Probabilistic high-density Particle-Tracking-Velocimetry of turbulent flow structures in a cubic Rayleigh-Bénard convection cell.

R. Barta[1,2,*], D. Schiepel[1], S. Herzog[3], C. Wagner[1,2]

1: Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Germany
2: Institute of Thermodynamics and Fluid Mechanics, Technische Universität Ilmenau, Germany
3: Department for Computational Neuroscience, University of Göttingen, Germany
*Corresponding author: robin.barta@dlr.de

Keywords: PTV processing, Rayleigh Bénard convection, Turbulence

ABSTRACT

We apply a new probabilistic python-based high-density Particle-Tracking-Velocimetry (pyHDPTV) method and the commercial *Shake-The-Box* (STB) algorithm from LaVision (DaVis v10.2.0) to measured data of a turbulent Rayleigh-Bénard (RB) convection in a cubic cell. The measurement was performed in water (Prandtl number $\mathrm{Pr} \approx 7$) with a seeding particle densities of $\rho \approx 0.04$ ppp at the hard turbulence regime with Rayleigh number $\mathrm{Ra} \approx 1 \cdot 10^{10}$. The comparison of the two PTV approaches reveals that the probabilistic pyHDPTV method leads to similar qualitative flow structures as with the state-of-the-art STB method. However, the pyHDPTV framework leads to better quantitative results such as longer track lengths, more active tracks and a higher track density over the whole measurement volume.

1. Introduction

In the last years Particle-Tracking-Velocimetry (PTV) (Dabiri & Pecora, 2020) became one of the most popular optical, non-invasive flow measurement techniques. PTV reconstructs the 3D path of individual seeding particles in a Lagrangian frame. Therefore, even complicated turbulent flow structures occurring on large as well as on small scales can be observed without interfering the flow. To perform PTV one needs a set of images per camera recoding the scattered light from in the flow seeded particles, which are neutral with respect to buoyancy. Generally, at least 2 cameras are needed with different viewing angles into the measurement volume. However, for best results 4-5 cameras are recommended (Maas et al., 1993). Next, every image should be processed to reduce noise and interfering influences that occur during the recording. The enhanced signal-to-noise ratio on the images makes it possible to identify the center position of the seeding particles in image coordinates. Generally, any PTV algorithm relies on the processing of three successive steps per set of camera images: triangulation of 3D particles from the particle center positions per image, generation of initial tracks by linking the triangulated 3D particles with the 3D particles from the last frame and predicting and extending the tracks into the next frame. The most prominent PTV framework is the commercial *Shake-The-Box* (STB) algorithm by LaVision (Schanz et al., 2016; Novara et al., 2016). Besides of the state-of-the-art code by Lavision other promising algorithms have been developed, e.g. the kernelized Lagrangian particle tracking algorithm based on a machine learning technique (Yang & Heitz, 2021), an open-source OpenLPT software (Tan et al., 2020), an improved iterative particle reconstruction algorithm (Jahn et al., 2021; Wieneke, 2013), the non-iterative double frame algorithm (Fuchs et al., 2017) and the Gaussian mixture model (GMM) based PTV method by Herzog et al. (2021).

In this paper we present the new pyHDPTV method which is an advanced python-based version of the above mentioned GMM based HDPTV method (Herzog et al., 2021). It is designed for generating long and stable particle tracks and uses a GMM to probabilistic predict the next particle positions during the tracking procedure. The accuracy of the prediction at high particle densities is improved by taking nearby particle trajectories and their histories into account. To demonstrate its performance both pyHDPTV and the commercial STB (DaVis v10.2.0) method are applied to experimentally measured data of a turbulent RB convection in a cubic cell (Schiepel et al., 2013). Do to its geometrically simple configuration, i.e. a cylindrical or rectangular shaped container, which is cooled from above and heated from below, RB convection is a common application to analyse turbulent thermal convection (Lappa, 2009). In a cubic RB cell one observes a meta-stable large scale circulation (LSC) (Brown & Ahlers, 2006; Wei, 2021) along one of the diagonals of the cell, rotating clock wise or counter clock wise (Vasiliev et al., 2016; Soucasse et al., 2019). The LSC arises from the interplay of warm fluid raising up from the heating plate and cold fluid descending from the cooling plate.

2. Experimental Data

2.1. Set-Up

A photograph of a cubic RB cell is shown in Fig. 1 as an example for the experiment. The cell used for the experiment equals the set-up used by Schiepel et al. (2013). It has an interior volume L^3, where $L = 500\,\mathrm{mm}$ is the side length of the cell. The side walls of the RB cell are made of OptiWhite glass with a strength of $8\,\mathrm{mm}$ and are fixed on a faint black anodised aluminium heating plate. The heating plate, highlighted with a red frame in Fig 1, is equipped with a heating mat. The water-cooled black anodised aluminium cooling plate is placed on top of the cell. It is highlighted with a blue frame in Fig 1. Both, the temperature of the heating and cooling plate are measured by 16 equidistant spread PT100 sensors implemented in the plates with the distance of $1\,\mathrm{mm}$ to the cell interior. To control the temperature of the heating plate to a set point $T_h = 27\,^{\circ}\mathrm{C}$, a Manson power supply is applied to the heating mat. The voltage output is controlled by a PID controller which is feed with the real time PT100 sensors temperature. Furthermore, the heating plate stands on a levelling plate for a horizontal alignment of the cell.

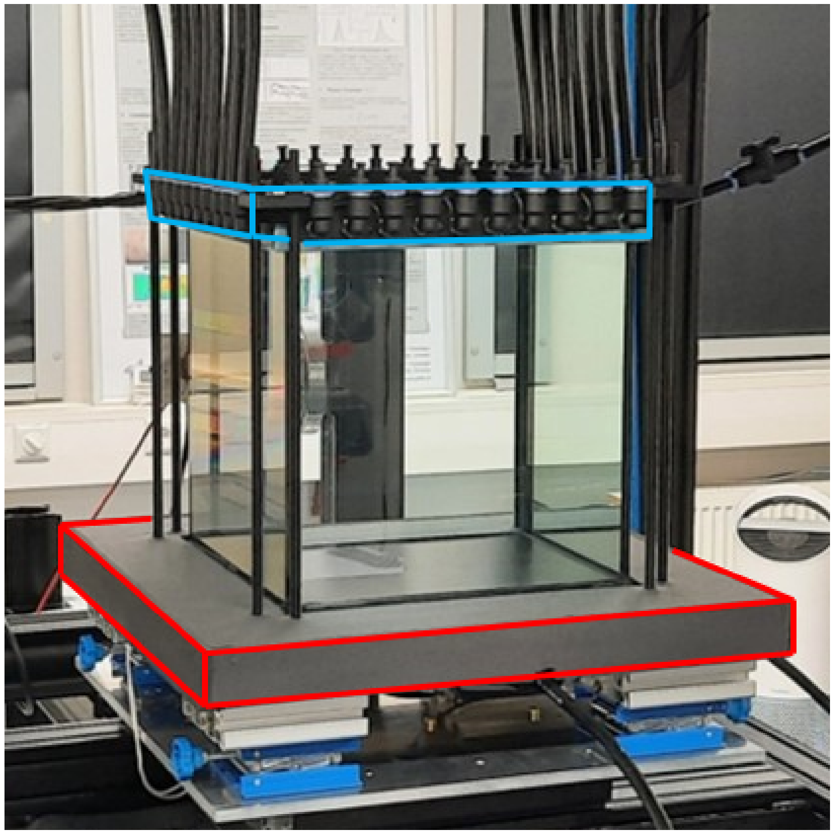

Figure 1. Photograph of a cubic RB cell. The heating and cooling plates are highlighted with red and blue frames, respectively.

Around the cooling plate there are in- and outlets for cooling water attached. The cooling water coming from a cooling water circuit is controlled by a Lauda with the set point $T_c = 21\,°C$. Due to the high thermal conductivity of the aluminium plates of $237\,W/(m\,K)$ at $25\,°C$ a homogeneous temperature distribution can be realized. The cooling plate is removable to allow attaching a calibration plate to a high precision traverse located above the cell so that it can be moved through the cell volume. In order to reduce the heat flux through the side walls, the room temperature is set to the mean RB cell temperature $\overline{T} = (T_h + T_c)/2$ using an environmental control system.
In the middle of the heating plate a small hole of $2\,mm$ diameter is placed to fill the RB cell with water. The water outlet of the RB cell is placed at the top of on one of the side walls and is connected via an aquarium pump with the inlet hole. This water circuit can be closed near the inlet and outlet during the experiment. Furthermore, the interior water circuit and the pump are used to distribute the seeding particles in the whole volume. For the measurement, high-quality polyamide particles from LaVision with a mean diameter of $60\,\mu m$ and a density of $\rho = 1.001\,\rho_{H_2O}$ are used to ensure buoyancy neutral seeding particles.
The cells interior is illuminated through one of the side walls by high power LED-arrays. The scattered light from the seeding particles is observed by four black and white PCO Pixelfly CCD cameras with a resolution of 1392×1024 pixels. The four cameras look inside the measurement volume through the same side wall which is in right angle to the LED position. Each camera is located at one of the corners of this side wall looking towards the cells center. During the experiment

black walls are placed around the set-up to avoid induced reflections due to scattered light.

2.2. Recording

During measurement the cell is filled with water to reach a Prandtl number $\mathrm{Pr} \approx 7$. Due to the temperature difference of the heating and cooling plate of $\Delta T = T_h - T_c = 6\,\mathrm{K}$ a RB convection in the hard turbulence regime at Rayleigh number $\mathrm{Ra} \approx 1 \cdot 10^{10}$ is realized.
To demonstrate the performance of pyHDPTV and STB, 100 frames are recorded at a recording frequency of $f = 6.6\,\mathrm{Hz}$. From the images one can estimate a seeding particle density of $0.04\,\mathrm{ppp}$. In Fig 2 the maximum image of all 100 frames of camera 1 after image processing is shown. The image processing procedure contains 4 steps: masking, subtraction of a minimum image, thresholding and sharpening. To ensure equal conditions, both PTV frameworks are applied on the same processed image samples.

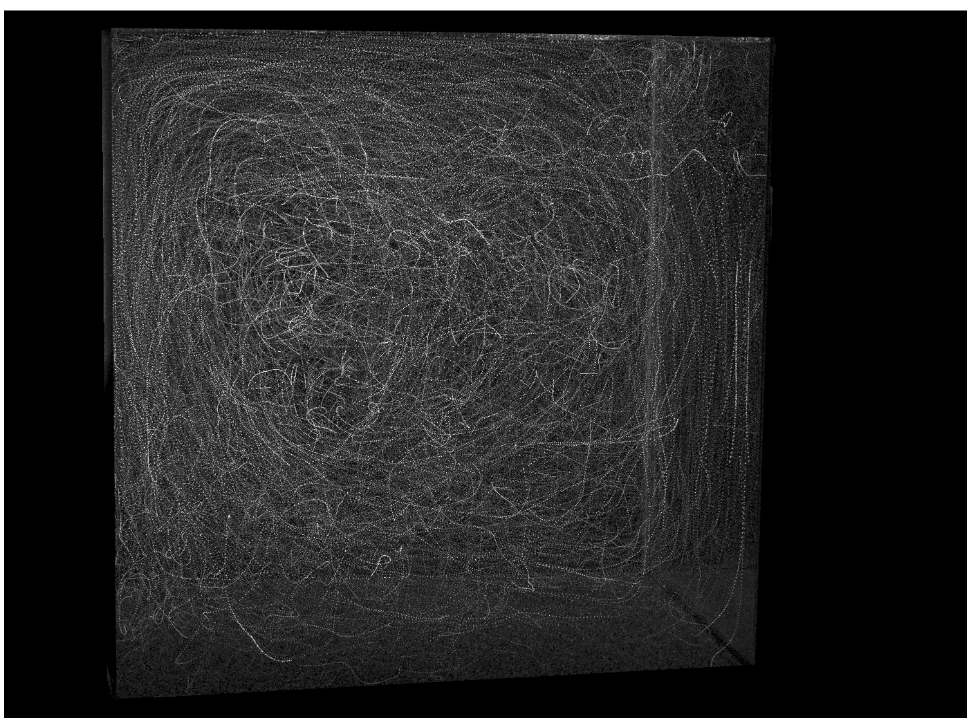

Figure 2. Maximum image of all 100 recorded frames of camera 1 after image processing.

2.3. Calibration

Before applying either PTV methode on the images a Volume-Self-Calibration (VSC) is mandatory, see Wieneke (2008). It must be noted that the pyHDPTV and STB method rely on different VSC approaches.
The commercial STB Software (DaVis v10.2.0) uses its own VSC which is only compatible with a pinhole model or a polynomial fit calibration model. For the here presented comparison the polynomial fit model of STB is used which is processed as follows. By loading the calibration plate

images in the STB software a polynomial fit between world coordinates and camera coordinates is performed for every calibration plane. The VSC of the STB method interpolates the polynomial between these planes and refines the polyomial based on the particle images of all 100 time steps. Thereby, an average disparity of 0.03 voxel and a maximum disparity of 0.1 voxel is reached.
In contrast, the here presented pyHDPTV framework uses the Soloff polynomial as calibration model as detailed in Herzog et al. (2021). The Soloff model is a polynomial that computes a 2D camera coordinate for a given 3D world coordinate. It is defined globally over the whole measurement volume and initially guessed from the calibration plate images. Then, the Soloff polynomial is optimized based on the particle images over all 100 time steps.

3. PTV Results

The processing of the particle images with both PTV methods relies on several control parameters. Besides setting the minimal allowed track length to 5, automatic predefined control parameters are used for processing both PTV methods. During processing with pyHDPTV the backtracking algorithm, where tracks are extended also backwards in time over all time steps, is not used because STB (DaVis v10.2.0) does not offer this option. Backtracking would lead to finding even longer tracks. After the processing with PTV methods often post-processing methods are used like track repairing, where suitable broken tracks are fitted together. This option is not used either because STB (DaVis v10.2.0) could not repair a single track.

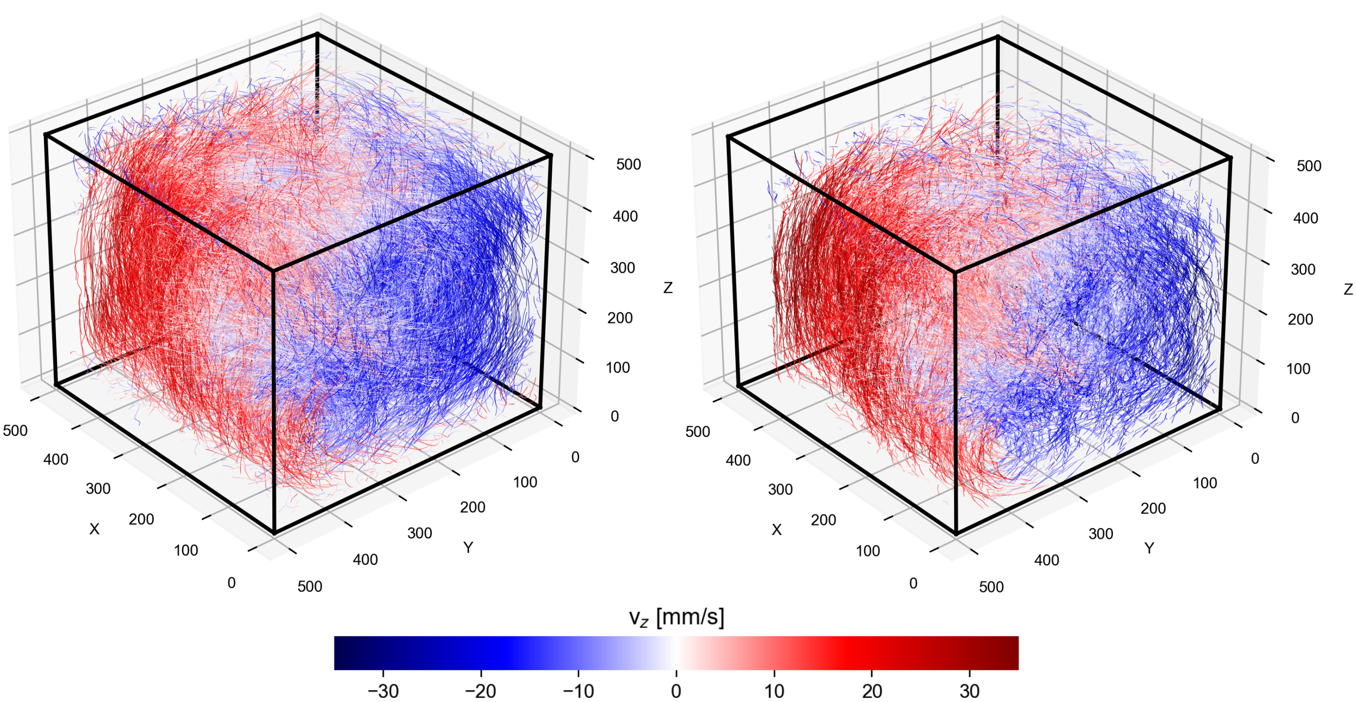

Figure 3. All reconstructed tracks color coded by the vertical velocity component v_z, **left** pyHDPTV and **right** STB.

In Fig 3 the tracks computed with both PTV methods are visualized. Both plots reflect a RB-like flow which is similar. In either 3D plot the LSC is located along the $X = Y$ diagonal and a corner vortex at $X = 0, Y = 500, Z = 0$ is visible. However, the STB method reconstructs less trajectories near the two left side walls and the cooling plate.

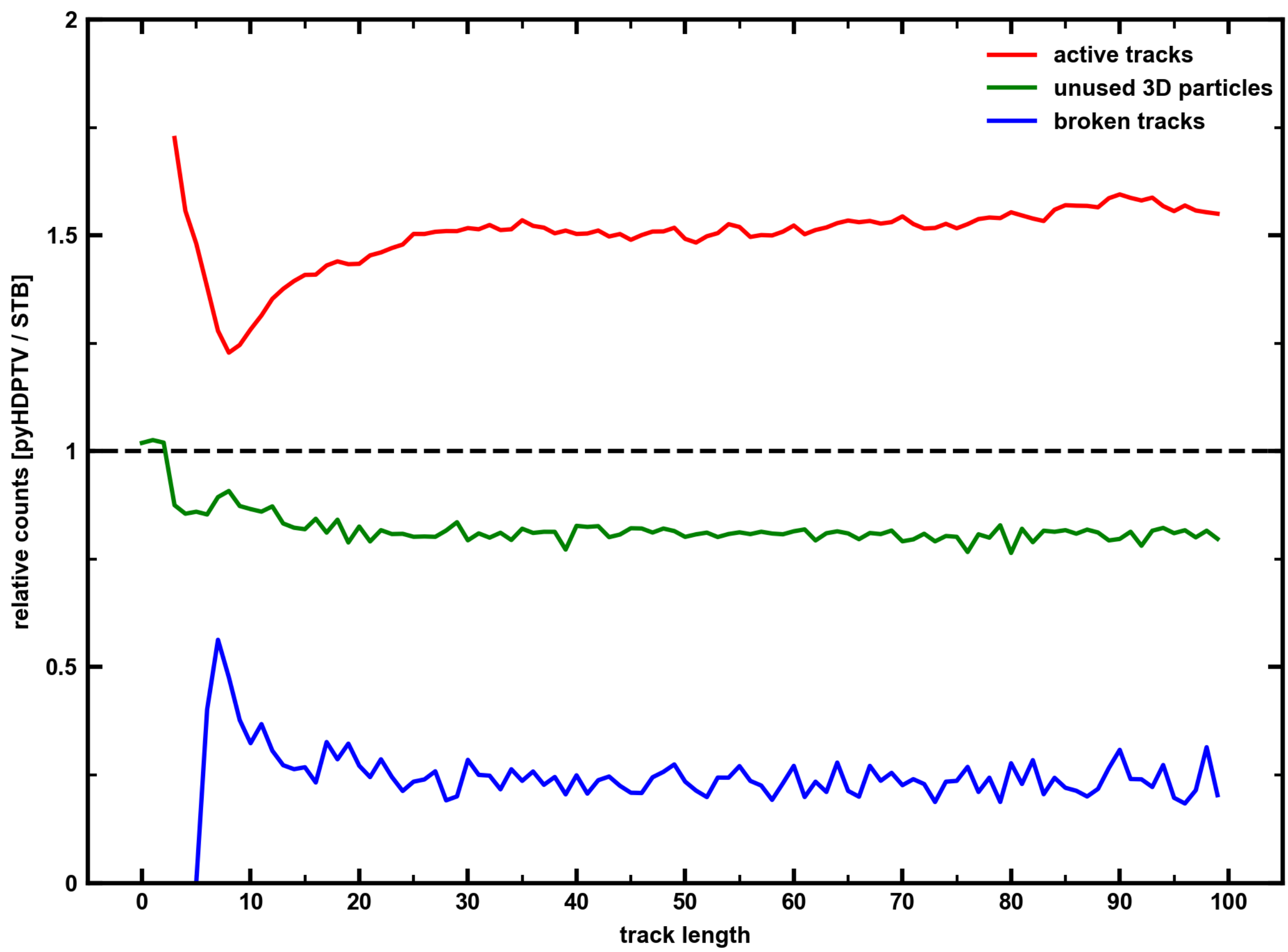

Figure 4. Performance overview during code processing. The relative counts of triangulated 3D points, active tracks and broken tracks per time step are visualized. Thereby, the counts from the pyHDPTV method are divided by the counts from STB.

This observation is supported by looking at the number of active tracks per time step shown in Fig. 4 which are less for the STB method, i.e. the relative counts of active tracks is greater than 1. Thereby, STB reaches a plateau of round 5000 active tracks at time step 10 while pyHDPTV converges to a plateau of round 7500 active tracks at time step 20. Interestingly both frameworks triangulate roughly the same number of 13000 3D particles per time, since the starting value of the green line in Fig. 4 is close to 1. When the plateau of active tracks is reached pyHDPTV triangulates round 7000 unused 3D particles per time step and STB triangulates round 9000 unused 3D particles per time step, so that the relative counts of unused 3D particles gets smaller than 1, as the green solid line indicates. At the plateau the number of new tracks per time step and broken tracks per time step equal each other. However, the number of broken tracks per time step is by STB a factor of 4 higher than with the pyHDPTV method, as the blue line in Fig. 4 indicates. For the pyHDPTV method the absolute number of broken tracks per time step is round 100.

Overall STB found 57800 tracks with a mean track length of 8. The pyHDPTV framework found a total amount of 18304 tracks with a mean track length of 40. The track length histograms are shown in Fig. 5. Due to the different track lengths distribution the histograms are plotted on different y-axes indicating the track counts.

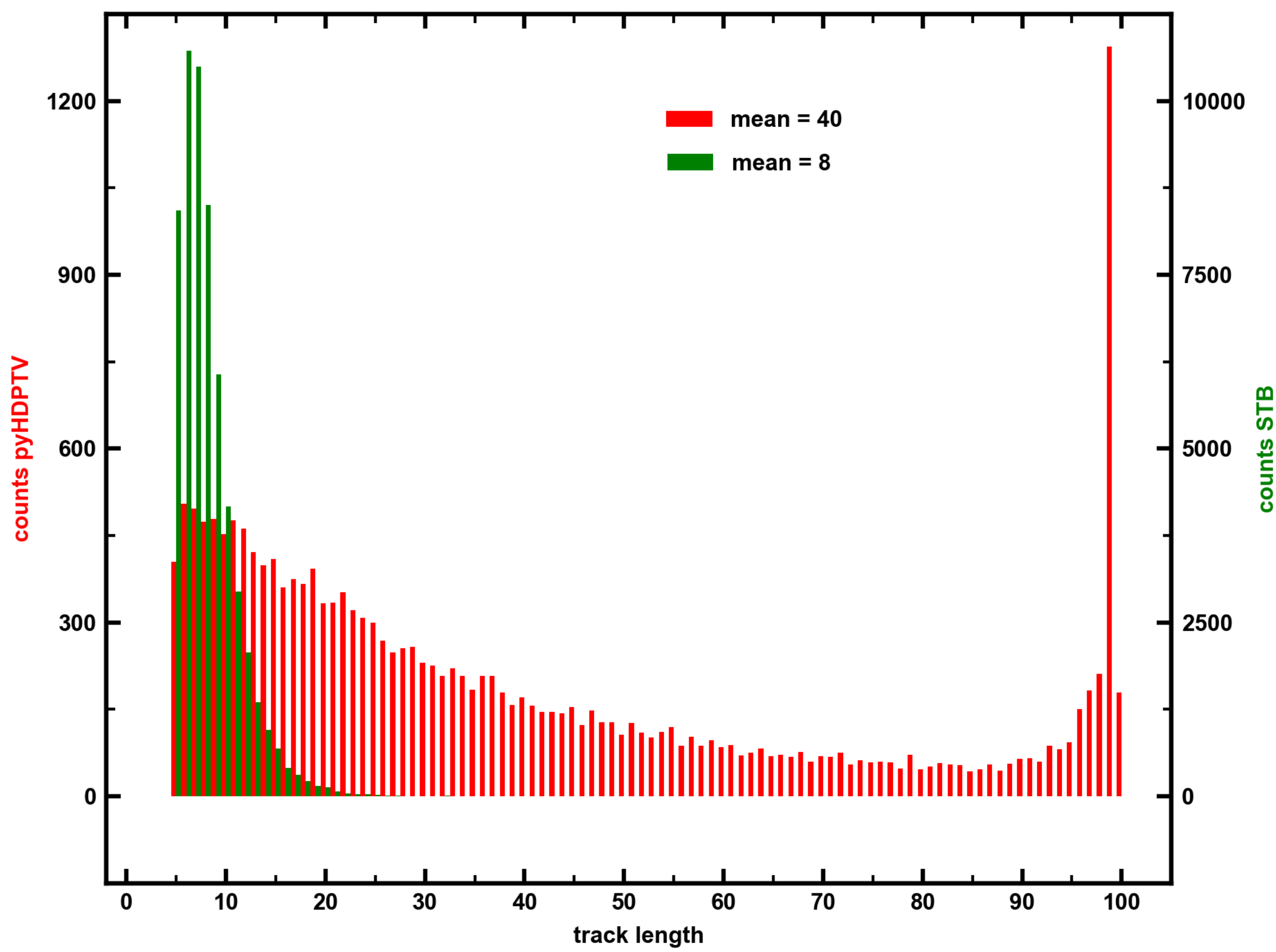

Figure 5. Track length histogram. The plot show the counts of which track length can be observed after applying the pyHDPTV and the STB method. The dotted lines indicate the weighted mean track length of each histogram.

Fig. 6 gives an overview of the track velocity distribution of each velocity component and the velocity magnitude. Besides of the v_z-, and the $||v||$-component both STB and pyHDPTV display similar shaped velocity distributions. Whereas the v_z-, and the $||v||$-component are shifted to higher positive velocities in the case of processing with STB. This may come from the thinned out particle resolution near the cooling plate and the two left side walls in Fig 3. The thinned out parts may follow from missing calibration plate positions in these areas. Because the polynomial calibration model from STB is defined only at the calibration plate positions and is then interpolated between them. The Soloff calibration model from pyHDPTV is more robust against missing calibration plane positions because it is optimized globally. The thinned out regions are visible by comparing the edges in Fig. 7.

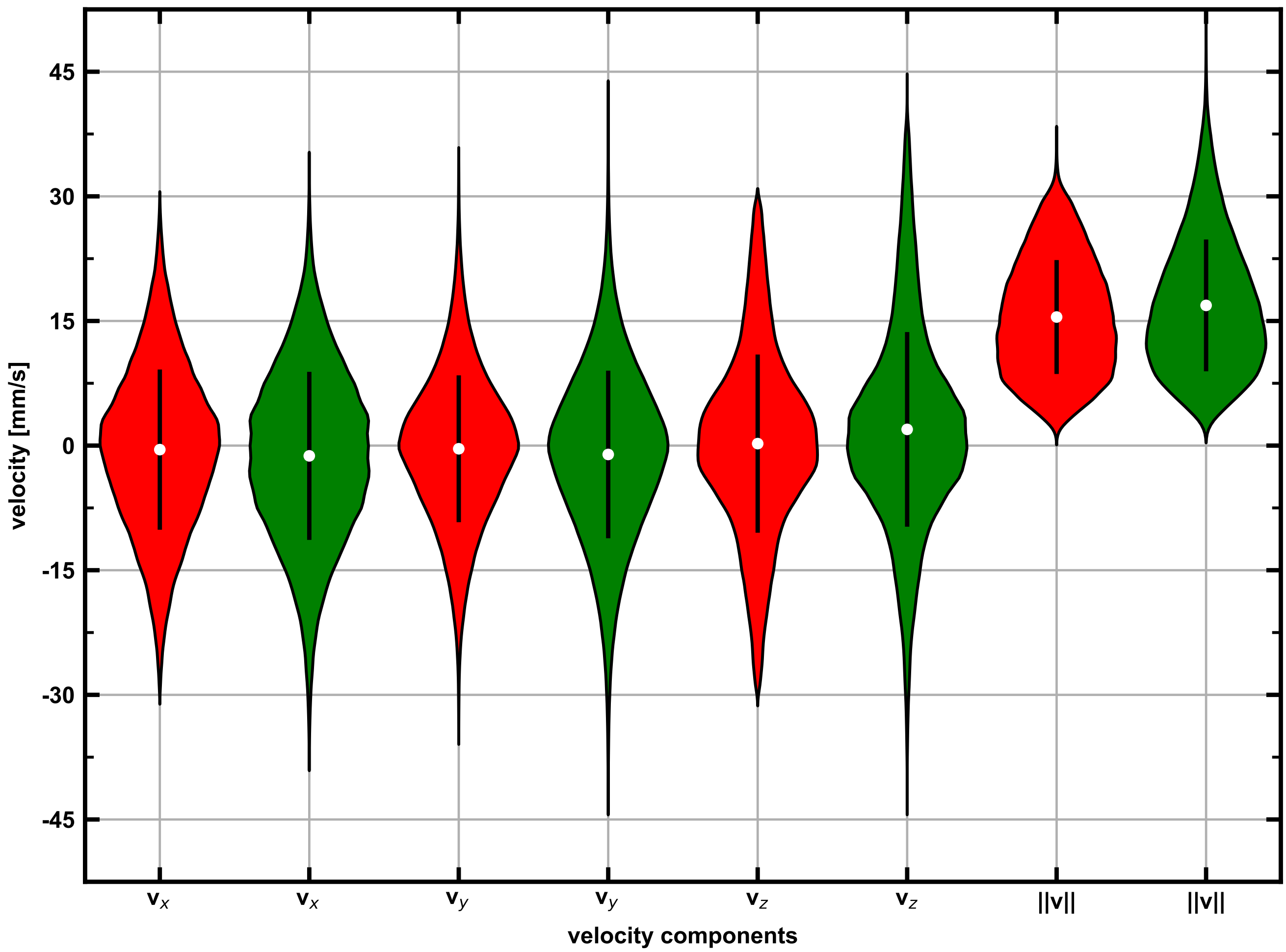

Figure 6. Velocity histograms. The plot show the distribution of each velocity component and the velocity magnitude over all tracks and all 100 time steps for the pyHDPTV and the STB method. The white nod indicates the mean velocity of each component. The black bar displays the standard deviation of the each velocity histogram.

To analyse how well the two frameworks resolve the LSC the Lagrangian velocity field from the trajectories is interpolated as an Eulerian velocity field on an equidistant grid with resolution of 25x25x25 points in the measurement volume. Fig. 7 shows the time averaged Eulerian velocity field along the LSC diagonal and its off-diagonal. Along the diagonal both mean fields look quite similar besides of the gaps of missing data in the mean field produced by the STB method near the cooling plate and in the corner next to the upward stream. The off-diagonal mean fields have similarities like the the corner vortex in the left bottom corner such as a large upward stream in the right hand side but generally they look different in the color coding.

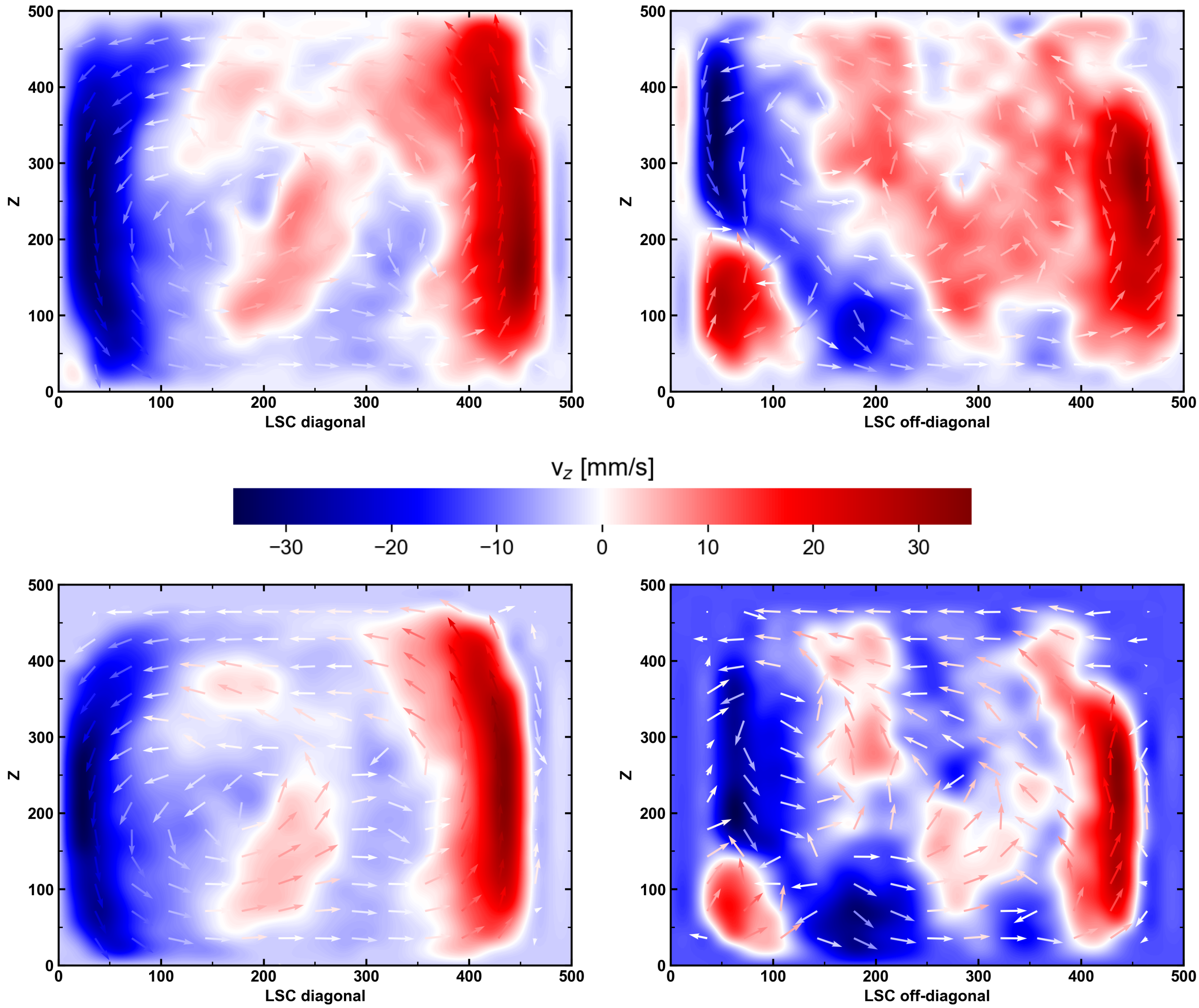

Figure 7. Time-averaged Eulerian velocity field along the diagonal and off-diagonal of the LSC in the measurement volume over all 100 time steps, **upper row** pyHDPTV **lower row** STB. The vertical velocity component v_z is used for color coding.

4. Conclusions

First of all, it must be noted that both algorithms performed well on the experimental data set. Because experimental data do not come with ground-truth informations it is not possible to precisely analyse qualitative differences in both frameworks. However, both methods were able to resolve the LSC and corner vortices. Inspecting Fig. 3 it is hard to spot major differences by naked eyes. Minor qualitative difference could only be found in the v_z-histogram of all tracks (Fig. 6), in the color coding of the Eulerian mean velocity field along the LSC off-diagonal (Fig. 7) and in the particle distribution throughout the RB cell. Especially the differences in the particle distribution between both codes at the left side-walls in Fig. 3 and cooling plate can be explained by the choice of the calibration method. The pyHDPTV method uses the more advanced Soloff model which also allows a camera calibration with completely different viewing angles. This is not the case when using the commercial STB software so that all cameras had to be located on the same side-wall of the RB cell.

The quantitative analysis showed that pyHDPTV leads to significantly longer tracks, as Fig. 5 indicates. The pyHDPTV framework reached a mean track length of 40 compared with a mean track length of 8 with STB over 100 time steps. Comparing the performance of both codes during the processing, see Fig. 4, one spots another quantitative difference between both codes. First of all, pyHDPTV reaches more active tracks and less unused triangulated 3D particles per time step. Second, the number of broken tracks per time step is for STB by a factor of 4 higher than with pyHDPTV.
Finally one can say the new pyHDPTY algorithm without backtracking performs at the same qualitative level as the state-of-the-art STB code from LaVision (DaVis v10.2.0) and a bit better in quantitative details.

References

Brown, E., & Ahlers, G. (2006). Rotations and cessations of the large-scale circulation in turbulent rayleigh–bénard convection. *Journal of Fluid Mechanics*, *568*, 351–386.

Dabiri, D., & Pecora, C. (2020). *Particle tracking velocimetry*. IOP Publishing Bristol.

Fuchs, T., Hain, R., & Kähler, C. J. (2017). Non-iterative double-frame 2d/3d particle tracking velocimetry. *Experiments in Fluids*, *58*(9), 1–5.

Herzog, S., Schiepel, D., Guido, I., Barta, R., & Wagner, C. (2021). A probabilistic particle tracking framework for guided and brownian motion systems with high particle densities. *SN Computer Science*, *2*(6), 1–20.

Jahn, T., Schanz, D., Gesemann, S., & Schröder, A. (2021). Advanced iterative particle reconstruction for lagrangian particle tracking. *Experiments in fluids*, *62*(179).

Lappa, M. (2009). *Thermal convection: patterns, evolution and stability*. John Wiley & Sons.

Maas, H., Gruen, A., & Papantoniou, D. (1993). Particle tracking velocimetry in three-dimensional flows. *Experiments in fluids*, *15*(2), 133–146.

Novara, M., Schanz, D., Gesemann, S., Lynch, K., & Schröder, A. (2016). Lagrangian 3d particle tracking for multi-pulse systems: performance assessment and application of shake-the-box. *18th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, 2638–2663.

Schanz, D., Gesemann, S., & Schröder, A. (2016). Shake-the-box: Lagrangian particle tracking at high particle image densities. *Experiments in fluids*, *57*(5), 1–27.

Schiepel, D., Bosbach, J., & Wagner, C. (2013). Tomographic particle image velocimetry of turbulent rayleigh-bénard convection in a cubic sample. *Journal of Flow Visualization and Image Processing*, *20*(1-2).

Soucasse, L., Podvin, B., Rivière, P., & Soufiani, A. (2019). Proper orthogonal decomposition analysis and modelling of large-scale flow reorientations in a cubic rayleigh–bénard cell. *Journal of Fluid Mechanics*, *881*, 23–50.

Tan, S., Salibindla, A., Masuk, A. U. M., & Ni, R. (2020). Introducing openlpt: new method of removing ghost particles and high-concentration particle shadow tracking. *Experiments in Fluids*, *61*(2), 1–16.

Vasiliev, A., Sukhanovskii, A., Frick, P., Budnikov, A., Fomichev, V., Bolshukhin, M., & Romanov, R. (2016). High rayleigh number convection in a cubic cell with adiabatic sidewalls. *International Journal of Heat and Mass Transfer*, *102*, 201–212.

Wei, P. (2021). The persistence of large-scale circulation in rayleigh–bénard convection. *Journal of Fluid Mechanics*, *924*.

Wieneke, B. (2008). Volume self-calibration for 3d particle image velocimetry. *Experiments in fluids*, *45*(4), 549–556.

Wieneke, B. (2013). Iterative reconstruction of volumetric particle distribution. *Meas Sci Technol*, *24*(2), 024008. Retrieved from `http://stacks.iop.org/0957-0233/24/i=2/a=024008`

Yang, Y., & Heitz, D. (2021). Kernelized lagrangian particle tracking.