



PAPER

OPEN ACCESS

RECEIVED
28 May 2021REVISED
6 August 2021ACCEPTED FOR PUBLICATION
20 August 2021PUBLISHED
2 September 2021

Fast 3D particle reconstruction using a convolutional neural network: application to dusty plasmas

Michael Himpel*  and André Melzer 

University Greifswald, 17489 Greifswald, Germany

* Author to whom any correspondence should be addressed.

E-mail: himpel@physik.uni-greifswald.de**Keywords:** 3D, particle, reconstruction, neural, networks, vision, dusty plasma

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Abstract

We present an algorithm to reconstruct the three-dimensional positions of particles in a dense cloud of particles in a dusty plasma using a convolutional neural network. The approach is found to be very fast and yields a relatively high accuracy. In this paper, we describe and examine the approach regarding the particle number and the reconstruction accuracy using synthetic data and experimental data. To show the applicability of the approach the 3D positions of particles in a dense dust cloud in a dusty plasma under weightlessness are reconstructed from stereoscopic camera images using the prescribed neural network.

1. Introduction

Machine learning currently is a rapidly growing field in its application to physics questions. Many challenging problems can now be addressed with different approaches from the constantly evolving artificial intelligence repository. Machine learning has been especially applied to image analysis or image classification [1–4].

The three-dimensional reconstruction of particle positions from multiple-view camera setup is another problem where machine learning can be of enormous help. Traditionally, the following different approaches for particle position reconstruction are often employed: volumetric reconstruction, triangulation, and iterative reconstruction. An example of a volumetric reconstruction method is tomographic PIV [5, 6]. There, the volumetric source field that contains the particles is computed by algebraic means using the measurement images as projections of the source field. Its drawback is a very high computational effort and thus the slow processing speed. Triangulation-based approaches [7, 8] are faster compared to tomographic PIV. Triangulation relies on a camera calibration and reconstructs the three-dimensional particle positions from known particle correspondences in the measurement images. The main problem that needs to be solved here is to find these corresponding particle projections in the different camera views. Usually this problem is addressed by means of epipolar geometry. The clear drawback of this approach is the ambiguity of the correspondences when the particle density in the measurement is high. The iterative reconstruction approaches [9, 10] optimize the particle positions to the given measurement images and use triangulation as well as volumetric reconstruction at certain moments. The Shake-the-Box (STB)-algorithm [10] is based on initial particle tracks, that are obtained using tomographic reconstruction. Then, these initial tracks are used to predict further locations of the particles that are refined to match the measurement images. Approved and optimized particles are erased from the measurement image and then new particles are detected using triangulation. STB is currently considered to be one of the state-of-the-art algorithms for high particle densities.

We want to apply the proposed AIPR algorithm in our field of research called *Dusty- or Complex Plasmas*. There, micrometer sized particles are injected into a plasma environment and attain a highly negative charge. This results in a variety of interesting collective effects like density waves or crystalline phases of the particle system. To our knowledge, machine learning has been only applied to analysis of two-dimensional investigations so far [11, 12].

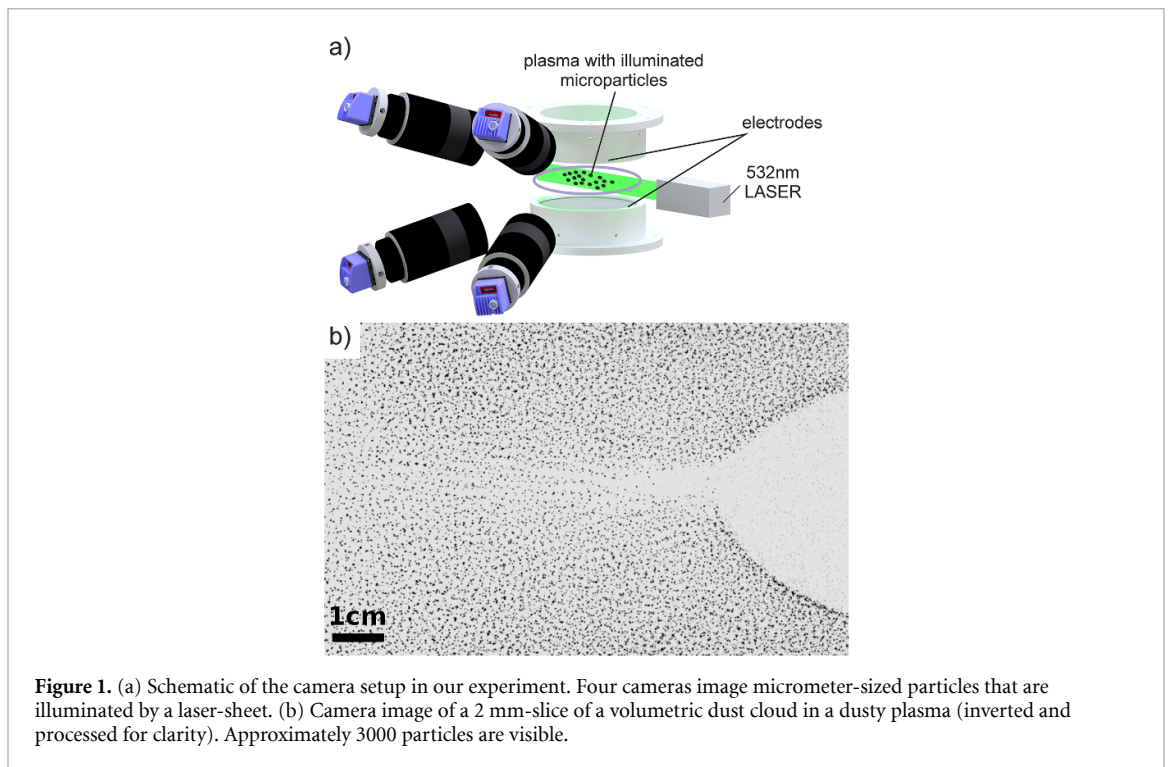


Figure 1. (a) Schematic of the camera setup in our experiment. Four cameras image micrometer-sized particles that are illuminated by a laser-sheet. (b) Camera image of a 2 mm-slice of a volumetric dust cloud in a dusty plasma (inverted and processed for clarity). Approximately 3000 particles are visible.

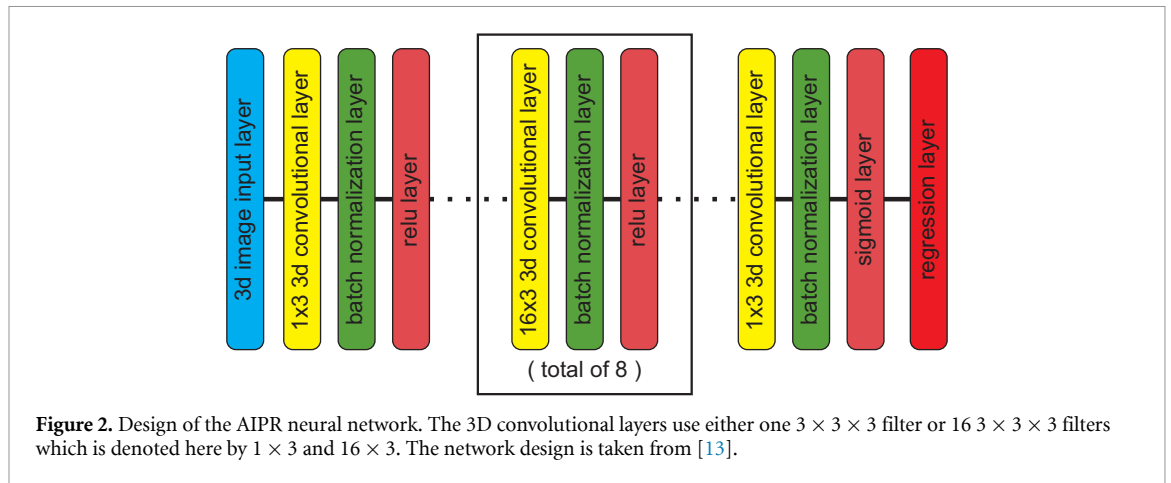
In this paper we will use a machine learning approach to reconstruct three-dimensional particle positions in a dusty plasma experiment. We adopt a deep learning algorithm, AIPR (Artificial Intelligence Particle Reconstruction) [13] that relies on a neural network to retrieve volumetric fields from preliminary (coarse) tomographic reconstructions. We advance AIPR by applying it to the experiment data from a parabolic flight campaign with about 3000 visible particles. Further, we extract the resulting three-dimensional particle positions from the computed volumetric field. Our approach will be tested and compared against the traditional STB approach, where a superior speed, maintaining a comparable accuracy, of AIPR is demonstrated. It is special to this algorithm that the computing workload is separated into two parts: the energy- and time consuming training, followed by the framewise non-intensive and fast reconstruction. This makes this algorithm especially suitable for remote applications. For example, future experiments on board the ISS can highly benefit from such a data analysis workflow.

The MATLAB source code of the implementation is available [14].

2. Experiment

The experimental data used in the AIPR reconstruction are from dusty plasmas under weightlessness on parabolic flights. Under such conditions the dust particles with a typical diameter of 4–8 μm attain a high negative charge and form a large and dense volume-filling cloud [15–18]. In the experiment, micrometer-sized grains are trapped in a low-temperature argon radio-frequency discharge. The plasma chamber is very similar to the IMPF-K2 design from earlier experiments [19, 20]. The argon pressure was 30 Pa and the plasma power was 3 W. Under such conditions, by injecting the microparticles using a dispenser, a dust cloud of about 10^6 microspheres can be confined in the plasma environment. There the particles interact via their electrostatic repulsion and via plasma-mediated forces [21]. A laser illuminates a volume of the dense dust cloud with an expanded beam of a few millimetres thickness. The light scattered by the particles is recorded with high-speed video cameras as sketched in figure 1. To reconstruct the three-dimensional positions of the particles, our camera setup consists of four synchronized cameras (MV-BlueFox3-2-2051). The measurements have been done at 200 fps. The pixel size of the sensors was 3.54 μm , but we used a 2×2 binning mode which results in an effective pixel size of 7.08 μm . The observed volume in the dust cloud was about $14 \times 9 \times 2$ mm. In that region, a few thousand particles were present.

For details regarding the setup and the calibration of this camera system, the reader is referred to [22]. In this paper, we will revisit a measurement that has already been analyzed with a state-of-the-art algorithm called STB [9, 10, 22]. The reconstructed three-dimensional motion of several thousands of particles has revealed that the particles arranged in two distinct layers within the observed volume [22]. This finding is confirmed by the AIPR approach, but the results are obtained much faster.



3. Outline of the AIPR approach

Here, first, the general idea for the machine learning reconstruction is described, the details are given in the following sections. The main processing chain of the proposed algorithm is as follows. The measurement volume is discretized into a number of voxels, a so called volumetric field. The field contains $N_x \times N_y \times N_z$ voxels, where N_x, N_y, N_z is of the order of 100–400, which in our case is limited by the GPU memory. Then, the measurement images of all four cameras are algebraically combined to a single initial volumetric field that contains a kind of ray-casting information of bright image pixels that are projected into the volumetric field. Then, a neural network is trained with synthetic data adopted to the measurement one wants to analyze. The task of the network will be to predict the final volumetric field from the initial volumetric field. This final field can then be used to extract the actual 3D positions of the particles.

When systems of indistinguishable particles (as in our case) are studied, there is a significant advantage compared to many complex neural network designs that are typically applied for three-dimensional reconstruction of real-life scenes. First, the objects are of simple shape (spherical) which can be easily modeled artificially. There is no need for the recognition or classification of objects as an individual task. Hence, the network does not need to compute feature maps but can be seen as a kind of sharpening filter in a 3D field. Another advantage lies in the possibility of artificial training data. A neural network generally needs a large number of training datasets to be properly trained. In our case, these are image sets for the four cameras and the desired final volumetric field. In many real-life applications of neural networks this training data is hard or expensive to obtain. In our situation, we can easily calculate artificial images from randomly chosen 3D positions with tuneable particle appearances. Also, the corresponding volumetric field is easily constructed. As a result, the neural network can be fine-tuned to match the exact measurement conditions such as particle sizes, brightness and the image noise of the cameras. In the following section we will present the actual network design and give detailed information regarding the training process.

4. Network design

The network design is taken from [13] and will be briefly outlined here. The neural network is designed to transform the volumetric initial input field I_i of size $N_x \times N_y \times N_z$ to the volumetric final output field I_f with the same size. Its design is depicted in figure 2. The dimensionality and voxel numbers of the volumetric field remain unchanged throughout the network. The network is built with a 3D image input layer matching the size of the initial field $N_x \times N_y \times N_z$. The first convolutional layer is chosen as $3 \times 3 \times 3$ sliding cuboidal convolution filter with a single filter followed by batch normalization and a ReLU layer. The following convolutional layers are set to maintain the data size—this is typically called *same padding*—and are followed each by a batch normalization layer and a ReLU layer. The first block is continued with eight 3D-convolutional layers with filters of size $3 \times 3 \times 3$ and a number of 16 filters each. The last convolutional layer has a filter size of 1 and a kernel size of $3 \times 3 \times 3$ followed by a batch normalization layer. To map the network onto an output field with values between zero and one, a sigmoid layer is used here rather than a ReLU layer.

The regression layer needs a suitable loss function to ensure that the network weights converge during training. The loss function is necessary to define a match or mismatch between the desired result and the actual result from the network. The originally proposed loss function used a fine tuning parameter ε to

ensure convergence of the network during training. Here, unlike the original network from [13], we employ a dice coefficient as a loss function \mathcal{L} [23]. This one is applicable without any input parameter or prior knowledge of the user side. It is defined by

$$\mathcal{L} = 1 - \frac{2 \sum (T \cdot Y)}{\sum T^2 + \sum Y^2}, \quad (1)$$

where T is the ground truth training data field (*Target*) and Y is the initial field. The summation is done over all voxels of the field and over all batches and the multiplication is done element-wise (Hadamard product). This loss coefficient is normalized so that it returns 0 for a perfect match between T and Y and 1 for mismatching T and Y . For clarity it can be thought of as an intersection-over-union loss function that is widely addressed in the literature [24, 25].

5. Network training details

The training of the neural network is very time consuming, especially when the volumetric field consists of a fine voxel grid and many training images are used. Each training dataset contains the ground truth 3D field and its corresponding projected camera images. The ground truth field is generated by choosing random particle positions in the reconstruction volume. To the voxels around the chosen particle position, Gaussian distributed voxel intensities are assigned. The spatial width of the Gaussian is chosen to significantly lift at least the particle containing voxel and the closest neighboring voxels above noise level. The artificial camera images are computed from the exact random particle positions using the camera projection matrices of each camera. The particles in the image are then again modeled as a Gaussian with a width of typically 5 px. This results in a particle diameter of 8–10 px in the image, whereas 1 px corresponds to about 12 μm in the investigated plasma volume. To account for difficult imaging situations in our experiment, each particle is given a random maximum intensity in the range of 0.7–1. This intensity is used in the images as well as in the volumetric field. To account for as realistic as possible training data, we also included background noise in the images. However, we found that noise is not necessary to prevent overfitting during the training process. Finally, the camera image data is stored as 8-bit data and the 3D field is stored in single precision to save GPU memory during training.

After training, the network should of course be capable to correctly processing unknown input data instead of just reproducing the training data. To obtain such a *generalizing* network, there is a minimum number of different training datasets necessary. The minimum number of these training images can only be estimated: every voxel should be covered by the initial field at least once in the whole training dataset. As a rule of thumb, the necessary number of images can be calculated by the number of voxels divided by the number of particles per training images. For our case, we found that a number of 1000 training images is sufficient for the training to converge towards a generalizing solution for the network coefficients. Thereby, the 1000 training images are synthesized with a number of 4000 particles randomly spread over the investigated measurement volume as shown in figure 3. As the network's memory footprint is usually large, we propose to use a batch size of 1, which means that only one training dataset is used at a time to optimize the network. We found that convergence is usually reached in 3 epochs with decreasing learning rates of 0.1, 0.01 and finally 0.001.

The training process is significantly faster using GPU-acceleration. We used a NVIDIA RTX-2080 Ti graphics adapter for the training. It takes approximately 6 h compared to a CPU training of 16 h. The crucial parameter is the amount of GPU memory. The volumetric grid on which the initial field is defined is the main reason for the need of a large amount of memory. In our case we use a $(332 \times 220 \times 68)$ -grid with a spacing of 40 μm and a finer $(456 \times 302 \times 91)$ -grid with a spacing of 30 μm . This results in almost full occupancy of the 11 GB GPU memory during the training process.

When a neural network is trained, it has to be ensured that the converged result generalizes well to unknown data instead of just learning the training data 'by heart'. To study the generalization behaviour, we ran different trainings with a varied amount of noise in the synthetic measurement images and never found a converged solution that did not generalize to unknown data. Hence, we conclude that the network is generalizing quite well by design.

6. From images to particle positions

In this section, we will give a detailed description of the necessary steps to retrieve particle positions based on AIPR from measured images. An accurate camera calibration [26–29] is a prerequisite for all further steps. The network defined in the previous section is trained using the same camera calibrations as for generating the training data and will be used in the corresponding processing step.

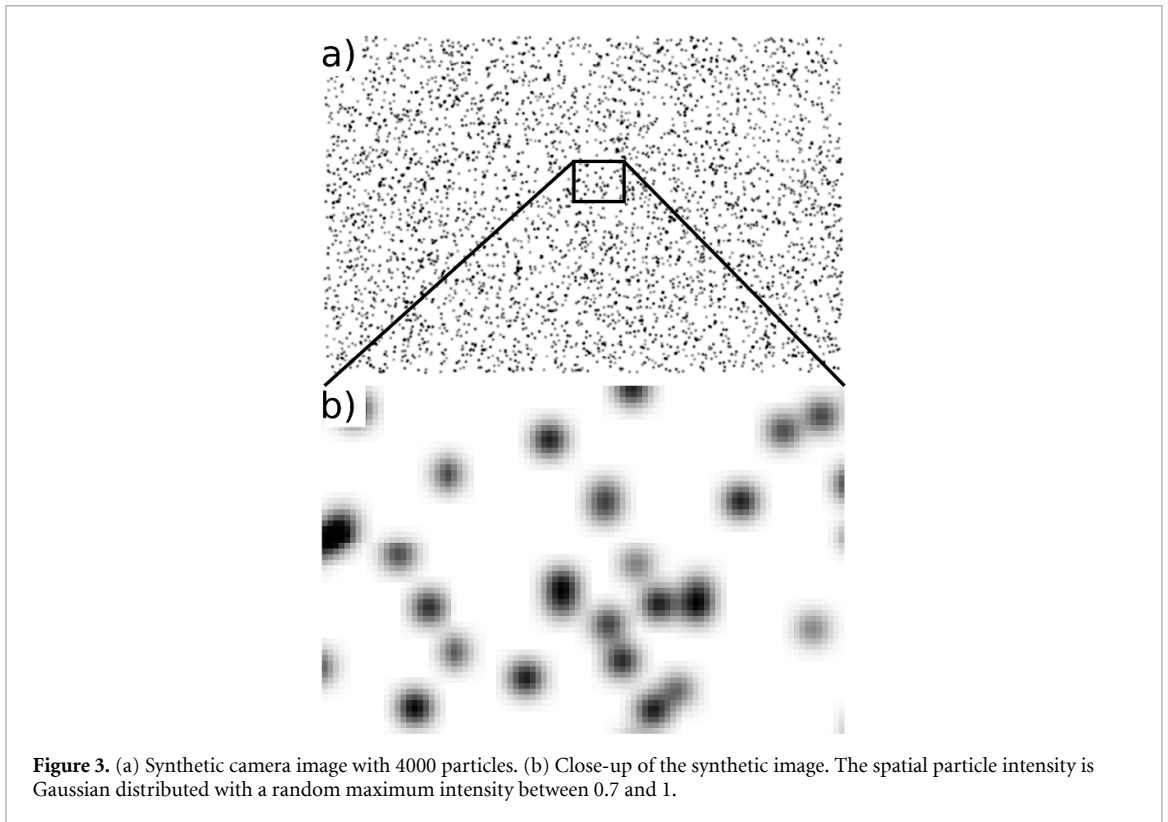


Figure 3. (a) Synthetic camera image with 4000 particles. (b) Close-up of the synthetic image. The spatial particle intensity is Gaussian distributed with a random maximum intensity between 0.7 and 1.

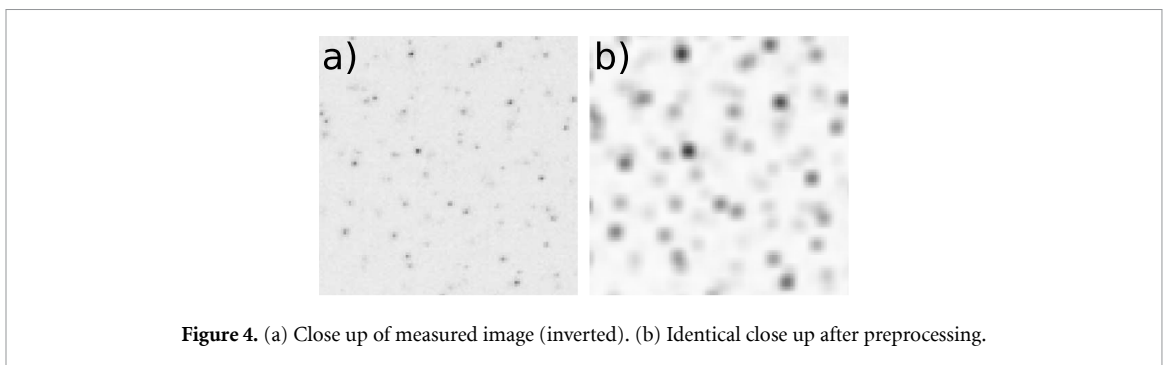


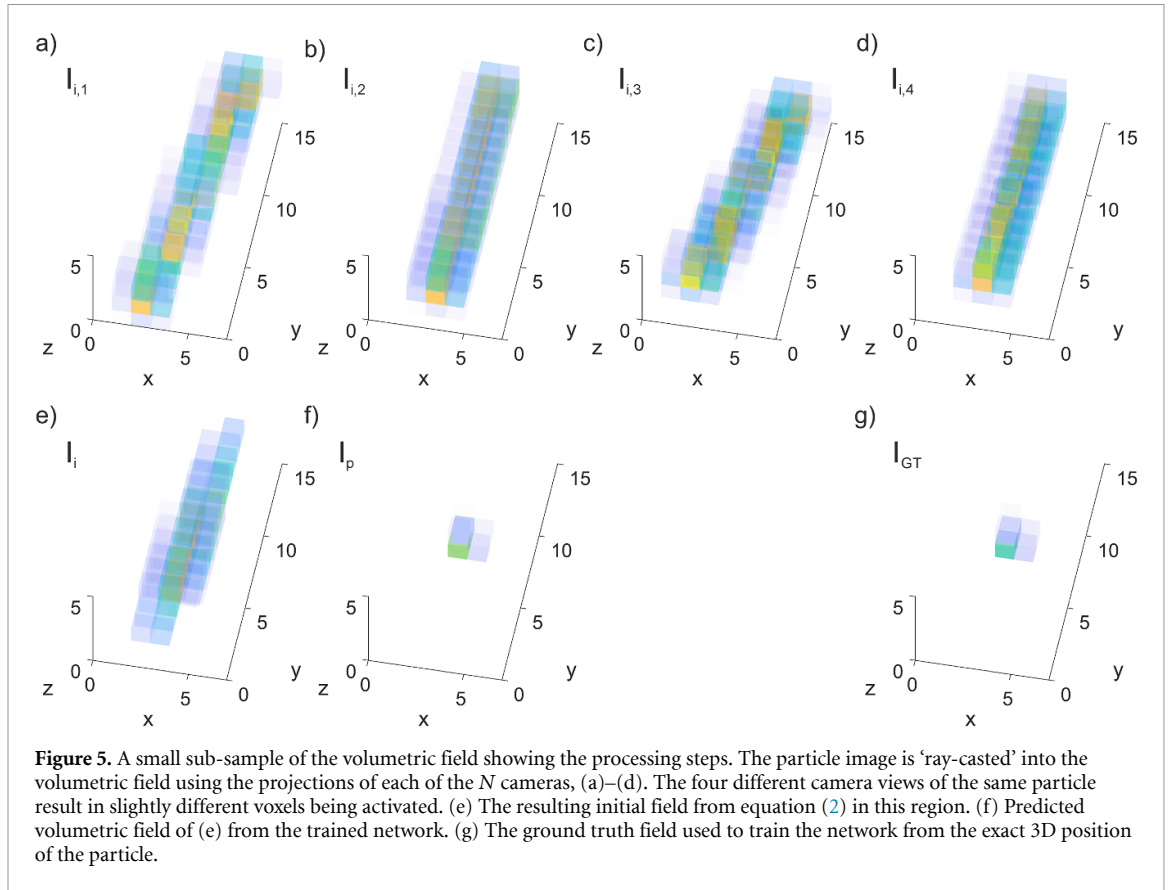
Figure 4. (a) Close up of measured image (inverted). (b) Identical close up after preprocessing.

6.1. Preprocessing of experimental images

In the presented design, the neural network processing needs the particles to have a common spherical shape with comparable brightness and size. This prerequisite is often hard to match in some experimental measurements. Hence, we applied the following image preprocessing steps to achieve uniform particle projections. The raw measurement image is processed by a Sobel filter which emphasizes intensity gradients. Afterwards, a two-dimensional Gaussian bandpass filter is applied to filter out noise and to turn the particle images into a Gaussian shaped spherical intensity profile. In figure 4 one can see the experimental raw image (a) and the processed image (b). After preprocessing the images, the particles have a nicely spherical shape but the intensity is still not uniform. To address this issue one can make the network learn to accept non-uniform brightness in a certain manner. During network training, one has to keep in mind to produce training images that also feature particle projections with non-uniform intensity. We found that enlarging the particles in the training images as well as the measurement images will improve the network detection outcome. This is probably due to our relatively coarse 3D grid (see next paragraph) which has a corresponding resolution of about 4 px in the measurement image.

6.2. Initial field generation

After the measurement images are preprocessed, a so called initial field $I_{i,N}$ is generated for every camera view ($N = 1, \dots, 4$ in our case). The field is defined on the same volumetric grid as the ground truth field. For a known projection matrix or mapping function, typically obtained by camera calibration, the initial field is generated as follows. First, for each voxel of the volumetric field, one needs to find that pixel onto



which the voxel center is projected. Then, the (scalar) entries of the initial field of each camera $I_{i,N}$ are given by the intensity of the corresponding pixels connected to each voxel (‘ray-casting’). We want to note that the algorithm is sped up for processing a large number of images when a lookup table is generated so that the projection is not computed again for every image. When this computation is done for all camera views, then the N initial fields are combined by

$$I_i = \left(\prod_N I_{i,N} \right)^{1/N}. \quad (2)$$

For clarity, this combination-process is sketched in figure 5. Images (a)–(d) show a small subsample of the volumetric fields $I_{i,N}$ from the projections of the sensor pixels. As the cameras have slightly different lines of sight into the volume, this ray-casting process results in slightly different directions of the activated voxels. After combining the fields (a)–(d) by using equation (2), the initial field I_i as shown in image (e) is ready to be processed by the neural network. The initial field combines the ray-casting information from each camera into a single volumetric field.

6.3. Network processing

The network processing itself is just the call of the trained network (in MATLAB this step is called *prediction*) with the initial field as an input. The predicted output field I_f produced by the neural network is again a volumetric field of the same size as the input field where the particle intensities range from zero to one. Particle locations are then encoded by contiguous bright voxels or ideally as 3D Gaussian distributed intensity regions (if the grid resolution is fine enough). The given example in figure 5(f) shows the result of an example prediction of the network. The elongated initial field from figure 5(e) seems to ‘collapse’ into a 3D spot. For comparison we also show the ground truth field in figure 5(g), which is known as test data from the network, but has not been used as learning data. The ground truth field and the predicted field are in good agreement demonstrating the capability of the network.

6.4. Particle position extraction

To extract the actual three-dimensional particle positions from this output volumetric field, we propose the following approach. In experiments it is not always possible to guarantee a uniform illumination of the

observed particles. This results in a volumetric field that contains voxels representing reconstructed particles, which have a non-uniform brightness. A simple threshold filter applied on the 3D field will thus not recover all particle positions in this case. To detect the positions of as many as possible particles, we propose a step-wise reduced threshold value followed by identifying contiguous regions in the volumetric data. In MATLAB, the regions above a certain intensity threshold can be found using the *regionprops* function. If a region with 7 or more voxels above the current threshold is found, their intensity-weighted mean is then associated with the particle position. Before proceeding with the next lower threshold to find particles with lower intensity, it is necessary to zero all previously identified connected regions from the volumetric dataset. In our measurements we found that reducing the intensity from 0.7 to 0.05 with a stepsize of 0.05 works well. These values, of course, depend on the noise level and the general data quality.

6.5. Position refinement based on STB

As the AI reconstruction algorithm is based on a volumetric grid with a limited spatial resolution, it is clear that the accuracy of the reconstructed particle positions is also limited. Due to hardware restrictions or to reduce computing time the volumetric field might be chosen coarser than the required spatial resolution. Nevertheless, in this case the coarser positions serve as a perfect starting point for a refinement step using the STB method [10]. In the following, we will carry out benchmark tests with and without the additional STB refinement to show its influence.

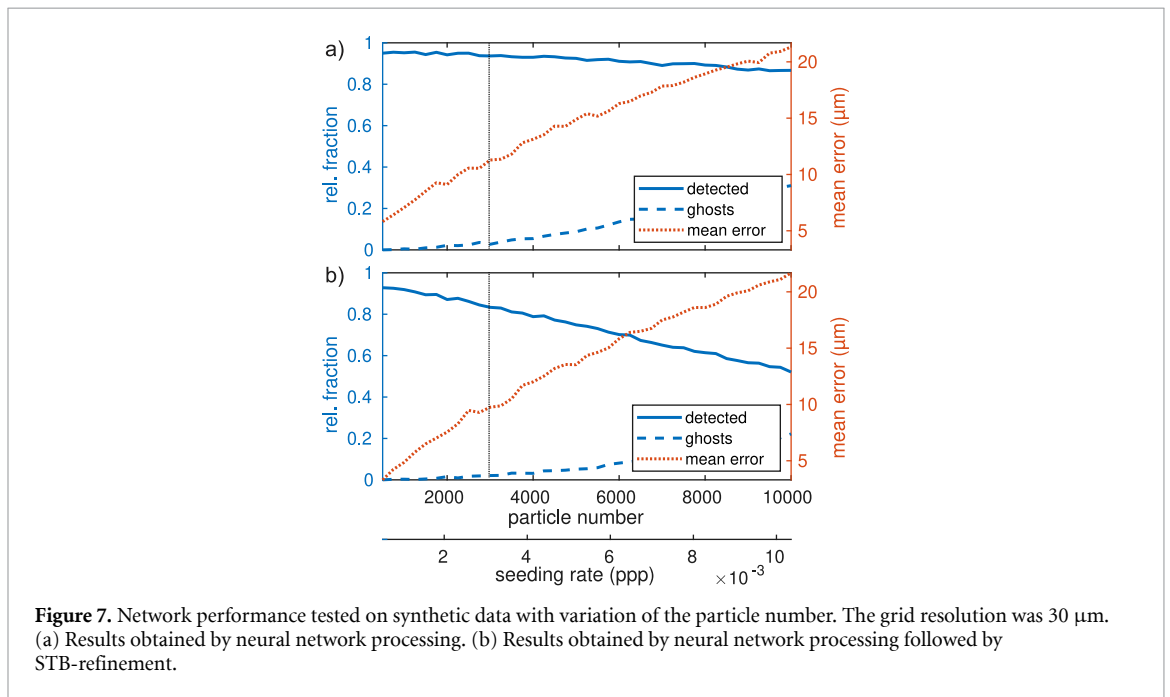
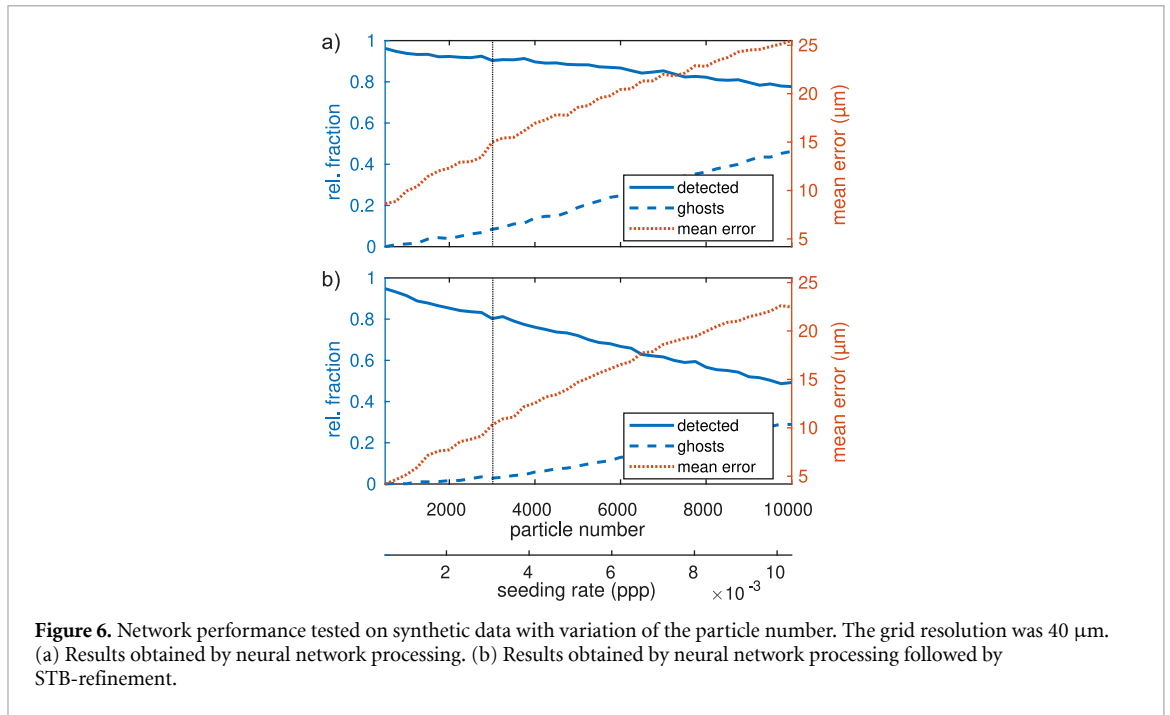
7. Results—synthetic data

In this step we generate test data in the same way as the learning data. However, these test data have not been used in the learning data set, hence, the network does not ‘know’ the test data. To estimate the performance of the trained network, we will show different measures for varied parameters. One measure will be called the mean error. This is defined as the distance (in μm) from a reconstructed position to its nearest neighbor in the ground truth of the test data. Another measure is the detection rate $R = N_{\text{NN}}/N_{\text{gt}}$, which is defined by the fraction of the correctly reconstructed number of particles N_{NN} that are closer than $50 \mu\text{m}$ to one of the N_{gt} ground truth particles. It tells us, how many of the synthetic particles are successfully reconstructed. The last important measure will be the ratio of ghost particles. Particles that are reconstructed, but do not match a ground truth particle within a distance of $50 \mu\text{m}$, will be considered as a ghost particle. The ratio $G = N_{\text{gh}}/N_{\text{NN}}$ is then defined as the ratio between the number of ghost particles N_{gh} and the total number of reconstructed particles N_{NN} . The following benchmarks on synthetic image data have been done using the neural network based on a $30 \mu\text{m}$ ($456 \times 302 \times 91$ voxels) and a $40 \mu\text{m}$ ($332 \times 220 \times 68$ voxels) grid. The image input data and the reconstruction volume was identical in both cases. The slight reduction of voxel size from 40 to $30 \mu\text{m}$ increases the number of voxels by a factor of 2.5. A further reduction of voxel size was not possible with our hardware.

The benchmark results using the coarser grid are shown in figure 6. The number of particles that has been used for each benchmark run is shown as a total number and as a seeding rate (particles per pixel or ppp), which is more useful to be comparable with other experiments and simulations. Figure 6(a) shows the results that are obtained by just using AIPR, figure 6(b) shows the same results followed by STB refinement. It can be seen that for particle numbers of about 3000 which we find typically in our experiments, 90% of the particles without refinement and 80% with STB-refinement are reconstructed. For clarity, the particle number of 3000 is also indicated by the vertical dashed line in the plot. Note that the higher reconstruction fraction without using the refinement comes at the cost of a higher ghost particle rate which is 7% without refinement and 3% with refinement. This means STB effectively reduces ghost particles, but also true particles. STB is therefore more restrictive. The positioning error (dotted line) of the reconstruction based on the $40 \mu\text{m}$ grid is about $15 \mu\text{m}$ without refinement and just $10 \mu\text{m}$ with refinement at a particle number of 3000.

The neural network based on the finer grid performs slightly better as shown in figure 7. The reconstruction rate is on a higher level for the non-refined results in figure 7(a). The ghost particle rate is slightly lower compared to the coarse grid from figure 6. This might be due to the fact that the volumetric field is now less sparse and contains more ‘hot voxels’ that contribute to a particle which eases the particle detection. The mean error is almost the same for the raw network processing and the STB-refinement pos-processing.

The network that was defined on a finer grid performs reasonably well. The prediction step of the neural network followed by the position extraction from the volumetric field is done in about one second or less on our GPU. The STB refinement, which takes several minutes per frame does not improve the reconstruction too much. The accuracy is comparable to the STB-refined approach while the ghost particle fraction is still low considering about 3000 visible particles. The reader should note that a further reduction of ghost particles is possible by tracking particles over many frames. Ghost particles would only ‘exist’ for a few

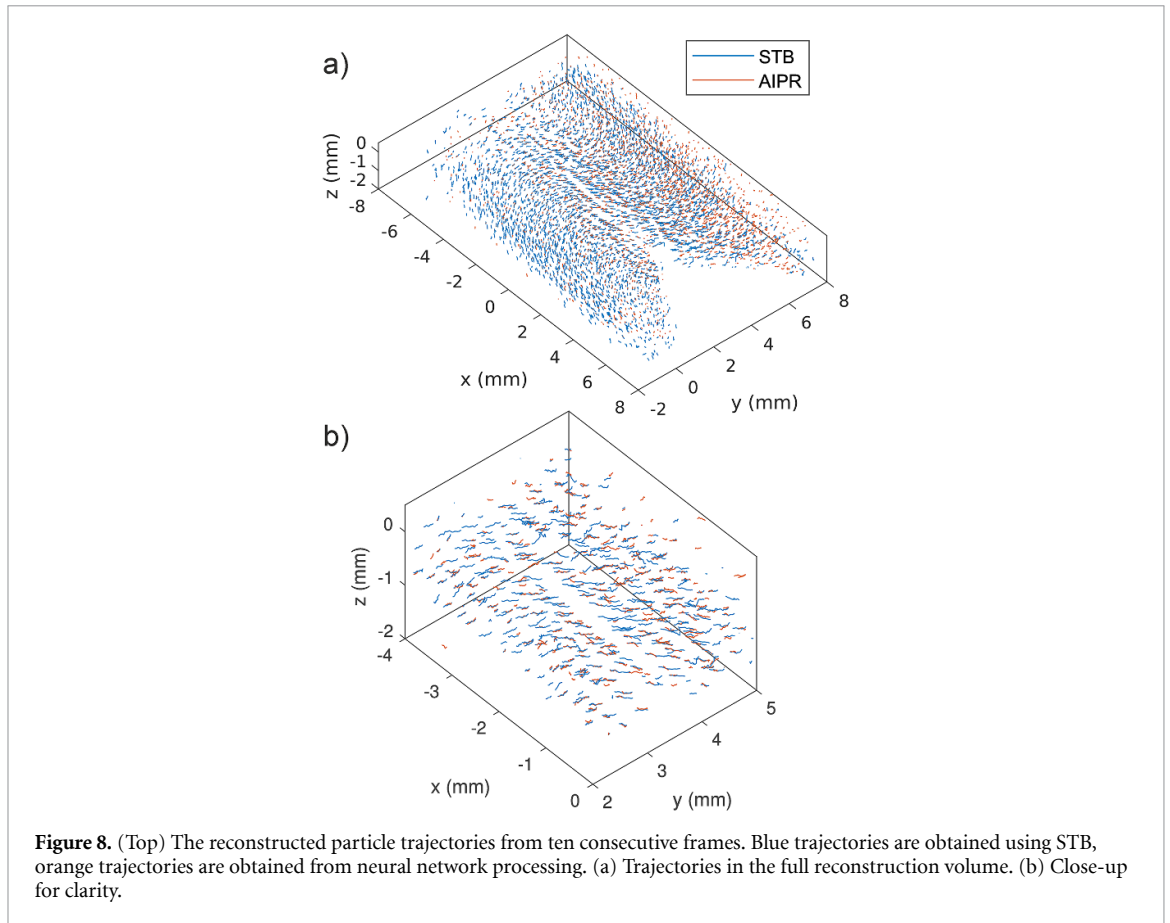


tracked frames. For the application to our experimental data, we will use the neural network with the finer (30 μm) resolution.

For both cases it can be seen that the performance gets worse when the particle number or seeding density gets too high. We think that this is based on the fact that we need a certain minimum particle distance in the volumetric field for our particle position extraction to work accurately. Unfortunately, this means that with increasing seeding density one would need an increasing grid resolution which is not possible in most cases due to hardware limitations.

8. Results—experimental data

Here, we now use experimental data from the parabolic flight as described in section 2. The neural network has been trained with a set of projection matrices that are obtained from an actual calibration of our experimental setup. The volume of the volumetric field was adjusted to match the investigated volume in the experiment. Thus, the trained network can be directly applied to reconstruct particles from a set of



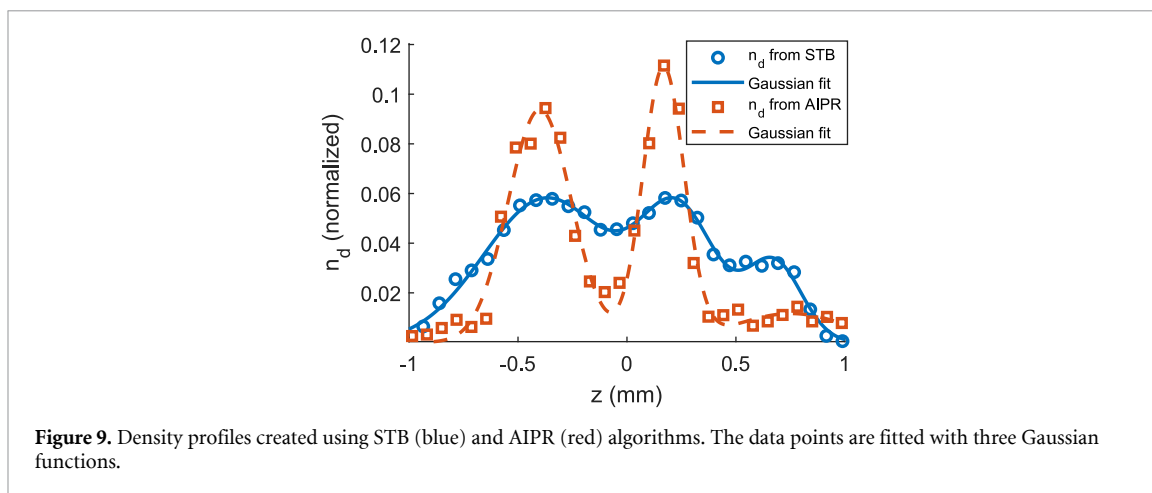
measurement images that have been done with the camera system. One image from the experiment is shown in figure 1.

As this is a measurement, there is no ground truth data available which can be used to verify the reconstruction results. However, these data has been previously analyzed using STB [22] and we can now compare the results from the neural network with the STB results. It should be noted that in this analysis of the experiment the STB approach was used completely independently of the neural network and not as a refinement step of neural network predictions. In figure 8 we show sample particle trajectories that have been tracked for ten frames. As one can see, the different methods seem to be sensitive to different aspects as the detected particles appear partially in different regions in the dust cloud. While AIPR has more detections in the positive y -direction which cannot be found in the STB data, there are less AIPR detections in the negative y -region compared to the STB results. To quantify this, we identified all matching particle positions that are characterized by a proximity between the reconstruction techniques of less than $40 \mu\text{m}$.

One result of this position matching is, that the mean distance of the particle positions from both approaches is $16 \mu\text{m}$. In other words, *when* the approaches reconstruct a particle, the positions agree quite well. On the one hand, 58% of the AIPR positions match the STB positions and 42% of the AIPR positions are exclusively detected in this approach. On the other hand, only 32% of the STB positions match the results from the AIPR processing and 68% of the positions are exclusively found by STB.

The difference in the results between both algorithms may be due to their basic principles. Whereas the AIPR algorithm makes a single snapshot-like detection in every single frame, the plain STB-algorithm tries to follow the particle path consecutively by using Kalman [30] or Wiener [31] filtering. Both approaches have pros and cons. The AIPR algorithm is thus insensitive to a sudden change of the particle system, that can be induced by vibrations in the measurement setup which is the case in our measurements on parabolic flights. The STB-algorithm has the advantage that particles are projected, reconstructed and then tracked for a relatively long time even if the brightness or imaging quality of this particle varies in consecutive frames.

Another possibility to compare both algorithms is to look at the physical properties obtained with either approach. The number density profile along z -direction of the observed dust cloud will now be compared. In earlier work on this dataset, we found that the z -profile of the number density n_d revealed a layered structure. In figure 9, this profile obtained by STB is shown by circles. The corresponding solid line represents a fit with three Gaussian distributions to the density data. The same dataset, but analyzed with AIPR, gives a quite



similar impression of the structure. The AIPR data in figure 9 shows two strong peaks with a much clearer separation of the layers. The third peak near $z \approx 0.7$ mm suggested by the STB analysis is only faintly present in the AIPR analysis. It can be suspected, that the AIPR algorithm seems to be less susceptible to ghost particles in between the peaks compared to STB. More ghost particles would result in an smeared out distribution. On the other hand, the faint third peak in the AIPR analysis may be caused by the poor imaging quality of particles that are not well focused or illuminated. As already said, it is difficult for the AIPR approach to handle particle projections with a poor signal-to-noise ratio. In contrast, some parameters in STB can be fine-tuned to also handle at least some of the weakly illuminated particles.

9. Conclusion

We have presented the application of a neural network to reconstruct three-dimensional particle positions from a multi-view imaging diagnostic. With an exemplary 4-camera setup the necessary steps for training and applying a neural network are described. It was shown that the neural network performs nearly as good as the Shake-the-box algorithm, whilst being extremely fast (once the trained network is available). The prediction step can be done on any modern office PC, but for training of the network a GPU with a large memory is recommended. For remote application of such reconstruction tasks, as e.g. on the International Space Station, this possibility to share the computation load is very welcome. The demanding computations, namely the network training, can be done before the measurement and on high-performance computers. After this energy and time consuming task, the analysis of the measurement images can be done with regular hardware in short time.

The reconstruction approach was benchmarked on synthetic data and applied to experimental data. The AIPR approach can be suggested for stereoscopic measurements of particles at a decently high seeding rate. With AIPR the successfully reconstructed particle fraction is in the range between 80% and 90% even at a high particle seeding rate. The number of ghost particles is still in an acceptable level and the position error is smaller than the voxel size.

AIPR has problems when the imaging conditions are not perfect. The influence of camera models and errors in camera positioning need to be addressed in future investigations. Nevertheless, we were able to reliably reconstruct 3D positions from experimental data of a dusty plasma. The results were very compatible with earlier analysis using STB. With AIPR we could verify the layering of the investigated dust cloud.

There is still work to be done to optimize the behaviour of the AIPR when imaging conditions are not perfect. Additionally, it is not yet clear how camera models and camera positioning may affect the performance of the neural network reconstruction. But as we presented in this paper, the speed of the reconstruction process which is nearly independent of the particle number is a sufficient reason to continue research in this field. Furthermore, the reconstruction of particles at higher seeding rates than we see in our experiment is still challenging. In future work we hope to get better results of higher particle density using more sophisticated position extraction from the final volumetric fields obtained by the neural network.

Data availability statement

The data generated and/or analysed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request. The data that support the findings of this study are available upon request from the authors.

Acknowledgment

Financial support from the Deutsches Zentrum für Luft- und Raumfahrt (DLR) under Project No. 50WM1962 is gratefully acknowledged.

ORCID iDs

Michael Himpel  <https://orcid.org/0000-0001-6710-0071>

André Melzer  <https://orcid.org/0000-0001-9301-9357>

References

- [1] Wei B, Wei W, Shaoyi B, Ben J and Bo L 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **787** 012002
- [2] Wang C, Sun X and Li H 2019 *J. Phys.: Conf. Ser.* **1176** 032028
- [3] Shen Z 2021 *J. Phys.: Conf. Ser.* **1881** 022005
- [4] Swalaganata G, Sulistyaningrum D R and Setiyono B 2017 *J. Phys.: Conf. Ser.* **893** 012062
- [5] Elsinga G E, Scarano F, Wieneke B and van Oudheusden B W 2006 *Exp. Fluids* **41** 933–47
- [6] Williams J D 2011 *Phys. Plasmas* **18** 050702
- [7] Akhmetbekov Y, Lozhkin V, Markovich D and Tokarev M 2011 Multiset triangulation 3D PTV and its performance compared to tomographic PIV *9th Int. Symp. On Particle Image Velocimetry-PIV* vol 11 pp 21–3
- [8] Mulsov M, Himpel M and Melzer A 2017 *Phys. Plasmas* **24** 123704
- [9] Wieneke B 2013 *Meas. Sci. Technol.* **24** 024008
- [10] Schanz D, Gesemann S and Schröder A 2016 *Exp. Fluids* **57** 70
- [11] Huang H, Schwabe M and Du C R 2019 *J. Imaging* **5** 36
- [12] Dietz C, Budak J, Kamprich T, Kretschmer M and Thoma M H 2021 *Contrib. Plasma Phys.* e202100079
- [13] Gao Q, Li Q, Pan S, Wang H, Wei R and Wang J 2019 Particle reconstruction of volumetric particle image velocimetry with strategy of machine learning (arXiv:1909.07815)
- [14] Himpel M 2021 AIPR Toolbox for MATLAB (available at: <https://physik.uni-greifswald.de/ag-melzer/aipr-toolbox/>)
- [15] Nefedov A P *et al* 2003 *New J. Phys.* **5** 33
- [16] Schwabe M, Zhdanov S, R ath C, Graves D B, Thomas H M and Morfill G E 2014 *Phys. Rev. Lett.* **112** 115002
- [17] Thoma M H, H ofner H, Kretschmer M, Ratynskaia S, Morfill G E, Usachev A, Zobnin A, Petrov O and Fortov V 2006 *Microgravity Sci. Technol.* **8** 47–50
- [18] Thomas H M *et al* 2008 *New J. Phys.* **10** 033036
- [19] Klindworth M, Arp O and Piel A 2006 *J. Phys. D: Appl. Phys.* **39** 1095–104
- [20] Klindworth M, Arp O and Piel A 2007 *Rev. Sci. Instrum.* **78** 033502
- [21] Ishihara O 2007 *J. Phys. D: Appl. Phys.* **40** R121–47
- [22] Himpel M, Sch utt S, Miloch W J and Melzer A 2018 *Phys. Plasmas* **25** 083707
- [23] Dice L R 1945 *Ecology* **26** 297–302
- [24] Jaccard P 1912 *New Phytol.* **11** 37–50
- [25] Rezatofighi S H, Tsoi N, Gwak J, Sadeghian A, Reid I D and Savarese S 2019 Generalized intersection over union: a metric and a loss for bounding box regression CoRR (arXiv:1902.09630)
- [26] Wieneke B 2008 *Exp. Fluids* **45** 549–56
- [27] Wieneke B 2018 *Meas. Sci. Technol.* **29** 084002
- [28] Zhang Z 2000 *IEEE Trans. Pattern Anal. Mach. Intell.* **22** 1330–4
- [29] Himpel M, Buttensch on B and Melzer A 2011 *Rev. Sci. Instrum.* **82** 053706
- [30] Kalman R E 1960 *Trans. ASME, J. Basic Eng.* **82** 35–45
- [31] Wiener N 1964 *Extrapolation, Interpolation and Smoothing of Stationary Time Series* (Cambridge, MA: MIT Press)