3D Fluid Flow Estimation with Integrated Particle Reconstruction

Katrin Lasinger[1][0000−0002−9329−2916], Christoph Vogel[2], Thomas Pock[2,3][0000−0001−6120−1058], and Konrad Schindler[1][0000−0002−3172−9246]

[1] Photogrammetry and Remote Sensing, ETH Zurich, Switzerland
[2] Institute of Computer Graphics and Vision, Graz University of Technology, Austria
[3] AIT Austrian Institute of Technology, Vienna, Austria

Abstract. The standard approach to densely reconstruct the motion in a volume of fluid is to inject high-contrast tracer particles and record their motion with multiple high-speed cameras. Almost all existing work processes the acquired multi-view video in two separate steps: first, a per-frame reconstruction of the particles, usually in the form of soft occupancy likelihoods in a voxel representation; followed by 3D motion estimation, with some form of dense matching between the precomputed voxel grids from different time steps. In this sequential procedure, the first step cannot use temporal consistency considerations to support the reconstruction, while the second step has no access to the original, high-resolution image data. We show, for the first time, how to *jointly* reconstruct both the individual tracer particles and a dense 3D fluid motion field from the image data, using an integrated energy minimization. Our hybrid Lagrangian/Eulerian model explicitly reconstructs individual particles, and at the same time recovers a dense 3D motion field in the entire domain. Making particles explicit greatly reduces the memory consumption and allows one to use the high-resolution input images for matching. Whereas the dense motion field makes it possible to include physical a-priori constraints and account for the incompressibility and viscosity of the fluid. The method exhibits greatly ($\approx 70\%$) improved results over a recent baseline with two separate steps for 3D reconstruction and motion estimation. Our results with only two time steps are comparable to those of state-of-the-art tracking-based methods that require much longer sequences.

1 Introduction

The capture and recovery of 3D motion in a transparent medium is a complex and challenging task, with important applications in different fields of science and technology: Observations of fluid motion and fluid-structure interaction form the basis of experimental fluid dynamics. Measuring and understanding flow and turbulence patterns is important for aero- and hydrodynamics in the automotive, aeronautic and ship-building industries, e.g. to design efficient shapes and to test the elasticity of components. Biological sciences are another application field, e.g., behavioral studies about aquatic organisms that live in flowing water [25].
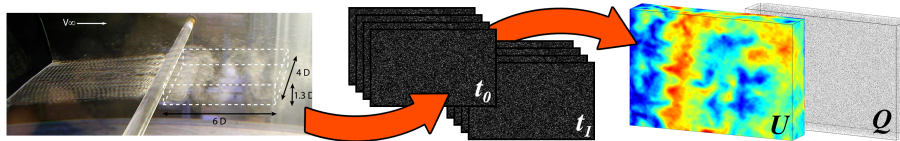
Fig. 1: From 2D images of a fluid injected with tracer particles, recorded at two consecutive time steps, we jointly reconstruct the particles $\mathcal{Q}$ and a dense 3D motion field $\mathcal{U}$. The example shows the experimental setup of [24].

A state-of-the-art technology to capture fluid motion in the laboratory is *particle image velocimetry* (PIV) [2, 32]. The underlying idea is to inject tracer particles into the fluid, which densely cover the illuminated volume of interest, and to record them with high-speed cameras from multiple viewpoints. Figure 1 shows the basic setup: 3D particle positions and a dense motion field are recovered from a set of input images from two consecutive time steps.

Due to computational limitations, early variants were restricted to 2D, by illuminating only a thin slice of the volume, thus neglecting interactions between different slices. More recent methods operate on the full 3D volume, but divide the problem in two independent steps. First, they recover the 3D particle distribution in the volume using all views at a single time step, by means of *tomographic PIV* (TomoPIV) [14], or sparse representations [29]. Second, they employ dense correspondence estimation between consecutive particle volumes to obtain the 3D motion of the fluid. The latter step is often simply an exhaustive matching of large enough local 3D windows [10, 13, 11] or, more recently, a variational flow estimation [20], with suitable priors on the motion field to limit the size of the matching window. We argue that any such two-step approach is sub-optimal, for several reasons. At each time step, the same particles are reconstructed without looking at the images at the other time step, effectively halving the available data. E.g., ignoring that a particle strongly supported by the later frame "must have come from somewhere". Similarly, valuable information is lost when discarding the input images after the first step. E.g., one cannot check whether a weak particle that unnaturally distorts the motion field "should be there at all"; particularly since physical limitations in practice restrict camera placement to narrow baselines, such that the initial reconstruction is ambiguous. Moreover, to achieve a resolution similar to the original images the volume must be discretized at high resolution, making the computation memory-hungry and expensive. Also, it is well documented that large 3D matching windows are needed for good performance, which undermines the benefits of high resolution.

In this work we propose a *joint energy model* for the reconstruction of the particles and the 3D motion field, so as to capture the inherent mutual dependencies. The model uses the full information – all available images from both time steps at full resolution – to solve the problem. We opt for a hybrid *Lagrangian/Eulerian* approach: particles are modeled individually, while the motion field is represented on a dense grid. Recovering explicit particle locations and intensities avoids the need for a costly 3D parameterization of voxel occupancy,

as well as the use of a large matching window. Instead, it directly compares evidence for single particles in the images, yielding significantly higher accuracy.

To represent the motion field, we opt for a trilinear finite element basis. Modeling the 3D motion densely allows us to incorporate physical priors that account for incompressibility and viscosity of the observed fluid [20]. This can be done efficiently, at a much lower voxel resolution than would be required for particle reconstruction, due to the smoothness of the 3D motion field [11, 20].

We model our problem in a variational setting. To better resolve particle ambiguities, we add a prior to our energy that encourages sparsity. In order to overcome weak minima of the non-convex energy, we include a proposal generation step that detects putative particles in the residual images, and alternates with the energy minimization. For the optimization itself, we can rely on the very efficient *inertial Proximal Alternating Linearized Minimization* (iPALM) [9, 30]. It is guaranteed to converge to a stationary point of the energy and has a low memory footprint, so that we can reconstruct large volumes.

Compared to our baselines [14] and [20], which both address the problem sequentially with two independent steps, we achieve particle reconstructions with higher precision and recall, at all tested particle densities; and greatly improved motion estimates. The estimated fluid flow visually appears on par with state-of-the-art techniques like [36], which use tracking over multiple time steps and highly engineered post-processing [38, 15].

2   Related Work

In experimental fluid mechanics, two strategies have evolved for flow estimation from tracer particles: PIV and *particle tracking velocimetry* (PTV) [2, 32]. PIV outputs a dense velocity field (Eulerian view), while PTV computes a flow vector at each particle location (Lagrangian view). The first method to operate in 3D was 3D-PTV [22], where individual particles are detected in different views, triangulated and tracked over time. Yet, as the particle density increases the particles quickly start to overlap in the images, leading to ambiguities. Therefore, 3D-PTV is only recommended for densities < 0.005 ppp (particles per pixel). To handle higher densities, [14] introduced Tomo-PIV. They first employ a tomographic reconstruction (e.g. MART) [3] per time step, to obtain a 3D voxel space of occupancy probabilities. Cross-correlation with large local 3D windows ($\geq 35^3$) [10, 13, 11] then yields the flow. Effectively, this amounts to matching particle constellations, assuming constant flow in large windows, which smoothes the output to low effective resolution. Recently, a new particle tracking method *Shake-the-Box* (StB) was introduced [36]. It builds on the idea of *iterative particle reconstruction* (IPR) [46], where triangulation is performed iteratively, with a local position and intensity refinement after every triangulation step.

None of the above methods accounts for the physics of (incompressible) fluids during reconstruction. In StB [36], a post-process interpolates sparse tracks to a regular grid, at that step (but not during reconstruction), physical constraints can be included. Variational approaches that impose physical consistent regular-

ization were first proposed for the 2D PIV setup, e.g. [35, 34]. [20] combine To-moPIV with variational 3D flow estimation and account for physical constraints, with a regularizer derived from the stationary Stokes equations, similar to [34]. However, their data term requires a huge, high-resolution intensity volume, and a local window of 11^3 voxels for matching, which lowers spatial resolution, albeit less than earlier local matching. [16] proposed a similar approach for dye-injected two-media fluids. Their aim are visually pleasing, rather than physically correct results, computed for relatively small volumes ($\approx 100^3$ voxels). We note that dye leads to data that is very different from tracer particles, e.g., it produces structures that can be matched more easily, but does not evenly cover the volume. [12] use compressive sensing to jointly recover the location and motion of a sparse, time-varying signal with a mathematical recovery guarantee. Results are only shown for small grids (256^3), and the physics of fluids is not considered. [4] introduce a joint approach for 3D reconstruction and flow estimation, however, without considering physical properties of the problem. Their purely Eulerian, voxel-based setup limits the method to small volume sizes, i.e., the method is only evaluated on a $61 \times 61 \times 31$ grid and a rather low seeding density of 0.02 ppp. [47] propose a joint formulation for their single-camera PIV setup. The volume is illuminated by rainbow-colored light planes that encode depth information. This permits the use of only a single camera with the drawback of lower depth accuracy and limited particle density. Voxel occupancy probabilities are recovered on a 3D grid. To handle the ill-conditioned problem from a single camera, constraints on particle sparsity and motion consistency (including physical constraints) are incorporated in the optimization. The method operates on a "thin" maximum volume of $512 \times 270 \times 20$. The single-camera setup does not allow a direct comparison with standard 3D PIV/PTV, but can certainly not compete in terms of depth resolution. In contrast, by separating the representation of particles and motion field, our hybrid Lagrangian/Eulerian approach allows for sub-pixel accurate particle reconstruction and large fluid volumes. Finally, [33] propose a hybrid discrete particle and continuous variational motion estimation approach. However, particle reconstruction and motion estimation are performed sequentially and without a physically motivated regularization of the flow.

Volumetric fluid flow is also related to variational scene-flow estimation, especially methods that parameterize the scene in 3D space [5, 42]. Like those, we search for the geometry and motion of a dynamic scene and exploit multiple views, yet our goal is a dense reconstruction in a given volume, rather than a pixel-wise motion field. Scene flow has undergone an evolution similar to the one for 3D fluid flow. Early methods started with a fixed, precomputed geometry estimate [44, 31], with a notable exception [17]. Later work moved to a joint reconstruction of geometry and motion [5, 40, 42]. Likewise, [14, 20] precompute the 3D tracer particles [14] before estimating their motion. The present paper is, to our knowledge, the first in multi-camera PIV to jointly determine the explicit particle geometry and (physically-constrained) motion of the fluid. Hence, our model can be seen either as an extension of the flow algorithm [20] to simultaneously reconstruct the particles, or an extension of TomoPIV with sparse

representations [28, 29], to simultaneously reconstruct the particles at both time steps, and the associated motion field.

Several scene flow methods [41, 43, 23] overcome the large state space by sampling geometry and motion proposals, and perform discrete optimization over those samples. In a similar spirit, we employ IPR to generate discrete particle proposals, but then combine them with a continuous, variational optimization scheme. We note that discrete labeling does not suit our task: The volumetric setting would require a large number of labels (3D vectors), and enormous amounts of memory. And it does not lend itself to sub-voxel accurate inference.

3 Method

To set the scene, we restate the goal of our method: densely predict the 3D flow field in a fluid seeded with tracer particles, from multiple 2D views acquired at two adjacent time steps.

We aim to do this in a direct, integrated fashion. The joint particle reconstruction and motion estimation is formulated as a hybrid Lagrangian/Eulerian model, where we recover individual particles and keep track of their position and appearance, but reconstruct a continuous 3D motion field in the entire domain. A dense sampling of the motion field makes it technically and numerically easier to adhere to physical constraints like incompressibility. In contrast, modeling particles explicitly takes advantage of the low particle density in PIV data. Practical densities are around 0.1 particles per pixel (ppp) in the images. Depending on the desired voxel resolution, this corresponds to $10-1000\times$ lower volumetric density. Our complete pipeline is depicted in Fig. 2. It alternates between generating particle proposals (Sec. 3.2) based on the current residual images (starting from the raw input images), and energy minimization to update all particles and motion vectors (Sec. 3.3). The correspondent energy model is described in Sec. 3.1. In the process, particle locations and flow estimates are progressively refined and provide a better initialization for further particle proposals.

Particle triangulation is highly ambiguous, so the proposal generator will inevitably introduce many spurious "ghost" particles (Fig. 3). A sparsity term in the energy reduces the influence of low intensity particles that usually correspond to such ghosts, while true particles, given the preliminary flow estimate, receive additional support from the data of the second time step. In later iterations, already reconstructed particles vanish in the residual images. This allows for a refined localization of remaining particles, as particle overlaps are resolved.

Notation and Preliminaries. The scene is observed by K calibrated cameras $\mathcal{K}_k, k = 1, \ldots, K$, recording the images $\mathcal{I}_k^t$ at time t. Parameterizing the scene with 3D entities obviates the need for image rectification. In our formulation we do not commit to a particular camera model and instead define a generic projection operator Π_k per camera. We note that fluid experiments typically need sophisticated models to deal with refraction (air-glass and glass-water), or an optical transfer function derived from volume self-calibration. Both are not the focus of this work and can be found in the respective literature, e.g. [45, 37].
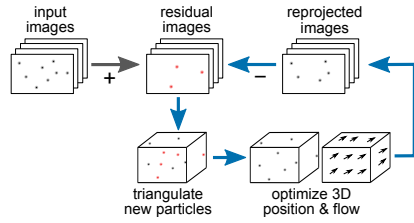
Fig. 2: Particle position and flow estimation pipeline. We alternate between joint optimization of 3D particle positions and dense flow vectors, and adding new candidate particles by triangulation from the residual images.
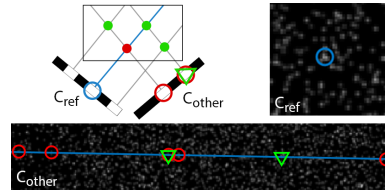
Fig. 3: A particle in the reference camera (circle) can lead to multiple epipolar-consistent putative matches (circles). However, only a subset of them represents true 3D particles (triangles). *Left*: 1D-illustration. *Right*: peak in reference camera (0.1ppp). *Bottom:* other camera view with 5 putative matches that are consistent over all 4 cameras.

The dependency on time is denoted via superscript t_0, t_1, and omitted when possible. We denote the set of particles $\mathcal{Q} := \{(p_i, c_i)\}_{i=1}^Q$, composed of a set of intensities $\mathcal{C} := \{c_i\}_{i=1}^Q$, $c_i \in \mathbb{R}_0^+$ and positions $\mathcal{P} := \{p_i\}_{i=1}^Q$, where each $p_i \in \mathbb{R}^3$ is located in the rectangular domain $\Omega \subset \mathbb{R}^3$. The 3D motion field at position $x \in \Omega$, between times t_0 and t_1, is $u(x, \mathcal{U})$. The set $\mathcal{U}$ contains motion vectors $u \in \mathbb{R}^3$ located at a finite set of positions $y \in \mathcal{Y} \subset \Omega$. If we let these locations coincide with the particle positions, we would arrive at a fully Lagrangian design, also referred to as *smoothed particle hydrodynamics* [26, 1]. In this work, we prefer a fixed set $\mathcal{Y}$ and represent the functional $u(x, \mathcal{U})$ by trilinear interpolation, i.e. we opt for an Eulerian description of the motion. Our model is, thus, similar to the so-called *particle in cell design* [48]. W.l.o.g. we assume $\mathcal{Y} \subset \Omega \cap \mathbb{Z}^3$, i.e., we set up a regular grid of vertices $\mathbf{i} \in \mathcal{Y}$ of size $N \times M \times L$, which induce a voxel covering $V(\Omega)$ of size $N-1 \times M-1 \times L-1$ of the whole domain. Each grid vertex $\mathbf{i} = (\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3)^\mathsf{T}$ is associated with a trilinear basis function: $\mathbf{b_i}(x) := \prod_{l=1}^3 \max(0, 1 - |x_l - \mathbf{i}_l|)$, for $x = (x_1, x_2, x_3)^\mathsf{T}$. The elements $u_\mathbf{i} \in \mathcal{U}$ now represent the coefficients of our motion field function $u(x, \mathcal{U}), x \in \Omega$ that is given by:

$$u(x,\mathcal{U}) = \big(u_1(x,\mathcal{U}), u_2(x,\mathcal{U}), u_3(x,\mathcal{U})\big)^\mathsf{T} \ , \quad u_l(x,\mathcal{U}) = \sum_{\mathbf{i} \in \mathcal{Y}} \mathbf{b_i}(x) u_{\mathbf{i},l} \, , l{=}1,2,3. \quad (1)$$

3.1   Energy Model

With the definitions above, we can write the energy

$$E(\mathcal{P}, \mathcal{C}, \mathcal{U}) := \frac{1}{2} E_\mathrm{D}(\mathcal{P}, \mathcal{C}, \mathcal{U}) + \frac{\lambda}{2} E_\mathrm{S}(\mathcal{U}) + \mu E_\mathrm{Sp}(\mathcal{C}), \qquad (2)$$

with a data term E_D, a smoothness term E_S operating on the motion field, and a sparsity prior E_Sp operating on the intensities of the particles.

Data Term. To compute the data term, the images of all cameras at both time steps are predicted from the particles' positions and intensities, and the 3D

motion field. E_D penalizes deviations between predicted and observed images:

$$E_D(\mathcal{P}, \mathcal{C}, \mathcal{U}) := \sum_{t \in \{t_0, t_1\}} \sum_{k=1}^{K} \int_{\Gamma_k} \left| \mathcal{I}_k^t(x) - \sum_{i=1}^{Q} \Pi_k(c_i \mathcal{N}(p_i + [t = t_1] \cdot u(p_i, \mathcal{U}), \sigma)(x)) \right|_2^2 \mathrm{d}x. \quad (3)$$

Following an additive (in terms of particles) image formation model, we integrate over the image plane Γ_k of camera k; $[\cdot]$ denotes the Iverson bracket. We model individual particles $(p, c) \in \mathcal{Q}$ as Gaussian blobs with variance σ^2. Particles do not exhibit strong shape or material variations. Their distance to the light source does influence the observed intensity. But since it changes smoothly and the cameras record with high frame-rate, assuming constant intensity is a valid approximation for our two-frame model.

In practice, the projection in (3) can be assumed to be almost orthographic, with little perspective distortion of the particles. The depth range of the volume Ω is small compared to the distance from the camera. Hence, we assume that particles remain Gaussian after a projection into the image. In that regime, and omitting constant terms, the expression for a projected particle simplifies to

$$\Pi\big(\mathcal{N}(\cdot, \sigma)(x)\big) \approx \mathcal{N}\big(\Pi(\cdot), \sigma\big)(x) \propto \sigma^{-1} \exp\big(-|\Pi(\cdot) - x|^2 \sigma^{-2}\big). \quad (4)$$

When computing the derivatives of (3) w.r.t. the set of parameters, we do not integrate the particle blobs over the whole image, but restrict the area of influence of (4) to a radius of 3σ, covering 99.7% of its total intensity.

Sparsity Term. The majority of the generated candidate particles do not correspond to *true* particles. To remove the influence of the huge set of low-intensity ghost particles one can exploit the expected sparsity of the solution, e.g. [29]. In other words, we aim to reconstruct the observed scenes with *few, bright* particle, by introducing the following energy term:

$$E_{\mathrm{Sp}}(\mathcal{C}) := \sum_{i=1}^{Q} |c_i|_\diamond + \delta_{\{\geq 0\}}(c_i). \quad (5)$$

Here, $\delta_\Delta(\cdot)$ denotes the indicator function of the set Δ. This term additionally excludes negative intensities. Although not directly related to sparsity, we identified (5) a convenient spot to include this constraint. Popular sparsity-inducing norms $|\cdot|_\diamond$ are either the 1- or 0-norm ($|\cdot|_\diamond = |\cdot|_1$, respectively $|\cdot|_\diamond = |\cdot|_0$). We have investigated both choices and prefer the stricter 0-norm for the final model. The 0-norm counts the number of non-zero intensities and rapidly discards particles that fall below a certain intensity threshold (modulated by μ in (2)). While the 1-norm only gradually reduces the intensities of weakly supported particles.

Smoothness Term. To define a suitable smoothness prior we follow [20] and employ a quadratic regularizer per component of the flow gradient, plus a term that enforces a divergence-free motion field:

$$E_{\mathrm{S}}(\mathcal{U}) := \int_\Omega \sum_{l=1}^{3} |\nabla u_l(x, \mathcal{U})|_2^2 + \delta_{\{0\}}\big(\nabla \cdot u(x, \mathcal{U})\big) \mathrm{d}x. \quad (6)$$

It has been shown in [20] that (6) has a physical interpretation, in that the stationary Stokes equations emerge as the Euler-Lagrange equations of the energy (6), including an additional force field. Thus, (6) models the incompressibility of the fluid, while λ represents its viscosity. [20] also suggest a variant in which the hard divergence constraint is replaced with a soft penalty:

$$E_{\mathrm{S},\alpha}(\mathcal{U}) := \int_{\Omega} \sum_{l=1}^{3} |\nabla u_l(x,\mathcal{U})|_2^2 + \alpha |\nabla \cdot u(x,\mathcal{U})|^2 \mathrm{d}x. \qquad (7)$$

This version simplifies the numerical optimization, trading off speed for accuracy. For adequate (large) α, the results are similar to the hard constraint in (6). Eq. (6) requires the computation of divergence $\nabla \cdot$ and gradients ∇ of the 3D motion field. Following the definition (1) of the flow field, both entities are linear in the coefficients $\mathcal{U}$ and constant per voxel $v \in V(\Omega)$. A valid discretization of the divergence operator can be achieved via the divergence theorem:

$$\int_v \nabla \cdot u(x) \mathrm{d}x = \int_{\partial v} \langle \nu(x), u(x,\mathcal{U}) \rangle \mathrm{d}x = \sum_{\mathbf{i}} \int_{\partial v} b_{\mathbf{i}}(x) \langle \nu(x), u_{\mathbf{i}} \rangle \mathrm{d}x = \frac{1}{4} \sum_{l,(\mathbf{i},\mathbf{j}) \in \mathcal{Y} \cap v : \mathbf{i}-\mathbf{j} = e_l} u_{\mathbf{i},l} - u_{\mathbf{j},l}, \quad (8)$$

where we let $\nu(x)$ denote the outward-pointing normal of voxel v at position $x \in v$ and e_l the unit vector in direction l. The final sum considers pairs of corner vertices $(\mathbf{i},\mathbf{j}) \in \mathcal{Y} \cap v$ of voxel v, adjacent in direction l. The definition of the per-voxel gradient follows from (1) in a similar manner.

3.2   Particle Initialization

To obtain an initial set of 3D particle locations we employ a direct detect-and-triangulate strategy like IPR [46] and iteratively triangulate putative particles, in alternation with the minimization of energy (2). Particle triangulation is extremely ambiguous and not decidable with local cues (Fig. 3). Instead, *all* plausible correspondences are instantiated. One can interpret the process as a proposal generator for the set of particles, which interacts with the sparsity constraint (5). This proposal generator creates new candidate particles where image evidence remains unexplained. The sparsity prior ensures that only "good" particles survive and contribute to the data costs, whereas those of low intensity that are inconsistent with our model become "zero-intensity" particles. Particles of low intensity are uncommon in reality. In each iteration the set of zero-intensity particles are actively discarded from $\mathcal{Q}$ to reduce the workload. Note that this does not change the energy of the current solution. After the first particles and a coarse motion field have been reconstructed, a better initialization is available to spawn further particles, in locations suggested by the residual maps between predicted and observed images. Particles that contribute to the data are retained in the subsequent optimization and help to refine the motion field, etc. The procedure is inspired by the heuristic, yet highly efficient, iterative approach of [46]. They also refine particle candidates triangulated from residual images. Other than theirs, our updated particle locations follow from a joint spatio-temporal objective, and

thus also integrate information from the second time step. In more detail, each round of triangulation proceeds as follows: first, detect peaks in 2D image space for all cameras at time step t_0. In the first iteration this is done in the raw inputs, then in the residual images $\mathcal{I}_{k,\mathrm{res}}^{t_0} := \int_{\Gamma_k} \mathcal{I}_k^{t_0}(x) - \sum_{i=1}^{Q} \Pi_k(c_i \mathcal{N}(p_i, \sigma)(x)) \mathrm{d}x$. Peaks are found by non-maximum suppression with a 3×3 kernel, followed by sub-pixel refinement of all peaks with intensity above a threshold I_{min}. We treat one of the cameras, $k = 1$, as reference and compute the entry and exit points to Ω for a ray passing through each peak. Reprojecting the entry and exit into other views yields epipolar line segments, along which we scan for (putatively) matching peaks (Fig. 3). Whenever we find peaks in all views that can be triangulated with a reprojection error below a tolerance ϵ, we generate a new candidate particle. Its initial intensity is set as a function of the intensity in the reference view and the number of candidates: if m proposals p_i are generated at a peak in the reference image, we set $c_i := \mathcal{I}_1\big(\Pi_1(p_i)\big) K/(K-1+m)$ for each of them.

3.3   Energy Minimization

Our optimization is embedded in a two-fold coarse-to-fine scheme. On the one hand, we start with a larger value for σ, so as to increase the particles' basins of attraction and improve convergence. During optimization, we progressively reduce σ until we reach $\sigma = 1$, meaning that a particle blob covers approximately the same area as in the input images. On the other hand, we also start at a coarser grid $\mathcal{Y}$ and refine the grid resolution along with σ.

To minimize the non-convex and non-smooth energy (2) for a given σ, we employ PALM [9], in its inertial variant [30]. Because our energy function is semi-algebraic [9] , it satisfies the Kurdyka-Lojasiewicz property [8], therefore the sequence generated by PALM globally converges to a critical point of the energy. The key idea of PALM is to split the variables into blocks, such that the problem is decomposed into one smooth function on the entire variable set, and a sum of non-smooth functions in which each block is treated separately. We start by arranging the locations and intensities of the particles $\mathcal{Q}$ into two separate vectors $\mathbf{p} := (p_1^\mathsf{T}, \ldots, p_Q^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{3Q}$ and $\mathbf{c} := (c_1, \ldots, c_Q)^\mathsf{T} \in \mathbb{R}^Q$. Similarly, we stack the coefficients of the trilinear basis $\mathbf{u} := (u_{\mathbf{i},1}^\mathsf{T}, u_{\mathbf{i},2}^\mathsf{T}, u_{\mathbf{i},3}^\mathsf{T})_{\mathbf{i} \in \mathcal{Y}}^\mathsf{T} \in \mathbb{R}^{3NML}$. With these groups, we split the energy functional into a smooth part H and two non-smooth functions, $F_{\mathbf{c}}$ for the intensities $\mathbf{c}$ and $F_{\mathbf{u}}$ for the motion vectors $\mathbf{u}$:

$$E(\mathbf{p}, \mathbf{c}, \mathbf{u}) := H(\mathbf{p}, \mathbf{c}, \mathbf{u}) + F_{\mathbf{c}}(\mathbf{c}) + F_{\mathbf{u}}(\mathbf{u}) + F_{\mathbf{p}}(\mathbf{p}), \text{ with}$$

$$H(\mathbf{p}, \mathbf{c}, \mathbf{u}) := E_D(\mathbf{p}, \mathbf{c}, \mathbf{u}) + \lambda \sum_{l=1}^{3} \|\nabla \mathbf{u}_l\|^2, \tag{9}$$

$$F_{\mathbf{c}}(\mathbf{c}) := \mu E_{Sp}(\mathbf{c}), \; F_{\mathbf{u}}(\mathbf{u}) := \delta_{\{0\}}(\nabla \cdot \mathbf{u}) \text{ and } F_{\mathbf{p}}(\mathbf{p}) := 0.$$

For notation convenience, we define $F_{\mathbf{p}}(\mathbf{p}) := 0$. The algorithm then alternates the steps of a proximal forward-backward scheme: take an explicit step w.r.t. one block of variables $z \in \{\mathbf{p}, \mathbf{c}, \mathbf{u}\}$ on the smooth part H of the energy function, then take a backward (proximal) step on the non-smooth part F_z w.r.t. the same

---

Algorithm 1 iPalm implementation for energy (9)

1: **procedure** IPALM($\mathbf{p}^0, \mathbf{c}^0, \mathbf{u}^0$)
2:     $\mathbf{p}^{-1} \leftarrow \mathbf{p}^0; \mathbf{c}^{-1} \leftarrow \mathbf{c}^0; \mathbf{u}^{-1} \leftarrow \mathbf{u}^0; \tau \leftarrow \frac{1}{\sqrt{2}}; L_\mathbf{p} \leftarrow 1; L_\mathbf{c} \leftarrow 1; L_\mathbf{u} \leftarrow 1;$
3:     **for** n:=0 to n_{steps} and while not converged **do**
4:         $\hat{\mathbf{p}} \leftarrow \mathbf{p}^n + \tau(\mathbf{p}^n - \mathbf{p}^{n-1});$ // inertial step
5:         **while** true **do**
6:             $\mathbf{p}^{n+1} := \hat{\mathbf{p}} - 1/L_\mathbf{p} \nabla_\mathbf{p} H(\hat{\mathbf{p}}, \mathbf{c}^n, \mathbf{u}^n);$
7:             **if** $\mathbf{p}^{n+1}, L_\mathbf{p}$ fulfill (12) **then break**; **else** $L_\mathbf{p} = 2L_\mathbf{p};$
8:         $\hat{\mathbf{c}} \leftarrow \mathbf{c}^n + \tau(\mathbf{c}^n - \mathbf{c}^{n-1});$ // inertial step
9:         **while** true **do**
10:             $\mathbf{c} := \hat{\mathbf{c}} - 1/L_\mathbf{c} \nabla_\mathbf{c} H(\mathbf{p}^{n+1}, \hat{\mathbf{c}}, \mathbf{u}^n); \mathbf{c}^{n+1} := \text{prox}_{L_\mathbf{c}}^{F_\mathbf{c}}(\mathbf{c});$ // Eq. (10)
11:             **if** $\mathbf{c}^{n+1}, L_\mathbf{c}$ fulfill (12) **then break**; **else** $L_\mathbf{c} = 2L_\mathbf{c};$
12:         $\hat{\mathbf{u}} \leftarrow \mathbf{u}^n + \tau(\mathbf{u}^n - \mathbf{u}^{n-1});$ // inertial step
13:         **while** true **do**
14:             $\mathbf{u} := \hat{\mathbf{u}} - 1/L_\mathbf{u} \nabla_\mathbf{u} H(\mathbf{p}^{n+1}, \mathbf{c}^{n+1}, \hat{\mathbf{u}}); \mathbf{u}^{n+1} := \text{prox}_{L_\mathbf{u}}^{F_\mathbf{u}}(\mathbf{u});$ // Eq. (10)
15:             **if** $\mathbf{u}^{n+1}, L_\mathbf{u}$ fulfill (12) **then break**; **else** $L_\mathbf{u} = 2L_\mathbf{u};$

---

variables. That is, we alternate steps of the form

$$z^{n+1} = \text{prox}_t^{F_z}(z) := \arg\min_y F_z(y) + \frac{t}{2}\|y - z\|^2, \text{ with } z = z^n - \frac{1}{t}\nabla_z H(\cdot, z^n, \cdot), \quad (10)$$

with a suitable step size $1/t$ for each block of variables. Here and in the following, the placeholder variable z can stand for $\mathbf{c}, \mathbf{p}$ or $\mathbf{u}$, as required.

A key property is that, throughout the iterations, the partial gradient of function H w.r.t. a variable block $z \in \{\mathbf{p}, \mathbf{c}, \mathbf{u}\}$ must be globally Lipschitz-continuous with some modulus L_z at the current solution:

$$\|\nabla_z H(\cdot, z_1, \cdot) - \nabla_z H(\cdot, z_2, \cdot)\| \leq L_z(\cdot, \cdot)\|z_1 - z_2\| \, \forall z_1, z_2. \quad (11)$$

In other words, before we accept an update z^{n+1} computed with (10), we need to verify that the step size t in (10) fulfills the descent lemma [7]:

$$E(\cdot, z^{n+1}, \cdot) \leq E(\cdot, z^n, \cdot) + \langle \nabla_z H(\cdot, z^n, \cdot), z^{n+1} - z^n \rangle + \frac{t}{2}\|z^{n+1} - z^n\|^2. \quad (12)$$

Note that Lipschitz continuity of the gradient of H has to be verified only locally, at the current solution. This property allows for a back-tracking approach to determine the Lipschitz constant, e.g. [6]. Algorithm 1 provides pseudo-code for our scheme to minimize the energy (9). To accelerate convergence we apply extrapolation (lines 4/8/12). These inertial steps, c.f. [30, 6], significantly reduce the number of iterations in the algorithm, while leaving the computational cost per step practically untouched. It is further convenient to not only reduce the step sizes (lines 7/11/15 in Alg. 1), but also to increase them, as long as (12) is fulfilled, to make the steps per iteration as large as possible.

One last thing needs to be explained, namely how we find the solution of the proximal steps on the intensities $\mathbf{c}$ and flow vectors $\mathbf{u}$. The former can be solved point-wise, leading to the following 1D-problem:

$$\text{prox}_t^{F_c}(\bar{c}) := \arg\min_c \mu|c|_\diamond + \delta_{\{\geq 0\}}(c) + \frac{t}{2}|c - \bar{c}|^2, \quad (13)$$

which admits for a closed-form solution for both norms ($\diamond \in \{0, 1\}$):

$$\text{prox}_t^{|\cdot|_0}(\bar{c}) := \begin{cases} 0 & \text{if } t\bar{c}^2 < 2\mu \text{ or } \bar{c} < 0 \\ \bar{c} & \text{else,} \end{cases}, \qquad \text{prox}_t^{|\cdot|_1}(\bar{c}) := \max(0, \bar{c} - \mu/t). \quad (14)$$

The proximal step for the flow vector $\mathbf{u}$, $\text{prox}_t^{F_\mathbf{u}}(\bar{\mathbf{u}}) := \arg\min_\mathbf{u} \delta_{\{0\}}(\nabla \cdot \mathbf{u}) + \frac{t}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2$, requires the projection of $\bar{\mathbf{u}}$ onto the space of divergence-free 3D vector fields. Given $\bar{\mathbf{u}}$, the solution is independent of the step size $1/t$, which we omit in the following. We construct the Lagrangian by introducing the multiplier ϕ, a scalar vector field whose physical meaning is the pressure in the fluid [20]:

$$\min_\mathbf{u} \max_\phi \frac{1}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2 + \phi^\mathsf{T}\nabla \cdot \mathbf{u}. \quad (15)$$

To prevent confusion, we introduce $D\mathbf{u}$ as matrix notation for the linear divergence operator ($\nabla \cdot \mathbf{u}$) in (8). The KKT conditions of the Lagrangian yield a linear equation system. Simplification with the Schur complement leads to a Poisson system, which we solve for the pressure ϕ to get the divergence-free solution:

$$\text{prox}_t^{F_\mathbf{u}}(\bar{\mathbf{u}}) := \bar{\mathbf{u}} - D^\mathsf{T}\phi \quad \text{with} \quad D^\mathsf{T}D\phi = D^\mathsf{T}\bar{\mathbf{u}}. \quad (16)$$

Again interpreted physically, the divergence of the motion field is removed by subtracting the gradient of the resulting pressure field. For our problem of fluid flow estimation, it is not necessary to exactly solve the Poisson system in every iteration. Instead, we keep track of the pressure field ϕ during optimization, and warm-start the proximal step. In this way, a few (10-20) iterations of preconditioned conjugate gradient descent suffice to update ϕ.

If we replace the hard divergence constraint with the soft penalty $E_{S,\alpha}$ from (7), we add $E_{S,\alpha}$ to the smooth function H in (9). Then only the proximal step on the intensities $\mathbf{c}$ is needed in Alg. 1. We conclude by noting that accelerating the projection step is in itself an active research area in fluid simulation [19, 39].

4   Evaluation

There is no other measurement technique that could deliver ground truth for fluid flow. We follow the standard practice and generate datasets for quantitative evaluation via *direct numerical simulations* (DNS) of turbulent flow, using the *Johns Hopkins Turbulence Database* (JHTDB) [21, 27]. This allows us to render realistic input images with varying particle densities and flow magnitudes, together with ground truth vectors on a regular grid. We evaluate how our approach performs with different smoothness terms, particle densities, initialization methods, particle sizes and temporal sampling rates. Additionally, we show results on "test case D" of the 4th International PIV Challenge [18]. We quantitatively compare to the best performing method [36] and refer to the supplementary material for further comparisons as well as additional visualizations, also with the experimental setup of Fig. 1.

Table 1: Endpoint error (AEE), angular error (AAE) and absolute divergence (AAD) for different regularizers (0.1 ppp).

	E_S	$E_{S,64}$	$E_{S,0}$	[20]
AEE	0.136	0.135	0.157	0.406
AAE	2.486	2.463	2.870	6.742
AAD	0.001	0.008	0.100	0.001

Fig. 4: Detail from an xy-slice of the flow in X-direction. *Left to right:* Ground truth, our method and result of [20].
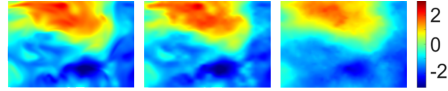
Simulated dataset. We follow the guidelines of the 4th International PIV Challenge [18] for the setup of our own dataset: Randomly sampled particles are rendered to four symmetric cameras of resolution 1500×800 pixels, with viewing angles of $\pm 35°$ w.r.t. the yz-plane of the volume, respectively $\pm 18°$ w.r.t. the xz-plane. If not specified otherwise, particles are rendered as Gaussian blobs with $\sigma = 1$ and varying intensity. We sample 12 datasets from 6 non-overlapping spatial and 2 temporal locations of the forced isotropic turbulence simulation of the JHTDB. Discretizing each DNS grid point with 4 voxels, identical to [18], each dataset corresponds to a volume size of $1024 \times 512 \times 352$. For our flow fields with flow magnitudes up to 8.8 voxels, we use 10 pyramid levels with downsampling factor 0.94. At every level we alternate between triangulation of candidate particles and minimization of the energy function (at most 40 iteration per level).

The effective resolution of the reconstructed flow field is determined by the particle density. At a standard density of ≈ 0.1 ppp and a depth range of 352 voxels, we get a density of ≈ 0.0003 particles per voxel. This suggests to estimate the flow on a coarser grid. We empirically found a particle density of 0.3 per voxel to still deliver good results. Hence, we operate on a subsampled voxel grid of 10-times lower resolution per dimension in all our experiments, to achieve a notable speed-up and memory saving. The computed flow is then upsampled to the target resolution, with barely any loss of accuracy.

We always require a 2D intensity peak in all four cameras to instantiate a candidate particle. We start with a strict threshold of $\epsilon = 0.8$ for the triangulation error, as suggested in [46], which is relaxed to $\epsilon = 2.0$ in later iterations. The idea is to first recover particles with strong support, and gradually add more ambiguous ones, as the residual images become less cluttered. We set $\lambda = 0.01$ for our dataset. Since λ corresponds to the viscosity it should be adapted for other fluids. We empirically set the sparsity weight $\mu = 0.0001$.

Regularization. Our framework allows us to plug in different smoothness terms. Following [20], we show results for hard (E_S) and soft divergence regularization ($E_{S,\alpha}$). *Average endpoint error* (AEE), *average angular error* (AAE), and *average absolute divergence* (AAD) are displayed in Tab. 1. Compared to our default regularizer E_S, removing the divergence constraint ($\alpha = 0$), increases the error by $\approx 15\%$. With the soft constraint at high $\alpha = 64$, the results are equal to those of E_S. We also compare to the method [20]. Our joint model improves the performance by $\approx 70\%$ over that recent baseline, on both error metrics. In Fig. 4 we visually compare our results (with hard divergence constraint) to those of [20]. The figure shows the flow in X-direction in one particular xy-slice of the volume. Our method recovers a lot finer details, and is clearly closer to the ground truth.

Table 2: Influence of particle density on our joint approach, as well as several baselines.

ppp	IPR joint			IPR sequential			MART			true particles		
	AEE	*prec.*	*recall*	AEE	*prec.*	*recall*	AEE	*prec.*	*recall*	AEE	*prec.*	*recall*
0.1	0.136	99.98	99.95	0.136	99.97	99.96	0.232	70.39	83.93	0.136	100	100
0.125	0.124	91.00	99.95	0.157	61.55	97.55	0.270	48.51	73.83	0.125	100	100
0.15	0.115	82.82	99.95	0.310	33.46	85.09	0.323	44.61	70.17	0.118	100	100
0.175	0.111	71.37	99.93	0.332	26.63	71.07	0.385	40.89	65.29	0.110	100	100
0.2	0.108	55.43	99.86	0.407	19.42	64.26	0.506	36.88	58.13	0.106	100	100

Particle Density & Initialization Method. There is a trade-off for choosing the seeding density: A higher density raises the observable spatial resolution, but at the same time makes matching more ambiguous. This causes false positives, commonly called "ghost particles". Very high densities are challenging for all known reconstruction techniques. The additive image formation model of Eq. (3) also suggests an upper bound on the maximal allowed particle density. Tab. 2 reports results for varying particle densities. We measure recall (fraction of reconstructed ground truth particles) and precision (fraction of reconstructed particles that coincide with true particles to < 1 pixel).

To provide an upper bound, we initialize our method with ground truth particle locations at time step 0 and optimize only for the flow estimation. We also evaluate a sequential version of our method, in which we separate energy (2) into particle reconstruction and subsequent motion field estimation. In addition to our proposed IPR-like triangulation, we initialize particles with a popular volumetric tomography method (*MART*) [14]. *MART* creates a full, high-resolution voxel grid of intensities (with, hopefully, lower intensities for ghost particles and very low ones in empty space). To extract a set of sub-voxel accurate 3D particle locations we perform peak detection, similar to the 2D case for triangulation. Since *MART* always returns the same particle set we run it only once, but increase the number of iterations for the minimizer from 40 to 160.

Starting from a perfect particle reconstruction (*true particles*) the flow estimate improves with increasing particle density. Remarkably, our proposed iterative triangulation approach achieves results comparable to the ground truth initialization, up to high particle densities and is able to resolve most particle ambiguities. In contrast, *MART* and the sequential baseline struggle with increasing particle density, which supports our claim that joint energy minimization can better reconstruct the particles.

Particle Size. For the above experiments, we have rendered the particles into the images as Gaussian blobs with fixed $\sigma = 1$, and the same is done when re-rendering for the data term, respectively, proposal generator. We now test the influence of particle size on the reconstruction, by varying σ. Tab. 3 shows results with hard divergence constraint and fixed particle density 0.1, for varying $\sigma \in [0.6 \ldots 1.6]$. For small enough particles, size does not matter, very large particles lead to more occlusions and degrade the flow. Furthermore, we verify the sensitivity of the method to unequal particle size. To that end, we draw an individual σ for each particle from the normal distribution $\mathcal{N}(1, 0.1^2)$, while still using a fixed $\sigma = 1$ during inference. As expected, the mismatch between actual and rendered particles causes slightly larger errors.

Table 3: Influence of particle size on reconstruction quality (0.1 ppp).

σ	0.6	0.8	1	1.2	1.4	1.6	$\mathcal{N}_{1,0.1}2$
AEE	0.194	0.135	0.136	0.140	0.217	0.235	0.155
AAE	3.388	2.465	2.486	2.561	4.002	4.575	2.879

Table 4: Varying the sampling distance between frames (0.1 ppp).

Temp. dist.	0.75x	1.0x	1.25x	1.5x
AEE	0.102	0.136	0.170	0.283
Max. $|u|_2$	6.596	8.795	10.993	13.192
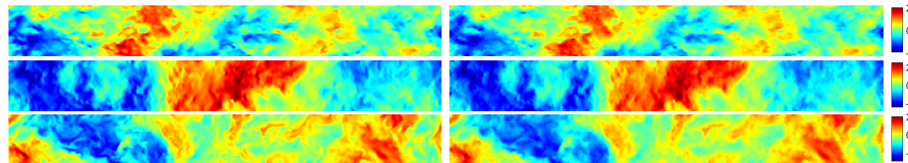
Fig. 5: xy-slice of the flow field for snapshot 10 of the 4th PIV Challenge. *Top to bottom: X, Y, Z* flow components. *Left:* multi-frame StB [36]. *Right:* our 2-frame method.

Temporal Sampling. To quantify the stability of our method to different flow magnitudes we modify the time interval between the two discrete time steps and summarize the results in Tab. 4, together with the respective maximum flow magnitude $|u|_2$. For lower frame rate (1.25x and 1.5x), and thus larger magnitudes, we set our pyramid downsampling factor to 0.93.

PIV Challenge. Unfortunately, no ground truth is provided for the data of the 4th PIV Challenge [18], such that we cannot run a quantitative evaluation on that dataset. However, Schanz et al. [36] kindly provided us results for their method, StB, for snapshot 10. StB was the best-performing method in the challenge with an endpoint error of $\approx$0.24 voxels (compared to errors >0.3 for all competitors). The average endpoint difference between our approach and StB is <0.14 voxels. In Fig. 5 both results appear to be visually comparable, yet, note that StB includes a tracking procedure that requires data of multiple time steps (15 for the given particle density 0.1). We show additional visualizations and a qualitative comparison with the ground truth in the supplementary material.

5   Conclusion

We have presented the first variational model that *jointly* solves sparse particle reconstruction and dense 3D fluid motion estimation in PIV data for the common multi-camera setup. The sparse particle representation allows us to utilize the high-resolution image data for matching, while keeping memory consumption low enough to process large volumes. Densely modeling the fluid motion in 3D enables the direct use of physically motivated regularizers, in our case viscosity and incompressibility. The proposed joint optimization captures the mutual dependencies of particle reconstruction and flow estimation. This yields results that are clearly superior to traditional, sequential methods [14, 20]; and, using only 2 frames, competes with the best available multi-frame methods, which require sequences of 15-30 timesteps.

References

1. Adams, B., Pauly, M., Keiser, R., Guibas, L.J.: Adaptively sampled particle fluids. ACM SIGGRAPH (2007)
2. Adrian, R., Westerweel, J.: Particle Image Velocimetry. Cambridge University Press (2011)
3. Atkinson, C., Soria, J.: An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. Experiments in Fluids **47**(4) (2009)
4. Barbu, I., Herzet, C., Mémin, E.: Joint Estimation of Volume and Velocity in TomoPIV. In: 10th Int'l Symp. on Particle Image Velocimetry - PIV13 (2013)
5. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: a view centered variational approach. CVPR (2010)
6. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences **2**(1), 183–202 (2009)
7. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice-Hall (1989)
8. Bolte, J., Daniilidis, A., Lewis, A.: The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J Optimiz **17**(4), 1205–1223 (2007)
9. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math Programming **146**(1), 459–494 (2014)
10. Champagnat, F., Plyer, A., Le Besnerais, G., Leclaire, B., Davoust, S., Le Sant, Y.: Fast and accurate PIV computation using highly parallel iterative correlation maximization. Experiments in Fluids **50**(4), 1169 (2011)
11. Cheminet, A., Leclaire, B., Champagnat, F., Plyer, A., Yegavian, R., Le Besnerais, G.: Accuracy assessment of a lucas-kanade based correlation method for 3D PIV. In: Int'l Symp. Applications of Laser Techniques to Fluid Mechanics (2014)
12. Dalitz, R., Petra, S., Schnörr, C.: Compressed motion sensing. SSVM (2017)
13. Discetti, S., Astarita, T.: Fast 3D PIV with direct sparse cross-correlations. Experiments in Fluids **53**(5) (2012)
14. Elsinga, G.E., Scarano, F., Wieneke, B., Oudheusden, B.W.: Tomographic particle image velocimetry. Experiments in Fluids **41**(6) (2006)
15. Gesemann, S., Huhn, F., Schanz, D., Schröder, A.: From noisy particle tracks to velocity, acceleration and pressure fields using B-splines and penalties. Int'l Symp. on Applications of Laser Techniques to Fluid Mechanics (2016)
16. Gregson, J., Ihrke, I., Thuerey, N., Heidrich, W.: From capture to simulation: connecting forward and inverse problems in fluids. ACM ToG **33**(4) (2014)
17. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. ICCV (2007)
18. Kähler, C.J., et al.: Main results of the 4th International PIV Challenge. Experiments in Fluids **57**(6) (2016)
19. Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., Gross, M.: Data-driven fluid simulations using regression forests. ACM ToG **34**(6) (2015)
20. Lasinger, K., Vogel, C., Schindler, K.: Volumetric flow estimation for incompressible fluids using the stationary stokes equations. ICCV (2017)
21. Li, Y., et al.: A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. J. of Turbulence **9** (2008)
22. Maas, H.G., Gruen, A., Papantoniou, D.: Particle tracking velocimetry in three-dimensional flows. Experiments in Fluids **15**(2) (1993)
23. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. CVPR (2015)

24. Michaelis, D., Poelma, C., Scarano, F., Westerweel, J., Wieneke, B.: A 3d time-resolved cylinder wake survey by tomographic piv. In: ISFV12 (2006)
25. Michalec, F.G., Schmitt, F., Souissi, S., Holzner, M.: Characterization of intermittency in zooplankton behaviour in turbulence. European Physical J **38**(10) (2015)
26. Monaghan, J.J.: Smoothed particle hydrodynamics. Reports on Progress in Physics **68**(8), 1703 (2005)
27. Perlman, E., Burns, R., Li, Y., Meneveau, C.: Data exploration of turbulence simulations using a database cluster. In: Conf. on Supercomputing (2007)
28. Petra, S., Schröder, A., Wieneke, B., Schnörr, C.: On sparsity maximization in tomographic particle image reconstruction. DAGM (2008)
29. Petra, S., Schröder, A., Schnörr, C.: 3d tomography from few projections in experimental fluid dynamics. In: Imaging Meas. Methods for Flow Analysis (2009)
30. Pock, T., Sabach, S.: Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. SIAM J Imaging Sci **9**(4) (2016)
31. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. ECCV (2010)
32. Raffel, M., Willert, C.E., Wereley, S., Kompenhans, J.: Particle image velocimetry: a practical guide. Springer (2013)
33. Ruhnau, P., Guetter, C., Putze, T., Schnrr, C.: A variational approach for particle tracking velocimetry. Measurement Science and Technology **16**(7), 1449 (2005)
34. Ruhnau, P., Schnörr, C.: Optical stokes flow estimation: an imaging-based control approach. Experiments in Fluids **42**(1), 61–78 (Jan 2007)
35. Ruhnau, P., Stahl, A., Schnörr, C.: On-line variational estimation of dynamical fluid flows with physics-based spatio-temporal regularization. GCPR (2006)
36. Schanz, D., Gesemann, S., Schröder, A.: Shake-The-Box: Lagrangian particle tracking at high particle image densities. Experiments in Fluids **57**(5) (2016)
37. Schanz, D., Gesemann, S., Schröder, A., Wieneke, B., Novara, M.: Non-uniform optical transfer functions in particle imaging: calibration and application to tomographic reconstruction. Measurement Science & Technology **24**(2) (2012)
38. Schneiders, J.F., Scarano, F.: Dense velocity reconstruction from tomographic PTV with material derivatives. Exp. in Fluids **57**(9) (2016)
39. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating eulerian fluid simulation with convolutional networks. CoRR **abs/1607.03597** (2016)
40. Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C., Theobalt, C.: Joint estimation of motion, structure and geometry from stereo sequences. ECCV (2010)
41. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. ICCV (2013)
42. Vogel, C., Schindler, K., Roth, S.: 3D scene flow estimation with a rigid motion prior. ICCV (2011)
43. Vogel, C., Schindler, K., Roth, S.: 3D scene flow estimation with a piecewise rigid scene model. IJCV **115**(1), 1–28 (2015)
44. Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., Cremers, D.: Stereoscopic scene flow computation for 3d motion understanding. IJCV **95**(1), 29–51 (2011)
45. Wieneke, B.: Volume self-calibration for 3d particle image velocimetry. Experiments in fluids **45**(4), 549–556 (2008)
46. Wieneke, B.: Iterative reconstruction of volumetric particle distribution. Measurement Science & Technology **24**(2) (2013)
47. Xiong, J., Idoughi, R., Aguirre-Pablo, A.A., Aljedaani, A.B., Dun, X., Fu, Q., Thoroddsen, S.T., Heidrich, W.: Rainbow particle imaging velocimetry for dense 3d fluid velocity imaging. ACM Trans. Graph. **36**(4), 36:1–36:14 (Jul 2017)
48. Zhu, Y., Bridson, R.: Animating sand as a fluid. ACM ToG **24**(3), 965–972 (2005)