

Enhanced PeakCNN: Peak detection with partial pre-knowledge from IPR

Philipp Godbersen^{1*}, Daniel Schanz¹, Andreas Schröder^{1,2}

¹ German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Göttingen, Germany

² Brandenburg Technical University, Institute of Transport Technology, Cottbus, Germany

* Philipp.Godbersen@dlr.de

Abstract

Peak detection, the identification and localization of particle images, is an important component in particle tracking methods. This can be a challenging task due to overlapping particle images and various methods exist to tackle this problem including our CNN based approach (Godbersen et al., 2024, PeakCNN). Peak detection is an important component of 3d particle reconstruction methods such as Iterative Particle reconstruction (Wieneke, 2012, IPR) and ultimately LPT approaches such as Shake-the-Box (Schanz et al., 2016, STB). A key improvement of IPR is its iterative nature. Particle images that have already been found in a prior iteration can be removed from the measurement image leading to an easier peak detection problem in the current iteration on this so called residual image. However, in real-world experiments, this removal generally is not perfect due to various factors such as slight positioning errors or variations in the actual particle image size. As a result, residual images frequently contain artifacts that can complicate peak detection, potentially leading to false positives or obscuring other particles. In this work we propose an extended PeakCNN approach that fully avoids the need of using residual images. Instead of explicitly "burning" the knowledge of already found peaks into a residual image which is irreversible, we provide the network with the original measurement image and provide the knowledge about already found peaks to the network as an additional separate input. The training process then allows the network to make its own decisions in how to best utilize this prior information to find the remaining particle peaks. This results in better peak detection quality.

1 Introduction

Peak detection is one of the first steps in particle tracking approaches and can thus have a large impact on particle reconstruction quality. Various algorithms exist for this, see (Dabiri and Pecora, 2019) for an overview, which typically look for local maxima in the image intensity and extract peak positions by fitting a Gaussian or approximation thereof. Detection of peaks can be challenging in dense particle clouds due to the strong overlaps of particle images. Particle reconstruction algorithms such as IPR are designed with an imperfect peak detection in mind but can achieve better results if given more accurate and complete inputs. An IPR starts out with an initial peak detection on the original camera image. A sufficient number of peaks must be found at this stage to successfully triangulate at least some particles in the measurement domain. These found particles are then backprojected onto the camera images and subtracted to form so called residual images. The residual images are generated through the projection of known 3D particle positions onto the 2D plane using the camera model and subtracting an estimated particle image at those positions from the measurement image. The estimated particle image is computed via an empirical point spread function (OTF) ((Schanz et al., 2012)) obtained during camera calibration, which ideally should remove the particle image entirely from the measurement image in the subtraction. As already known particles have been subtracted from the residual image the effective particle image density has been decreased and the peak detection algorithm may then find additional particles that had been previously hidden in overlap situations in the original image during following IPR iterations. An example of a residual image is shown in fig 1. Even though this approach relaxes the demands on the peak detection somewhat its performance is still of interest. Sufficient and well enough positioned peaks must be found to enable triangulation of some 3d particles without producing too much false or "ghost" particles. The less accurate the peak positioning is, the larger the allowed triangulation errors have to be set increasing the prevalence of ghosts. Likewise it is not sufficient to detect the peak of a particle on one camera alone. A peak for a given particle must be found on multiple cameras in

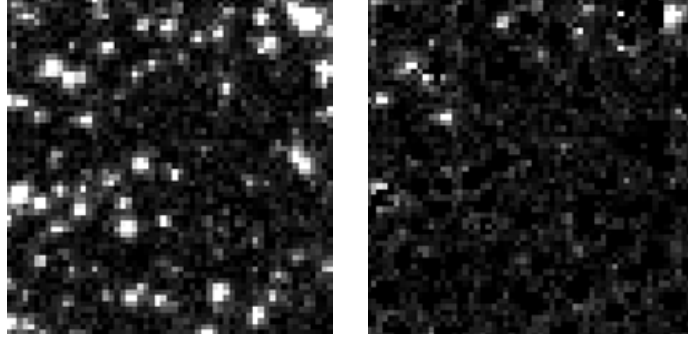


Figure 1: Cutout of particle image example: (left) The original measurement image. (right) The residual image after many found particles have been subtracted from the measurement image.

order to enable triangulation. Due to these factors the quality of peak detection is still relevant even in advanced algorithms such as IPR which motivated our development of the neural network based detector PeakCNN.

2 PeakCNN concept

The original PeakCNN (Godbersen et al., 2024) consists of a single stage Convolutional Neural Network approach. The model is trained, ideally on the same optical setup, and can then identify peak positions in images. A key idea in order to use a single CNN for this task is the mapping of the list of particle positions into an internal representation that can be understood by the CNN architecture which is otherwise not well suited for the variable number of outputs required. Peak positions are discretized onto the image pixel grid and the task interpreted as a joint classification and regression problem in each pixel. For each pixel we have a binary variable corresponding to whether there is a peak in the respective pixel or not. We then have two additional continuous variables in each pixel that correspond to the subpixel offset positions of the peak within the pixel if one is present. This concept is illustrated in fig. 2 and results in a mapping of the peak position list to a three channel image that forms the internal representation of the model. This gives an internal representation of peak locations in an image that can be utilized by a Convolutional Neural Network as the size stays the same no matter how many peaks there are in the image. This internal representation can be unpacked again into a list of peak coordinates by obtaining the coordinates of all pixels that were marked and modifying them by their respective subpixel offsets. A more detailed explanation of the architecture and simultaneous classification and regression loss design as well as training strategies is provided in the original paper (Godbersen et al., 2024).

The training of the model consists of obtaining labeled training data in the form of images together with a list of the peak positions. The position labels are transformed into the internal representation and the corresponding image is shown to the model. The output of the model is then compared with the labeled target data and a loss value computed. Model parameters are then modified in order to minimize this loss using a form of stochastic gradient descent.

Once the model has been trained this way it can then be applied to new images to obtain the output in the internal representation which is then unpacked to a list of peak positions in the image. This approach works very well and allows peaks to be discovered even in very dense images. Such a PeakCNN detector can then be utilized in an IPR processing pipeline where the improved detection performance pushes the limit of seeding densities which can be handled towards higher than possible before.

In these tests it became clear that embedding the PeakCNN in such an iterative processing required changes to the training procedure. Only in the first iteration does the peak detector see an original camera image. In all further iterations it is applied to the residual image formed by removing particle images that already have been successfully reconstructed. These are not simply images at a lower particle concentration as the removal process is not perfect introducing some image artefacts. For the best performance of the PeakCNN such imperfect residual image should also be introduced during the training process.

This is easily implemented as an image augmentation approach during training. For each of the original labeled images we also generate a set of residual images at several completion percentages. This gives a set of training images where 100%, 50%, 10%, or 1% of the original particles remain. We create such a residual image by starting from the original image together with a list of ground truth 3D positions of the particles and their intensities. We then designate the desired percentage of the particles as previously found particles and move them into a different list. Pretending these particles to have been found by some prior IPR iteration we add artificial random errors of 0.1 px standard deviation to their

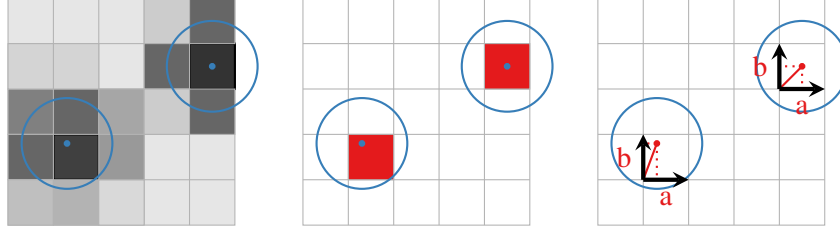


Figure 2: Internal representation of peak positions used by the model

positions. These "found" particles are then used to render a residual image just as would be done in a real IPR processing. The actual labeled training data is then the residual image together with the list of the other set of particles that still need to be found in it. With this image augmentation approach each labeled training image results in four images at various residual levels and thus ensures the model does see a variety of residual images and their artifacts during training.

3 Residual free PeakCNN enhancement

This approach quite successfully extends IPR to even higher seeding densities than with the conventional peak detectors used before (see evaluation in (Godbersen et al., 2024)). But the flexibility of the neural network based approach also opens up new opportunities to push the capabilities of IPR even further by feeding in additional information from the iterations into the network. One such idea we wish to present here is the move away from the use of residual images in IPR. The residual images are generated solely for the benefit of peak detection by iteratively reducing the apparent particle image density leading to easier detection of the remaining particles. The downside of this approach is the creation of image artefacts as particles can generally not be stamped out perfectly from an image either leaving some traces of themselves behind or removing more than necessary thereby damaging neighboring or overlapping particle images. Such damaged particle images may then not be found as accurately or even at all. This effect of damaged particle images in the residual is addressed by PeakCNN somewhat due to the presence of such images in the training data. The model can learn to handle such situations to some extent but it will never be able to do so perfectly as the residual image generation is a destructive operation. Information is lost and can not be recovered anymore.

To avoid this the modification of PeakCNN proposed here avoids actually forming a residual image. The model is always shown the original image so the undamaged information is fully available to it. Instead we pass the information about already found peaks to the model directly. This list of peaks is converted into the internal representation and simply also fed into the model alongside the image data. This forms an implicit residual image as all the steps necessary to create one are taken just up until actually burning the information into the image data.

Only limited changes to the PeakCNN architecture are needed to enable this approach. An additional input is created in form of the three channel known peaks data in internal representation as shown in fig. 3. The rest of the network is left the same although there is likely room for further improvements by adjusting the number of layers in the model. The only other necessary change is in the training data generation where the image augmentation process needs to be adjusted. When generating the labeled data at the various residual density levels the information is now not burned into the image data but provided separately as the additional known peaks data input using a modified internal representation format. We slightly modify the internal representation format used by the original PeakCNN in order to better handle very close particles. The original approach was unable to handle the rare case of two peaks occupying the same pixel as at most one could be addressed with any prior data being overwritten. To be able to represent such cases we modify the underlying algorithm to possibly also consider empty neighboring pixels. If the target pixel is already occupied we attempt to anchor the peak in a neighboring empty pixel and adjust the subpixel offset to match as shown in fig. 3. This way even multiple peaks falling into the same pixel can be addressed by the network internal representation.

4 Test case

We evaluate the proposed approach on a synthetic test case similar to "Case C" in the 1st LPT-Challenge (Sciacchitano et al., 2021) also used in the original PeakCNN paper. For the present study

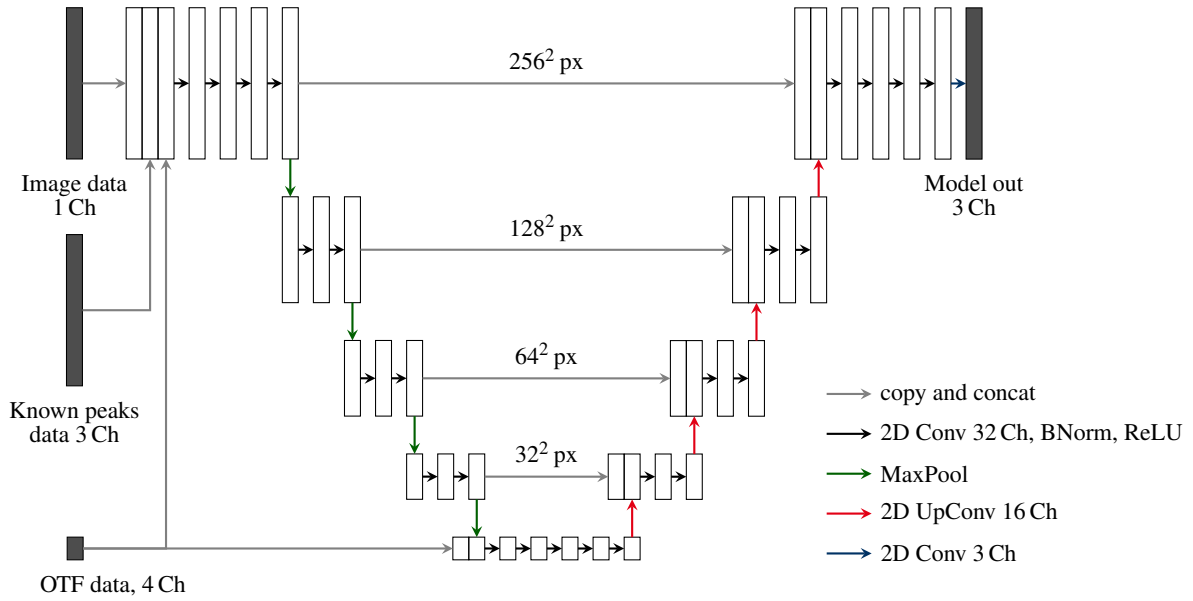


Figure 3: Architecture of the extended PeakCNN model now accepting information of already known peaks. The model then only needs to detect those peaks that are not known yet

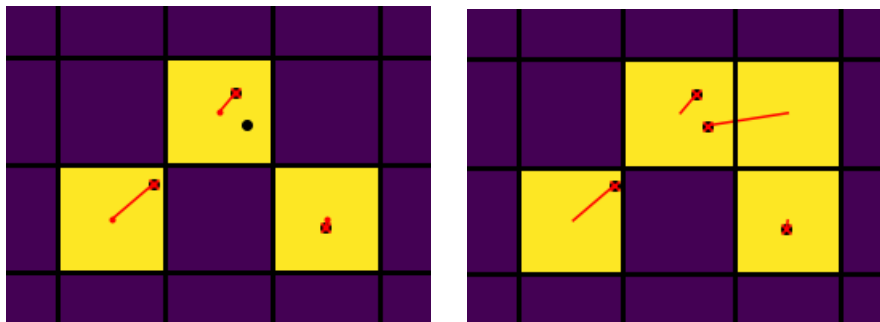


Figure 4: Modification of internal representation to handle multiple peaks within one pixel. Peaks drawn as black circles, subpixel offsets as red line starting from pixel center. (Left): With the prior method only three pixels have been activated since two of the peaks fall into the same pixel. (Right): The modified approach fixes this by anchoring the additional peak in a neighboring pixel.

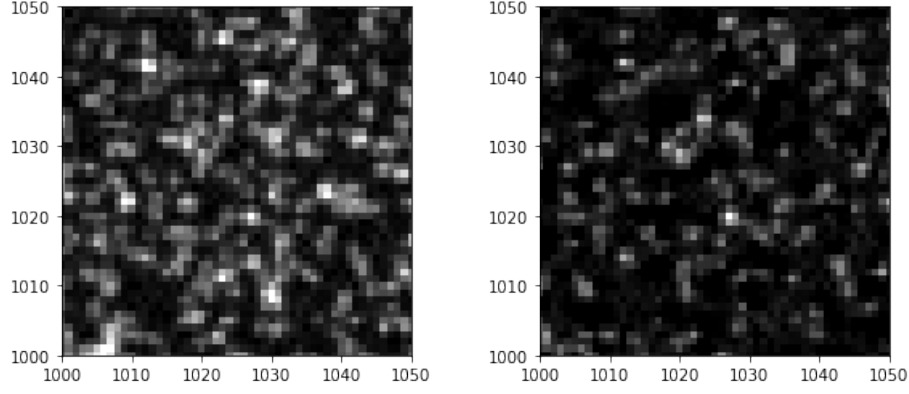


Figure 5: Cutout of images from the test case. (Left): The original synthetic measurement image at 0.15 ppp. (Right): The residual image used for evaluation with half of the peaks subtracted from the image.

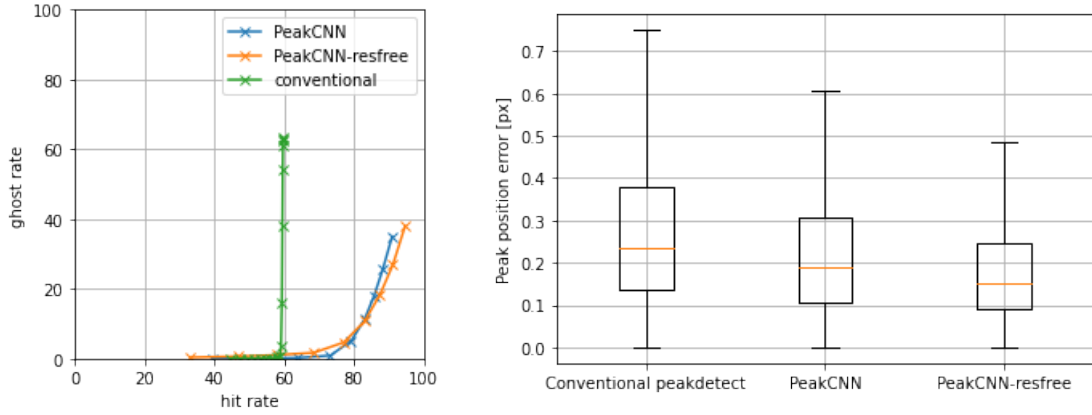


Figure 6: Evaluation of peak detectors on the synthetic test case: (Left) Found and False peak rates obtained by varying detector aggressiveness. (Right) Position errors of found peaks for the three methods.

we desire a setup that allows to create residual images which show some remaining artefacts from imperfectly removed particles. In the original version this case used variations in particle intensity but had identical particle image shapes throughout the volume. Synthetic data with perfectly uniform particle image shapes makes it easy to exactly subtract predicted peaks leading to unrealistically clean residuals. The now added particle shape noise leads to more realistic residual images as particles are not always exactly stamped out and some image artefacts remain. We create such an residual image by first creating an original synthetic experiment image at 0.15 ppp giving us the image together with a list of ground truth 3D positions of the particles and their intensities. We then designate half of the particles as previously found particles and move them into a different list. Pretending these particles to have been found by some prior IPR iteration we add artificial random errors of 0.1 px standard deviation to their positions. These "found" particles are then used to render a residual image just as would be done in a real IPR processing. The actual test case for this evaluation is then the residual image together with the list of the other half of particles that have yet to be found in the image. For evaluations of the new residual free variant we utilize the original image and provide the model with the list of found particles directly instead of using them to create the residual image.

We apply three different peak detectors to this test case. The first two are the conventional peak detector and PeakCNN architecture from the original paper now compared to the third detector in form of the residual free PeakCNN variant. For each of the detectors a key parameter modifying its aggressiveness is varied. For the conventional detector this is the minimum peak height or intensity required for a peak varied between 10 and 200 counts. For the two PeakCNN variants this is the certainty threshold of the binary classifier between 0.1 and 0.9. A more detailed description of the metrics discussed here can be found in the PeakCNN paper. The curves showing found and ghost rate show the familiar behavior of the conventional peak detector running into the cliff edge of overlapping particles where decreasing the intensity threshold does to yield any more real peaks, only ghosts. Both PeakCNN detectors are able to discover much more peaks and show a more gradual decline towards increasing ghost rate. The curves for the original and residual free version are very similar, showing nearly the same number of

false particles for a yield of 80%. The main difference can be seen in the peak positioning errors of the correctly found peaks. As expected the conventional detector shows the highest errors as it often can not distinguish between closely overlapping particles. The peak position error of the original PeakCNN is further improved upon by the residual free version. These findings are in line with our original motivation for the residual free peak detector approach. There is not necessarily an improvement in the yield of peaks but their positioning is more accurate as the model is able to work with the original measurement image undamaged by destructive subtraction.

5 Conclusion

The initial evaluation of the residual free PeakCNN concept has shown promising results on synthetic image data. A further reduction in peak positioning error on residual images over the original PeakCNN concept was achieved allowing for tighter bounds on triangulation radii during IPR. This is directly related to the ability to target higher seeding densities where such tight bounds are required to minimize the creation of ghost particles during triangulation. The synthetic test performed here can only be the first step in further investigating the usability and benefits of the approach. Some care has been taken to replicate real world residual images but further investigation should be done on real world images to fully capture all effects. The analysis of the effect on the full IPR is complicated by the strong interdependence between iterations. The state of the residual image presented to a peak detector depends directly on the performance of that detector in the previous iteration. Further investigation along this direction is needed. Beyond the concept investigate here feeding in even more information from the IPR process could be an interesting opportunity. Cost function gradients from the "shaking" of the particles could be communicated to the peak detector this way to potentially allow the model to discern the reliability of previously found peaks.

References

- Dabiri D and Pecora C (2019) *Particle Tracking Velocimetry*. IOP Publishing
- Godbersen P, Schanz D, and Schröder A (2024) Peak-cnn: improved particle image localization using single-stage cnns. *Experiments in Fluids* 65:153
- Schanz D, Gesemann S, and Schröder A (2016) Shake-the-box: Lagrangian particle tracking at high particle image densities. *Experiments in fluids* 57:1–27
- Schanz D, Gesemann S, Schröder A, Wieneke B, and Novara M (2012) Non-uniform optical transfer functions in particle imaging: calibration and application to tomographic reconstruction. *Measurement Science and Technology* 24:024009
- Sciacchitano A, Leclaire B, and Schröder A (2021) Main results of the first lagrangian particle tracking challenge. in *14th International Symposium on Particle Image Velocimetry*
- Wieneke B (2012) Iterative reconstruction of volumetric particle distribution. *Measurement Science and Technology* 24:024008