

PAPER • OPEN ACCESS

Deep learning based instance segmentation of particle streaks and tufts

To cite this article: C Tsalicoglou and T Rösger 2022 *Meas. Sci. Technol.* **33** 114005

View the [article online](#) for updates and enhancements.

You may also like

- [Obstacle detection in dangerous railway track areas by a convolutional neural network](#)
Deqiang He, Kai Li, Yanjun Chen et al.
- [On-line monitoring of multi-component strain development in a tufting needle using optical fibre Bragg grating sensors](#)
Edmon Chehura, Giuseppe Dell'Anno, Tristan Huet et al.
- [Brain tumor segmentation using 3D Mask R-CNN for dynamic susceptibility contrast enhanced perfusion imaging](#)
Jiwoong Jeong, Yang Lei, Shannon Kahn et al.

Deep learning based instance segmentation of particle streaks and tufts

C Tsalicoglou*  and T Rösgen* 

ETH Zurich, Zurich, Switzerland

E-mail: ctsalico@ethz.ch and roesgen@ethz.ch

Received 29 April 2022, revised 6 July 2022

Accepted for publication 10 August 2022

Published 24 August 2022



CrossMark

Abstract

3D particle streak velocimetry (3D-PSV) and surface flow visualization using tufts both require the detection of curve segments, particle streaks or tufts, in images. We propose the use of deep learning based instance segmentation neural networks Mask region-based convolutional neural network (R-CNN) and Cascade Mask R-CNN, trained on fully synthetic data, to accurately identify, segment, and classify streaks and tufts. For 3D-PSV, we use the segmented masks and detected streak endpoints to volumetrically reconstruct flows even when the imaged streaks partly overlap or intersect. In addition, we use Mask R-CNN to segment images of tufts and classify the detected tufts according to their range of motion, thus automating the detection of regions of separated flow while at the same time providing accurate segmentation masks. Finally, we show a successful synthetic-to-real transfer by training only on synthetic data and successfully evaluating real data. The synthetic data generation is particularly suitable for the two presented applications, as the experimental images consist of simple geometric curves or a superposition of curves. Therefore, the proposed networks provide a general framework for instance detection, keypoint detection and classification that can be fine-tuned to the specific experimental application and imaging parameters using synthetic data.

Keywords: instance segmentation, deep learning, particle streak velocimetry, tufts

(Some figures may appear in colour only in the online journal)

1. Introduction

In optical measurement techniques for fluid dynamics, the flow field properties are usually observed through tracers that indicate the flow behavior. These tracers, which can be dyes, particles, excited molecules, smoke, tufts, oil substances, or liquid crystals, are detected in the recorded images, and their apparent properties (shape, displacement, intensity, color) are analyzed to infer information about the flow.

Therefore, the localization and segmentation of tracer objects from images is often an initial task in the processing chain of experimental flow visualization techniques. Image segmentation can be tackled with classical computer vision methods that rely on detecting engineered features, such as edges, clusters of intensity values or colors, known sets of gradients, thresholding, and other algorithms that are often developed or adapted for specific applications [1].

Recently, deep learning based methods for image processing using convolutional neural networks (CNNs) [2] and Transformers [3] have proven to be extremely valuable in detecting and segmenting objects in complex images with a large number of classes. These networks are specialized to perform various tasks such as object detection, classification, and instance, semantic or panoptic segmentation [2–7]. They are often trained on standardized benchmark datasets, such

* Authors to whom any correspondence should be addressed.



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.

as Microsoft COCO [8], Cityscapes [9] or Pascal VOC [10], which contain natural images with 20 to 80 annotated classes. For applications that contain object types not represented in the training dataset, application-specific natural or synthetic annotated images are used to fine-tune the networks' parameters. Therefore, these methods have been employed for a variety of segmentation tasks: for example, in [11], Mask region-based convolutional neural network (R-CNN) [2] is fine-tuned for crop seed phenotyping, and a review of deep learning based segmentation for biological-image analysis is provided in [12].

Deep learning is also finding its way into an increasing number of applications in experimental fluid dynamics. In [13], two CNNs are used to first detect particles in an image obtained with an astigmatic 3D Particle Tracking Velocimetry (3D-PTV) setup and then regress their 3D coordinates from a single image. End-to-end particle image velocimetry (PIV) is performed using a CNN in [14], and in [15] dynamic masking of objects is performed for PIV images, using a convolutional autoencoder. A CNN with a fully connected regression head is also employed for planar particle streak velocimetry (PSV) [16] to regress a representative streak orientation and length from an image patch containing multiple streaks obtained by long-exposure particle imaging.

Here, we evaluate the performance of state-of-the-art CNNs on instance segmentation, keypoint detection, and classification tasks for two different applications in experimental fluid dynamics: volumetric 3D particle streak velocimetry (3D-PSV) and flow visualization using tufts.

3D-PSV is a variant of 3D-PTV, where a longer exposure time is used when recording the tracer particle images, so that the particles' pathlines, called 'streaks', are recorded instead of the 'frozen' particle signatures required in 3D-PTV [17–20]. This method has been researched by [21–25], where different illumination and 3D reconstruction methods are proposed. 3D-PSV allows the use of fewer and lower frame rate cameras than 3D-PTV, it does not require displacement assumptions as those required for tracking, and it can result in fewer reconstruction ambiguities than particle-based methods [25]. However, the 3D reconstruction in volumetric 3D-PSV requires the localization of each individual streak in a set of images obtained from different camera views. As the volume depth and seeding density increase, streak intersections increase, posing a challenge in the accurate localization of all streak instances. Therefore, the segmentation of the streaks acquired for 3D-PSV is an interesting problem to address with deep learning based methods. Related work on streak segmentation is presented in section 2.1.

The second application for which we use deep learning based instance segmentation is flow visualization using tufts. In this application, pieces of string are applied to a model's surface and their motion under an applied flow is recorded to deduce the behavior of the surface flow [26]. As with streaks, tufts produce streak-like features in the recorded images: curve segments that sometimes appear blurred. In contrast to streaks, tufts should not intersect so as to not interfere with their neighbors' motion, and they have a fixed point, so their segmentation is significantly less challenging than that of streaks. However, depending on their shape, recorded in long-exposure

images, tufts can be classified as stationary or fluttering. This classification task is here posed as an instance segmentation task. Related work on tuft segmentation and classification is presented in section 2.2.

The methods used for training the models, conducting the experiments, and post-processing the data are discussed in section 3. The evaluation of the CNNs' performance on the tasks of instance segmentation and keypoint detection is described in section 4.1. In section 4.2, we fine-tune one of the networks that we use for the streaks to detect, segment, and classify instances of tufts as stationary or fluttering. Both networks are trained exclusively on synthetic data, eliminating the need for manual image annotation. The performance on the segmentation, keypoint detection, and classification tasks demonstrates a successful domain shift from synthetic to real data. The accurate segmentation of streak instances enables the use of 3D-PSV at high seeding densities, and the automated segmentation and classification of tufts using neural networks can speed up the processing and increase the accuracy of flow visualization methods using tufts.

2. Related work

2.1. 3D-PSV

In 3D-PSV, a calibrated multi-camera setup is used to record the trajectories of particles seeded in a fluid. A long exposure time results in images of pathlines, or 'streaks', which are then reconstructed in 3D space using the known camera calibration. To reconstruct the 3D streaks, it is necessary to first identify the streak instances in the individual images.

In [25], curved streaks are reconstructed in a volume by simultaneously optimizing the detected streaks' shapes and checking if they are projections of the same 3D curve. In previous works, the segmentation of streaks relies on binarization using a global [21] or local adaptive thresholding using the Otsu method [24], it is based on the detection of local oriented structures [27], or iterative processes using the structure tensor and region growing methods [28]. An ellipse enclosing all the segmented pixels is fitted in [27] to provide an indication of the streak shape based on the size of the fitted ellipse's axes. In [22], after segmentation by thresholding, temporal information is used to improve the streak detection and a spline is fitted through the skeletonized outline of the streak.

During reconstruction, the above methods assume linear segments between the matched corresponding points and work well as long as there are few streak intersections, which effectively limits the maximum allowable seeding density and volume depth. Here, we propose the use of a deep learning based method to segment the individual streaks from the images. The neural network predicts the endpoints and a mask containing the pixel coordinates belonging to each streak instance detected in the image.

2.2. Flow visualization with tufts

Tufts are small pieces of string often used to visualize surface flows and identify regions of flow separation. When recorded

in video or images with a relatively long exposure time, the rapid, unsteady movement of tufts in regions where flow separation occurs results in a blurry appearance, which can serve as an indicator for flow separation [26].

Visually inspecting images of tufted models is still a common way to deduce the flow behavior in the different regions of the examined model, e.g. by identifying reversed or blurry tufts [29–32]. In [33], a quantitative analysis of tufts' behavior was performed, where statistics about each tuft's orientation were used to derive information about the local flow state. Edge detection and the Hough transform can be used to fit a line through the detected tufts and deduce their orientation, as in [34], though this method becomes inaccurate for highly curved tufts. Other methods detect the tufts' shapes by recognizing one end of the tuft and progressing pixel-by-pixel along the high-intensity ridge that defines the tuft's centerline [35].

Non-moving tufts can be segmented from images using filtering and thresholding. If a time series of images is provided, background subtraction can improve the thresholding performance, though obtaining a background image is not always possible if, e.g. the model is moving to sample different angles of attack. Visual inspection can be used to assess the occurrence of blurred ends, though this is limited by the human capacity to process and label the different flow regions accurately. Therefore, training a neural network to detect tufts and label them according to their range of movement can help (a) in the segmentation of tufts from a series of image frames and (b) in the classification of the different flow regions. We fine-tune a network developed for instance segmentation to predict the tuft masks and classify them as residing in a region of attached or separated flow.

2.3. Instance segmentation

Instance segmentation is the task of identifying individual instances of a class in an image and predicting a segmentation mask for each instance. As each instance is detected individually, and a mask is derived per instance, networks dealing with instance segmentation can handle overlapping objects. In contrast, another method that can identify distinct instances, panoptic segmentation, assigns an instance ID and class to each image pixel and is unable to deal with overlapping objects [36].

Two of the most commonly used CNNs that can perform instance segmentation are Mask R-CNN [2] and Cascade Mask R-CNN [5]. Mask R-CNN is a two-stage R-CNN that first proposes regions of interest (RoI) with a fully convolutional region proposal network (RPN) [4] and then regresses a bounding box and class, and predicts a mask with a fully convolutional branch for each RoI (figure 1). The two-stage detector works on a feature map extracted from a convolutional 'backbone', typically a ResNet [37] or ResNeXt [38]. To extract and process features at different scales, Mask R-CNN optionally employs a feature pyramid network (FPN) [39].

The region-based CNN architecture was introduced by [40] to enable translation-invariant object detection by first detecting RoIs and then performing bounding box and class regression for each RoI. At the time, the region proposals were still

performed with classical methods for object recognition [41], and features were extracted from each RoI individually. Building on R-CNN, Fast R-CNN [42] increased the speed and performance of the original R-CNN by introducing a multi-task loss to minimize the class and bounding box prediction loss jointly and moved the feature map extraction step to be performed on the whole image instead of on each proposed region. The network's performance was increased significantly in Faster R-CNN [4], when a convolutional RPN that shares features with the detection network was introduced, resulting in a two-stage detector that learns both how to propose RoI and provide refined bounding box coordinates and classification predictions. Mask R-CNN used the same architecture as Faster R-CNN, introduced a new method to improve the alignment of RoIs to the input, and added a mask prediction head in parallel to the class and bounding box prediction to provide a binary mask for each RoI. In recent years, improvements to Mask R-CNN performance have come mainly from exploiting new backbone architectures, such as transformers, as demonstrated in [43].

Cascade Mask R-CNN [5] uses a sequence of detectors based on the Mask R-CNN architecture and trained on increasing intersection over union (IoU) thresholds to make use of close false positives during training and not overfit on the very high-confidence positive samples. While this network is slower during inference compared to Mask R-CNN, it achieves higher precision and recall values on the reported cases.

Keypoint detectors are often trained for human pose estimation, typically estimating keypoints for the joints, eyes, nose, etc. Keypoint R-CNN [2] is an extension of Mask R-CNN that treats keypoints as a single-pixel mask and follows the same architecture as Mask R-CNN to detect this mask. As Keypoint R-CNN is an additional branch parallel to the bounding box, class and mask prediction branches of Mask R-CNN, these features are predicted jointly for each instance, thus eliminating the need to assign keypoints to masks at a later step.

To circumvent the need for obtaining and annotating large numbers of natural images, fine-tuning these networks for new applications often relies on fully synthetic data [13], synthetic data generated from augmentations of a pool of real data [11] or real data augmented with synthetic data generated using generative adversarial networks [44, 45]. However, fully synthetic data is often not representative of the complexity of real data, and the shift from the synthetic to the real domain can result in performance loss. Domain randomization [46] is a common method to bridge the simulation-to-real world gap, and it is based on the idea that by introducing variability in the synthetic data, the shift to the real data domain 'may appear to the model as just another variation' [46].

3. Methods

In the following, we present the synthetic data generation, training, and evaluation process for Mask R-CNN and Cascade Mask R-CNN on the tasks of instance segmentation, keypoint detection, and classification for two different applications of fluid flow visualization: 3D-PSV and flow visualization

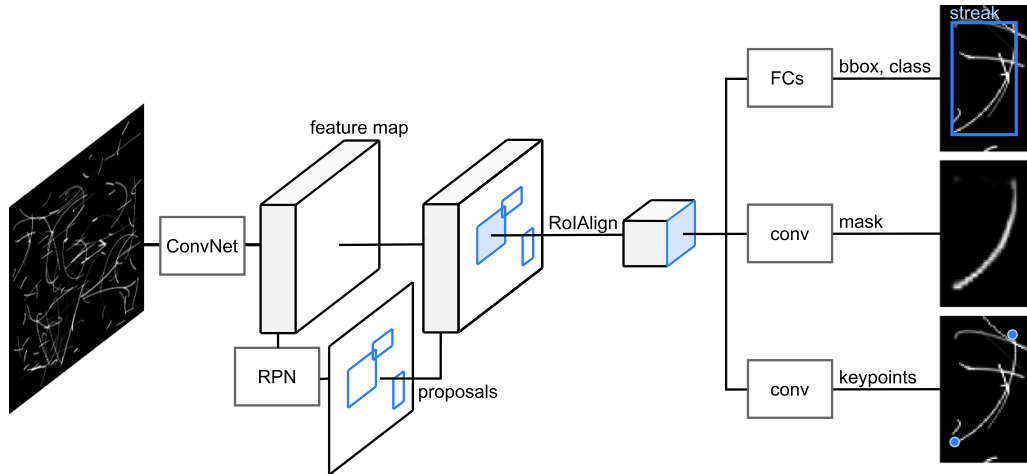


Figure 1. Outline of Mask R-CNN [2]. Here, ConvNet stands for the convolutional backbone, FCs for fully connected layers and ‘conv’ for a convolutional branch.

using tufts. We validate and test the methods on synthetic data and demonstrate their applicability on experimental data in section 4.

3.1. Networks

We use Mask R-CNN [2] and Cascade Mask R-CNN [5] on the Detectron2 platform [47]. For the streak detection we include a Keypoint R-CNN head to multi-task instance segmentation and keypoint detection. The settings used during training and inference are reported in appendix C.

The keypoint detection head is trained to detect two keypoints per instance, representing a streak’s endpoints. During training, the cost function compares the heatmaps generated by the predicted keypoints and the ground truth keypoints to calculate the cross-entropy loss.

In our case, the two endpoints are interchangeable. Therefore, we modify the input to the keypoint loss function by first rearranging the predictions relative to the ground truth before calculating the keypoint loss, so that the sum of the Euclidean distance of the two keypoint proposals to the ground truth is minimized.

3.2. 3D-PSV

3.2.1. Training data. Ground truth data of images containing streaks and the corresponding bounding boxes, masks, and keypoints are required to train the networks. As a typical image for 3D-PSV might consist of 10^3 – 10^4 individual streak instances, annotating the data manually would be challenging. Additionally, streak images of many different flows with different acquisition and experimental settings would have to be obtained to cover the range of different streak appearances. On the other hand, using synthetic data for particle images has a long history in PIV, as they allow the reliable evaluation of reconstruction algorithms [48], and are easy to modify to simulate optical aberrations, particle sizes, and illumination intensities.

Therefore, we use exclusively synthetic ground truth data during training and use the principle of domain randomization to enable generalization to real data by varying the generation parameters of the synthetic streaks. The streaks are generated as particles that follow a path along a given conic section segment, initialized at random locations in the images. The conic section segment parameters, axes, orientation, segment length, width, and brightness are chosen randomly from a given uniform distribution. The curve segments are generated from concatenated particles that follow the conic section path, using the best practice method described for particle image generation for PIV [48] and summing the particle intensities at each time step. For each training dataset, we generate 10 000 images with random initializations.

We use images of size 250×250 px². This choice allows the longest required streaks to fit in the training images while keeping the number of streaks per image low enough for our memory constraints, with 118 streaks per image on average. Additionally, training on smaller images produces better results on the keypoint detection accuracy when we scale up the images to the default 800×800 px², as the resolution around the keypoints increases. The performance for different image sizes is reported in tables 1 and 2.

Finally, we use overlapping masks so that each instance’s mask is a set of connected pixels despite possible overlaps and intersections with other instances. An intensity threshold defines each mask’s extent, and the mask is saved as a set of pixel coordinates and their corresponding intensity. The endpoints of each streak are the known start and end positions of the traveling particle.

3.2.2. Experimental setup. We acquire images of a vortex ring in air with three Photron AX-100 high-speed cameras with image resolution of 1024×1024 px² and pixel size of 20×20 μm². All cameras use Nikon Micro-NIKKOR 55 mm 1:2.8 lenses with the aperture set to $f_{\#} = 11$. The magnification factor is $M = 0.06$. The particle images are acquired at 1000 fps with an exposure time of 1 ms.

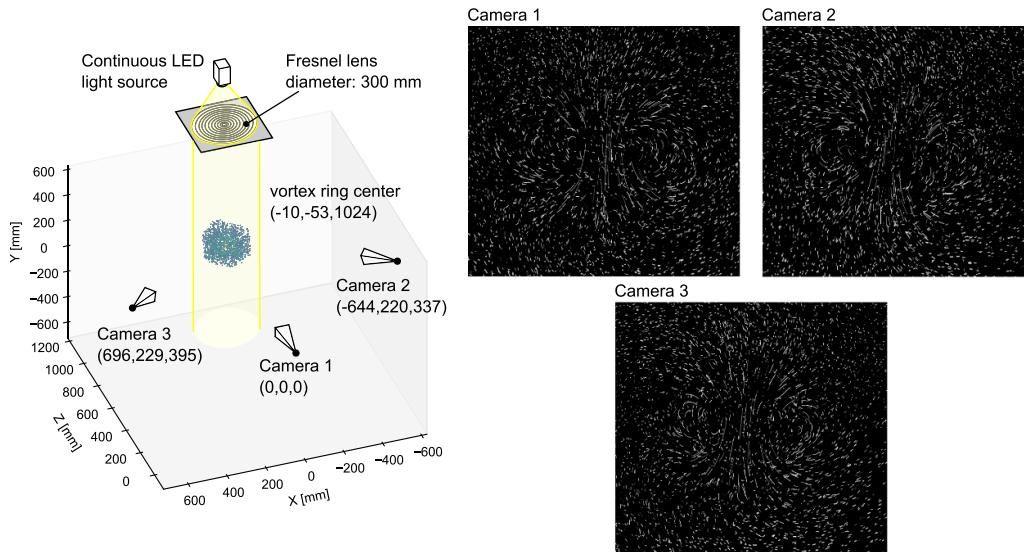


Figure 2. Sketch of the experimental setup for the acquisition of vortex ring images and view from the three cameras. The camera coordinates and the location of the vortex ring's center are given in mm. The coordinate system is defined by the position and orientation of Camera 1. The grayscale images are cropped around the region of interest, with dimensions $650 \times 600 \text{ px}^2$.

The center of the recorded vortex ring is at a distance of 1024 mm from the center of the first camera. The other two cameras are arranged as shown in figure 2. The field of view covers approximately $350 \times 350 \text{ mm}^2$ but for the presented results only the region shown in figure 2 is processed, which covers an area of $650 \times 600 \text{ px}^2$, corresponding to a field of view of $224 \times 218 \text{ mm}^2$.

The measurement volume is illuminated by a continuous 50 W white LED light source, and the light is collimated with a Fresnel lens with a diameter and focal length of 300 mm. The tracers, which are helium-filled soap bubbles (HFSBs), are generated using an in-house built bubble generator [22] with a modified nozzle based on [49]. The HFSB size is approximately $300 \mu\text{m}$.

The particle images are filtered and thresholded to the same intensity value used to threshold the images of the training dataset, and they are subsequently summed to obtain images of particle streaks. This acquisition method allows an evaluation of the same data with a 3D-PTV algorithm, as in [25]. For the results presented in section 4.1.3, 60 frames are summed, resulting in an effective exposure time of 60 ms.

The cameras are calibrated using the pinhole camera model with radial and tangential distortion correction. A custom 2D target with regularly arranged dots is moved within the field of view to obtain 300 images of the target at different positions and orientations, covering the field of view of the three cameras. The fit error obtained by the calibration is 0.018 px. For the evaluation with the 3D-PTV type software, a self-calibration is performed, but this is not used for the streak evaluation. The fit error after the self-calibration is 0.032 px.

3.2.3. Post-processing. Instances are inferred from the experimental images using the trained networks. For the inference, we resize the images by the same ratio of 800:250 used during training (table C1). Given the predicted keypoints,

multi-camera endpoint matching is performed across the views using a tolerance of 2 px for the epipolar constraint. Subsequently, we perform 2-view streak matching for all image pairs and transfer the matched endpoints to the remaining camera view. If both transferred endpoints overlap with a predicted mask, and the path connecting them has a large IoU with this mask, we assign the transferred endpoints to the predicted mask and consider the new triplet of streaks a successful match.

For the reconstruction of the experimental data (section 4.1.3) we subsequently perform conic section segment matching [25] for all matched triplets of streaks: a multi-view correspondence criterion is imposed to optimize conic section fits through the predicted masks while ensuring that the fits are 3D-consistent.

3.2.4. Baseline method for linear streaks. Our baseline method is used to compare the performance of the deep learning based segmentation to a classical segmentation method. The images are thresholded to generate enclosures of individual or intersecting streaks. For each enclosure, the probabilistic Hough transform [50] returns lines that fit through the enclosure points, which are then grouped in clusters [51] to derive the most dominant line orientations in the enclosure. The endpoints of these lines are then refined and used as endpoints for the further processing steps and masks are generated from the fitted lines.

3.3. Tufts

3.3.1. Training data. We generate synthetic data for tufts by calculating the displacements for the first four eigenmodes of a cantilever beam and superposing the first mode and one more, chosen randomly, with different amplitude ratios to obtain various shapes of moving tufts. The fluttering is simulated by

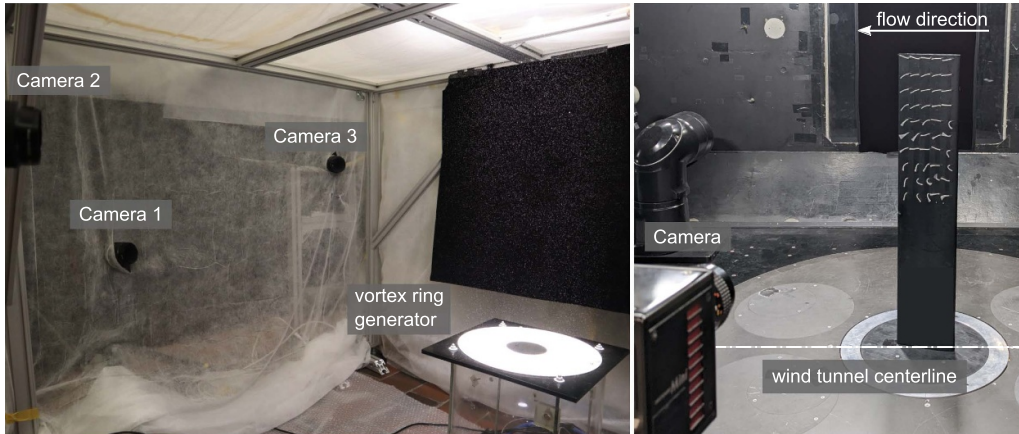


Figure 3. Experimental setup for the acquisition of vortex ring images (left) and surface flow visualization using tufts (right).

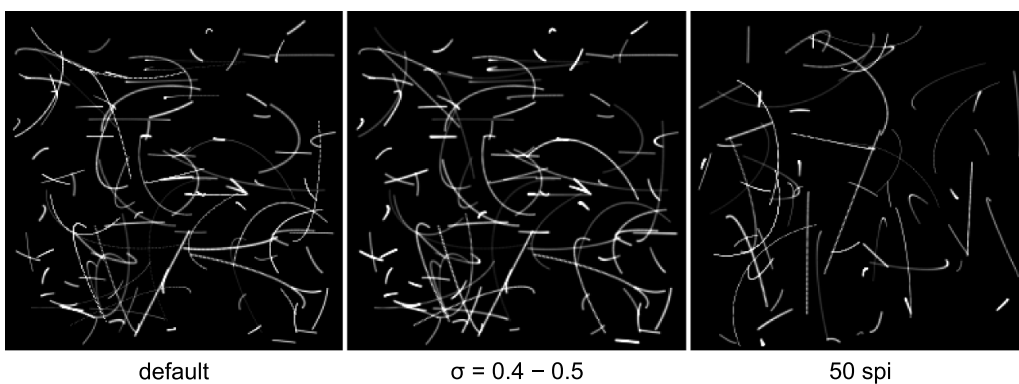


Figure 4. Examples of images used for the different training cases. The default settings are described in section 3.2. Images for the case ' $\sigma = 0.4 - 0.5$ ' contain streaks generated with a reduced range of variable thickness. For the case '50 spi', the training images contain 50 streaks per image on average, approximately half as many as in the 'default' case.

calculating the cantilever beam's position for different amplitudes, scaling it to keep the arc length of the resulting curve constant, and averaging the resulting curve intensities.

The tufts' lengths, intensities, and positions on the image are varied within pre-defined ranges but, contrary to the streaks, the synthetic tufts do not overlap, as they also should not overlap or cross during experiments. The masks are obtained by thresholding and the classes are assigned based on the range of motion of each simulated tuft. We use images of size $250 \times 250 \text{ px}^2$.

Finally, as tufts are usually placed on a surface, it is common that some surface reflections, tapes and other objects might be visible in the images. Therefore, we introduce random background shapes and noise in the images to enable accurate segmentation despite background objects (figure 8). The network settings for the training and evaluation are provided in table C2.

3.3.2. Experimental setup. We use a NACA0012 [52] airfoil with a chord of 0.2 m and span length of 1 m for our tuft visualization tests. The wing is installed vertically in the subsonic wind tunnel of the Institute of Fluid Dynamics at ETH Zurich (figure 3), at the center of the 3 m wide, 2.1 m high test section. We test at a free stream velocity of 6.9 m s^{-1} , resulting in a

Reynolds number of $Re = 98000$, and set the wing to angles of attack of $8^\circ - 12.4^\circ$ to record states of attached and separated flow.

Fifty tufts, made of white yarn of $\approx 2 \text{ mm}$ thickness and a length of 25–30 mm, are applied to the model. The images are acquired with a Photron AX100 high-speed camera at 500 fps and exposure time of 1 ms to obtain the instantaneous tuft positions. The time series of images is then averaged to obtain simulated long-exposure images. The camera is placed at a distance of 1.5 m from the model, at the edge of the test section.

4. Results and discussion

4.1. Streaks

4.1.1. Validation. We validate the performance of Mask R-CNN and Cascade Mask R-CNN on streak detection, segmentation and keypoint detection for training with different backbones, and image size and complexity. Our default settings use a ResNet-101 as backbone, and networks trained on the default training dataset (figure 4), with overlapping masks. Two more datasets are used to assess the effect of training with a smaller variation in streak thickness (' $\sigma = 0.4 - 0.5$ ') and fewer streaks per image ('50 spi'). The datasets of 10 000

Table 1. Instance segmentation mean precision (P_{50}) and recall (R_{50}), and average precision (AP_{50}) for masks with $\text{IoU} > 0.5$: n/o: non-overlapping masks; min 500: minimum image size set to 500, images rescaled to $500 \times 500 \text{ px}^2$ during training; 50 spi: trained with 50 streaks per image on average; $\sigma = 0.4-0.5$: trained with data with reduced streak thickness range.

Network	P_{50}	R_{50}	AP_{50}
Mask R-CNN	79.8	80.1	81.7
Cascade M R-CNN	84.4	81.7	84.6
Cascade M R-CNN, n/o	80.1	77.6	80.2
Cascade M R-CNN, ResNet-50	83.2	80.7	83.9
Cascade M R-CNN, min 500	84.5	79.1	81.7
Cascade M R-CNN, 50 spi	81.7	78.8	82.3
Cascade M R-CNN, $\sigma = 0.4-0.5$	81.2	78.2	82.3

Table 2. Keypoint detection: number of detected streaks with mask $\text{IoU} > 0.5$, whose endpoints are both closer than 1 px (d_1) and 2 px (d_2) from the ground truth endpoints, as percentage of the number of ground truth streaks.

Network	d_1	d_2
Mask R-CNN	64.3	70.7
Cascade M R-CNN	65.9	74.8
Cascade M R-CNN, n/o	64.8	72.6
Cascade M R-CNN, ResNet-50	65.5	74.0
Cascade M R-CNN, min 500	61.8	70.5
Cascade M R-CNN, 50 spi	64.1	71.8
Cascade M R-CNN, $\sigma = 0.4-0.5$	61.3	69.5

images are split into 9900 images for training and 100 images for validation and all models are trained for 15 000 iterations with a mini-batch size of four images. For the results of tables 1 and 2 we use the validation data from the hardest ('default') dataset with 118 streaks per image on average.

A commonly used metric to evaluate segmentation performance is the IoU between the predicted and ground truth masks, that describes by how much the masks overlap relative to their size and alignment (appendix A). For a given IoU threshold, the recall and precision of the network for a given dataset are calculated, with recall describing how many of the ground truth instances were identified correctly and precision telling how many of the predicted instances were matched to ground truth instances for the given IoU threshold (appendix A).

We report the mean precision, P_{50} , and recall, R_{50} , for bounding box and mask detections with $\text{IoU} > 0.5$ (table 1). The average precision, AP_{50} , is also evaluated so that the models' performance can be compared to benchmarks found in literature. The AP_{50} metric is commonly used in object detection, and includes an evaluation of different score thresholds by measuring the area under the precision-recall curve [10]. For P_{50} and R_{50} , we use a constant score threshold of 0.5. Finally, to evaluate the performance on keypoint detection, we report the number of streaks whose endpoints are both within 1 px (d_1) and 2 px (d_2) from the matched ground truth (table 2).

Cascade Mask R-CNN with overlapping masks and trained with our default settings described above exhibits the best

performance of the examined cases. The trained Mask R-CNN has lower precision than any of the other test cases, and using Cascade Mask R-CNN with non-overlapping masks (case 'n/o') results in the lowest recall of all examined cases. Further, Cascade Mask R-CNN with our default settings and a ResNet-50 backbone only slightly underperforms both in instance segmentation and keypoint detection compared to ResNet-101. A network trained on smaller images (case 'min 500') performs well on instance segmentation, but the keypoint detection deteriorates significantly (table 2). Introducing fewer streaks per image in the training dataset (case '50 spi') results in slower learning and the network probably does not learn sufficiently how to detect difficult intersections, resulting in lower precision and recall on the validation dataset. Finally, the importance of introducing variability in the training data is clear in the results of case ' $\sigma = 0.4-0.5$ ', which performs significantly worse than the default Cascade Mask R-CNN case.

4.1.2. Testing: synthetic flow field data. Synthetic images and ground truth data of streaks for the flow field describing Hill's spherical vortex [53] are generated in a volume and projected to three camera views. Streaks and their endpoints are detected from the images and reconstructed as described in section 3.2.3 for an end-to-end evaluation of the instance segmentation and keypoint detection on images of a realistic flow field. We perform the evaluation for Mask R-CNN, Cascade Mask R-CNN and Cascade Mask R-CNN with non-overlapping masks. The same synthetic flow field data is used to detect lines with the classical line detection method described in section 3.2.4, as the streaks are nearly linear in the synthetic data.

Three seeding densities of 1000, 2000 and 3000 streaks per image are evaluated (figure 5). The image size is $1024 \times 1024 \text{ px}^2$ and the spherical vortex and surrounding flow are within a volume of $300 \times 300 \times 300 \text{ mm}^3$ at a distance of 1 m from the cameras. For each seeding density we evaluate P_{50} and R_{50} on 10 datasets consisting of three views each where the particles are initialized at different random positions and assigned random intensities and thicknesses (table 3).

Cascade Mask R-CNN with the default settings has the best performance across all metrics, as with the validation dataset. All models perform better than the baseline, which performs particularly poorly in terms of recall, which is detrimental when performing 3D reconstruction. Indeed, as faulty matches can often be eliminated through the multi-view constraints, high recall is more desirable than high precision. Recall can be increased at the expense of reduced precision by reducing the strictness of non-maximum suppression for the RPN proposals (case 'more proposals' in table 3). However, faulty multi-view matches will increase when the instances are detected with lower precision, also causing a drop in the overall 3D reconstruction precision.

Following detection, the streaks are reconstructed using the methods described in section 3.2.3 and the mean reconstruction precision (P_{3D}) and recall (R_{3D}) are evaluated (table 4). The final step of conic section reconstruction is

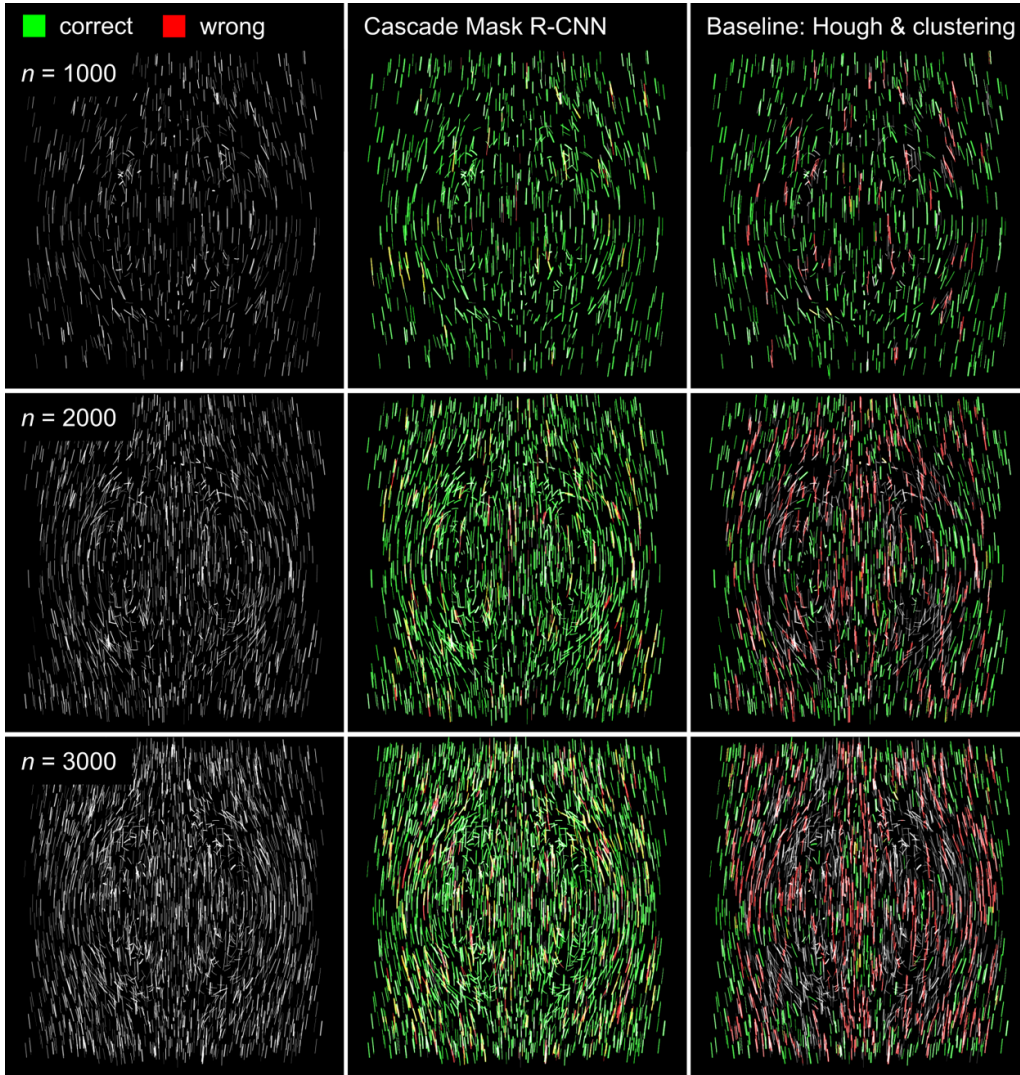


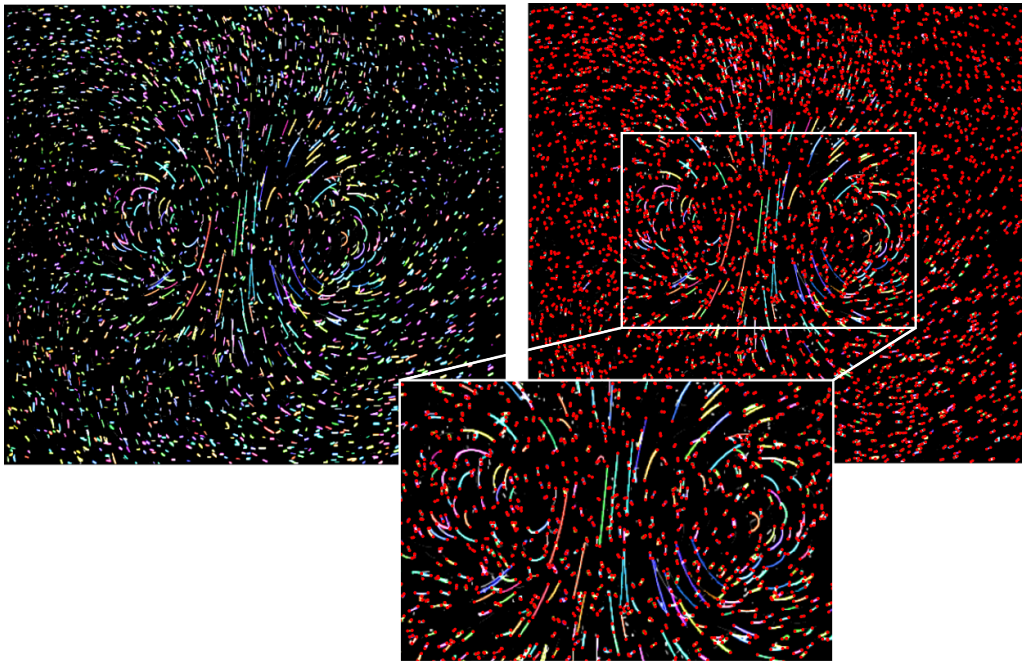
Figure 5. Synthetic images of Hill’s spherical vortex for different numbers of streaks per image, n , and predicted masks with Cascade Mask R-CNN and with the classical baseline method. Correct mask predictions are shown in green and wrong in red. The IoU threshold is set to 0.5.

Table 3. Instance segmentation mean precision (P_{50}) and recall (R_{50}) for masks with $\text{IoU} > 0.5$ and number of detected streaks whose endpoints are both closer than 1 px (d_1) and 2 px (d_2) from the ground truth endpoints, as percentage of the number of ground truth streaks. Evaluation for images with n streaks per image.

Network	n	P_{50}	R_{50}	d_1	d_2
Mask R-CNN	1000	90.3	89.7	64.8	72.2
	2000	84.2	78.0	53.5	59.6
	3000	78.8	64.7	42.0	47.0
Cascade M R-CNN	1000	93.1	89.9	67.8	76.1
	2000	88.3	78.3	55.7	63.2
	3000	84.5	65.7	44.7	50.9
Cascade M R-CNN n/o	1000	90.8	87.0	66.2	74.6
	2000	84.1	73.1	54.2	62.2
	3000	77.8	58.8	42.7	49.4
Cascade M R-CNN more proposals	2000	74.1	82.1	57.2	64.9
Baseline	1000	89.5	55.5	33.6	41.3
	2000	77.8	27.8	15.7	19.3
	3000	70.5	13.7	7.7	9.5

Table 4. Mean reconstruction precision (P_{3D}) and recall (R_{3D}) for masks with $\text{IoU} > 0.5$. Evaluation for images with n streaks per image.

Network	n	P_{3D}	R_{3D}
Mask R-CNN	1000	88.9	72.9
	2000	76.3	52.4
	3000	66.3	36.7
Cascade M R-CNN	1000	91.2	76.4
	2000	82.8	57.1
	3000	74.1	40.7
Cascade M R-CNN, n/o	1000	93.2	75.4
	2000	86.2	53.6
	3000	81.0	36.3
Cascade M R-CNN, more proposals	2000	56.8	65.6
Baseline	1000	88.5	28.5
	2000	84.5	7.7
	3000	84.7	2.0

**Figure 6.** Mask and keypoint predictions from inference with Cascade Mask R-CNN on experimentally acquired images. The predicted keypoints are shown in red.

not performed for the synthetic flow field data, as the streak curvature is very small. As the number of streaks per image increases, R_{3D} drops significantly, since corresponding streaks must be well detected in all three views for a valid streak to be reconstructed. Using more, less reliable predictions in the case ‘more proposals’ increases recall by 8.5% but causes a reduction of 26% in reconstruction precision, with about half of the reconstructed streaks not corresponding to a ground truth streak.

4.1.3. Testing: experimental data. The vortex ring images obtained from the setup described in section 3.2.2 are processed in the same way as the synthetic data, and the matched masks are processed and reconstructed in 3D with

our conic section matching method. As ground truth data are not available, the results are inspected visually (figure 6) and compared to particle-based reconstruction using commercial 3D-PTV software [54] (figure 7).

Using a Cascade Mask R-CNN model, trained with our default settings, 2438 to 2539 streaks are detected in each image. The average number of particles per image, detected from the short-exposure particle images is 2852. Therefore, the number of detected streaks is comparable to the number of detected particles, but as it is unknown which of these particles form streaks we cannot directly evaluate what percentage of the imaged streaks is actually detected. After performing endpoint and conic section matching, 2070 streaks are reconstructed in the 3D volume, while the 3D-PTV method reconstructs 810 streaks on average per time step. The settings

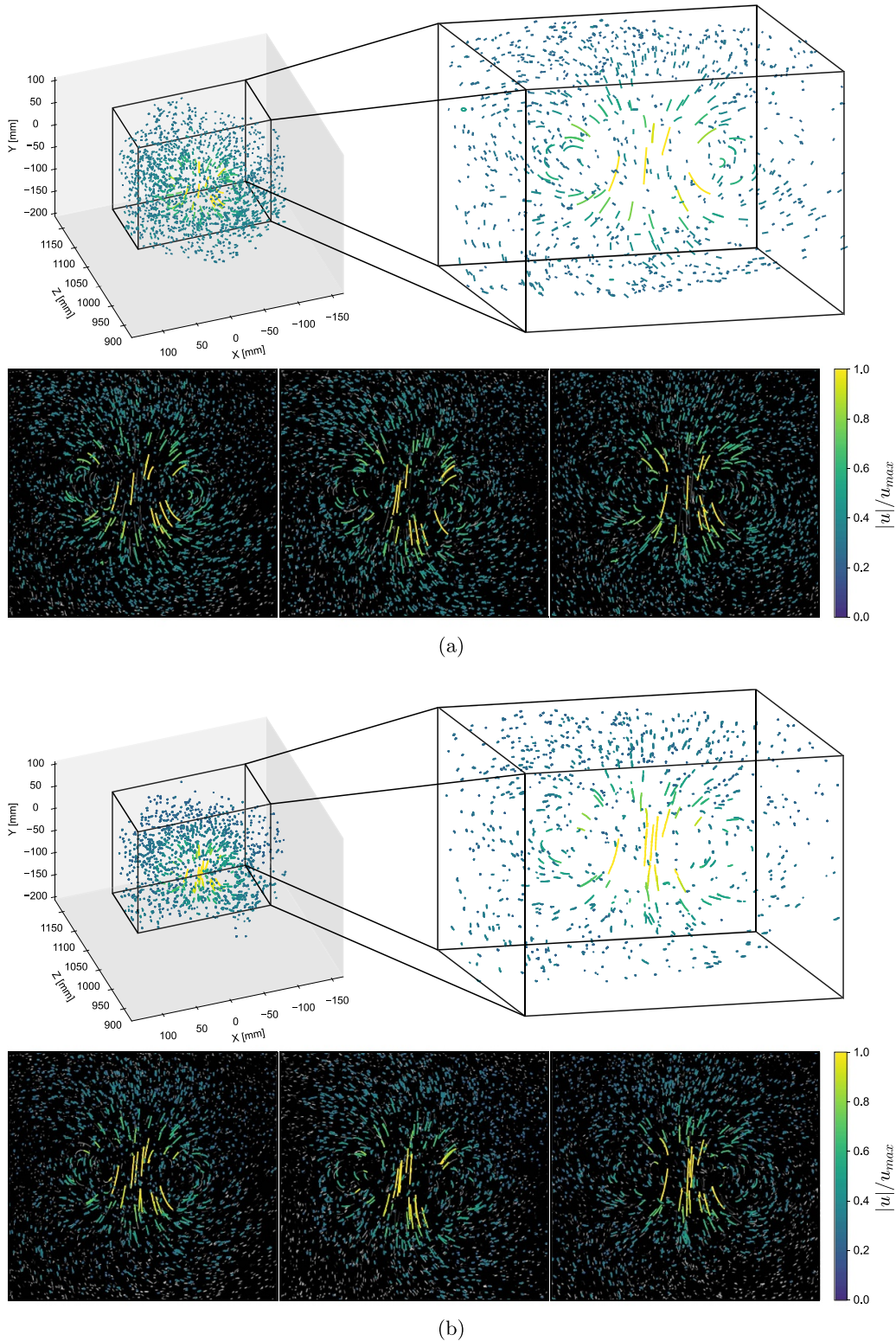


Figure 7. (a) Reconstructed conic section segments and their projections on the three camera views, from streak detections with Cascade Mask R-CNN and (b) reconstructed trajectories and their projections using commercial 3D-PTV software. A section of the reconstructed 3D (a) streaks and (b) trajectories is zoomed in.

for the evaluation with 3D-PTV can be found in appendix B. While our method seems to perform better across a wider range of displacements and identifies correctly many streaks

that remain undetected by 3D-PTV, some of the longer streaks detected by 3D-PTV are not reconstructed with our method. Additional processing steps, such as iterative elimination of

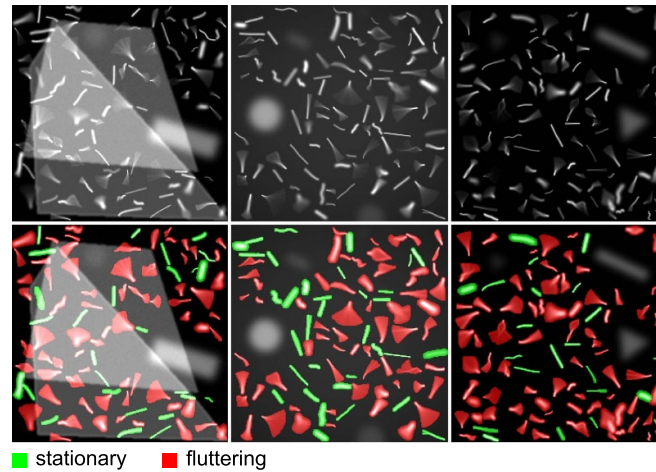


Figure 8. Top row: original images of three samples from the synthetic validation data with random synthetic tuft arrangements. The images include tufts and background features introduced to encourage the model to neglect background objects. Bottom row: detected and classified tuft instances. The various background features are successfully ignored.

the matched streaks from the images and renewed detection of the remaining streaks, could help to increase recall. Finally, it must be noted that the streak direction cannot be derived from a single frame with the presented method, unless additional experimental techniques such as colored light flashes are employed, as for example, proposed by [24]. However, provided the 3D reconstruction of multiple time frames, the direction of each streak can also be inferred by its position in the next time frame which requires some form of tracking based on the known shape and average speed obtained from each 3D streak.

4.2. Tufts

4.2.1. Validation. Mask R-CNN is used for the detection and classification of tufts, as the segmentation of tufts is an easier task than that of streaks and Mask R-CNN is faster at inference time. Wherever not defined, a ResNet-101 backbone is used. The models are trained for 6000 iterations with 4900 images containing random background shapes and polygons with different edge intensities and degree of blurriness (figure 8). The goal of this augmentation is to avoid that the network associates all edges and blurry parts of the image with tufts. 100 images of this dataset are used for the validation of the models described below. The best performing training settings are used to train a model up to 15 000 iterations. The inference results from this model are shown in figure 8.

The model trained with a ResNet-101 backbone outperforms ResNet-50, and using training data without background augmentations results in lower performance (table 5). Finally, training up to 15 000 iterations results in a small increase in performance.

4.2.2. Testing: experimental data. The default Mask R-CNN model, trained for 15 000 iterations is used for inference on the images of a NACA 0012 airfoil on which 50 tufts are applied. The network segments the streaks and predicts the

Table 5. Instance segmentation mean precision (P_{50}) and recall (R_{50}), and average precision (AP_{50}) for masks with $IoU > 0.5$ and correct class assignment, for the case of tuft segmentation.

Network	P_{50}	R_{50}	AP_{50}
Mask R-CNN	93.3	93.9	98.5
Mask R-CNN, ResNet-50	91.1	92.2	97.7
Mask R-CNN, black background	88.1	90.8	94.5
Mask R-CNN, 15 000 iterations	94.0	94.4	98.7

Table 6. Instance segmentation precision (P_{50}) and recall (R_{50}) for masks with $IoU > 0.5$ and correct class assignment for the three angles of attack shown in figure 9.

Angle of attack	P_{50}	R_{50}
8°	94.3	100.0
11°	89.3	100.0
12.4°	87.7	100.0

classes accurately, with 100% recall for all cases and 94.3, 89.3 and 87.7% precision for 8, 11, and 12.4° angle of attack respectively (table 6, figure 9). On the classification task, tufts with a wide range of motion are classified correctly as ‘fluttering’, while those for which both classes are predicted tend to be difficult to classify even by visual inspection. Some entirely faulty detections remain, as small, blurry clusters of high intensity pixels whose scale fits that of the tufts are detected.

At an angle of attack of 8° the flow is attached and all tufts are recognized as stationary. For angles of attack of 11 and 12.4° flow separation occurs, and the tufts display a fluttering, unsteady motion on the wing’s suction side (figure 9). The top row of tufts remains attached for all angles of attack. The gradual separation toward the tip of the airfoil as the angle of attack increases is consistent with finite wing flow separation patterns [55, 56].

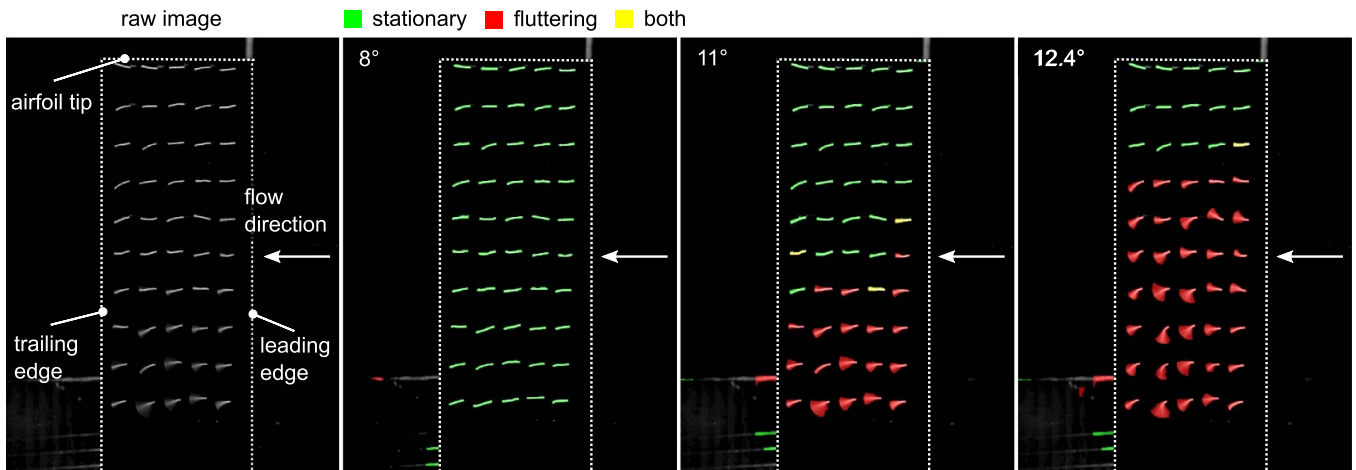


Figure 9. Predictions with Mask R-CNN for three angles of attack for the NACA 0012 airfoil. Some false detections remain, though these could be masked out as they are outside of the airfoil geometry. Stationary tufts are shown in green, fluttering tufts in red show regions of separated flow and yellow colored tufts are those where the network predicts both classes.

5. Conclusion and outlook

3D-PSV and flow visualization with tufts both require the detection of small sections of curves or a superposition of curves: the particle streaks and stationary or fluttering pieces of string, the tufts. In this work, we presented the training strategies, synthetic data generation process, and evaluation of two CNNs built for instance segmentation on the tasks of (a) particle streak segmentation and endpoint detection and (b) tuft detection and classification based on the tufts' range of motion. We used two state-of-the-art instance segmentation networks, Mask R-CNN and Cascade Mask R-CNN, and trained them on synthetically generated training data. The training strategies were evaluated on synthetic data and, for the streaks, an end-to-end evaluation of the complete processing chain, from segmentation to 3D reconstruction, was performed. The networks clearly outperformed our classical segmentation baseline, especially for high seeding densities. Though trained exclusively on synthetic images, the neural networks performed well on all tasks, as shown by the high reconstruction quality of experimental data using 3D-PSV and the high precision and recall on the tuft images. Finally, while the accuracy of 2D detections is high, the fact that the corresponding streaks must be well segmented on all camera views results in a reduction in recall after 3D reconstruction. Pair-wise endpoint matching and transfer to the third view, as described here, can be performed, but additional methods can be considered to make the method more robust, such as mask transfer to the additional views or iterative deletion of the detected streaks from the images and repeated inference on the residual images.

The proposed segmentation method for streak-like features, as presented here for particle streaks and tufts, can enable the use of 3D-PSV with higher seeding densities than before,

making the method a viable alternative to 3D-PTV when few cameras or not high-speed cameras are available and seeding densities of the order of 0.01 ppp are acceptable. Provided the ease in the segmentation step when neural networks are employed, one can focus on efficient reconstruction and tracking methods, as well as on informing the segmentation with constraints available from previous time steps and the camera geometry.

On the other hand, the segmentation and classification of tufts using neural networks can enable the automated detection of separation regions, which were previously detected manually, allowing the processing of a larger amount of data with higher accuracy.

Other areas in experimental fluid dynamics that can profit from the segmentation of streak-like features are 3D-PTV images with streak regions, where a hybrid particle-streak method for reconstruction could be employed, the segmentation of streaks of colored oil used for surface flow visualization, the detection of rigid or non-rigid non-spherical particles in flows, or the segmentation of microorganisms in biofluidics. The main challenge in applying the proposed method lies in generating sufficiently diverse and representative training data to close the synthetic-to-real domain gap.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors would like to thank Berk Giz for his initial work on Mask R-CNN, and Oliver Häuselmann for building the vortex generator that was used for the experiments.

Appendix A. Evaluation metrics

A.1. IoU

To evaluate the similarity between the ground truth and predicted masks, we use the IoU metric (figure A1). The ground truth is a binary mask encompassing all the pixels that belong to the specific instance. The prediction consists of pixels whose intensity corresponds to a score in the range $[0 \dots 1]$ that shows how confident the network is that a pixel belongs to the mask. This mask is thresholded at a confidence of 0.5 for the IoU calculation. The ratio of the intersection area of the two masks to the union of the two masks' areas is the IoU.

A.2. P_{50}

The mean precision at an IoU threshold of 0.5 (P_{50}) is calculated from all mask predictions that can be matched to ground truth instances with an IoU of 0.5 or higher, as the ratio of true positive to the sum of true positive and false positive predictions. For cases where more than one classes participate in the evaluation, P_{50} is the mean P_{50} over all classes. To calculate P_{3D} , we use the ratio of the number of 3D reconstructed streaks that can be matched to ground truth streaks to the number of total reconstructed streaks. It is not a direct measure of ghost streak generation, as some of the 3D reconstructions are due to faulty 2D detections and not due to reconstruction ambiguities.

A.3. R_{50}

The mean recall at an IoU threshold of 0.5 (R_{50}) is calculated from all mask predictions that can be matched to ground truth instances with an IoU of 0.5 or higher, as the ratio of true positive to the sum of true positive and false negative predictions. To calculate R_{3D} we use the ratio of the number of 3D reconstructed streaks that can be matched to ground truth streaks to the number of total ground truth streaks.

A.4. d_1, d_2

d_1 and d_2 are the number of streaks for which the sum of both endpoints' Euclidean distance to the ground truth endpoints is below 1 or 2 px respectively. Both endpoints are used in the metric to provide a realistic indication of how well these streaks could be matched across multiple views, as only streaks whose endpoints can both be matched to endpoints of streaks in reconstructions.

Finally, the average precision, AP_{50} is commonly used in object detection, and its value is the area under the

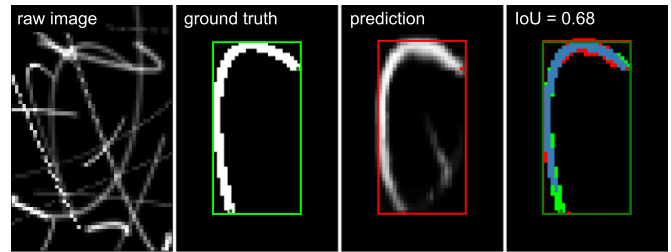


Figure A1. Example of IoU calculation for mask detection. On the right-most image, the green pixels are those that belong exclusively to the ground truth, red pixels belong only to the predicted mask, and blue pixels belong to both the ground truth and prediction. The ratio of the number of blue pixels to the sum of blue, green, and red pixels results in an IoU of 0.68 in this example.

Table B1. Settings for the reconstruction with 3D-PTV type commercial software.

Name	Assigned value
Allowed triangulation error	2 voxels
Minimum track length required	6 time steps
Threshold for 2D particle detection	1000 counts
Voxel to mm ratio	4

precision-recall curve other views can result in valide. The curve is obtained by evaluating the precision and recall values at different score threshold levels [10].

Appendix B. 3D reconstruction settings

B.1. 3D-PSV

For a triplet of streaks on the tree camera views to be a valid pair, all endpoints muss fulfill the epipolar constraint with a maximum distance tolerance set to 2 px. The conic sections are then matched using the predicted masks.

B.2. 3D-PTV

The settings for the flow reconstruction using the commercial 3D-PTV software listed in table B1.

Appendix C. Network settings

Models that are pre-trained on ImageNet are used for Mask R-CNN and Cascade Mask R-CNN, and the following changes were made to train the networks for streak and tuft detection. Wherever not mentioned, the default settings of the configuration 'Base-RCNN-FPN' in Detectron2 are used.

Table C1. Settings for Cascade Mask R-CNN and Mask R-CNN for the steaks application.

Name	Assigned value
INPUT	
MASK_FORMAT	Bitmask
RANDOM_FLIP	None
MIN_SIZE_TRAIN	800/3276 ^a
MODEL	
KEYPOINT_ON	True
MASK_ON	True
RESNETS	
DEPTH	50 or 101
ROI_BOX_HEAD	
CLS_AGNOSTIC_BBOX_REG ^b	True
SMOOTH_L1_BETA	1.0
ROI_HEADS	
NAME ^b	CascadeROIHeads
NUM_CLASSES	1
SCORE_THRESH_TEST	0.5
ROI_KEYPOINT_HEAD	
NUM_KEYPOINTS	2
POOLER_SAMPLING_RATIO	2
RPN	
POST_NMS_TOPK_TRAIN	4000
PRE_NMS_TOPK_TRAIN	9600
POST_NMS_TOPK_TEST	16 000/100 000 ^a
PRE_NMS_TOPK_TEST	16 000/100 000 ^a
SMOOTH_L1_BETA	0.2
SOLVER	
MAX_ITER	15 000
TEST	
DETECTIONS_PER_IMAGE	1000/5000 ^a

^a Settings used for the validation/test datasets, due to different image sizes and number of instances.

^b Setting not modified for Mask R-CNN.

Table C2. Settings for Mask R-CNN for the tufts application.

Name	Assigned value
INPUT	
MASK_FORMAT	Bitmask
RANDOM_FLIP	None
MIN_SIZE_TRAIN	800/3276 ^a
MODEL	
MASK_ON	True
ROI_HEADS	
NUM_CLASSES	1
SCORE_THRESH_TEST	0.5
SOLVER	
BASE_LR	0.002
MAX_ITER	15 000
TEST	
DETECTIONS_PER_IMAGE	1000

^a Settings used for the validation/test datasets, due to different image sizes and number of instances.

ORCID iDs

C Tsalicoglou  <https://orcid.org/0000-0003-2282-3975>

T Rösger  <https://orcid.org/0000-0003-2702-5876>

References

- [1] Shapiro L G *et al* 2001 *Computer Vision* vol 3 (Englewood Cliffs, NJ: Prentice Hall)
- [2] He K, Gkioxari G, Dollár P and Girshick R 2017 Mask R-CNN *Proc. IEEE Int. Conf. on Computer Vision* pp 2961–9
- [3] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A and Zagoruyko S 2020 End-to-end object detection with transformers *European Conf. on Computer Vision* pp 213–29
- [4] Ren S, Kaiming H, Girshick R and Sun J 2015 Faster R-CNN: towards real-time object detection with region proposal networks *Advances in Neural Information Processing Systems* vol 28
- [5] Cai Z and Vasconcelos N 2021 Cascade R-CNN: high quality object detection and instance segmentation *IEEE Trans. Pattern Anal. Mach. Intell.* **43** 1483–98
- [6] Bowen Cheng M D, Collins Y Z, Ting Liu T S, Huang H A and Chen L-C 2020 Panoptic-DeepLab: a simple, strong and fast baseline for bottom-up panoptic segmentation *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 12475–85
- [7] Cheng B, Misra I, Schwing A G, Kirillov A and Girdhar R 2021 Masked-attention mask transformer for universal image segmentation (arXiv:2112.01527)
- [8] Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Lawrence Zitnick C 2014 Microsoft COCO: common objects in context *European Conf. on Computer Vision* pp 740–55
- [9] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S and Schiele B 2016 The cityscapes dataset for semantic urban scene understanding *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 3213–23
- [10] Everingham M, Luc V G, Williams C K I, Winn J and Zisserman A 2010 The Pascal visual object classes (VOC) challenge *Int. J. Comput. Vis.* **88** 303–38
- [11] Toda Y, Okura F, Ito J, Okada S, Kinoshita T, Tsuji H and Saisho D 2020 Training instance segmentation neural network with synthetic datasets for crop seed phenotyping *Commun. Biol.* **3** 1–12
- [12] Moen E, Bannon D, Kudo T, Graf W, Covert M and David V V 2019 Deep learning for cellular image analysis *Nat. Methods* **16** 1233–46
- [13] König Jorg, Chen M, Rösing W, Boho D, Mäder P and Cierpka C 2020 On the use of a cascaded convolutional neural network for three-dimensional flow measurements using astigmatic PTV *Meas. Sci. Technol.* **31** 074015
- [14] Rabault J, Kolaas J and Jensen A 2017 Performing particle image velocimetry using artificial neural networks: a proof-of-concept *Meas. Sci. Technol.* **28** 125301
- [15] Vennemann B and Rösger T 2020 A dynamic masking technique for particle image velocimetry using convolutional autoencoders *Exp. Fluids* **61** 1–11
- [16] Grayver A V and Noir J 2020 Particle streak velocimetry using ensemble convolutional neural networks *Exp. Fluids* **61** 1–12
- [17] Nishino K, Kasagi N and Hirata M 1989 Three-dimensional particle tracking velocimetry based on automated digital image processing *J. Fluids Eng.* **111** 384–91
- [18] Kobayashi T, Saga T and Sekimoto K 1989 Velocity measurement of three-dimensional flow around rotating parallel disks by digital image processing *ASME FED* **85** 29–36
- [19] Maas H G, Gruen A and Papantoniou D 1993 Particle tracking velocimetry in three-dimensional flows *Exp. Fluids* **15** 133–46

- [20] Schanz D, Gesemann S and Schröder A 2016 Shake-the-box: Lagrangian particle tracking at high particle image densities *Exp. Fluids* **57** 1–27
- [21] Müller D, Müller B and Renz U 2001 Three-dimensional particle-streak tracking (PST) velocity measurements of a heat exchanger inlet flow a new method to measure all three air-flow velocity components in a plane is applied to a steady-state three-dimensional flow *Exp. Fluids* **30** 645–56
- [22] Machacek M and Rösgen T 2002 Photogrammetric and image processing aspects in quantitative flow visualization *Ann. New York Acad. Sci.* **972** 36–42
- [23] Sun Y and Zhang Y 2003 Development of a stereoscopic particle image velocimetry system for full-scale room airflow studies, part II: experimental setup *ASHRAE Trans.* **109** 540–8
- [24] Wang H, Xianting Li, Shao X, Wang B and Lin Y 2017 A colour-sequence enhanced particle streak velocimetry method for air flow measurement in a ventilated space *Buil. Environ.* **112** 77–87
- [25] Tsalicoglou C and Rösgen T 2022 Using 3D topological connectivity for ghost particle reduction in flow reconstruction *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 1839–47
- [26] Crowder J P 2001 Tufts *Handbook of Flow Visualization* ed W-J Yang (New York: Taylor & Francis) 131–79
- [27] Rosenstiel M and Grigat R-R 2010 Segmentation and classification of streaks in a large-scale particle streak tracking system *Flow Meas. Instrum.* **21** 1–7
- [28] Voss Born, Stapf J, Berthe Ae and Garbe C S 2012 Bichromatic particle streak velocimetry bPSV Interfacial, volumetric three-component velocimetry using a single camera *Exp. Fluids* **53** 1405–20
- [29] Koklu M, Lin J C, Hannon J A, Melton L P, Andino M Y, Paschal K B and Vatsa V N 2021 Investigation of the nacelle/pylon vortex system on the high-lift common research model *AIAA J.* **59** 3748–63
- [30] Koklu M, Melton L G P, Lin J C, Hannon J, Andino M, Paschal K and Vatsa V N 2019 Surface flow visualization of the high lift common research model *AIAA Aviation 2019 Forum* p 3727
- [31] Karpovich E, Gueraiche D, Sergeeva N and Kuznetsov A 2021 Investigation of a light boxplane model using tuft flow visualization and CFD *Fluids* **6** 451
- [32] Gkiolas D and Mathioulakis D 2021 Aerodynamic study of a NACA 64418 rectangular wing under forced pitching motions *Fluids* **6** 394
- [33] Vey S, Lang H M, Nayeri C N, Paschereit C O and Pechlivanoglou G 2014 Extracting quantitative data from tuft flow visualizations on utility scale wind turbines *J. Phys.: Conf. Ser.* **524** 012011
- [34] Steinfurth B, Cura C, Gehring J and Weiss J 2020 Tuft deflection velocimetry: a simple method to extract quantitative flow field information *Exp. Fluids* **61** 1–11
- [35] Chen L, Suzuki T, Nonomura T and Asai K 2019 Characterization of luminescent mini-tufts in quantitative flow visualization experiments: surface flow analysis and modelization *Exp. Therm. Fluid Sci.* **103** 406–17
- [36] Kirillov A, Kaiming H, Girshick R, Rother C and Dollár P 2019 Panoptic segmentation *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 9404–13
- [37] Kaiming H, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 770–8
- [38] Xie S, Girshick R, Dollár P, Tu Z and He K 2017 Aggregated residual transformations for deep neural networks *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 1492–500
- [39] Lin T-Y, Dollár P, Girshick R, Kaiming H, Hariharan B and Belongie S 2017 Feature pyramid networks for object detection *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 2117–25
- [40] Girshick R, Donahue J, Darrell T and Malik J 2016 Region-based convolutional networks for accurate object detection and segmentation *IEEE Trans. Pattern Anal. Mach. Intell.* **38** 142–58
- [41] Uijlings J R R, Van De Sande K E A, Gevers T and Smeulders A W M 2013 Selective search for object recognition *Int. J. Comput. Vis.* **104** 154–71
- [42] Girshick R 2015 Fast R-CNN *Proc. IEEE Int. Conf. on Computer Vision* pp 1440–8
- [43] Liu Z, Lin Y, Cao Y, Han H, Wei Y, Zhang Z, Lin S and Guo B 2021 Swin transformer: hierarchical vision transformer using shifted windows *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)* pp 10012–22
- [44] Frid-Adar M, Klang E, Amitai M, Goldberger J and Greenspan H 2018 Synthetic data augmentation using GAN for improved liver lesion classification *2018 IEEE 15th Int. Symp. on Biomedical Imaging (ISBI 2018)* pp 289–93
- [45] Giuffrida M V, Scharr H and Tsaftaris S A 2017 ARIGAN: synthetic arabidopsis plants using generative adversarial network *Proc. IEEE Int. Conf. on Computer Vision Workshops* pp 2064–71
- [46] Tobin J, Fong R, Ray A, Schneider J, Zaremba W and Abbeel P 2017 Domain randomization for transferring deep neural networks from simulation to the real world *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* pp 23–30
- [47] Yuxin W, Kirillov A, Massa F, Wan-Yen L and Girshick R 2019 Detectron2 (available at: <https://github.com/facebookresearch/detectron2>)
- [48] Lecordier B and Westerweel J 2004 The EUROPIV synthetic image generator (S.I.G.) *Particle Image Velocimetry: Recent Improvements* (Berlin: Springer) pp 145–61
- [49] Gibeau B and Ghaemi S 2018 A modular, 3D-printed helium-filled soap bubble generator for large-scale volumetric flow measurements *Exp. Fluids* **59** 1–11
- [50] Galamhos C, Matas J and Kittler J 1999 Progressive probabilistic hough transform for line detection *Proc. 1999 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (Cat. No PR00149)* vol 1 (IEEE) pp 554–60
- [51] Ester M, Kriegel H-P, Sander Jorg and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise *KDD '96* pp 226–31
- [52] Gregory N and O'Reilly C L 1970 *Low-speed aerodynamic characteristics of NACA 0012 aerofoil section, including the effects of upper-surface roughness simulating hoar frost 3726* (Aeronautical Research Council)
- [53] Hill M J M 1894 VI. On a spherical vortex *Phil. Trans. R. Soc. A* **185** 213–45
- [54] LaVision GmbH 2020 *DaVis 10.1.2*
- [55] Winkelmann A 1981 An experimental study of separated flow on a finite wing *7th Atmospheric Flight Mechanics Conf.* p 1882
- [56] Vasanth Kumar G and Mukherjee R 2021 Investigation of flow separation lines over a finite wing *Proc. 16th Asian Congress of Fluid Mechanics* pp 247–55