

Stereo Event-based Particle Tracking Velocimetry for 3D Fluid Flow Reconstruction (Supplemental Material)

Yuanhao Wang, Ramzi Idoughi, and Wolfgang Heidrich

King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
{yuanhao.wang, ramzi.idoughi, wolfgang.heidrich}@kaust.edu.sa

1 Implementation details

We illustrate in the following a pseudo code of our proposed framework.

Algorithm 1 Framework of the Stereo Event-based PIV

```

1: procedure STEREO_EVENT-BASED_PIV( $\{e_i\}_{i=1..N}^1, \{e_i\}_{i=1..N}^2, \mathbf{M}_1, \mathbf{M}_2, \rho$ )
2:   // Step 1: Event-based particle tracking using the algorithm presented in [8,9].
3:    $\mathbf{v}_1, \mathbf{v}_2 \leftarrow \text{PARTICLETRAKING}(\{e_i\}_{i=1..N}^1, \{e_i\}_{i=1..N}^2)$ 
4:
5:   // Step 2: Stereo matching performed using a triangulation procedure ([1,3]).
6:   // The output is a list of matched particles' center ( $\{X_c(P_j), Y_c(P_j), Z_c(P_j)\}_j$ ).
7:    $\{X_c(P_j), Y_c(P_j), Z_c(P_j)\}_j \leftarrow \text{STEREOMATCHING}(\{e_i\}_{i=1..N}^1, \{e_i\}_{i=1..N}^2, \mathbf{M}_1, \mathbf{M}_2)$ 
8:
9:   // Step 3: 3D velocimetry reconstruction.
10:  // Initialization
11:   $\mathbf{u} \leftarrow \mathbf{0}, \mathbf{p} \leftarrow \text{project}(P_j, \mathbf{M}_1) \odot \text{project}(P_j, \mathbf{M}_2) \quad \forall j$ 
12:  // Estimation of  $\mathbf{u}$  using a multi-scale strategy.
13:  // Generate the multi-scale data from the coarsest to the finest.
14:   $\mathbf{u}^1 \leftarrow \mathbf{u}, \mathbf{p}^1 \leftarrow \mathbf{p}$ 
15:  for  $s$  from 1 to  $Nb_{scales} - 1$  do
16:     $\mathbf{u}^{s+1} \leftarrow \rho \downarrow \mathbf{u}^s$ 
17:     $\mathbf{p}^{s+1} \leftarrow \downarrow \mathbf{p}^s$ 
18:  end for
19:  for  $s$  from  $Nb_{scales}$  to 1 do
20:     $\mathbf{u}^s \leftarrow \text{ESTIMATEVELOCITYFIELD}(\mathbf{u}^s)$ 
21:     $\mathbf{u}^{s-1} \leftarrow \frac{1}{\rho} \uparrow \mathbf{u}^s$ 
22:  end for
23:   $\mathbf{u} \leftarrow \mathbf{u}^1$ 
24:
25:  return  $\mathbf{u}$ 
26: end procedure

```

In the line 20 of the algorithm 1 ($\mathbf{u}^s \leftarrow \text{ESTIMATEVELOCITYFIELD}(\mathbf{u}^s)$), at each scale we estimate the velocity \mathbf{u}^s using the Conjugate Gradient method.

For a given iteration k , we update the velocity as follows: $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}^k$. The update $\Delta\mathbf{u}^k$ is obtained by solving the following linear system of the form $A \times \Delta\mathbf{u}_t^k = b$, for each time step t :

$$\begin{aligned} & (\mathbf{p}^2 \odot \mathbf{M}_1^T \mathbf{M}_1 + \mathbf{p}^2 \odot \mathbf{M}_2^T \mathbf{M}_2 + \lambda_1 \nabla_S^2 + 2\lambda_2 \mathbf{p}^2 \odot) \Delta\mathbf{u}_t^k + \lambda_3 \text{div}^T(\text{div}(\Delta\mathbf{u}_t^k)) = \\ & - \mathbf{p} \odot \mathbf{M}_1^T (\mathbf{p} \odot (\mathbf{M}_1 \mathbf{u}_t^k) - \mathbf{v}_1) - \mathbf{p} \odot \mathbf{M}_2^T (\mathbf{p} \odot (\mathbf{M}_2 \mathbf{u}_t^k) - \mathbf{v}_2) \\ & - \lambda_1 \nabla_S^2 \mathbf{u}_t^k - \lambda_2 \mathbf{p}^2 \odot (\mathbf{u}_t^k - \text{advect}(\mathbf{u}_{t-1}^k, \Delta t)) - \lambda_2 \mathbf{p}^2 \odot (\mathbf{u}_t^k - \text{advect}(\mathbf{u}_{t+1}^k, -\Delta t)) \\ & - \lambda_3 \text{div}^T(\text{div}(\mathbf{u}_t^k)) \end{aligned} \quad (1)$$

2 Stereo calibration

To implement the calibration, we attached an 17×16 checkerboard, where each square has an edge length of 0.5 mm , on a glass slide. By using the pinhole model proposed by [7], we could use a controllable translation stage to control the slide moving perpendicularly to its plane, 0 to 10 mm with a step of 0.5 mm in our experiments, as illustrated in Figure 1. We generated 42 gray-scale images for two cameras through the exposure measurement (EM) events in the camera [5]. Examples are shown in Figure 2.

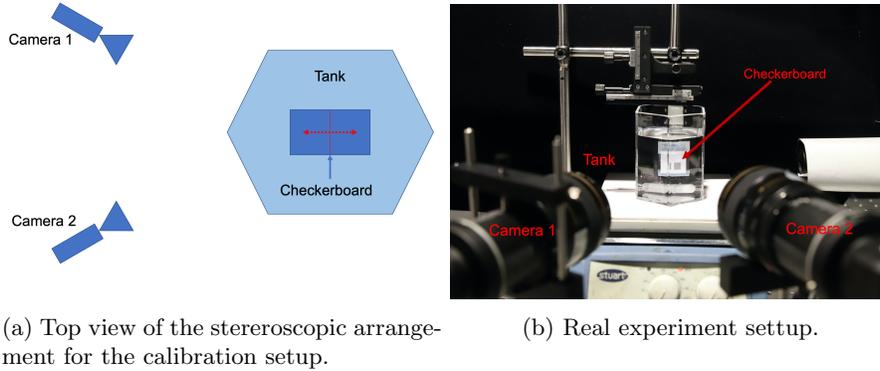
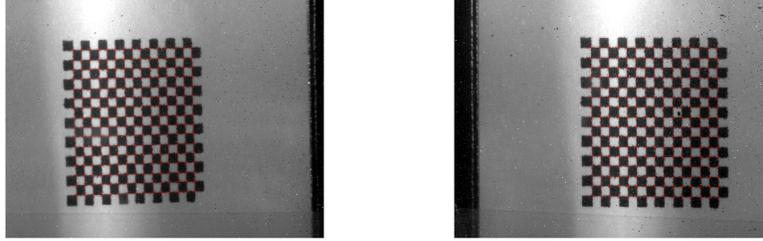


Fig. 1: Calibration Setup.

Each time when conducting the experiment, we also implemented the calibration for the two cameras. One calibration result is as follows.

$$\mathbf{M}_1 = \begin{bmatrix} 25.7972 & -0.5223 & 0.0011 \\ 0.6822 & 28.9551 & -0.0002 \\ 12.6402 & -1.3181 & -0.0035 \\ 95.7501 & 78.4601 & 1.0000 \end{bmatrix} \quad \mathbf{M}_2 = \begin{bmatrix} 26.6957 & -0.0081 & -0.0011 \\ 0.2530 & 29.6012 & -0.0002 \\ -13.5633 & 0.2220 & -0.0034 \\ 213.0987 & 73.4363 & 1.0000 \end{bmatrix} \quad (2)$$



(a) Camera 1

(b) Camera 2

Fig. 2: Gray-scale images with detected corners.

3 Additional experiments

3.1 Additional synthetic experiments

Soft Constraint or Hard Constraint? To verify the efficiency of the soft constraint used in this paper, we conducted comparisons among the four different methods: our method without the use of the temporal coherence and the divergence terms (**w/o \mathbf{E}_{TC} & \mathbf{E}_{div}**), our method without the divergence term (**w/o \mathbf{E}_{div}**), hard constraint (**Hard Constraint**) and our proposed method (**Ours**). The result is shown in Figure 3. It seems that the hard constraint did not working better with the increase of the voxel numbers. And also the increase of the volume size will also induce the increase of computation. A 3D velocimetry vector in double for $320 \times 320 \times 320$ voxels will take around 6.4 GB memory, and this also means more computational time. In this paper, we utilized the soft constraint.

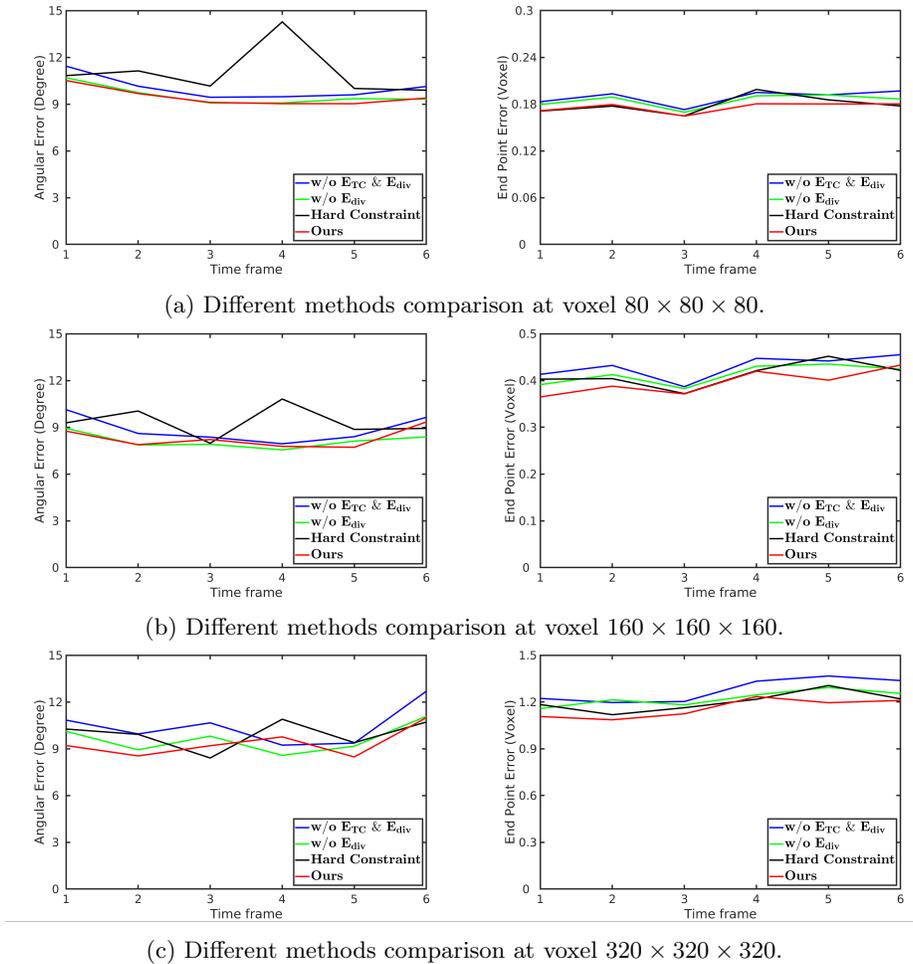
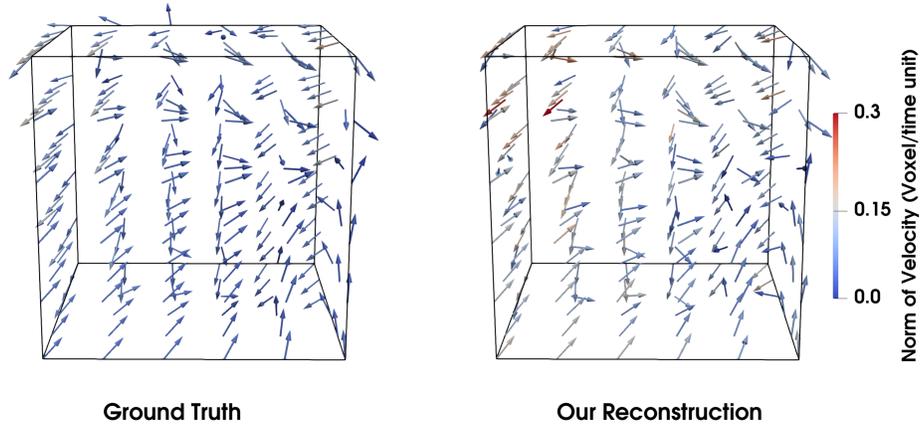


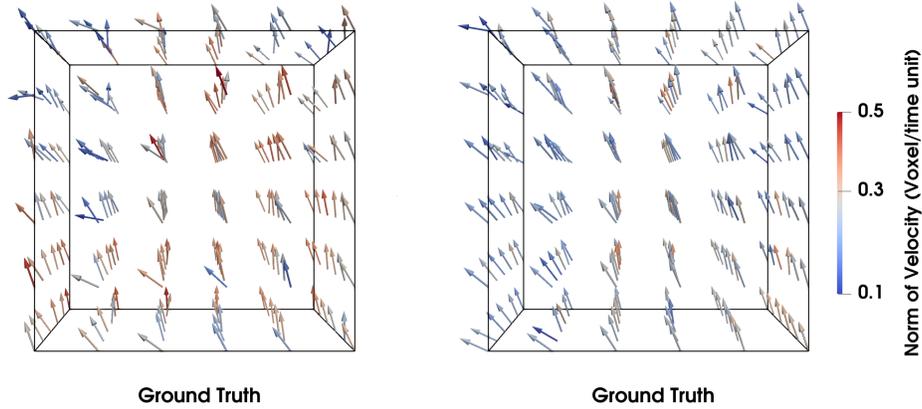
Fig. 3: Comparison of different methods in different voxels. (EPE errors increase with precision of the voxels.)

3.2 Experiments on Johns Hopkins Turbulence Database

We also conducted some experiments using the Johns Hopkins Turbulence Databases (JHTDB) [4,2]. We use two different turbulence datasets (Homogeneous buoyancy driven turbulence and Forced isotropic turbulence datasets) to generate our data, where random particles are generated in a volume ($80 \times 80 \times 80$ voxels), with a density of 0.001 particles/voxels. These particles are then advected using the velocity fields of the turbulence datasets. Finally, two rendered stereo sequences are generated and the corresponding events are obtained by applying E-sim [6]. The reconstruction results are shown in Figure 4.



(a) Ground truth (left) and reconstruction result using our method (right) for the Johns Hopkins Homogeneous buoyancy driven turbulence dataset.



(b) Ground truth (left) and reconstruction result using our method (right) for the Johns Hopkins Forced isotropic turbulence dataset.

Fig. 4: Our reconstruction results using Johns Hopkins Turbulence Datasets

In the figures 5 and 6, we can see that the end point error is better for the Homogeneous buoyancy driven turbulence dataset in comparison to the Forced isotropic turbulence dataset. Indeed, in our framework we assume that there is no external forces that apply on the fluid. Moreover, we can notice that the angular error is large in the boundary of the turbulence where neighboring particles will have opposite directions. This could be improved by increasing the resolution of the volume.

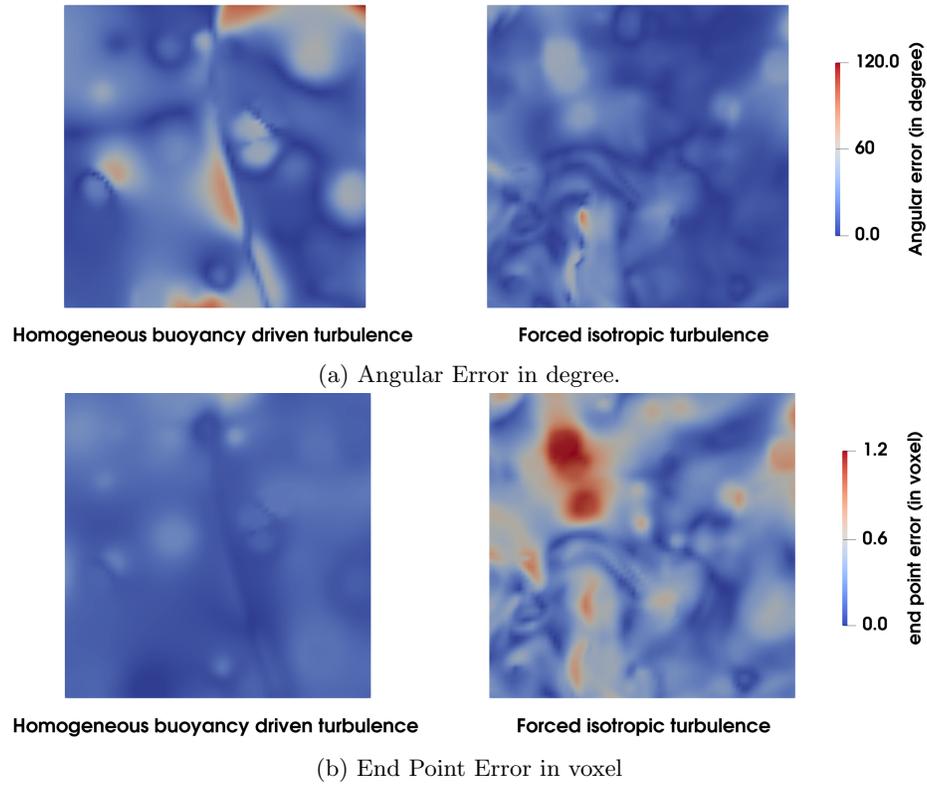
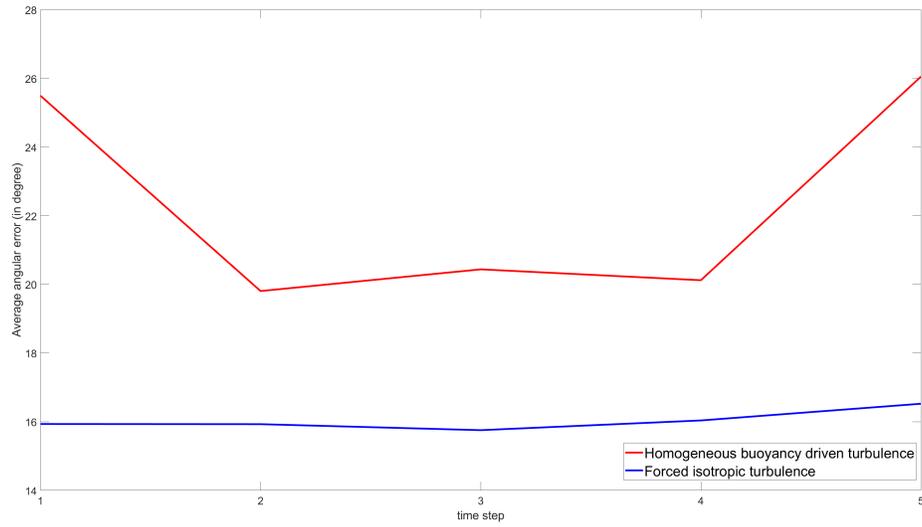


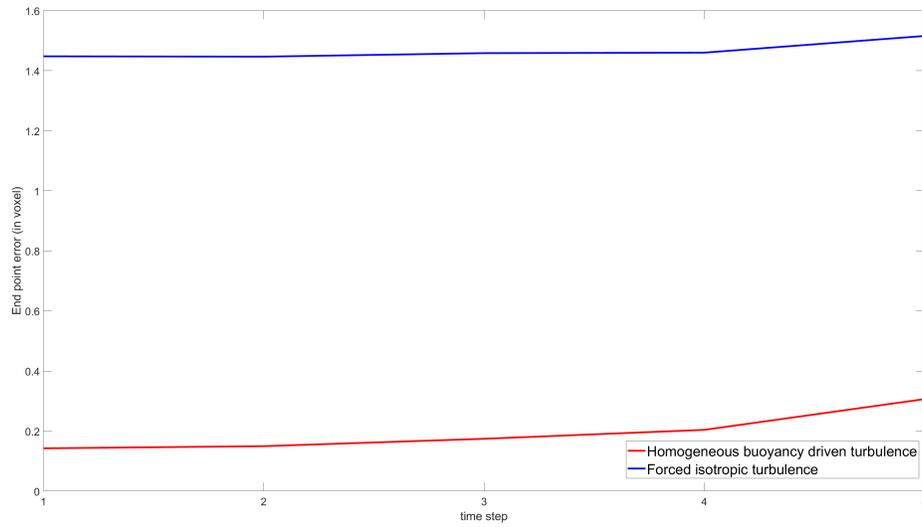
Fig. 5: Error estimation of our reconstruction of the Johns Hopkins Datasets

3.3 Additional captured data

We present two more experiment results in Figure 7. One is stirring stopping from 2, and the other is another fast injection. One is from side view and the other is from top view.



(a) Average Angular Error in degree.



(b) End Point Error in voxel

Fig. 6: Average error estimation of our reconstruction of the Johns Hopkins Datasets for the different time steps.

References

1. Knutsen, A.N., Lawson, J.M., Dawson, J.R., Worth, N.A.: A laser sheet self-calibration method for scanning PIV. *Experiments in Fluids* **58**(10), 145 (2017)
2. Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., Chen, S., Szalay, A., Eyink, G.: A public turbulence database cluster and applications to study la-

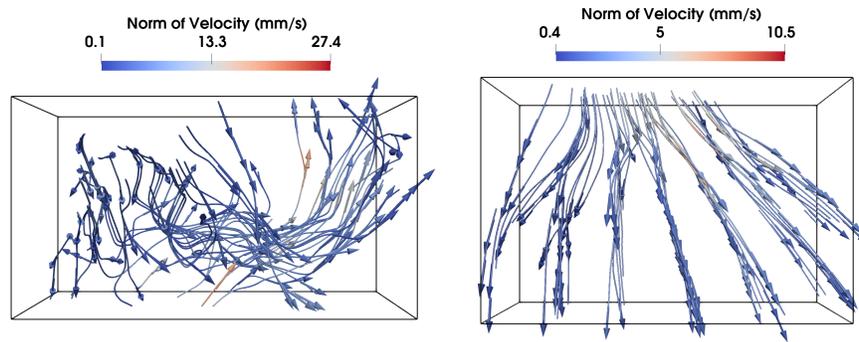


Fig. 7: Two additional experiment results. Left: Streamline visualization for vortex with the stirrer slowing down from 2 to 0 (Side view). Right: Streamline visualization for another fast injection using a syringe (Top View).

- grangian evolution of velocity increments in turbulence. *Journal of Turbulence* (9), N31 (2008)
3. Machicoane, N., Aliseda, A., Volk, R., Bourgoïn, M.: A simplified and versatile calibration method for multi-camera optical systems in 3D particle imaging. *Review of Scientific Instruments* **90**(3), 035112 (2019)
 4. Perlman, E., Burns, R., Li, Y., Meneveau, C.: Data exploration of turbulence simulations using a database cluster. In: *Proc. of the 2007 ACM/IEEE conference on Supercomputing*. pp. 1–11 (2007)
 5. Posch, C., Matolin, D., Wohlgenannt, R.: A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits* **46**(1), 259–275 (2010)
 6. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an open event camera simulator. In: *Conference on Robot Learning*. pp. 969–982 (2018)
 7. Tsai, R.Y.: An efficient and accurate camera calibration technique for 3d machine vision pp. 364–374 (1986)
 8. Zhu, A.Z., Atanasov, N., Daniilidis, K.: Event-based feature tracking with probabilistic data association. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4465–4470. IEEE (2017)
 9. Zihao Zhu, A., Atanasov, N., Daniilidis, K.: Event-based visual inertial odometry. In: *Proc. CVPR*. pp. 5391–5399 (2017)