c.	h.c.
ADMM iter.	n.a.	n.a.	5	5
ADMM λ	n.a.	n.a.	1.0	1.0

Table 1: *Optimization parameters for ground truth comparisons. 'n.a.' stands for 'not applicable', 'h.c.' for 'hard constraint'. The H&S solvers compute a solution to the linear system, corresponding to Eq. 11 using CG, whereas ADMM uses the algorithms described in the paper. The multi-scale parameters are common for all methods: 3 levels, σ =0.5, η =0.65.*

almost random, our proposed algorithm computes reasonable estimates.

To test the effect of the kinetic energy regularizer, Fig. 4 also includes a version "ADMM no kinetic" of our proposed algorithm with deactivated kinetic energy term. The numerical performance is worse than the full method for most of the sequence. It is interesting to note that once the data has reached the full domain, the kinetic energy term has a slightly negative effect: when data is available everywhere over-regularization occurs. An adaptive determination of the corresponding parameter may increase the accuracy of our algorithm in future implementations.

We also tested the performance of the method on flows with solid obstacles and with varying degrees of data corruption. Input data was generated for a domain with 100x200x100 cells, consisting of a spherical dye inflow with randomized initial velocities and a spherical obstacle that triggers turbulent mixing (shown in Figure 6). We enforce the solid boundary conditions via the variational approach of Batty et al. [2007] in the pressure solve and as a penalty in the velocity estimation subproblem.

We evaluated the fluid tracking performance at the full resolution and on a half-resolution grid with varying levels of blur added to the input. This was done to approximate the quality of data obtained by optical tomography capture setups (e.g. as used by [Gregson et al. 2012]) which typically yield fairly low spatial resolution with substantial filtering. The velocity norm errors and angular errors are plotted in Figure 6.

The results in Figure 6 show that reasonable velocity predictions may be obtained even from coarse input volumes with moderate levels of blur. High levels of blur cause the reconstructions to diverge more quickly over time in both metrics.

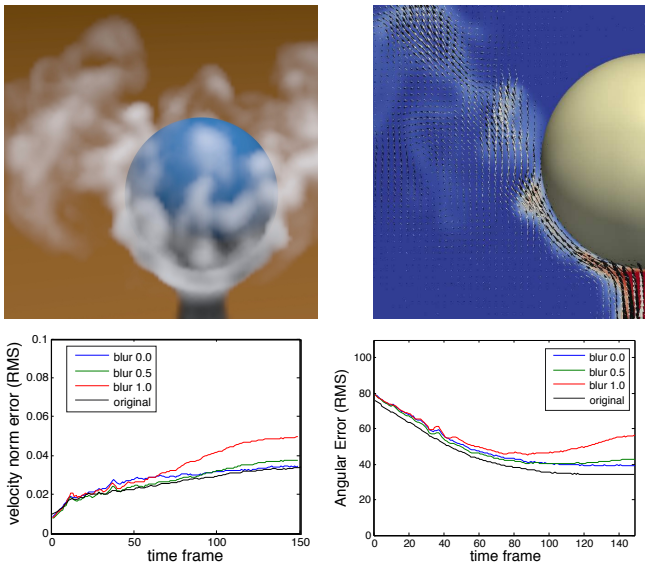
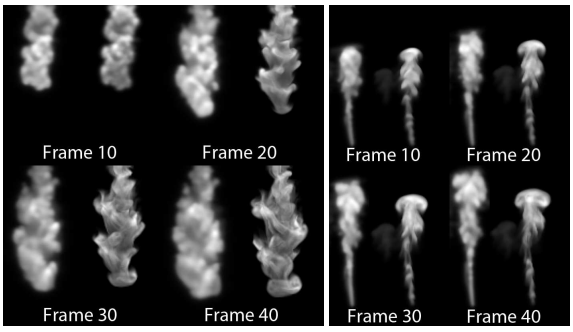


Figure 6: *Fluid tracking for a flow interacting with a solid obstacle. Top row: dye enters from a spherical inflow and impinges upon a spherical obstacle. A slice through the domain shows dye concentration (fringe color), ground-truth velocity vectors (black) and reconstructed velocity vectors (white). Bottom row: velocity norm and angular errors for native resolution ("original") and half-resolution reconstructions with varying levels of blur ($\sigma_{blur} = [0.0, 0.5, 1.0]$ voxels) applied to the input density fields.*



(a) bloom (b) smoke

Figure 7: *Resimulation of the bloom and smoke datasets at 10 frame intervals. Despite fine-scaled turbulent mixing, the simulator preserves many features of the flow for over a second. In contrast, the smoke capture with its buoyancy induced instabilities causes more rapid divergence of the simulated flow.*

6.2 Validation with Resimulation

Having developed a method to estimate the velocity fields of incompressible flows and assessed its accuracy on simulated data, we are now able to restart fluid simulations using initial conditions computed from captured data. In this way we are trying to answer a question that is interesting for scientific applications of our method: based on our velocity reconstructions, how close can we get to the real phenomenon with a simulation? We begin with some discussion on what can be expected in such a situation.

The results in Section 6.1 demonstrate that tightly coupling simulation and fluid tracking significantly improves the accuracy of the computed velocity fields. Passive advection of marker fields through the velocity fields computed by our method confirms that results reproduce the medium and coarse scales of the velocity fields well, Fig. 5. We see that the majority of significant flow features are reproduced in the advected marker fields, despite being transported over several seconds through turbulent mixing. In comparison, an advection with the velocities computed by the Horn-Schunck derived techniques are not able to reproduce the behavior of the capture. Temporal tracking artifacts lead to flicker in the resimulations, which is most easily seen in the supplemental video.

It is well known that turbulent flows exhibit considerable energy transfer between spatial scales in the velocity field, including scales that are significantly finer than those that can be resolved by flow tracking. We consequently expect that the resimulated behavior matches better for flows without effects such as buoyancy that cause high frequency behavior. In the former case, numerical and physical viscosity rapidly dampens out high frequencies that cannot be estimated accurately, an effect that is advantageous for longer resimulation similarity.

This expectation is confirmed in the resimulated "bloom" capture, Figure 7 (a), which involves dye being poured into still water. Although high-frequencies are quickly lost, the medium and large scale behavior persists for over a second of turbulent mixing since there are no density gradients driving the flow. The energy transfer in this case is primarily coarse-to-fine, and our resimulation gives a motion close to the real flow.

The opposite effect can be observed for the "smoke" capture, shown in Figure 7 (b). Here the buoyancy term results in kinetic energy being introduced in high spatial frequencies of the flow, resulting in energy being transferred from fine-to-coarse scales. The captured data is under-resolved for this type of flow and diverges more quickly. As such it highlights a limitation that our method shares with many other methods for simulating fluids: when our solver and the chosen discretization are unable to represent the physics correctly, we obtain results that deviate from the desired behavior, although they can sometimes be improved by optimizing simulation parameters.

6.3 Stylistic Capture Modifications

A benefit of computing velocity fields for capture data is that now the full flow parameters are known and can be used for artistic modifications. This measure makes captured flows amenable for use in fluid simulations. In the following, we demonstrate exemplary changes that can be applied.

Resolution Enhancement can be achieved by two means: The first is a resimulation of the captured velocity fields in conjunction with passive advection that carries initial frame densities forward in time. The availability of full flow information enables a resampling in both space and time. For an illustration of this approach, see Figs. 8 and 9. The apparent resolution increase can be appreciated by comparing the columns 'Passive Advection' and 'Tomography'. The advected densities are not guaranteed to match the capture partially because we model the density inflow conditions from a single frame of the sequence, which can lead to changes of the overall shape (e.g. on the left side of Figure 9).

The second approach for apparent resolution increase is the application of a synthetic turbulence model. As an example, we apply wavelet turbulence [Kim et al. 2008] in the column labeled 'Super-resolved' (calculated with *mantaflow* [Pfaff and Thuerey 2013]). Since the original characteristics of the flow are not changed, the

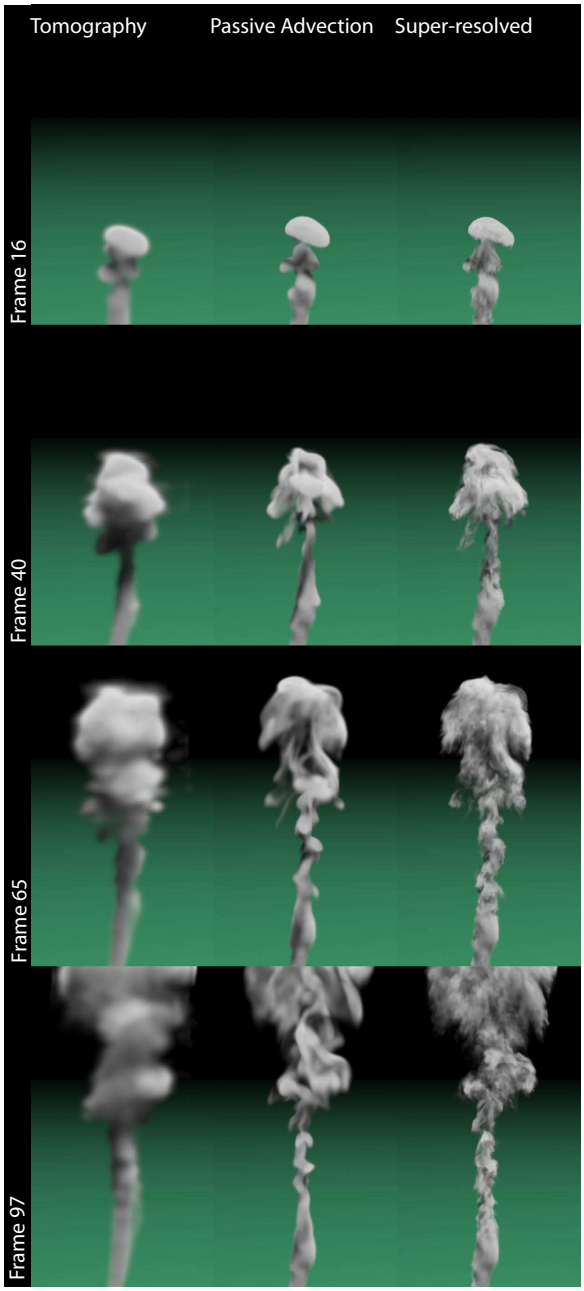


Figure 8: *Proposed fluid-tracking applied to the 'smoke' capture consisting of dye and alcohol rising under buoyancy in still water. Left column: tomographically reconstructed concentration fields, center column: passive advection using fluid tracking velocities, right column: additional synthetic turbulence. The reconstruction was performed on a* $50 \times 150 \times 50$ *grid, and took 2.35 minutes per frame on average.*

global behavior of the capture remains intact.

Domain Change can be affected by running the resimulation with advection (discussed above) up to a desired time-frame, after which the simulation is switched to a dynamic setting that takes into account the new boundary conditions. This way, the computed velocity fields can be used to transfer captured flows into modified domains. Figure 10 shows the 'smoke' dataset resimulated after adding a large sphere to the domain. The resimulation is robust despite a significant modification of the domain.

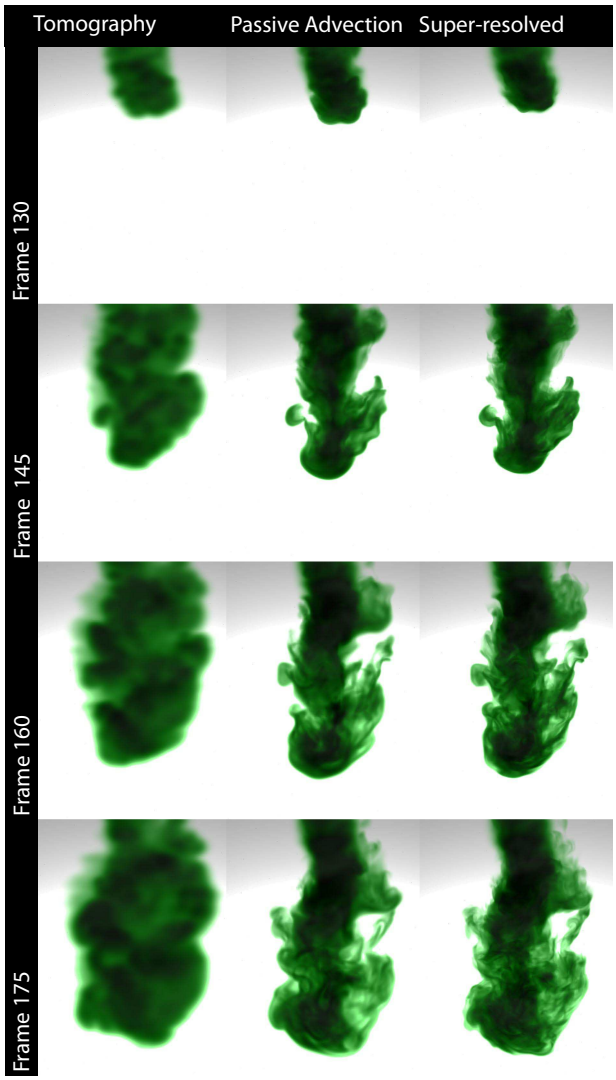


Figure 9: *Proposed fluid-tracking applied to the 'bloom' capture consisting of dye poured into still water. Left column: tomographically reconstructed concentration fields, center column: passive advection using fluid tracking velocities, right column: additional synthetic turbulence. The flow velocities were reconstructed on a* $100 \times 150 \times 100$*, taking 12.5 minutes on average per frame.*

Guided Simulation can be achieved by exploiting a second connection to optimization techniques from image processing: Given an initial simulation with velocity fields $\mathbf{u}$, and a target state $\mathbf{u}_{ref}$ that may e.g. result from flow tracking, we seek an updated state $\mathbf{u} + \mathbf{\Delta u}$ of the original simulation that reproduces the coarse scale behavior of the target flow while maintaining some characteristics of the original simulation.

We may formulate the coarse scale requirement as a low frequency blur affected by a matrix $\mathbf{B}$ on the simulation data. With these prerequisites, we can express a new optimization that seeks the necessary update $\mathbf{\Delta u}$ as:

$$\begin{aligned} \underset{\mathbf{\Delta u}}{\arg\min}\ & \frac{1}{2} \| \left(\mathbf{B}\mathbf{\Delta u} - \mathbf{B}(\mathbf{u}_{ref} - \mathbf{u}) \right) \|_2^2 \\ & + \alpha E_{SM}(\mathbf{\Delta u}) + \beta E_{KE}(\mathbf{\Delta u}) + \gamma E_{DIV}(\mathbf{u} + \mathbf{\Delta u}), \end{aligned} \tag{19}$$

i.e. the update $\mathbf{\Delta u}$, when blurred, should mimic the differences in the original and the target flows, as before subject to physical constraints. The data term now resembles a deblurring problem, but, as

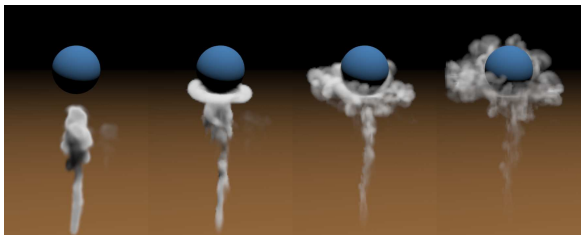


Figure 10: *The ability to estimate fluid velocities allows us to restart simulations in modified domains as compared to the capture setting. The example uses the 'smoke' data set.*

(a) regular simulation (b) with guiding

Figure 11: *As an application of the guided simulation approach, we add high-frequency curl noise to a resimulation of the smoke dataset in order to affect its visual appeal. Since no measures are taken to avoid affecting the global flow behavior, the overall low-frequency characteristics of the captured flow is lost (left). Incorporating the guided simulation approach helps to recover the low-frequency motions (right).*

outlined above, has a different interpretation in fluid problems. The regularizers, again, ensure physicality of the flow. The proximal operators only change in the data term, which can be derived in a similar manner to Eq. 18.

The fluid guiding scheme is then performed by interleaving a simulation and the guiding step Eq. 19 until the full sequence is computed. An example of this application is shown in Fig. 11. Note that for this example we are effectively running the ADMM framework twice, first to reconstruct the guiding velocities, and then a second time to guide the dynamic simulation.

7 Limitations

The present work has several limitations that bear consideration. The most significant one is that the method depends on the flows having time-varying detail that can be used to infer the underlying velocities. Consequently the performance of the method suffers when such texture is unavailable. This occurs most frequently in laminar flows which develop a static or near-static distribution of the marker fluid. This limits the applicability of the approach in such areas as laminar boundary layers.

Less severe limitations include the fact that through the use of simulation as a prediction step frames must be reconstructed in sequence, preventing reconstructing multiple frames in parallel. Finally the weights of the smoothness and kinetic energy penalties must be tuned for a given flow and capture setup to achieve a good balance between reconstructed detail and noise levels.

8 Conclusions & Future Work

In this paper we have shown that the pressure solve that is integral to many fluid simulators can be cast as a proximal operator. This formulation allows us to implement highly optimized versions of this operator, and re-use it in another context using a proximal optimization framework such as ADMM.

There are many applications for this approach. In this paper we have focused primarily on tracking fluid velocities based on densities captured by prior art fluid imaging methods. We have shown that these tracked fluid velocities can serve as the basis for resimulation, for example to add details not captured in the original data (fluid super-resolution), or to change the domain of the fluid flow. We have also shown some initial results of a guided simulation application of the same framework, which is akin to a deblurring problem in image processing. These applications show how the proximal operator framework can be used to connect forward and inverse problems in fluids, and therefore provides an avenue for using captured fluid data in graphics applications.

We believe that the applications we have explored so far are only touching upon the possibilities offered by this approach. In general, casting the fluid problems into this convex optimization framework using proximal operators allows us to separately implement and optimize different components of both forward and inverse problems, and to combine them in a modular and efficient way. We believe that it is possible to implement additional local constraints on the flow field as proximal operators, for example to allow for flexible boundary conditions such as the wall separation boundaries proposed by Batty et al. [2007]. This approach could be used to implement a valve, or to allow fluid flows to peel of the surface of a solid in a fluid simulation. We believe that many more such applications are possible.

Acknowledgements

This was in part supported by NSERC and the GRAND NCE, Ivo Ihrke has been supported by the German Research Foundation (DFG) through Emmy-Noether Grant IH 114/1-1.

References

AKINCI, N., IHMSEN, M., AKINCI, G., SOLENTHALER, B., AND TESCHNER, M. 2012. Versatile Rigid-fluid Coupling for Incompressible SPH. *ACM Transactions on Graphics 31*, 4, article 62.

ANDO, R., THUEREY, N., AND WOJTAN, C. 2013. Highly Adaptive Liquid Simulations on Tetrahedral Meshes. *ACM Transactions on Graphics 32*, 4 (August), article 10.

ATCHESON, B., IHRKE, I., HEIDRICH, W., TEVS, A., BRADLEY, D., MAGNOR, M., AND SEIDEL, H.-P. 2008. Time-resolved 3D Capture of Non-stationary Gas Flows. *ACM Transactions on Graphics 27*, 5, article 132.

BARBU, I., HERZET, C., AND MÉMIN, E. 2013. Joint Estimation of Volume and Velocity in TomoPIV. In *International Symposium on Particle Image Velocimetry*.

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics 26*, 3 (July).

BRIDSON, R. 2008. *Fluid Simulation for Computer Graphics*. AK Peters/CRC Press.

ELSINGA, G., SCARANO, F., WIENEKE, B., AND VAN OUDHEUSDEN, B. 2006. Tomographic Particle Image Velocimetry. *Experiments in Fluids 41*, 6, 933–947.

FOSTER, N., AND METAXAS, D. 1996. Realistic Animation of Liquids. *Graphical Models and Image Processing 58*, 5 (Sept.), 471–483.

GREGSON, J., KRIMERMAN, M., HULLIN, M. B., AND HEIDRICH, W. 2012. Stochastic Tomography and its Applications in 3D Imaging of Mixing Fluids. *Proceedings ACM SIGGRAPH 2012 31*, 4, article 52.

GU, J., NAYAR, S., GRINSPUN, E., BELHUMEUR, P., AND RAMAMOORTHI, R. 2013. Compressive Structured Light for Recovering Inhomogeneous Participating Media. *IEEE Pattern Analysis and Machine Intelligence 35*, 3, 555–567.

HARLOW, F., AND WELCH, E. 1965. Numerical Calculation of Time-dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids 8*, 12, 2182–2189.

HASINOFF, S. W., AND KUTULAKOS, K. N. 2007. Photo-consistent Reconstruction of Semitransparent Scenes by Density-sheet Decomposition. *IEEE Trans. PAMI 29*, 5, 870–885.

HAWKINS, T., EINARSSON, P., AND DEBEVEC, P. 2005. Acquisition of Time-Varying Participating Media. *ACM Transactions on Graphics 24*, 3, 812–815.

HORN, B., AND SCHUNCK, B. 1981. Determining Optical Flow. *Artificial Intelligence 17*, 1, 185–203.

IHRKE, I., AND MAGNOR, M. 2004. Image-Based Tomographic Reconstruction of Flames. In *Proc. of ACM/Eurographics Symposium on Computer Animation (SCA)*, 367–375.

KADRI HAROUNA, S., DÉRIAN, P., HÉAS, P., AND MEMIN, E. 2013. Divergence-free Wavelets and High Order Regularization. *International Journal of Computer Vision 103*, 1 (May), 80–99.

KIM, B., LIU, Y., LLAMAS, I., AND ROSSIGNAC, J. 2005. FlowFixer: Using BFECC for Fluid Simulation. In *Proceedings of the First Eurographics conference on Natural Phenomena*, 51–56.

KIM, T., THÜREY, N., JAMES, D., AND GROSS, M. 2008. Wavelet Turbulence for Fluid Simulation. *ACM Transactions on Graphics 27*, 3, article 50.

LI, C., PICKUP, D., SAUNDERS, T., COSKER, D., MARSHALL, D., HALL, P., AND WILLIS, P. 2013. Water Surface Modeling from a Single Viewpoint Video. *IEEE Transactions on Visualization and Computer Graphics 19*, 7, 1242–1251.

LIU, T., AND SHEN, L. 2008. Fluid Flow and Optical Flow. *Journal of Fluid Mechanics 614* (11), 253–291.

MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid Control Using the Adjoint Method. *ACM Transactions on Graphics 23*, 3, 449–456.

MEINHARDT-LLOPIS, E., PÉREZ, J. S., AND KONDERMANN, D. 2013. Horn-Schunck Optical Flow with a Multi-Scale Strategy. *Image Processing On-Line 2013*, 151–172.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based Fluid Simulation for Interactive Applications. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 154–159.

NARAIN, R., SEWALL, J., CARLSON, M., AND LIN, M. C. 2008. Fast Animation of Turbulence Using Energy Transport and Procedural Synthesis. *ACM Transactions on Graphics 27*, 5, article 166.

NIELSEN, M. B., CHRISTENSEN, B. B., ZAFAR, N. B., ROBLE, D., AND MUSETH, K. 2009. Guiding of Smoke Animations through Variational Coupling of Simulations at Different Resolutions. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 217–226.

PANIGRAHI, P. K., AND MURALIDHAR, K. 2012. *Schlieren and Shadowgraph Methods in Heat and Mass Transfer*. Springer.

PARIKH, N., AND BOYD, S. 2013. Proximal Algorithms. *Foundations and Trends in Optimization 1*, 3, 123–231.

PFAFF, T., AND THUEREY, N., 2013. MantaFlow. *http://mantaflow.ethz.ch*.

SETTLES, G. S. 2001. *Schlieren and Shadowgraph Techniques*. Experimental Fluid Mechanics. Springer.

SHI, L., AND YU, Y. 2005. Taming Liquids for Rapidly Changing Targets. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 229–236.

STAM, J. 1999. Stable Fluids. In *Proceedings ACM SIGGRAPH*, 121–128.

STOMAKHIN, A., SCHROEDER, C., CHAI, L., TERAN, J., AND SELLE, A. 2013. A Material Point Method for Snow Simulation. *ACM Transactions on Graphics 32*, 4, article 102.

SU, L. K., AND DAHM, W. J. A. 1996. Scalar Imaging Velocimetry Measurements of the Velocity Gradient Tensor Field in Turbulent Flows. I. Assessment of Errors. *Physics of Fluids 8*, 7, 1869–1882.

SUN, D., ROTH, S., AND BLACK, M. J. 2014. A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles behind Them. *International Journal of Computer Vision 106*, 2, 115–137.

THUEREY, N., KEISER, R., RUEDE, U., AND PAULY, M. 2006. Detail-Preserving Fluid Control. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 7–12.

TROPEA, C., YARIN, A. L., AND FOSS, J. F. 2006. *Handbook of Experimental Fluid Mechanics*. Springer.

VAN VLIET, E., VAN BERGEN, S., DERKSEN, J., PORTELA, L., AND VAN DEN AKKER, H. 2004. Time-Resolved, 3D, Laser-Induced Fluorescence Measurements of Fine-Structure Passive Scalar Mixing in a Tubular Reactor. *Experiments in Fluids 37*, 1–21.

VLASENKO, A., AND SCHNÖRR, C. 2009. Variational Approaches to Image Fluid Flow Estimation with Physical Priors. In *Imaging Measurement Methods for Flow Analysis*, vol. 106. Springer, 247–256.

WANG, H., LIAO, M., ZHANG, Q., YANG, R., AND TURK, G. 2009. Physically Guided Liquid Surface Modeling from Videos. *ACM Transactions on Graphics 28*, 3, article 90.

YUAN, J., SCHNÖRR, C., AND MÉMIN, E. 2007. Discrete Orthogonal Decomposition and Variational Fluid Flow Estimation. *Journal of Mathematical Imaging and Vision 28*, 1, 67–80.

ZHU, Y., AND BRIDSON, R. 2005. Animating Sand as a Fluid. *ACM Transactions on Graphics 24*, 3, 965–972.