

Snapshot Space–Time Holographic 3D Particle Tracking Velocimetry

Ni Chen,* Congli Wang, and Wolfgang Heidrich

Digital inline holography is an amazingly simple and effective approach for 3D imaging, to which particle tracking velocimetry is of particular interest. Conventional digital holographic particle tracking velocimetry techniques are computationally separated in particle and flow reconstruction, plus the expensive computations. Usually, the particle volumes are recovered first, from which fluid flows are computed. Without iterative reconstructions, This sequential space–time process lacks accuracy. This paper presents a joint optimization framework for digital holographic particle tracking velocimetry: particle volumes and fluid flows are reconstructed jointly in a higher space–time dimension, enabling faster convergence and better reconstruction quality of both fluid flow and particle volumes within a few minutes on modern GPUs. Synthetic and experimental results are presented to show the efficiency of the proposed technique.

1. Introduction

Since its invention in 1948, inline holography^[1] offers an easy means of measuring 3D information. Many variants have been proposed, including off-axis holography,^[2] phase-shifting holography,^[3] optical scanning holography,^[4] and many others. However, the simplicity and high space-bandwidth product of inline holography remain attractive and competitive. Particle imaging velocimetry (PIV) and particle tracking velocimetry (PTV) allow fluid velocities to be measured quantitatively across volumetric regions (typically at cm scales), and is an important technique for fluid visualization and understanding. Current widely-used 3D particle velocimetry acquisition setups include scanning PIV,^[5] defocusing PIV,^[6] tomographic PIV,^[7] Rainbow PIV,^[8] and single-camera structured light PTV.^[9] These photometric techniques inevitably suffer from the tradeoff between depth of field and exposure for fast motions, which in turn increases the

uncertainty for depth localization. To overcome these limitations, holographic PTV has been proposed.^[10] Digital holography^[11] offers a new alternative to the time consuming development process of the exposed hologram and the evaluation of the particle distribution, where holograms are recorded with an image sensor followed by numerical reconstruction of the object wave. Compared to digital inline holographic PTV (DIH-PTV),^[12] off-axis holography^[13] offers a smaller depth of focus by separating the twin and DC terms in the reconstructions. However, DIH-PTV remains attractive because of its simple and compact setup (only laser illumination with a bare sensor), single-view measurement

(low-cost), and the ability for depth-resolved particle reconstruction. This technique allows direct integration with a microscopic setup,^[14] and provides superior resolution for 3D flow measurements with the relevant scales ranging from μm to sub-mm.^[15,16]

Despite the drastic improvements in hardware, the software for domain-specific, application-oriented numerical reconstruction algorithms has remained mostly untouched. First, conventional algorithms are separate, in that particle and fluid flow reconstructions are sequential, with possible error accumulations throughout the computation pipeline. Second, there are works pushing forward for a better holographic particle/volume reconstruction by hand-crafted or trained priors, including depth-of-field extension with wavelet transform,^[17] depth-resolved reconstruction with 3D deconvolution,^[18] compressing sensing approach with fused lasso regularization^[19] or sparsity,^[20] digital filtering followed by one-pass 3D deconvolution,^[21] and the deep learning approaches,^[22–24] but little has been done in improving the fluid flow reconstruction.

Here we present a joint optimization scheme for particle flow reconstructions, outlined in **Figure 1**. Instead of reconstructing the particle volume and flow velocities sequentially,^[19,21,25] the proposed method recovers 3D particle volumes and flow velocities in a joint fashion, that is, flow velocities are utilized as a regularization prior for 3D particle reconstructions, and reversely flow velocities are further estimated given the previously updated 3D particles, with additional physically plausible priors incorporated, including particle sparsity, flow smoothness and Tikhonov regularization. These two computational processes keep iterating in alternation until convergence. No segmentation^[25] or further post-processing is needed.

Compared to conventional approaches, our proposed technique improves in both particle reconstruction quality (in terms

N. Chen, C. Wang, W. Heidrich
Visual Computing Center
King Abdullah University of Science and Technology
Thuwal 23955, Saudi Arabia
E-mail: ni.chen@kaust.edu.sa

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/lpor.202100008>

© 2021 The Authors. Laser & Photonics Reviews published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

DOI: 10.1002/lpor.202100008

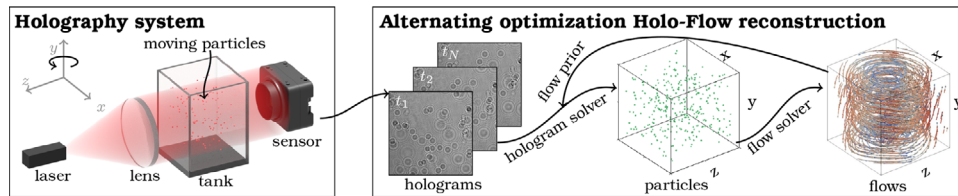


Figure 1. Overall pipeline of our DIH-PTV space-time particle-flow reconstruction algorithm. Given single-shot hologram images, we obtained simultaneously spatial particle volumes and temporal fluid flows by solving the challenging inverse problem Equation (6) via alternating optimization of custom solvers with domain-specific priors.

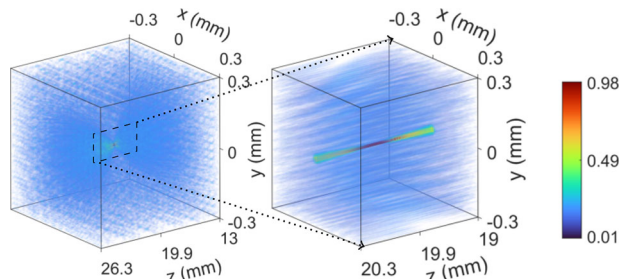


Figure 2. Hologram reconstruction of a single particle. Diffraction PSF spans across more regions in z than in x, y .

of reducing particle reconstruction uncertainties) and flow velocities. In addition to the algorithmic contributions, our method is highly parallelizable. The computational speed has increased significantly, reducing reconstruction times from typically hours on CPUs down to seconds on a single modern GPU. The applicability of our proposed numerical approach has been further validated using both synthetic and experimental flow fields. Such improvements will substantially enhance the implementation of DIH-PTV for 3D flow measurements and enable the potential commercialization of this technique.

2. Numerical Verification

In this section, we evaluate our proposed approach based on synthetic holograms for ground-truth comparisons.

2.1. Uncertainty of Holographic Particle Reconstruction

According to the Abbe diffraction limit, the resolution of an optical imaging system is fundamentally limited by the nature of diffraction.^[26] The extent of the diffraction-limited point spread function (PSF) along the optical axis is much larger than in the traversal direction. **Figure 2** shows a back-propagation of a single particle reconstruction. The long-tail PSF distribution along z -axis defines the uncertainty of conventional particle reconstructions, deteriorating particle reconstruction qualities in back-propagation.

2.2. Uncertainty Improvement with Holo-Flow

Previously mentioned uncertainty could be greatly reduced using the proposed Holo-Flow framework. **Figure 3** shows a reconstruction of a single particle, numerically performed by back-

propagation, FASTA deconvolution,^[19,37] and the proposed Holo-Flow reconstruction, respectively. Lateral and axial resolution were set to $5\ \mu\text{m}$, with particle diameter being $20\ \mu\text{m}$, corresponds to two lateral and one axial voxels. We performed 50 iterations for FASTA, and 5 (alternating iterations) and 10 (inner loop FASTA) for Holo-Flow reconstruction, with the total inner loop iterations being equal to a single-run FASTA. **Figure 3** visualizes the overall reconstruction quality (a), along x -axis (b) and z -axis (c). The ground-truth (GT), back-propagation (BP) and FASTA reconstructions are presented for comparison. From **Figure 3b**, We conclude both FASTA and Holo-Flow match the ground-truth perfectly, whereas our approach (Holo-Flow) outperforms FASTA in the z -axis.

It should be mentioned that noise-level affects the uncertainty of the particle reconstruction. **Figure 4** shows the uncertainty with respect to noise-level. Considering readout noise of the image sensor that would exist in real experiments, we added Gaussian white noise to the holograms. **Figure 4a** shows that the reconstructed particles along the x axis changes slightly according to varying noise level for both the conventional FASTA method and the proposed Holo-Flow. In **Figure 4b**, the reconstructed particle along the z axis with the FASTA performance degrades as the noise level increases, while for Holo-Flow it maintains the same. From this synthesis experiment we conclude that in comparison with alternative methods, Holo-Flow is more robust against noise.

2.3. Particle-Flow Joint Improvement with Holo-Flow

For a quantitative assessment, in **Figures 5** and **6**, we generated synthetic particle volumes and the measurement holograms ($t = 1, 2$) with two known fluid flows $\mathbf{v}(x, y, z) = (-y, x, 0)$ and $\mathbf{v}(x, y, z) = (0, z, -y)$, that is, two vortexes with the rotation axes aligned or orthogonal with the optical axis z . The particle volumes are of voxel number $128 \times 128 \times 128$, with a physical voxel size of $5\ \mu\text{m} \times 5\ \mu\text{m} \times 5\ \mu\text{m}$. We generated uniformly random-distributed single-voxel sized particles with a density of particles per pixel (ppp) at 2×10^{-2} , at wavelength $660\ \text{nm}$ with the sensor located at $z_1 = 1\ \text{mm}$ away from the volume of interest. The results of **Figures 5b** and **6b** are consistent with ground-truth flows in **Figures 5c** and **6c**.

Compared to a sequential flow reconstruction following the state-of-the-art hologram deconvolution method using FASTA,^[19] our joint Holo-Flow optimization framework performs a better particle and flow reconstruction. The particle and flow reconstruction error comparisons are shown in **Figures 7**

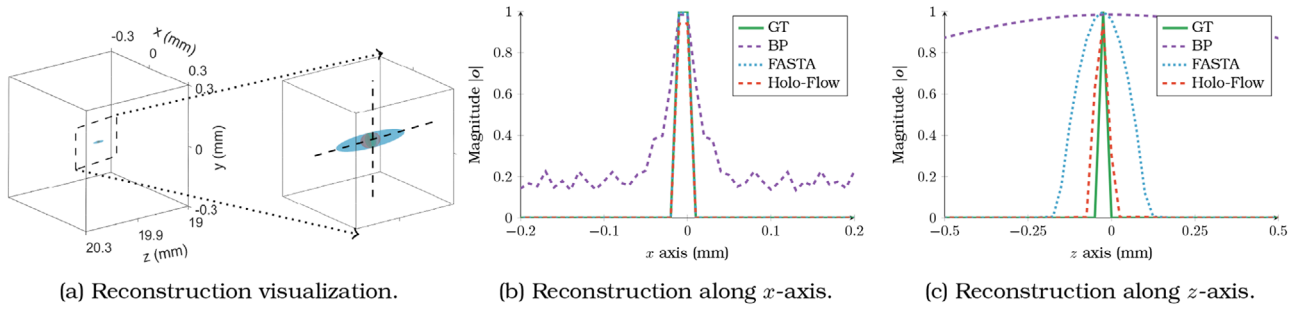


Figure 3. Single particle reconstruction uncertainty comparison using different methods.

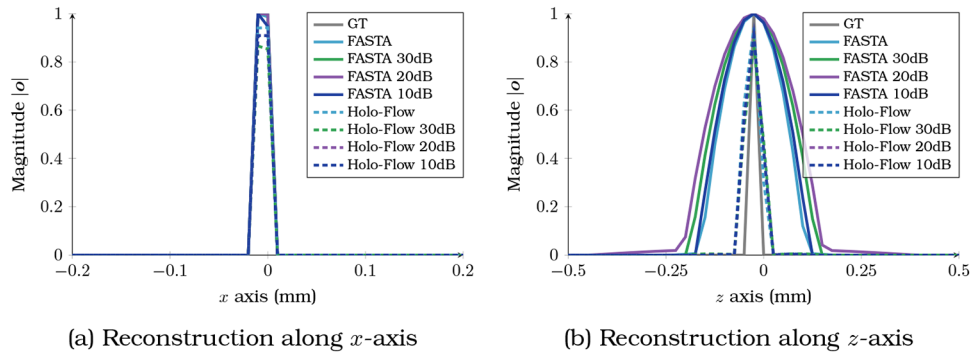


Figure 4. Single particle reconstruction uncertainty comparison under varying noise-level.

and 8. Figure 7a shows a general increase of particle localization (i.e., decreasing of axial size) with respect to the increase of iterations. Figure 7b contains particle size distribution and location errors for the above two simulations. The reconstructed particles' lateral sizes are with a mean value of one, which is the same as the ground-truth in both conventional and the Holo-Flow methods, refer to Supporting Information. In Figure 7b, the axial length of the reconstructed particles is no larger than three pixels in both simulations, comparing to five pixels in the conventional reconstructions. In the right column of Figure 7b, the number of particles with location error of zero in the Holo-Flow reconstruction is larger than in the conventional method, in other words, the location error of the Holo-Flow reconstruction is more concentrate around zero, indicating a more accurate particle localization for the Holo-Flow method. These results verify that the Holo-Flow method achieves a better particle reconstruction than the conventional way.

The joint Holo-Flow optimization method also improves in terms of flow reconstructions. We make use of two quantitative evaluation metrics for the assessment of the proposed Holo-Flow method. The metrics are the flow average angular error (AAE)^[27] and the average endpoint error (EPE)^[28] defined as:

$$AAE(\mathbf{v}_g, \mathbf{v}_e) = \arccos \left(\frac{\mathbf{v}_g \cdot \mathbf{v}_e + 1}{\sqrt{(\|\mathbf{v}_g\|_2^2 + 1)(\|\mathbf{v}_e\|_2^2 + 1)}} \right) \quad (1)$$

$$EPE(\mathbf{v}_g, \mathbf{v}_e) = \|\mathbf{v}_g - \mathbf{v}_e\|_2$$

where \mathbf{v}_g and \mathbf{v}_e are the ground-truth and estimated velocity, respectively. The AAE measure is convenient because it handles both large and very small speeds without the amplification inherent in a relative measure of vector differences. Although

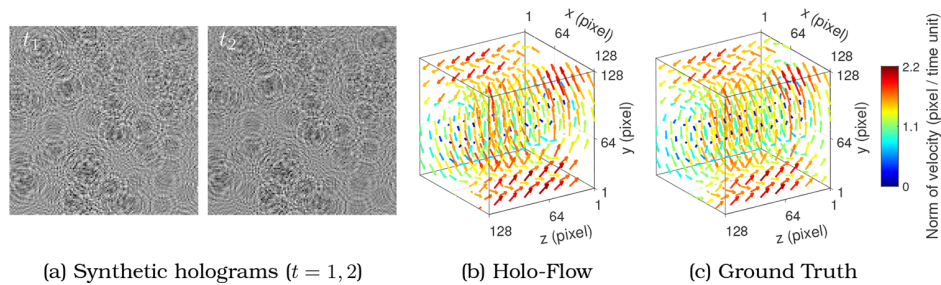


Figure 5. b) The reconstructed velocity vector fields from a) two holograms, c) induced by rotating the particle volume with a vector flow of $\mathbf{v}(x, y, z) = (-y, x, 0)$.

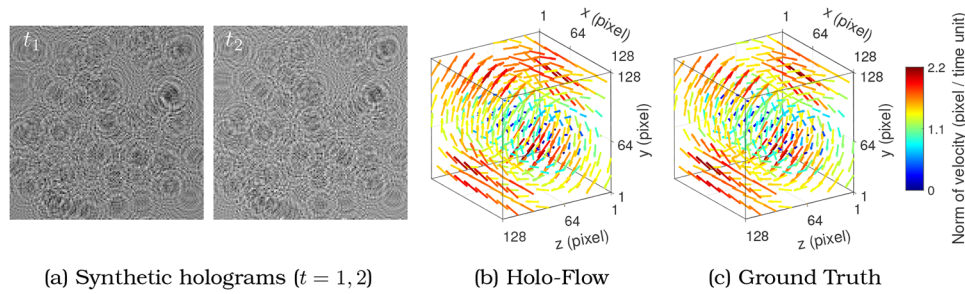


Figure 6. b) The reconstructed velocity vector fields a) from two holograms, c) induced by rotating the particle volume with a vector flow of $\mathbf{v}(x, y, z) = (0, z, -y)$.

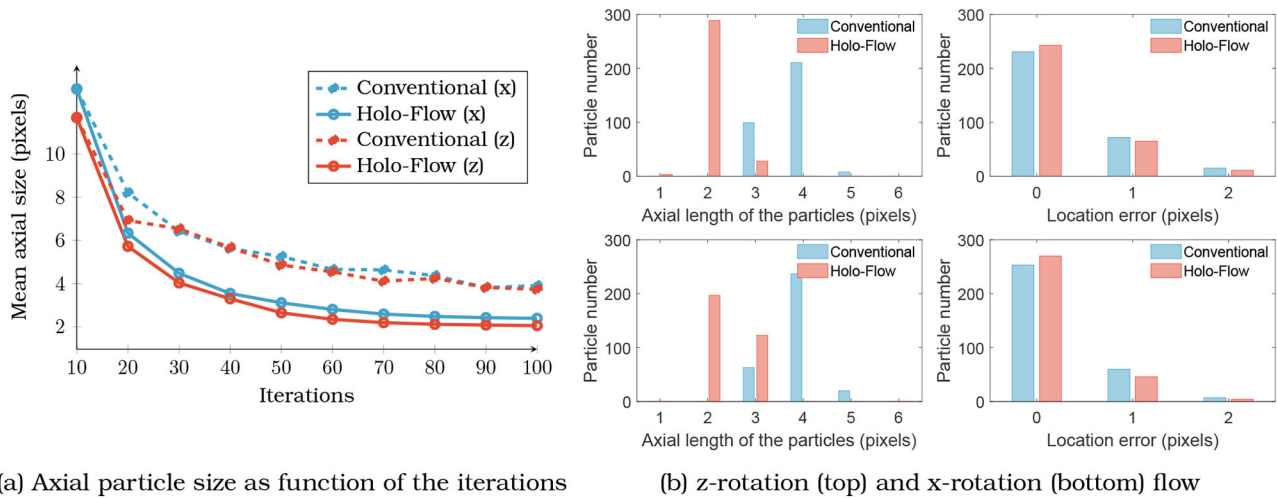


Figure 7. Particle size and localization comparisons of the reconstructed particle with ground truth for conventional and the proposed Holo-Flow methods.

the AAE is prevalent, it is unclear whether errors in a region of smooth non-zero motion should be penalized less than errors in regions of zero motion. The AAE also contains an arbitrary scaling constant (1.0) to convert the units from pixels to degrees. Hence, we also compute the EPE^[29]. Figure 8 shows the AAE and EPE with respect to the iterations for the conventional and Holo-Flow methods. In both error metrics, Holo-Flow achieves a better performance than the conventional sequential reconstructions.

2.4. Synthetic Turbulent Flow with Holo-Flow

We further verified the proposed method to a 3D turbulent flow, which is more challenging as it involves velocity fluctuations across a broad dynamic range of scales. Here we take a homogeneous isotropic turbulent flow as an example. **Figure 9a** shows the turbulence flow that was queried from by the forced isotropic turbulence data from the Johns Hopkins Turbulence Database^[30] with the default parameters of the direct numerical simulations

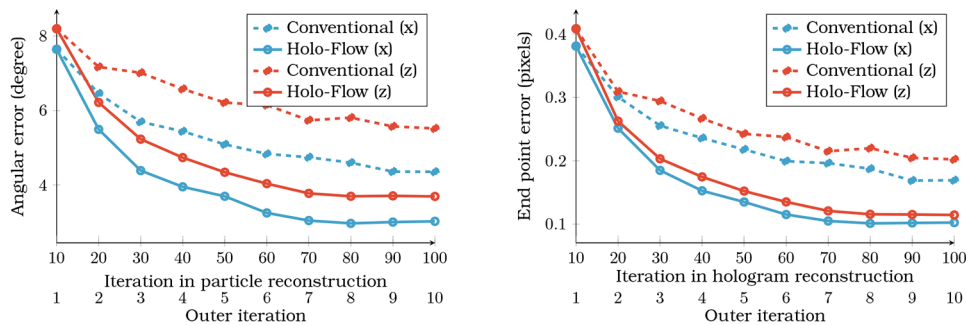


Figure 8. Flow reconstruction comparison in terms of angular errors (degrees) and end point errors (pixels).

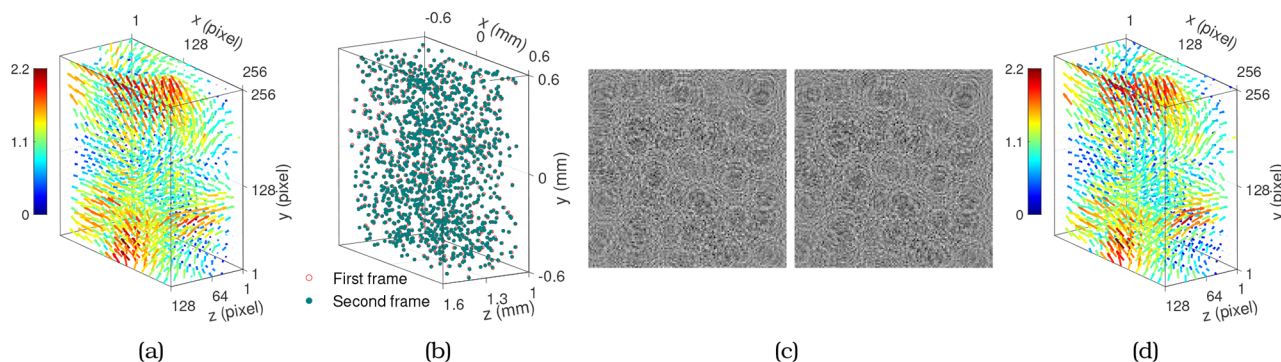


Figure 9. d) The reconstructed velocity vector fields from c) two hologram frames of b) particle volumes induced by a) a ground-truth turbulence flow.

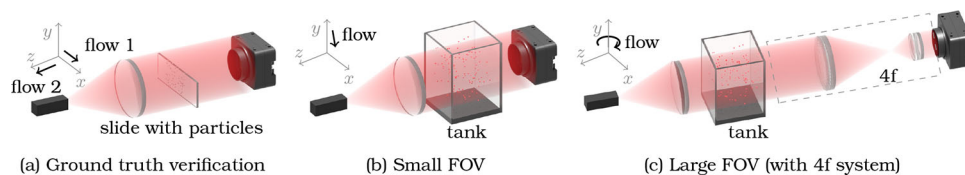


Figure 10. a) Schematic experimental setups for ground-truth (a), b) small-scale and c) larger-scale. The black arrow-headed lines show the flow directions in each case. In (a) flow 1 and flow 2 represent the flows along the x axis and z axis respectively, in (b) and (c), the arrow-headed lines represent injection and spinned flow respectively.

(DNS) data. The volume resolution of the turbulence flow is $256 \times 256 \times 128$, scaled to $1.28 \times 1.28 \times 0.62 \text{ mm}^3$. Particles with a radius of $2.5 \text{ }\mu\text{m}$ were randomly spatially seed throughout the volume, with their positions at subsequent time were determined by the turbulence flow, as shown in Figure 9b. The number of particles in the volume was 1310, which corresponds to a concentration of 1250 mm^{-3} . A padding in each dimension was applied to the particles to ensure that the number of objects in the field of view is constant. Figure 9c shows the two corresponding hologram frames with a resolution 256×256 pixels, simulated at a wavelength of 660 nm .

Compare the Holo-Flow reconstruction that shown in Figure 9d to the ground-truth of Figure 9a, the AAE and EPE were 5.736 and 0.1495, respectively. This indicates that the proposed method can also perform very well for a turbulence flow.

3. Experimental Results

We have conducted three experimental setups, as shown in Figure 10. In Figure 10a, particles were mounted on a microscopy slide that translates in fixed directions serving as ground truth flow. Figure 10b,c are the experimental capture setups with different field-of-view (FOV) configurations to image fluids at different scales. Particles are white micro-spheric polyethylene microspheres with diameters of $\approx 45\text{--}53 \text{ }\mu\text{m}$ and $\approx 150\text{--}180 \text{ }\mu\text{m}$. The black arrow-headed lines the coordinates of the figures show the flow directions. The light source is a laser diode of wavelength 660 nm , and the camera is a CMOS sensor with a pixel pitch of $3.45 \text{ }\mu\text{m}$ (PointGrey GS3-U3-51S5C in gray mode). The acquisition image resolution as well as the frame rate of the image sensor can be configured to different values via inbuilt operational modes. In Figures 10(a,b), the FOV is with the same size as the camera sensor, while in Figure 10c, the focal lengths of the first f_1

and second lens f_2 can be changed to adjust the FOV. Specifically, the FOV equals to f_1/f_2 multiplied by the sensor area.

3.1. Experiments with Ground-truth

To evaluate our proposed method's effectiveness on real captured data, we conduct the experiments with a microscopy slide with particles stuck in it. The particle size is $45\text{--}53 \text{ }\mu\text{m}$. The slide was put on a two-axis linear translation stage (Thorlabs XYT1/M) such that the proposed reconstructions can be compared with ground truth movements. The scheme of the experimental setup is shown in Figure 10a. Two independent tests were performed: 1) Two holograms were acquired while a translation of $50.8 \text{ }\mu\text{m}$ in the x direction (perpendicular to the optical axis) was applied. 2) Two holograms were acquired while a translation of $101.6 \text{ }\mu\text{m}$ in the z direction (along the optical axis and far away from the camera) was applied.

Before processing the captured holograms, we subtract the background image, then downsample them by a factor of 8. The downsampled image pixel size is $27.6 \text{ }\mu\text{m}$. We discretize the volume of interest ($\approx 15\text{--}60 \text{ mm}$) into 80 z -layers equivalent to a depth interval of $100 \text{ }\mu\text{m}$.

The reconstructed velocity vector fields and the corresponding holograms are visualized in Figure 11. The reconstructed flow's overall structures in both cases reveal that a significant part of the real flow structures is reproduced. We can further analyze the reconstructed results with respect to the ground truth movements. In the experiments, the x -axis and z -axis translations move $50.8 \text{ }\mu\text{m}$ and $101.6 \text{ }\mu\text{m}$ in one step, which corresponds to 1.8 and 1 pixels in the reconstructed images. The mean of the norm of the velocity in Figure 11a is 1.73 pixel sizes with a standard deviation of 0.16, while the mean of that in Figure 11b is 0.98 pixel

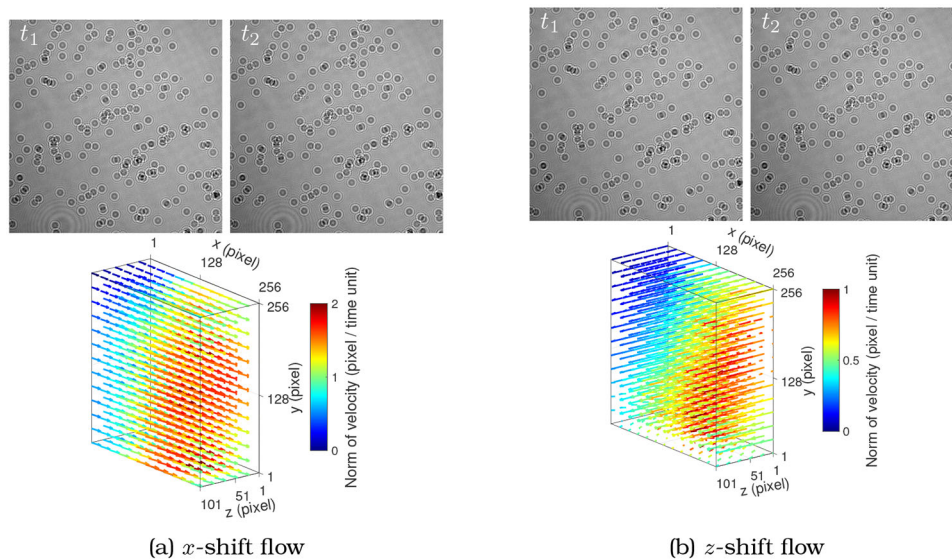


Figure 11. The reconstructed velocity vector fields from the hologram frames induced by moving the particle volume with a translation stage.

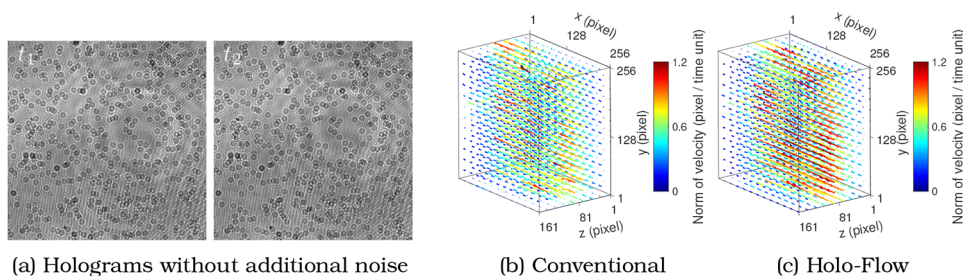


Figure 12. Comparison without noise and with noise (induced by adjust the gain of the CMOS sensor).

sizes with a standard deviation of 0.02, demonstrating a closely matched quantity compared to the theoretical values.

Particles with diameters of $\approx 45\text{--}53\text{ }\mu\text{m}$ and $\approx 96\text{--}605\text{ }\mu\text{m}$ were adhered to two glass slides in varying layers. Compared to the previous example, different particle densities have also been applied. To compare the reconstruction flow, we had made a flow from the reconstructed particles with the known flow magnitude. **Figure 12** shows the EPE for the conventional and Holo-Flow method are 0.38 and 0.36, and AAE are 18.47 and 16.98, respectively. These quantitative values may suffer from misalignment errors or setup systematic errors, but at least provide a sense of the improvements that Holo-Flow has achieved compared to the prior conventional approach.

3.2. Experiments without Ground-truth

We also tested the proposed technique on three real practical flows. Parameter details are listed in **Table 1**.

In the first experiment, a tank containing seeded particles in high viscosity liquid was illuminated by a laser beam, as shown in **Figure 10b**. We injected water from the top of the tank and captured hologram image sequence at a frame rate of 38 Hz. In the flow reconstruction, we downsampled the holograms by a factor of 8 from an original resolution of 2048×2048 to 256×256 .

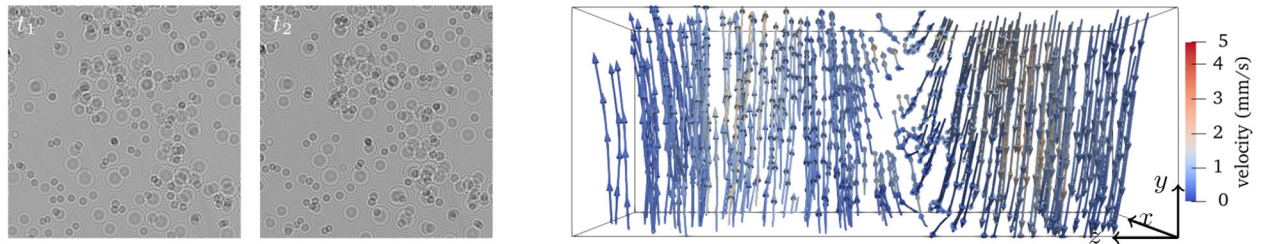
The depth was discretized into 201 layers (0.15 mm per layer), completing the final grid resolution at $256 \times 256 \times 333$, with a corresponding voxel size of $27.6\text{ }\mu\text{m} \times 27.6\text{ }\mu\text{m} \times 150\text{ }\mu\text{m}$. The path-line visualization of the recovered flow field is shown in **Figure 13a**. In the part of the tank that near the camera, the flow direction is downward, which is induced by the downward water injection. On the contrary, in the tank's portions that are farther away from the camera, the flow is upward. This is due to the bounces when the particles approach the bottom of the tank.

In the second experiment, we spun the water clock-wisely from top of the tank and captured hologram sequences through a 4f system that consists of two achromatic lenses (focal length of 200 and 60 mm), as shown in **Figure 10c**. Holograms were downsampled by the same factor as in the previous experiment. The depth was discretized into 301 layers (0.1 mm per layer with a voxel resolution of $55.2\text{ }\mu\text{m} \times 55.2\text{ }\mu\text{m} \times 100\text{ }\mu\text{m}$). **Figure 13b** shows the path-line visualization of the reconstructed vector flow. As expected, the swirl rotates clock-wisely along the y axis. The reconstruction only shows a part of the swirl because of the limited FOV of the experiment (around 23.6 mm), about one half of the total tank width.

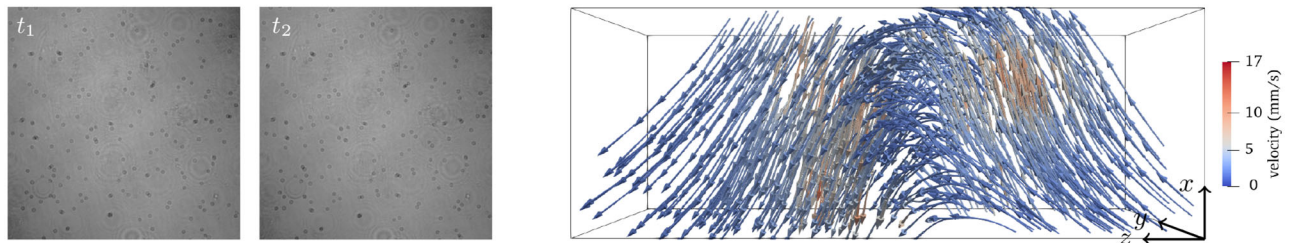
We further made a more complicated flow in the third experiment. We injected water with a needle from the middle bottom of the tank. The water bounced back from the water surface of the tank makes a vortex-like flow. Holograms were downsampled by

Table 1. Parameters of the experiments.

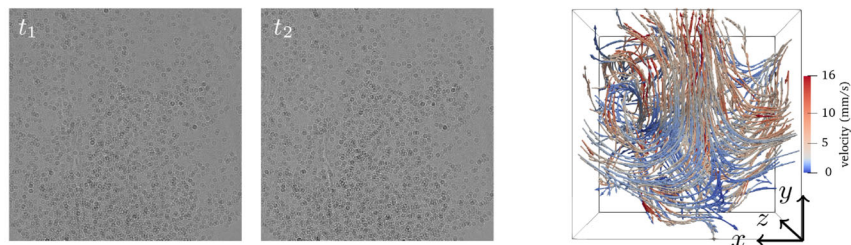
| | | Injection flow | Spun flow | Vortex-like flow |
|--------------------|----------------|---|---|---|
| Holography system | Setup | Figure 10b | Figure 10c | Figure 10c |
| | Specifications | — | $f_1 = 200 \text{ mm}, f_2 = 60 \text{ mm}$ | $f_1 = 300 \text{ mm}, f_2 = 60 \text{ mm}$ |
| | FOV | $7.1 \times 7.1 \text{ mm}^2$ | $23.67 \times 23.67 \text{ mm}^2$ | $35.5 \times 35.5 \text{ mm}^2$ |
| | Frame rate | 15 Hz | 38 Hz | 25 Hz |
| Particles | Size | $\approx 45\text{--}53 \mu\text{m}$ | $\approx 150\text{--}180 \mu\text{m}$ | $\approx 45\text{--}53 \mu\text{m}$ |
| Original hologram | Resolution | 2048×2048 | 2048×2048 | 1024×1024 |
| | Pixel pitch | $3.45 \mu\text{m}$ | $3.45 \mu\text{m}$ | $6.9 \mu\text{m}$ |
| Processed hologram | Resolution | 256×256 | 256×256 | 512×512 |
| | Pixel pitch | $27.6 \mu\text{m}$ | $55.2 \mu\text{m}$ | $13.8 \mu\text{m}$ |
| Imaging volume | Resolution | $256 \times 256 \times 334$ | $256 \times 256 \times 334$ | $512 \times 512 \times 251$ |
| | Size | $7.1 \times 7.1 \times 50 \text{ mm}^3$ | $23.67 \times 23.67 \times 50 \text{ mm}^3$ | $35 \times 35 \times 25 \text{ mm}^3$ |



(a) Injected fluid flow (small FOV configuration)



(b) Rotation fluid flow (large FOV configuration)



(c) Vortex fluid flow (large FOV configuration)

Figure 13. Path line visualization of the reconstructed fluid flows (see also Visualizations 1 and 2^[31]).

the a factor of two. The depth was discretized into 251 layers (0.1 mm per layer with a voxel resolution of $13.8 \mu\text{m} \times 13.8 \mu\text{m} \times 100 \mu\text{m}$). Figure 13c shows the path-line visualization of the reconstructed vector flow. As the left part of the streamline figure shows, the vortex rings rotate along the water flow, that is, upright in the middle part of the tank and downright in the side parts.

The recovered flow field in all the cases shown in Figure 13 are consistent with both the expectations and the observed hologram frames. For all experimental cases, the outer iterations are 3, and the inner iteration of the hologram solver is 20, and $\tau = 0.001$. Specific prior weights are tuned to plausible values. The reconstruction of the second flow (Figure 13c), in which the volume is with a voxel number of $256 \times 256 \times 334$, cost around 1 min on a Ubuntu 18.04 Linux workstation with 2.70GHz Intel(R) Xeon(R) CPU E5-2680, 62.9GB RAM, and a NVIDIA TITAN X (Pascal) GPU.

4. Discussions and Conclusion

We have introduced a novel holographic imaging method that is capable of recovering 3D fluid flows and particle volumes using an inline holography system, meanwhile significantly reducing the hardware setup requirements and relaxing calibration complexity. We formulate an inverse problem to reconstruct the particle volumes. A further step is taken to reconstruct the fluid velocity vector fields with the recovered particle volumes at different times. The fluid velocity vector fields further improve the particle volume reconstructions.

We demonstrate our method both on synthetic flows, ground-truth flows induced by moving a frozen particle volume, and a real stirred flow. Overall, our method can robustly reconstruct a significant part of the flow structures at good accuracy.

The primary drawback of our method is the limited FOV. Even with the help of a 4f system, the inevitable wavefront degradation would reduce the reconstructions' spatial resolution, introducing systematic errors into the image formation model. One way to overcome this FOV limitation is to illuminate the volume with spherical wavefronts (known as inline holography with spherical illumination^[32]), resulting in a modification of the system PSF, so that the full FOV can be covered by a limited size image sensor. Also, the iterative nature of our algorithm requires a good approximate initialization to start, in that particle volume and fluid flows are constrained to each other and may prevent high-quality reconstructions. Finally, the tolerance of our fluid pyramid optical flow reconstruction algorithm, by its nature, only allows for subtle particle displacements (less than 4 pixels), and thus preventing the method itself to be applied to fast fluid flows when the camera capture frame rate is low. Future works may revise the current pyramid downsampling to be a maximum projection operator to allow for deeper pyramids to conserve particles without losing them due to blurring.

There are several ways to elaborate on the spirit of our approach further. In software, reconstruction quality can be potentially improved by varies basis functions or learned ones.^[22–24] The framework is eligible in principle for different hardware configurations as well, resulting in a change of the forward image formation model (matrix **A**) in Equation (6).

Despite these current limitations, on account of the simple setup and good accuracy, our system can be easily implemented

and applied to investigate new types of fluid flows in the future. Our proposed framework could also be inspiring in other space-time diffraction imaging applications with different data manifolds other than the particle volumes presented here, for example optical diffraction tomography^[33] and X-ray ptychography,^[34] and Fourier ptychography,^[35] reducing the demanding measurements in a conventional way.

5. Methods

We conduct two types of experiments, where the first one is to move particles with known motion vector, verifying the accuracy of our methods on real data; The second one is to reconstruct practical fluid flows.

5.1. Digital Inline Holography

Digital inline holography aimed to recover a 3D objective transfer function from a single 2D image. Particles were absorptive and the amplitudes of the objective transfer functions were always less than one. For simplicity, the reciprocity field of objective transfer functions were referred to as particle volume, denoted as $o(\mathbf{r}, z) \in \mathbb{R}$ with $\mathbf{r} = (x, y)$, and hence in this work only amplitude reconstructions were considered. In inline holography, under a planar uniform wavefront, the scalar field $u_z(\mathbf{r}) \in \mathbb{C}$ at the sensor plane can be formulated as an interference between reference wave $r_z(\mathbf{r})$ and the convolution of $o(\mathbf{r}, z)$ and free-space propagation kernel $h_0(\mathbf{r}, z) \in \mathbb{C}$ (see Appendix A for derivation). The 2D image $I(\mathbf{r})$ is written as:

$$I(\mathbf{r}) = |r_z(\mathbf{r}) + u_z(\mathbf{r})|^2 \quad \text{of} \quad u_z(\mathbf{r}) = \int_{z_1}^{z_2} o(\mathbf{r}, z) \otimes h_0(\mathbf{r}, z) dz \quad (2)$$

where \otimes denotes convolution. In practice, the particle volume was sparse, that is, for most regions $o(\mathbf{r}, z) = 0$. This observation led to a plausible negligence of the squared term in Equation (2), with $\{\cdot\}$ denoting the complex conjugate:

$$I(\mathbf{r}) = |r_z(\mathbf{r})|^2 - 2\Re\{u_z(\mathbf{r})\} + |u_z(\mathbf{r})|^2 \approx 1 - 2\Re\{u_z(\mathbf{r})\overline{r_z(\mathbf{r})}\} \quad (3)$$

For the sake of brevity, all measurements were post-processed and constant-free, that is, $b(\mathbf{r}) = [I(\mathbf{r}) - 1]/2$. Also $r_z(\mathbf{r})$ can be rewritten as a convolution (see Appendix A), resulting in a change of convolution kernel $h_0(\mathbf{r}, z) \rightarrow h(\mathbf{r}, z)$. This completed the linear image formation model, through which the Gabor holography related the 3D particle volumes $o(\mathbf{r}, z)$ to the 2D measurements $b(\mathbf{r})$:

$$b(\mathbf{r}) \approx \Re\left\{\int_{z_1}^{z_2} o(\mathbf{r}, z) \otimes h(\mathbf{r}, z) dz\right\} \quad (4)$$

5.2. Motion Flow

The temporal evolution of particles defines an apparent movement of the underlining particle volume, known as fluid velocities. Motion flows can be numerically reconstructed from consecutive, rapid measurements of changing particle volumes. Under

the assumption of incompressibility, that is, divergence-free flow, the motion flow $\mathbf{v}_t \in \mathbb{R}^3$ related temporally neighboring particle volumes from time t to $t + 1$.^[36] That is, with $\mathbf{x} = (\mathbf{r}, z)$:

$$o_t(\mathbf{x} + \mathbf{v}_t) \approx o_{t+\Delta t}(\mathbf{x}) \quad (5)$$

where subscripts (t and $t + \Delta t$) denote for two neighboring time frames.

5.3. Inverse Problem

Given a sequence of indexed 2D holograms $b_t(\mathbf{r})$ of $t = 1, 2, \dots, T$, at each frame t , its 3D particle volume $o_t(\mathbf{x})$ and the associated 3D flow vector $\mathbf{v}_t(\mathbf{x})$ would simultaneously be likely to be recovered. One obvious and conventional approach was to reconstruct frame-by-frame. Exploiting Equations (4) and (5), in a least-squares sense, and sequentially at each frame t :

$$\min_{o_t, \mathbf{v}_t} \|\mathbf{A}o_t - \mathbf{b}_t\|_2^2 + \tau \|o_{t+\Delta t}(\mathbf{x}) - o_t(\mathbf{x} + \mathbf{v}_t)\|_2^2 \quad \text{for} \quad t = 1, 2, \dots, T \quad (6)$$

where $\tau > 0$ is a trade-off parameter, bold fonts are discretization from their continuous variants, and matrix \mathbf{A} denotes the linear mapping of Equation (4).

Considering the dimensionality gap from 2D to 3D, inversion of Equation (6) was highly ill-posed, and data priors (i.e., domain knowledge) were important for plausible reconstructions. For volumes of interest, only very small portions were occupied by particles, indicating a natural sparsity for particle volumes, as noticed by ref. [19] Fluid flows, on the other hand, were practically smooth and were usually incompressible,^[8,36] but the reconstruction of fluid flows depended on available particles, which were presumably sparse. This dilemma led to “null” flow reconstructions as a result of numerical ill-conditioning, and could be mitigated by the introduction of Tikhonov regularization. Given these domain-knowledge, particle sparsity (priors(\mathbf{o})), flow smoothness, and Tikhonov regularization (priors(\mathbf{v})) were added to the original reconstruction of Equation (6). Specifically:

$$\text{priors}(\mathbf{o}) = \underbrace{\mu \|\mathbf{o}\|_1}_{\text{sparsity}} \quad (7a)$$

$$\begin{aligned} \text{priors}(\mathbf{v}) = & \underbrace{\alpha_{xy} (\|\nabla \mathbf{v}_x\|_2^2 + \|\nabla \mathbf{v}_y\|_2^2) + \alpha_z \|\nabla \mathbf{v}_z\|_2^2}_{\text{smoothness prior}} \\ & + \underbrace{\beta_{xy} (\|\mathbf{v}_x\|_2^2 + \|\mathbf{v}_y\|_2^2) + \beta_z \|\mathbf{v}_z\|_2^2}_{\text{Tikhonov regularization}} \end{aligned} \quad (7b)$$

where μ , α_{xy} , α_z , β_{xy} , and β_z are trade-off parameters, and are manually tuned for specific scenes. These prior terms can handle numerical ill-conditioning, which could happen when particles were too few to fully cover the whole volume, since it is problematic to estimate flow for volumes where there were no particles present. A larger value of μ encouraged such a sparse particle situation. Similarly, larger values of $\alpha_{xy,z}/\beta_{xy,z}$ helped stabilize the flow reconstruction, but at the cost of smoothing/reducing the resulting flow magnitudes, anisotropically in x , y , and z .

Further, by taking into consideration of the temporal coherence between neighboring frames, particle volumes and fluid flows were jointly estimated and this strategy was optimized for neighboring frames together, and hence closing the numerical reconstruction loop. These yielded the following optimization problem:

$$\begin{aligned} \min_{\mathbf{o}, \mathbf{v}} \quad & \sum_{t=1}^T \|\mathbf{A}o_t - \mathbf{b}_t\|_2^2 + \tau \sum_{t=1}^{T-1} \|o_{t+\Delta t}(\mathbf{x}) - o_t(\mathbf{x} + \mathbf{v}_t)\|_2^2 \\ & + \text{priors}(\mathbf{o}) + \text{priors}(\mathbf{v}) \end{aligned} \quad (8)$$

Specifically, Equation (8) was solved by alternating between solving particle volume \mathbf{o} and solving volumetric movement flow \mathbf{v} , as in Algorithm 1:

$$\begin{aligned} \text{Solve for } \mathbf{o}: \quad & \min_{\mathbf{o}} \sum_{t=1}^T \|\mathbf{A}o_t - \mathbf{b}_t\|_2^2 \\ & + \tau \sum_{t=1}^{T-1} \|o_{t+\Delta t}(\mathbf{x}) - o_t(\mathbf{x} + \mathbf{v}_t)\|_2^2 + \text{priors}(\mathbf{o}) \end{aligned} \quad (9a)$$

$$\begin{aligned} \text{Solve for } \mathbf{v}: \quad & \min_{\mathbf{v}} \tau \sum_{t=1}^{T-1} \|o_{t+\Delta t}(\mathbf{x}) - o_t(\mathbf{x} + \mathbf{v}_t)\|_2^2 + \text{priors}(\mathbf{v}) \end{aligned} \quad (9b)$$

Specifically, particles were solved using FASTA iterative shrinkage algorithm,^[37] and flow was solved using standard Horn–Schunck optical flow.^[38] See Appendix B for more details.

Appendix A: Holographic Particle Image Velocimetry

A.1. Propagation Point Spread Function

Let $\mathbf{r} = (x, y)$. Consider in free space monochromatic scalar field $u_0(\mathbf{r})$ propagates through a short distance z perpendicular to (x, y) and becomes $u_z(\mathbf{r})$. The relationship between $u_0(\mathbf{r})$ and $u_z(\mathbf{r})$, in the spatial and frequency domain, is:^[26,39]

$$\begin{aligned} u_z(\mathbf{r}) &= h_0(\mathbf{r}, z) \otimes u_0(\mathbf{r}) \quad \text{and} \\ U_z(\boldsymbol{\rho}) &= \exp \left[jkz(1 - \lambda^2 |\boldsymbol{\rho}|^2)^{1/2} \right] U_0(\boldsymbol{\rho}) \end{aligned} \quad (A1)$$

where \otimes denotes convolution, $\boldsymbol{\rho}$ is the Fourier dual of \mathbf{r} , and $U_0(\boldsymbol{\rho})$ and $U_z(\boldsymbol{\rho})$ are Fourier transforms of $u_0(\mathbf{r})$ and $u_z(\mathbf{r})$, respectively. We follow the convention that $|\cdot|$ denotes the ℓ_2 -norm of a vector, for example, $|\boldsymbol{\rho}| = (\rho_x^2 + \rho_y^2)^{1/2}$. In practice, spatial frequency satisfies $|\boldsymbol{\rho}| \ll 1/\lambda$, and $(1 - \lambda^2 |\boldsymbol{\rho}|^2)^{1/2}$ is numerically close to 1 for any given $\lambda^2 |\boldsymbol{\rho}|^2$, making the kernel numerically hard for inversion. Hence we prefer an approximation to Equation (A1) for enhanced numerical stability. Taylor expanding Equation (A1):

$$\begin{aligned} U_z(\boldsymbol{\rho}) &= \exp(jkz) \exp \left[-jkz \left(\frac{1}{2} \lambda^2 |\boldsymbol{\rho}|^2 + \frac{1}{8} \lambda^4 |\boldsymbol{\rho}|^4 + \frac{1}{16} \lambda^6 |\boldsymbol{\rho}|^6 + \dots \right) \right] U_0(\boldsymbol{\rho}) \end{aligned} \quad (A2)$$

Algorithm 1 Holo-Flow solver for Equation (9)

function Reconstruct Volume and Flow $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_T$

Initialize \mathbf{o}^0 and \mathbf{v}^0 ;

while not converge **do**

▷ Alternating loop

$\mathbf{o}^{k+1} \leftarrow \arg \min_{\mathbf{o}} \sum_{t=1}^T \|\mathbf{A}\mathbf{o}_t - \mathbf{b}_t\|_2^2 + \tau \sum_{t=1}^{T-1} \|\mathbf{o}_{t+\Delta t}(\mathbf{x}) - \mathbf{o}_t(\mathbf{x} + \mathbf{v}_t^k)\|_2^2 + \text{priors}(\mathbf{o})$;

▷ \mathbf{o} -solver: FASTA [37]

$\mathbf{v}^{k+1} \leftarrow \arg \min_{\mathbf{v}} \tau \sum_{t=1}^{T-1} \|\mathbf{o}_{t+\Delta t}^k(\mathbf{x}) - \mathbf{o}_t^k(\mathbf{x} + \mathbf{v}_t)\|_2^2 + \text{priors}(\mathbf{v})$;

▷ \mathbf{v} -solver: 3D optical flow [38]

return \mathbf{o}^K and \mathbf{v}^K ;

▷ At final iteration K

The Fresnel diffraction formula can be derived by preserving only the first term $\frac{1}{2}\lambda^2|\rho|^2$ in Equation (A3). An expansion of two or three terms suffice in most of our situations. Excluding the $\exp(jkz)$ term, we define a new convolution kernel $h(\mathbf{r}, z)$:

$$u_z(\mathbf{r}) = h(\mathbf{r}, z) \otimes u_0(\mathbf{r}) \quad \text{and}$$

$$U_z(\rho) = \exp \left[-jkz \left(\frac{1}{2}\lambda^2|\rho|^2 + \frac{1}{8}\lambda^4|\rho|^4 + \frac{1}{16}\lambda^6|\rho|^6 \right) \right] U_0(\rho) \quad (\text{A3})$$

A.2. Inline Holography Under Plane Wave Illumination

Suppose there is a 3D object $o(\mathbf{r}, z)$ illuminated by a plane wave under the assumption of a transmissive background of unique magnitude, where we consider the object of interest $o(\mathbf{r}, z)$ to be a small differential quantity to the background, and is sparsely distributed over the volume space. The captured hologram $I_h(\mathbf{r})$ is the propagated 2D scalar field $o_h(\mathbf{r})$ at z_h ,^[26] which is a self-interference of the two fields:

$$\begin{aligned} I_h(\mathbf{r}) &= \left| \int_{z_1}^{z_2} h_0(\mathbf{r}, z) \otimes (1 + o(\mathbf{r}, z)) dz \right|^2 \\ &= |r_h(\mathbf{r}) + o_h(\mathbf{r})|^2 \\ &= 2\Re \left\{ o_h(\mathbf{r}) \overline{r_h(\mathbf{r})} \right\} + |o_h(\mathbf{r})|^2 + |r_h(\mathbf{r})|^2 \end{aligned} \quad (\text{A4})$$

where according to the properties of Fourier transform $r_h(\mathbf{r}) = \int_{z_1}^{z_2} \exp(jkz) dz$ is the reference field when there are no objects present, $\{\cdot\}$ denotes complex conjugate operator, $|r_h(\mathbf{r})|^2$ could be easily pre-calibrated before any measurements and simply assumed as 1, and $|o_h(\mathbf{r})|^2 \ll 1$. This leads to a linear model^[19]:

$$I_h(\mathbf{r}) \approx 2\Re \left\{ o_h(\mathbf{r}) \overline{r_h(\mathbf{r})} \right\} \quad (\text{A5})$$

where $o_h(\mathbf{r})$ at the sensor plane is a superposition of all 3D object fields $o(\mathbf{r}, z)$ across z . That said, with Equation (A1):

$$o_h(\mathbf{r}) \overline{r_h(\mathbf{r})} = \int_{z_1}^{z_2} [h_0(\mathbf{r}, z) \otimes o(\mathbf{r}, z)] \overline{r(\mathbf{r}, z)} dz \quad (\text{A6})$$

In frequency domain, according to Equations (A2) and (A3):

$$\begin{aligned} &\mathcal{F} \left\{ [h_0(\mathbf{r}, z) \otimes o(\mathbf{r}, z)] \overline{r(\mathbf{r}, z)} \right\} \\ &= \exp \left[-jkz \left(\frac{1}{2}\lambda^2|\rho|^2 + \frac{1}{8}\lambda^4|\rho|^4 + \frac{1}{16}\lambda^6|\rho|^6 + \dots \right) \right] \mathcal{F}\{o(\rho, z)\} \\ &= \mathcal{F}\{h(\mathbf{r}, z) \otimes o(\mathbf{r}, z)\} \end{aligned} \quad (\text{A7})$$

where \mathcal{F} denotes the Fourier transform, $h(\mathbf{r}, z)$ is the free space point spread function of distance z in spatial domain, and \otimes denotes convolution operator. With Equations (A5), (A6), and (A7), we have:

$$I_h(\mathbf{r}) \approx 2\Re \left\{ \int_{z_1}^{z_2} h(\mathbf{r}, z) \otimes o(\mathbf{r}, z) dz \right\} \quad (\text{A8})$$

With an additional assumption that $o(\mathbf{r}, z) \in \mathbb{R}$, the above is rewritten as:

$$I_h(\mathbf{r}) \approx \int_{z_1}^{z_2} 2\Re \{ h(\mathbf{r}, z) \} \otimes o(\mathbf{r}, z) dz \quad (\text{A9})$$

Here, we end up with a linear convolutional model in real domain with respect to $o(\mathbf{r}, z)$.

Appendix B: 4D Particle-flow Optimization

We now discuss further details regarding the optimization for Equations (8) and (9). We solve Equation (8) by alternating optimization, refer to Algorithm 1. All computations are highly parallelizable, and can be efficiently implemented. The algorithms were implemented in two versions: CPU and GPU version. For the CPU version, pure MATLAB code (MATLAB 2020a) is prototyped for this purpose, with the parallelization done intrinsically via vectorization operations. For the GPU version, code is refactorized in pure C++17 (compiled via Clang 9) and CUDA 10.2, with the parallelization done mostly by the implicit mechanism of CUDA and its associated acceleration libraries (CuFFT and Thrust), as well as code optimization to reduce GPU memory usage and to improve locality for speed. We believe the major acceleration difference comparing the CPU and the GPU version, is primarily because the C++17 and CUDA programming language offer a better way for fine memory/instruction control, leveraging the parallelization spirit of the proposed algorithms.

B.1. \mathbf{o} -update (Equation (9a))

We solve \mathbf{o} -update (Equation (9a)) using proximal gradient descents, or known as iterative thresholding algorithms.^[40] For such schemes to work, it requires for a splitting strategy as:

$$\begin{aligned} &\min_{\mathbf{o}} f(\mathbf{o}) + g(\mathbf{o}), \\ &\text{s.t.} \quad f(\mathbf{o}) = \sum_{t=1}^T \|\mathbf{A}\mathbf{o}_t - \mathbf{b}_t\|_2^2 + \tau \sum_{t=1}^{T-1} \|\mathbf{C}_t \mathbf{o}_t - \mathbf{o}_{t+\Delta t}\|_2^2, \\ &\quad g(\mathbf{o}) = \text{priors}(\mathbf{o}) \end{aligned} \quad (\text{B1})$$

Received: January 7, 2021

Revised: March 18, 2021

Published online:

where C_t denotes a linear mapping containing all interpolation coefficients for computing $\mathbf{o}_t(\mathbf{x} + \mathbf{v}_t)$ from $\mathbf{o}_t(\mathbf{x})$, and we neglect \mathbf{x} and \mathbf{v}_t to simplify notations. The gradient of $f(\mathbf{o})$ is, for each frame t :

$$\nabla f(\mathbf{o}_t) = \begin{cases} 2\mathbf{A}^T(\mathbf{A}\mathbf{o}_t - \mathbf{b}_t) \\ + 2\tau \mathbf{C}_t^T(\mathbf{C}_t\mathbf{o}_t - \mathbf{o}_{t+\Delta t}) & \text{for } t = 1, 2, \dots, T-1, \\ 2\mathbf{A}^T(\mathbf{A}\mathbf{o}_t - \mathbf{b}_t) & \text{for } t = T \end{cases} \quad (\text{B2})$$

Computation for \mathbf{C}^T is not easy. For efficiency, we approximate \mathbf{C}^T as \mathbf{C}^{-1} , and the matrix-vector multiplication becomes the interpolation by inverse \mathbf{v}_t . This approximation is known as the forward-backward warping^[41,42]:

$$\mathbf{C}_t\mathbf{o}_t = \mathbf{o}_t(\mathbf{x} + \mathbf{v}_t) \quad \text{and} \quad \mathbf{C}_t^T\mathbf{o}_t \approx \mathbf{C}_t^{-1}\mathbf{o}_t = \mathbf{o}_t(\mathbf{x} - \mathbf{v}_t) \quad (\text{B3})$$

B.2. \mathbf{v} -update (Equation (9b))

Notice \mathbf{v} -update (Equation (9b)) is a variant of the classical Horn-Schunck optical flow^[38] formulation. To cope with multi-resolution and keep robust, using a pyramid and warping scheme,^[43] we linearize the original problem to linear least squares, followed by median filtering.^[44] The linear least squares problem is the following, with $\nabla\mathbf{o}_t$ (spatial) and $\nabla_t\mathbf{o}_t$ (temporal) gradients of particle volumes \mathbf{o}_t at time t being computed from the last \mathbf{o} -update:

$$\min_{\mathbf{v}} \tau \sum_{t=1}^{T-1} \|\nabla\mathbf{v}_t \cdot \nabla\mathbf{o}_t + \nabla_t\mathbf{o}_t\|_2^2 + \alpha_{xy} (\|\nabla\mathbf{v}_x\|_2^2 + \|\nabla\mathbf{v}_y\|_2^2) \\ + \alpha_z \|\nabla\mathbf{v}_z\|_2^2 + \beta_{xy} (\|\mathbf{v}_x\|_2^2 + \|\mathbf{v}_y\|_2^2) + \beta_z \|\mathbf{v}_z\|_2^2 \quad (\text{B4})$$

The associate linear system is readily solved by conjugate gradients, and typically converged within hundreds of iterations at sufficiently small residual errors.

Acknowledgements

N.C. and C.W. contributed equally to this work. The authors thank Jinhui Xiong and Guangming Zang for constructive discussions, Prof. Sigurdur Thoroddsen and Ziqiang Yang from High-Speed Fluids Imaging Laboratory at King Abdullah University of Science and Technology for preparing the particles, and design the flow experiments. This work was supported by the KAUST individual baseline funding.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in KAUST repository at <http://doi.org/10.25781/KAUST-724NT>, reference number [31].

Keywords

holography, optical imaging, optofluidics

- [1] D. Gabor, *Nature* **1948**, 161, 777.
- [2] E. N. Leith, J. Upatnieks, *J. Opt. Soc. Am. A* **1962**, 52, 1123.
- [3] I. Yamaguchi, T. Zhang, *Optics Letters* **1997**, 22, 1268.
- [4] T.-C. Poon, M. H. Wu, K. Shinoda, Y. Suzuki, *Proc. IEEE* **1996**, 84, 753.
- [5] T. Hori, J. Sakakibara, *Measurement Sci. Technol.* **2004**, 15, 1067.
- [6] C. E. Willert, M. Gharib, *Exp. Fluids* **1992**, 12, 353.
- [7] L. Schäfer, W. Schröder, *J. Phys.: Conf. Ser.* **2011**, 318, 022019.
- [8] J. Xiong, R. Idoughi, A. A. Aguirre-Pablo, A. B. Aljedaani, X. Dun, Q. Fu, S. T. Thoroddsen, W. Heidrich, *ACM Trans. Graphics* **2017**, 36, 1.
- [9] A. Aguirre-Pablo, A. B. Aljedaani, J. Xiong, R. Idoughi, W. Heidrich, S. T. Thoroddsen, *Exp. Fluids* **2019**, 60, 25.
- [10] J. D. Trolinger, R. A. Belz, W. M. Farmer, *Appl. Opt.* **1969**, 8, 957.
- [11] U. Schnars, W. Jueptner, *Digital Holography*, Springer-Verlag, Berlin, Heidelberg **2005**.
- [12] H. Meng, F. Hussain, *Fluid Dyn. Res.* **1991**, 8, 33.
- [13] D. H. Barnhart, R. J. Adrian, G. C. Papen, *Appl. Opt.* **1994**, 33, 7159.
- [14] J. Sheng, E. Malkiel, J. Katz, *Exp. Fluids* **2008**, 45, 1023.
- [15] J. Katz, J. Sheng, *Annu. Rev. Fluid Mech.* **2010**, 42, 531.
- [16] S. Talapatra, J. Katz, *J. Fluid Mech.* **2012**, 711, 161.
- [17] W. Yingchun, W. Xuecheng, Y. Jing, W. Zhifeng, G. Xiang, Z. Binwu, C. Linghong, Q. Kunzan, G. Gréhan, C. Kefa, *Appl. Opt.* **2014**, 53, 556.
- [18] T. Latychevskaia, H.-W. Fink, *Opt. Express* **2014**, 22, 20994.
- [19] K. Mallery, J. Hong, *Opt. Express* **2019**, 27, 18069.
- [20] S. Yoshida, K. Itakura, *Opt. Rev.* **2020**, 27, 441.
- [21] B. Sun, A. Ahmed, C. Atkinson, J. Soria, *Meas. Sci. Technol.* **2020**, 31, 104002.
- [22] T. Shimobaba, T. Takahashi, Y. Yamamoto, Y. Endo, A. Shiraki, T. Nishitsuji, N. Hoshikawa, T. Kakue, T. Ito, *Appl. Opt.* **2019**, 58, 1900.
- [23] S. Shao, K. Mallery, S. S. Kumar, J. Hong, *Opt. Express* **2020**, 28, 2987.
- [24] N. Chen, C. Wang, W. Heidrich, *IEEE Trans. Computat. Imaging* **2021**, 7, 288.
- [25] M. Toloui, J. Hong, *Opt. Express* **2015**, 23, 27159.
- [26] J. W. Goodman, *Introduction to Fourier Optics*, Roberts and Company Publishers, Greenwood Village, CO **2006**.
- [27] J. L. Barron, D. J. Fleet, S. S. Beauchemin, *Int. J. Comput. Vision* **1994**, 12, 43.
- [28] M. Otte, H. H. Nagel, In *European Conference on Computer Vision*, Springer, Berlin, Heidelberg **1994**, pp. 49–60.
- [29] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, R. Szeliski, in *2007 IEEE 11th Int. Conf. on Computer Vision*, IEEE, Piscataway, NJ **2007**, pp. 1–8.
- [30] Forced isotropic turbulence dataset (extended), http://turbulence.pha.jhu.edu/Forced_isotropic_turbulence.aspx (accessed: July, 2020), **2020**.
- [31] N. Chen, C. Wang, W. Heidrich, Snapshot space-time holographic three-dimensional particle tracking velocimetry: video, **2021**, <http://hdl.handle.net/10754/668685>.
- [32] T. Latychevskaia, H.-W. Fink, *Appl. Optics* **2015**, 54, 2424.
- [33] F. Merola, P. Memmolo, L. Miccio, R. Savoia, M. Mugnano, A. Fontana, G. D'Ippolito, A. Sardo, A. Iolascon, A. Gambale, P. Ferraro, *Light: Sci. Appl.* **2016**, 6, e16241.
- [34] F. Pfeiffer, *Nat. Photonics* **2018**, 12, 9.
- [35] G. Zheng, R. Horstmeyer, C. Yang, *Nat. Photonics* **2013**, 7, 739.
- [36] J. Gregson, I. Ihrke, N. Thuerey, W. Heidrich, *ACM Trans. Graphics* **2014**, 33, 1.
- [37] T. Goldstein, C. Studer, R. Baraniuk, *arXiv:1411.3406*, **2014**.

- [38] B. K. Horn, B. G. Schunck, In *Techniques and Applications of Image Understanding*, Vol. 281, International Society for Optics and Photonics, Bellingham, WA **1981**, pp. 319–331.
- [39] M. R. Teague, *J. Opt. Soc. Am. A* **1983**, 73, 1434.
- [40] A. Beck, M. Teboulle, *SIAM J. Imaging Sci.* **2009**, 2, 183.
- [41] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, W. Heidrich, *ACM Trans. Graphics* **2019**, 38, 100.
- [42] M. Odstrcil, M. Holler, J. Raabe, A. Sepe, X. Sheng, S. Vignolini, C. G. Schroer, M. Guizar-Sicairos, *Nat. Commun.* **2019**, 10, 2600.
- [43] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, In, *European Conf. on Computer Vision*, Springer, Berlin, Heidelberg **2004**, pp. 25–36.
- [44] D. Sun, S. Roth, M. J. Black, in, *2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2010**, pp. 2432–2439.