Automatic Yarn Path Extraction of Large 3D Interlock Woven Fabrics with Confidence Estimation

Yuriy Sinchuk, Yanneck Wielhorski, Arturo Mendoza Quispe, Samy Blusseau, Santiago Velasco-Forero

Automatic Yarn Path Extraction of Large 3D Interlock Woven Fabrics with Confidence Estimation

Yuriy Sinchuk[a], Yanneck Wielhorski[b], Arturo Mendoza[c], Samy Blusseau[a,*], Santiago Velasco-Forero[a]

[a] *Center for Mathematical Morphology, Mines Paris, PSL University, 35, rue Saint Honoré - F-77305 Fontainebleau Cedex, France*
[b] *Safran Aircraft Engines, Rond-point René Ravaud - Réau, 77550 Moissy-Cramayel, France*
[c] *Safran Tech, Rue des Jeunes Bois, 78772 Magny-les-Hameaux, France*

Abstract

Modeling realistic textile composite structures remains a challenging task due to their complex geometry. In this paper, a novel method for reconstructing yarn paths based on micro-computed tomography is proposed. A deep learning approach is employed to convert µ-CT scan into an appropriate distance map, which is used for extracting yarn paths with a tracking algorithm. An ablation study is performed to understand the hyperparameters that matter the most. This study includes variation of the target images, selection of spatial dimension of the U-Net (2D, 2.5D and 3D), dataset sampling strategies and loss terms weighting. Additionally, a robust method for estimating the quality of the predictions without the need for annotation is introduced. The accuracy of the reconstruction method is demonstrated through the analysis 15 test µ-CT images, with 5 devoted to the optimal post-processing evaluation and 10 for assessing the final test results.

Keywords: Fabrics/textiles, Computational modeling, Deep learning, CT analysis

1. Introduction

Textile composite materials have a substantial impact on a multitude of industries, including sectors like aircraft manufacturing and their engines. For instance, in the case of contemporary commercial aircraft, the extensive integration of composite materials leads to notable advancements in both fuel efficiency and the reduction of weight, when compared to older ones with all-metal counterparts. Modeling textile composites is essential for comprehending their behavior, optimizing their performance and ensuring their reliability in diverse applications. Particularly, finite element modeling stands out as a cost-effective and efficient method for designing, analyzing and refining composite structures.

*Corresponding author.
Email address: samy.blusseau@minesparis.psl.eu (Samy Blusseau)

A recent work [1] provides a thorough review of numerical models for 3D woven composite reinforcements at the mesoscale or sub-mesoscale. These modeling approaches are categorized into predictive and descriptive groups based on the method of textile geometry creation. Many predictive (idealized) models for different types of textile patterns employ tools like WiseTex [2] or TexGen [3, 4] to create a geometrical model of a material unit cell. The study [5] compares results produced using idealized and realistic geometries, revealing oversimplification in the former. A descriptive model geometry, created to match the µ-CT image, tends to be more realistic.

Image-based modeling of textile composites has gained significant attention in recent years. Integration of mesoscale finite element modeling with experimental imaging techniques demonstrates a comprehensive approach to understanding the mechanical behavior of textile composites [6]. A straightforward way to create an exceptionally realistic model involves voxel-mesh modeling, used in studies [7, 8]. This method basically relies on the segmentation of µ-CT images. Pidou-Brion and Le Guilloux [9] introduced active yarn meshes for mesoscopic-scale segmentation on X-ray computed tomography, presenting an approach for accurate representations of internal structures. Wintiba et al. [10] emphasized automation for the reconstruction and conformal discretization of 3D woven composite µ-CT scans, providing precise control over local fiber volume fractions. Huang et al. [11] utilized µ-CT aided geometric modeling for reconstructing mesostructural material twin models in engineering textiles, enhancing geometric accuracy. Bénézech and Couégnat [12] explored variational segmentation techniques for textile composite preforms from X-ray computed tomography, introducing advanced methods for segmentation and analysis. Methodologies for generating conformal meshes of woven composites at the mesoscopic scale using µ-CT scans are outlined in numerous previous studies [13, 14, 15].

Previous efforts utilizing CT scans on 3D architectures, including orthogonal weaves with vertical binders, have used image segmentation based on structure tensors to extract yarns (see works by [16, 17]). By isolating yarns using this method, centerlines can be obtained through slicing different families of yarns. However, gradient-based segmentation (as to other morphological approaches) is usually much more sensitive to the quality of the input CT data - such as noise, resolution, fiber gradient, and contrast - compared to a deep learning processing.

The deep-learning-based instance segmentation of images has seen significant advancements through various innovative approaches. The work [18] introduces a skeleton-aware distance transform to overcome challenges in instance segmentation of objects with complex structures, addressing issues related to boundary maps. In the field of biomedical image segmentation, the U-Net convolutional architecture [19] and its 3D

version [20] as well as U-Net++ [21] have been groundbreaking. This method swiftly gained popularity owing to its effectiveness, versatility and simplicity. Other prominent deep learning methods in biomedical image analysis are Cellpose [22] and StarDist [23]. Cellpose specializes in versatile cellular segmentation, while StarDist excels in object detection and segmentation. The aforementioned methods are renowned for their accuracy and effectiveness, making substantial contributions to the progress of image analysis. They excel in tackling a wide range of segmentation challenges and find extensive applications in various fields, beyond biology.

Presently, segmentation methods based on deep learning are widely employed as powerful tools to address segmentation challenges in material imagery. For instance, a modular U-Net was recently used for segmentation in composites material [24], including testing of 2D, 2.5D and 3D models (2D or 3D convolutions). The aim is to provide a conceptually more compact representation of this neural network family by identifying three essential blocks of a U-Net (convolutional, down-sampling and up-sampling blocks) so as to facilitate their implementation. The analysis of woven fabric composites is also widely studied nowadays. Notably, the works of [14, 25, 26] showcase the high accuracy achieved in the segmentation of 2D layered textiles through convolutional neural networks. Zheng et al. [26] used SwinT as the backbone to extract features which will feed a multi-task framework called UPerNet [27] to convert the features to segmentation results in order to create artificial learning datasets, hence improving the real learning datasets. Additionally, the segmentation of 3D woven CT in [28] is accomplished using a dense neural network approach, which is supplied with features derived from the monogenic signal and learned by a convolutional neural network. Mendoza et al. [29] introduced a complete description for yarn instance segmentation by extracting yarn contours as well as centerlines from µ-CT images of woven composites. The main idea was to use the key points estimation method through the Mask R-CNN [30] neural network. Moreover, it is noteworthy that an astute manner for providing data augmentation to the model was the utilization of U-Net to generate synthetic µ-CT images from their corresponding FE simulations. Subsequently, these centerlines are transformed into a voxel-mesh model, which is further converted into a tetrahedral finite element mesh as presented in [15]. Later, in the work of [31] a 2D U-Net is used to predict binary images of the yarn centerlines. The resulting textile description is converted into yarn instances using distance transform prediction for the yarn envelopes (2D cross sections). These multiple studies highlight the wide and rich literature about deep learning processing of composites images, showcasing its growing importance in understanding, analyzing and modeling material properties based on µ-CT data.

Very recently, Koptelov et al. [32] used a particular type of recurrent neural network (RNN), namely a

Long Short-Term Memory (LSTM) network, or an autoencoder architecture for spatiotemporal prediction of voxel mesh. The idea is to predict a compacted voxelized model from an initial kinematic model in a loose configuration (as-woven state) instead of performing heavy compaction simulation.

In the continuity of previous researches [29, 31], this study introduces a novel method, employing convolutional neural networks, for reconstructing yarn centerlines from the large tomography data of textile composite structures, addressing the challenges posed by the complex geometry at the mesoscale. The concept of this instance segmentation relies on the use of distance transforms and heat flows for extracting yarn centerlines through a customized tracking algorithm as post-processing in the fashion of previous methods [31, 33].

The focus of the present of work is to overcome the multiple challenges associated with the processing of substantial volumes of tomography data. Indeed, the method have to tackle typical challenges associated with large-scale data, including variations in image quality, woven architecture, cross section size, to name but a few, and yarn lengths, as well as constraints related to computational resources. These challenges require robustness in accuracy and tolerance towards unseen input data, scalability for seamless adaptation to larger datasets, high computational speed and simplicity in configuring setup parameters. This paper includes also sensitive studies covering diverse neural network training experiments with different samplings of the training dataset, varied dimensionality (2D/3D convolutions) and multiple types of target images as well.

The article is structured as follows. Section 2 describes the dataset used for training and testing which is composed of 25 large CT scans coming from fan blades of aircraft engines. The used deep learning approach has been subjected to an ablation study on 10 full CT scans to evaluate the importance of individual hyperparameters, contributing to enhance the robustness of the method. The data pre-processing and the deep learning models are respectively detailed in Section 3.1 and Section 3.2. The post-processing of the inferred images is outlined in Section 3.3. The selection of a straightforward but high-performance and robust post-processing method for centerlines tracking is justified through a comparison with alternative solutions. Then, Section 3.4 presents methods to estimate the quality of centerline predictions with and without ground-truth annotations and taking into account the yarn cross section orientation. The computational experiment results and corresponding discussions are finally presented in Section 4 and Section 5 respectively.

2. Materials

The material under investigation is a carbon-fiber-reinforced epoxy composite featuring a 3D interlock architecture. A set of 25 µ-CT images of the material are employed, each annotated with the centerline yarn paths (trajectories). These scans correspond to a section from the fan blade dovetail from the LEAP-1A and LEAP-1B engines.

The voxel size of µ-CT images is in the range from 36 to 40 µm. In Figure 1a, a 3D shape of the typical µ-CT image (size: $1749 \times 2162 \times 772$ voxels, voxel size: 39.4 µm) is presented. In this and other µ-CT images, the 0-axis represents the thickness direction, the 1-axis corresponds to the warp direction (Figure 1c), and the 2-axis represents the weft direction (Figure 1d).

The average cross-sectional shape of the yarns is elliptical. In particular, the average dimensions of the warp yarns is 2×1 mm^2 (major and minor axes of the ellipse) whereas the weft cross section size is more varied. For example, weft yarns with a small height (about 0.5 mm) and large width (about 4 mm) occur frequently. A large majority of the yarns in every µ-CT image are annotated. For each yarn, these annotations consist of a sequence of 3D point coordinates, manually positioned at the yarn cross section center. On average, there are about 300 annotated "ground-truth" (GT) yarn paths in the warp direction and about 400 in the weft direction per µ-CT image.

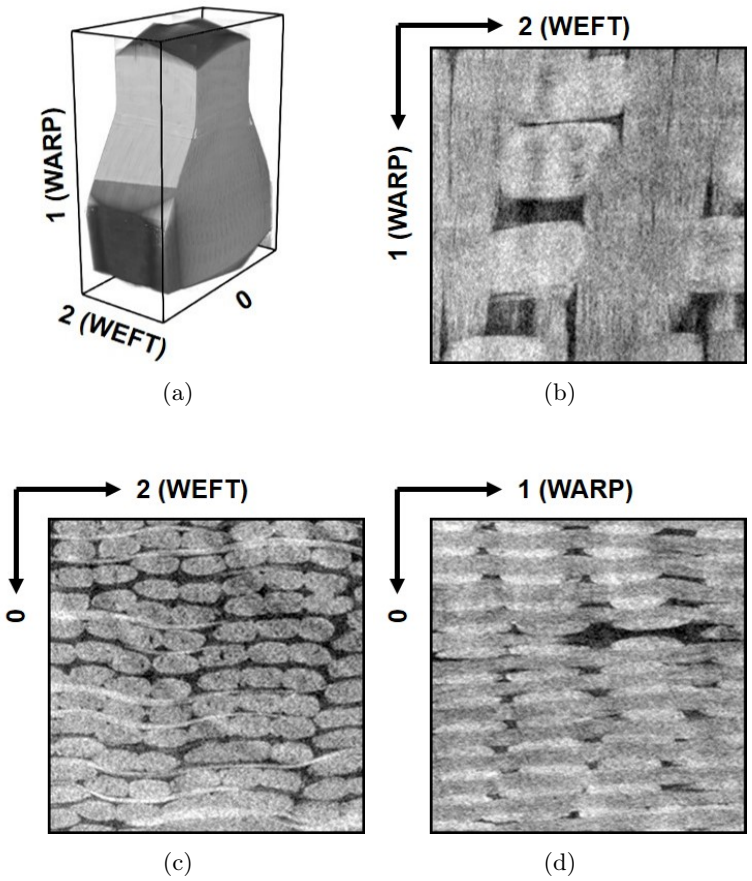


Figure 1: µ-CT scan of a root-blade structure: (a) 3D shape and crops of the top view (b) and (c) warp and (d) weft slices.

3. Methods

The proposed reconstruction of yarn centerlines is outlined as a sequential process comprising the following stages: data preparation for the training and following post-treatment, training of a deep learning model, inference, reconstruction of yarn paths and quality estimation.

Each stage of the pipeline will be detailed in the following. Additionally, Figure 2 illustrates the interplay between the different elements that constitute it.

3.1. Data preparation

First, the input images undergo some simple preprocessing with the goal of reducing computational load and minimize GPU memory usage. During the preprocessing phase we also generate a binary mask of the object within the scan and estimate a general orientation field for the textile layers. Finally, the target

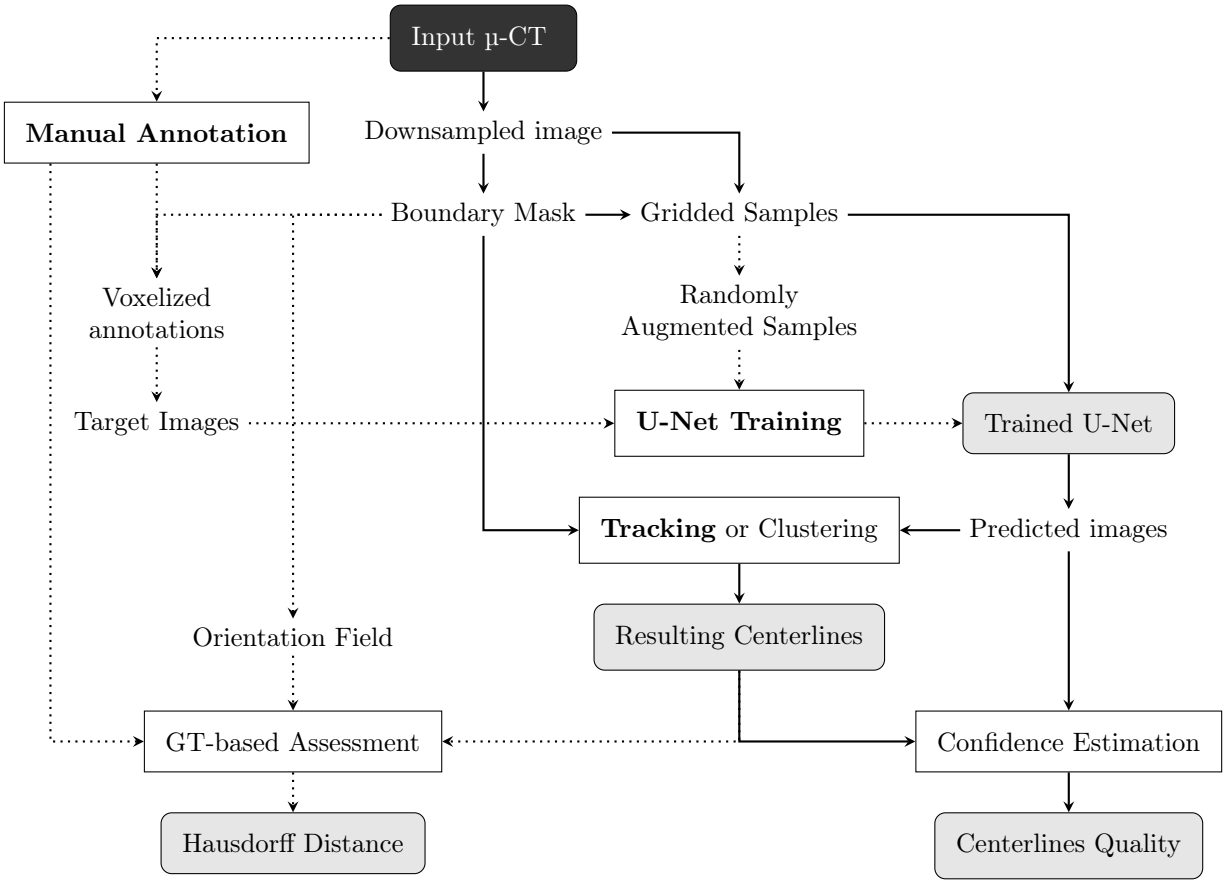


Figure 2: Flowchart of the proposed pipeline, with the outputs highlighted with gray rounded rectangles. Moreover, the important processes in the pipeline appear inside white rectangles, whereas the other elements are simply displayed without any background. It is important to note that the dotted lines indicate the operations that are only required for training and development of the method, whereas the solid lines indicate the steps to follow if one wishes to only perform inference on novel images.

images are created from the manual annotations. This allows setting-up the input/output image pairs and the corresponding regions of interest.

3.1.1. Source and mask images

First, a binary mask is created using the Otsu threshold method on the input µ-CT images. This mask is further refined using morphological operations (closing and hole-filling [34, 35]). This binary mask is used to identify the smallest bounding box that contains all useful voxels. This defines the region of interest used for cropping the input µ-CT images (and the mask).

Next, an anisotropic scaling is applied to both input and mask images. This transformation aims at converting the mostly elliptical yarn cross sections into almost circular ones. Previous studies [36] have shown that if the information sought to extract from the image is measured at the textile scale (*i.e.*, one does not seek to precisely determine the yarn envelope), this transformation guarantees that all useful information is kept. As such, given that our interest are the yarn paths (*i.e.*, the mean yarn positioning), the anisotropic scaling can be safely applied to the µ-CT images. In particular, a scaling factor of $(1, 0.5, 0.5)$ will perform this task as the major axes of the elliptical yarn cross sections are oriented along the axes 1 and 2 of the µ-CT image (this can be seen in Figure 1). An example of the resulting downsampled µ-CT image is show in Figure 3, and the corresponding downsampled binary mask M_X is shown in Figure 4a. It may be worth noting an additional benefit of this transformation: the resulting images contain 4 times less voxels than the original ones. This reduction in volume to analyze, is extremely useful for accelerating the subsequent calculations.

3.1.2. Textile orientations

Due to the shape of the dovetail, the yarn cross sections follow different orientations according to their location within the part. Furthermore, the density of yarns can be variable depending on zones. Therefore, both yarn neighborhood and position lead to more or less "compressed" cross sections (elliptical shape) with different orientations. In particular, this phenomenon is highlighted for the weft yarns as shown in the diagram Figure 4b.

So, the assessment of our method needs an anisotropic distance to centerlines (described in Section 3.4.1), adapted to each yarn, which seems more suitable than a classical isotropic Euclidean metric. The idea is to penalize differently the errors according to the minor and the major axes of the ellipse.

This orientation field is computed by interpolating between the orientations of the top and bottom boundaries of the 2D projection of the mask image, displayed in Figure 4c. Note that this orientation field is

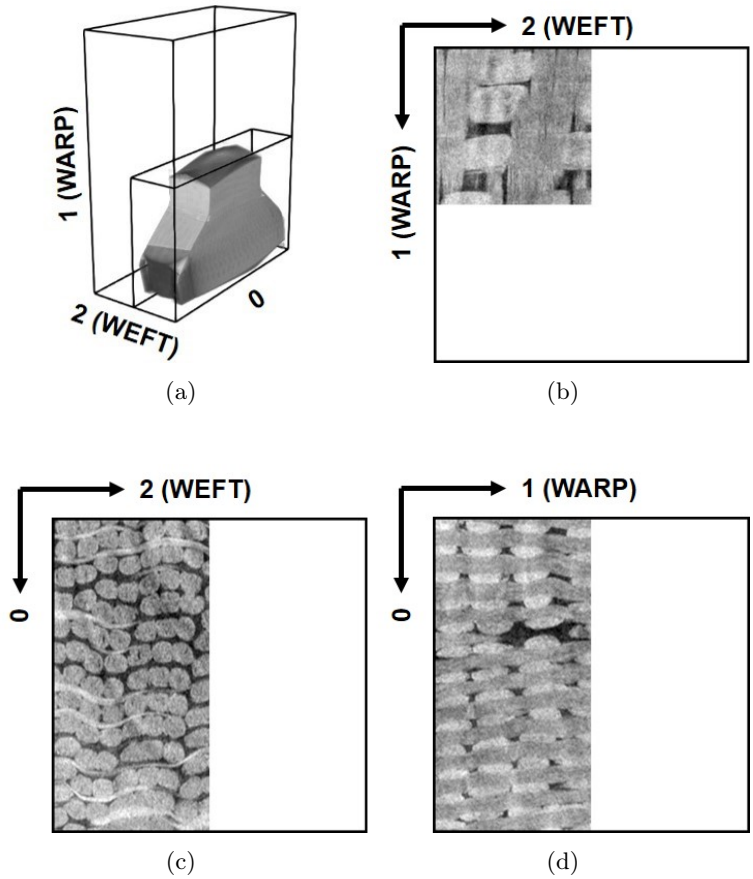
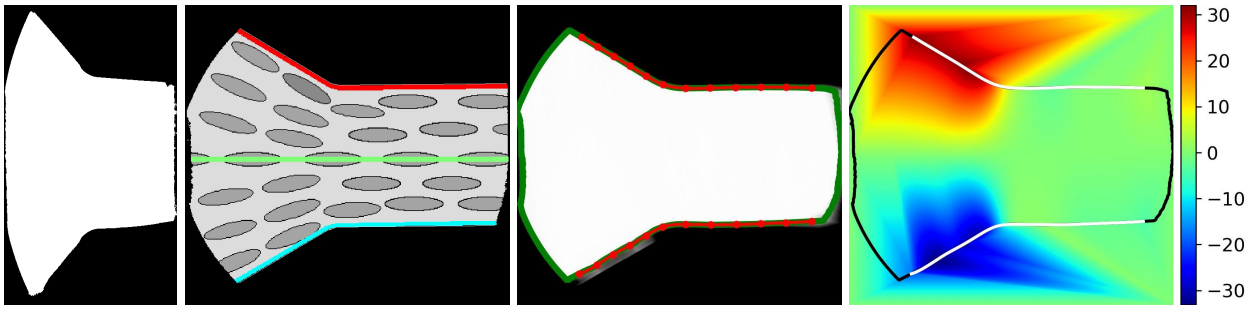


Figure 3: Results of anisotropic scaling for the images shown in Figure 1

computed only as the yarn cross section slope while the axial orientation of the weaving is not accounted here. Figure 4d provides a typical representation of the resulting interpolation field. Although this orientation field of weft yarns also gives an information on the warp weaving, it was not used at all in our tracking method, but only for evaluation purpose.

3.1.3. Input dimensions

In this study we deal with 3D images, as such it makes sense to employ the 3D nature of the source µ-CT images as inputs for the neural network. However, most computer vision applications deal with 2D images, hence most deep learning models and architectures are designed for two dimensions and not three. It is thus natural to also explore the use of 2D-like images extracted from the 3D volumes as inputs for the neural networks as well. Now, given that there are two main orientations (warp and weft), this 2D strategy implies "looking" at the volume twice. Once using the warp axis orthogonal to the inspection plane (*i.e.*,

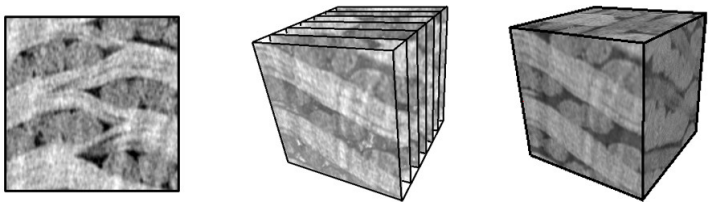


(a) (b) (c) (d)

Figure 4: (a) view of mask M_X; Orientation field approximation: (b) diagram of weft yarns arrangement, (c) mask projection with contour and spline approximation of top and bottom contour parts, (d) resulting orientation (in degrees).

the thickness and weft axes would conform the 2D images) and a second one using the weft axis. Under this 2D paradigm, there are no longer warp and weft orientations but rather a horizontal dimension aligned with the major axis of the elliptical cross sections and a vertical dimension aligned with the minor axis.

Additionally, we propose an "intermediate" approach using 2.5 input dimensions. Here, we take as input a multi-channel 2D image, built as a sequence of 2D images from an ordered set of slices (in the 3D spaced) spaced by a given step in one given direction (warp or weft). This means that the slices are sampled with a sampling rate possibly larger than one (*i.e.*, two successive slices in the 2.5D input may not be adjacent in the original scan). Such 2.5D technique embeds the local depth orientation as "channels" that will become feature vectors for the neural networks. Further details into the construction of these multi-channel images will be given in Section 3.2.3. Figure 5 illustrates this 2D, 2.5D and 3D propositions.



(a) 2D (b) 2.5D (c) 3D

Figure 5: Illustrations of the input dimensions to be used in this study

3.1.4. Target images

Different strategies will be explored in this work, as such different "target images" will be constructed. Each type of target image, paired along the corresponding input µ-CT images, will form the training pairs to be used for constructing the neural networks.

The target images are derived, slice by slice, from the 2D Euclidean distance of each voxel to the closest annotated (ground-truth) yarn centerline in the cross section plane. That means that they only consider yarns orthogonal to the measurement direction and that the procedure is applied independently for the weft and warp orientations. Moreover, they are computed using a discretization equivalent to that of the downsampled input images and only inside the binary mask defined earlier.

Formally, let $H : \mathbb{R}^3 \to \mathbb{R}^2$ be the function mapping a voxel x to the vector $c(x) - x$, projected in the cross section plane, where $c(x)$ is the closest annotated voxel within the cross section plane.

We can then compute the "heat flow" image (vector field) as

$$Y_H(x) = \frac{H(x)}{D(x)} \quad \forall x \in \Omega \tag{1}$$

and the "distance transform" image (scalar field) as

$$Y_D(x) = \left(1 - \min\left\{1, \frac{D(x)}{D_{max}}\right\}\right)^{\alpha} \quad \forall x \in \Omega \tag{2}$$

where $\Omega = \{x \,|\, M_X(x) = 1\}$ limits the computation to useful regions of the image and $D_{max} = 35$ voxels is chosen based on the downsampled yarn cross section size. Moreover α helps controlling the *steepness* of the distance transform by introducing a certain nonlinearity of the brightness variation between the center and the edge of the cross section, avoiding hence unstable training when the ground-truth annotation is marked with only one pixel. Figure 6 showcases its effect, where that it can be seen a delta Dirac is formed for $\alpha \to \infty$. Here, the value $\alpha = 4$ is chosen after observations of yarn separability.

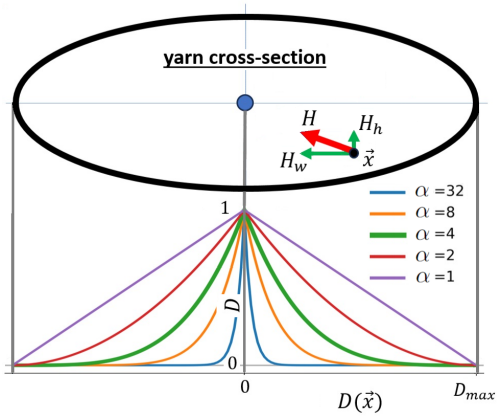


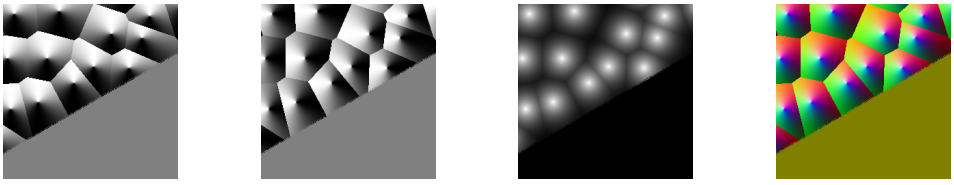
Figure 6: Explanation of the target image channels for a cross section region.

Let us note that, by definition, $Y_H(x)$ produces unit vectors. Then, starting at any position x ($\forall x \in \Omega$),

one can reach the closest yarn centerline iteratively following the directions pointed by the corresponding the 2D vectors $Y_H(x)$. Furthermore, given that $Y_H(x)$ and $Y_D(x)$ are computed independently for each yarn orientation, we actually obtain four images: $Y_{H1}(x)$, $Y_{H2}(x)$, $Y_{D1}(x)$ and $Y_{D2}(x)$.

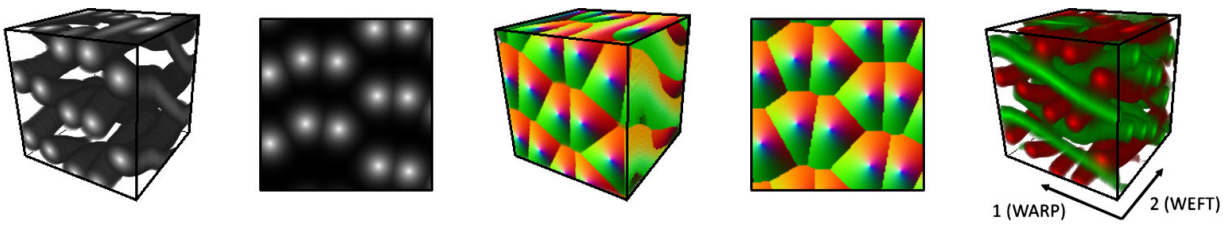
Next, the four computed images can be discretized into 4-D arrays of dimensions (C, H, W, D) where C represents the number of "channels", while H, W and D represent the corresponding spatial dimensions (height, width and depth). In particular, $C = 2$ for Y_{H1}, Y_{H2} and $C = 1$ for Y_{D1}, Y_{D2}. Then, the array of each yarn orientation (warp or weft) can be concatenated (along their channel dimension) so as to obtain the three-channel arrays Y_{HD1} and Y_{HD2} (*i.e.*, $Y_{HD1} = [Y_{H1}, Y_{D1}]$, $Y_{HD2} = [Y_{H2}, Y_{D2}]$ and $C = 3$). Alternatively Y_{D1} and Y_{D2} can be concatenated into the Y_{DD} array (with $C = 2$) that encodes the distance to both warp and weft yarns along their respective orthogonal planes in each channel.

Figure 7 shows the horizontal and vertical components of the vector field Y_{H2} (in Figures 7a and 7b) as well as the corresponding Y_{D2} for some weft yarns in one extremity of the blade. Moreover, Figure 7d illustrates Y_{HD2} by assigning each of the $C = 3$ channels to red, green and blue accordingly. Similarly, Figure 8 shows 3D visualizations of Y_{D2}, Y_{HD2} and Y_{DD} alongside to their respective $2D$ slices (axes 0 and 1) for a region completely inside the blade. It should be noted that, since Y_{D2} and Y_{HD2} correspond to the weft orientation, no warp yarn is visible at all. This is not the case for Y_{DD} which is the only construction that accounts for both yarn orientations simultaneously.



(a) Y_{H2} vertical (b) Y_{H2} horizontal (c) Y_{D2} (d) Y_{HD2}

Figure 7: Examples of 2D visualization for some target images Y_{H2} (heatflow), Y_{D2} (distance transform) and Y_{HD2} using RGB as mapping of the three channels.



(a) Y_{D2} (b) Y_{D2} (c) Y_{HD2} (d) Y_{HD2} (e) Y_{DD}

Figure 8: Examples of 2D visualization for some target images Y_{D2}, Y_{HD2} and Y_{DD} using RG as mapping of the two channels.

3.2. Deep learning model

The goal of this section is to detail the development the pipelines that will convert input µ-CT images (using either of the 2D, 2.5D or 3D approaches) into one of the target images presented previously (Y_D, Y_{HD} or Y_{DD}). Given that there exist three types of input dimensions and three types of target images, one could expect up to nine combinations of input dimensions and target images. However, there exist two impractical combinations: 2D with Y_{DD} and 2.5D with Y_{DD}. Then, there exist seven useful combinations, such as using 2.5 input dimensions with the goal of predicting the corresponding Y_D target image. For the sake simplicity, let us use the notations $\mathbb{D}$, $\mathbb{HD}$ and $\mathbb{DD}$ to highlight the type of target image that a given model will seek.

3.2.1. Neural network architecture

A base U-Net [19, 20] architecture with depth three levels is employed for all configurations. The diagram in Figure 9 shows the main blocks of the U-Net. Each block denotes the number of output channels.

The "Conv-ReLU" layers perform convolutions and use the Rectified Linear Unit (ReLU) as activation function. This operation is always repeated. The "MaxPool" layers help condensing the information so that only the most prevalent features are kept. The "TransConv" operation is equivalent to a simple upscaling (by bilinear/trilinear interpolation) followed by a convolution. The goal of this layer is to expand useful features to larger spatial dimensions. Finally, the "Conv-Linear" layer simply condenses the found feature maps into the sought output (from 64 to N_{out_ch} channels). Here, unit-kernel convolution is used without any activation function.

All these operations are implemented in 2D and 3D (convolutions, max-pooling, upscaling). This results in around 22 million parameters for the 3D U-Net and about eight million for the 2D one. The 3D U-Net is used for the 3D models ($\mathbb{D}$, $\mathbb{HD}$, $\mathbb{DD}$). On the contrary, the 2D U-Net is used for the 2D and 2.5D models ($\mathbb{D}$, $\mathbb{HD}$). The only modification required is on the first convolution layer that accounts for a varying number of N_{ch} channels. Always $N_{ch} = 1$ for the 2D case and $N_{ch} > 1$ for the 2.5D case.

3.2.2. Training set-up

This study examines the role of weighting on the mean square error (MSE). As such, two loss functions are explored: MSE and WMSE (weighted-MSE).

The goal of this weighted loss function is to penalize more strongly prediction errors closer to the

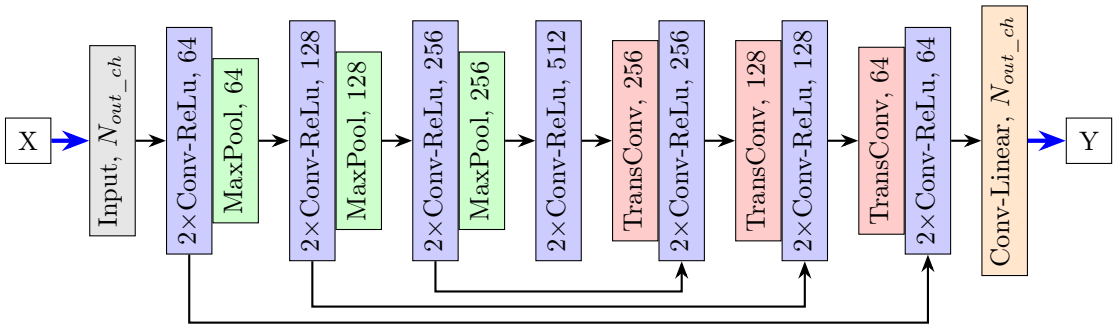


Figure 9: General architecture of the U-Net network

centerline. It is defined for each model

$$WMSE_{\mathbb{D}} = \frac{1}{N_v} \sum_x (y_D(x) + \varepsilon)(y_D(x) - \widetilde{y}_D(x))^2 \tag{3}$$

$$WMSE_{\mathbb{HD}} = \frac{1}{3N_v} \sum_{i=1}^{3} \sum_x (y_D(x) + \varepsilon)(y_{HD}^{(i)}(x) - \widetilde{y}_{HD}^{(i)}(x))^2 \tag{4}$$

$$WMSE_{\mathbb{DD}} = \frac{1}{N_v} \sum_{i=1}^{2} \sum_x (y_{DD}^{(i)}(x) + \varepsilon)(y_{DD}^{(i)}(x) - \widetilde{y}_{DD}^{(i)}(x))^2 \tag{5}$$

where y are the ground-truths and $\widetilde{y}$ are the predicted samples, N_v is the count of voxels in the sample, i is the channel index, x is voxel position. A small value of $\varepsilon = 0.05$ is used (5% of the maximum value of y) to avoid any pathological scenario. Without ε, the loss function would ignore errors at voxels with zero ground-truth values. This could potentially lead to decrease training performance, especially in regions where the ground-truth values are low or zero.

During training, the values of MSE or WMSE are minimized. In order to do so, an Adam optimizer with a learning rate coefficient of 3×10^{-4} and a batch size of one are used. Larger batches were not tested due to GPU memory constraints.

The early stopping technique with patience of 20 epochs is employed to automatically determine the optimal moment at which to finalize training. Detailed information regarding the training time and the total number of epochs for the studied configurations is provided in Table 1 (cf. Section 4).

Finally, the network weights are initialized randomly but with an identical initialization across all training sessions. Indeed, to ensure reproducibility and eliminate random factors stemming from parallelism, the *determinism* feature of TensorFlow [37] is enabled. This guarantees that when the training is executed multiple times on the same hardware with identical inputs and initialization, it consistently produces the exact same outputs.

3.2.3. Dataset sampling

First of all, 10 of the 25 available µ-CT images are selected for the training and validation procedures. The remaining 15 samples will be used for optimizing the post-processing and "blind validation" of the method (testing).

Each of the 10 training images (after preprocessing) is split into non-overlapping gridded samples of size $160 \times 160 \times 160$. This operation (shown in Figure 10a) results, on average, in 124 samples per CT image.

Only samples containing "useful" information (non zero standard deviation) or those containing at least one annotation point (with at least 20% of the target image non zero) were kept. This filtering operation removes around 33% of all samples, and only 826 remain (from 1238). All samples from all blades are assembled into a single dataset, from which a random 20% subset is selected for validation and the remaining 80% for training. Each sample of $\mathbb{D}$ and $\mathbb{HD}$ model is rotated so that the warp and weft orientations are aligned to the 0-axis of each sample (*i.e.*, two separate samples are created).

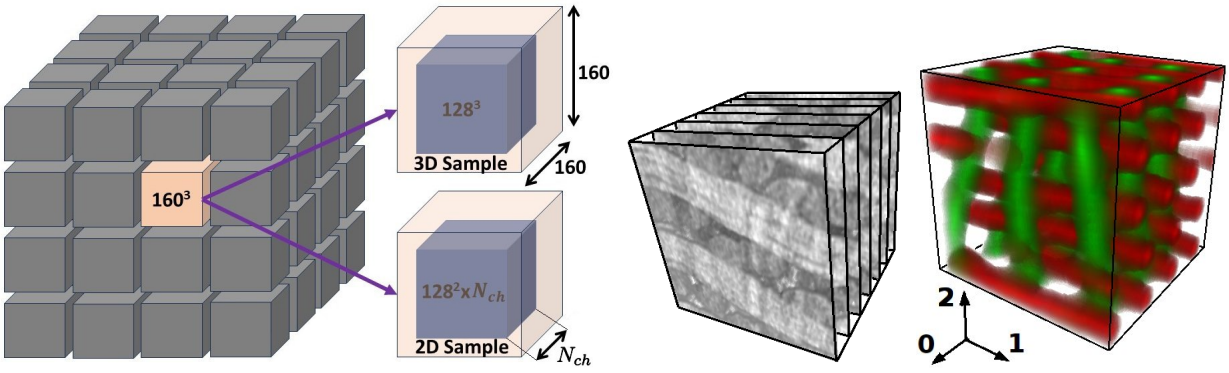
The following "on-the-fly" operations are applied to each sample: normalization, random gray level variation, random flip and random crop. The normalization procedure simply seeks a zero mean and unit standard variation. Next, the random gray level variation can be written as $(b_1 + 1) \cdot (b_2 + x)$, for the normalized input x, where parameters b_1 and b_2 take a random value from the range $[-0.2, 0.2]$. Then, the random flip operation operates on any of the spatial dimensions with a 50% probability. The sign of the target images Y_H must be applied accordingly so that the heat flow vectors point in the correct direction.

The random crop operation consists in randomly selecting an origin (within the sample) and then performing a subset selection procedure, detailed hereafter. Note that the random selection of the origin is constrained so that the selected subset fits completely within the initial sample. The sample center is used as origin for all samples in the validation dataset.

The subset selection procedure is adapted for each of the 2D, 2.5D or 3D scenarios. They either extract a 2D slice of 128×128 pixels or a 3D sub-volume of $128 \times 128 \times 128$ voxels around the previously defined origin. Note that the 2D slice is selected respecting the yarn axial direction. Now, the 2.5D selection procedure consists in extracting a 3D sub-volume of dimension $128 \times 128 \times N_{depth}$, with $N_{depth} = (N_{ch} - 1) \cdot S_{ch} + 1$, only retaining N_{ch} slices with step S_{ch} voxels, and assigning the slices as channels of a 2D image. Setting $N_{ch} = 127$ and $S_{ch} = 1$ results in $N_{depth} = 127$, a 3D region (almost) equivalent to that of the 3D scenario (but seen as a 2D image with multiple channels). The depth of 127 can also be achieved with $N_{ch} = 19$ and $S_{ch} = 7$ or $N_{ch} = 7$ and $S_{ch} = 21$. The idea for these combinations is to encompass equivalent but progressively more spaced-out domains. A fourth combination of $N_{ch} = 19$ and $S_{ch} = 3$ that results in

$N_{depth} = 55$ is also explored.

Finally, the corresponding sample size for the target images is $128 \times 128 \times 128 \times N_{out\_ch}$ for 3D and $128 \times 128 \times N_{out\_ch}$ for the 2D and 2.5D models. As stated previously, $N_{out\_ch} = 1$ for $Y_{\mathbb{D}}$, $N_{out\_ch} = 2$ for $Y_{\mathbb{DD}}$ and $N_{out\_ch} = 3$ for $Y_{\mathbb{HD}}$. Note that the random cropping is applied on the input and output images.



(a) (b) (c)

Figure 10: (a) Dataset sampling and random cropping augmentation schemes for 2D sample size of 128×128 voxels with N_{ch} channels and 3D sample size of $128 \times 128 \times 128$ voxels; (b) 2.5D sample of size $128 \times 128 \times N_{ch}$ (for $N_{ch} = 7$); (c) training sample of Y_{DD} target image (size of $128 \times 128 \times 128$ voxels).

3.2.4. Weighted inference

Even if the convolutional model could infer on input samples of different size than those used for training, here they are chosen to be identical (*i.e.*, $128 \times 128 \times 128 \times 1$ for 3D model and $128 \times 128 \times N_{ch}$ for 2.5D and 2D).

In order to improve the signal-to-noise ratio on the predictions, the samples are taken from an overlapping grid. The distance between consecutive samples is 25% of the sample size for each dimension (*e.g.*, 32 for the spatial dimensions of 128). This results in an overlap of 75% and each voxel being analyzed (at most) 64 times. The resulting output for a given voxel at position x is obtained as

$$\widetilde{Y}(x) = \frac{\sum_s w(x - x_s)\widetilde{y}_s(x - x_s)}{\sum_s w(x - x_s)} \quad \forall x \in \omega \tag{6}$$

with $\omega = \{x \,|\, M_X(x) = 1\}$ and $\widetilde{y}_s$ is the output of the neural network for a given sample s centered at x_s

and the weighting function

$$w(x) = \begin{cases} 100\% & \text{if} \quad \|x\|_\infty < 32; \\ 10\% & \text{if} \quad \|x\|_\infty < 64; \\ 0\% & \text{otherwise} \end{cases} \tag{7}$$

with $\|x\|_\infty = \max_i |x_i|$ representing the largest component of the 3D position vector. Figure 11a depicts a slice of $\widetilde{Y}_D(x)$.

3.3. Yarn paths reconstruction

This post-processing step consists in exploiting the target images predicted by the neural network as inputs for extracting the yarn centerlines.

In this study, three strategies have been explored: *distance transform tracking*, *heat flow tracking* and `DBSCAN` clustering [38, 39]. Distance transform tracking and `DBSCAN` exploit the single-channel image $\widetilde{Y}_D$, while heat flow tracking uses the two-channel image $\widetilde{Y}_H$.

In all methods, a constraint on the minimum distance between yarn centerlines is set to $\delta = 0.4$ mm. This value was chosen after analyzing the average proximity between centerlines, computed over all annotated yarns.

As a final step of the reconstruction, a 1D Gaussian filter with a sigma parameter of 0.4 mm is applied to the centerline points (along the yarn orientation) so that small perturbations do not affect the quality. The resulting centerlines are mapped back into the original µ-CT images resolution (before downscaling).

3.3.1. Refinement of the predicted image

In order to highlight regions that can be safely assumed to correspond to a correct identification of a yarn center, a template of the "ideal" output of the network for a single yarn center is constructed using Equation (2) so as to perform a cross-correlation. The template is denoted as D_{ideal} and is shown in Figure 11b.

Then, the normalized cross-correlation [40] between this template and the weighted inference image results in an refined image (shown in Figure 11c) defines as

$$\overline{Y}_D(x) = \widetilde{Y}_D(x) * D_{ideal}(x) \tag{8}$$

It may be worth noting that the dynamic range of the image is now defined by $|\overline{Y}_D(x)| \leq 1$. With all negative values indicating a negative correlation (*i.e.*, indicative of wrong yarn center identification by the neural network).

3.3.2. Tracking strategies

Both tracking algorithms act on the warp and weft directions independently and operate on sequential 2D slices of the obtained volumes. They both start at an initial 2D plane with an set of yarn centers (one for each orientation). Then, these center points are propagated towards the edges of the sample so as to create the yarn centerlines.

The initial set of yarn points can be manually obtained (annotation) or by exploiting the $\overline{Y}_D$ image and identifying its peaks (local maxima of the cross-correlation). Moreover, this initial set must respect the minimal distance δ condition.

The *distance transform tracking* is detailed in Algorithm 1 for a slice i, a centerline P, a tracking point $p_i \in P$ and a window W (2D region with center at p_i). The initial points p_0 is given on an initial slice. The window W was chosen to be of size 7×7 pixels (after calibration on the 5 dedicated CT images). Moreover, $t_{end} = 0$ is used to stop the tracking as only positively (cross-)correlated values can be considered as trustworthy.

Algorithm 1: Distance transform tracking

1 Initialize by the previous slice value: $p_i := p_{i-1}, \ \forall P$;
2 Find the closest local maximum position within the window: $p_{\texttt{trial}} = \arg\max_{p \in W} \overline{Y}_D(p), \ \forall P$;
3 Find collided yarns $\mathbb{P}_{\texttt{collision}} := \{p_{\texttt{trial}} \mid \exists q_{\texttt{trial}} \neq p_{\texttt{trial}}, \ \|p_{\texttt{trial}} - q_{\texttt{trial}}\| < \delta\}$;
4 **If** $\mathbb{P}_{\texttt{collision}} \neq \emptyset$: $p_{\texttt{trial}} := p_{i-1}, \ \forall p_{\texttt{trial}} \in \mathbb{P}_{\texttt{collision}}$ and go to step **3**,
else $p_i := p_{\texttt{trial}}$;
5 Stop when an end condition is reached : $\overline{Y}_D(p_i) < t_{end}$ or i is defined as "last" slice.

On the other hand, the *heat flow tracking* procedure follows a similar structure as the previous one (initialization, stopping criteria, collision condition) but differs on the update strategy (step 2 of the algorithm). Let us recall that the ground truth $Y_H(x)$ image is composed of unit vectors that consecutively lead to the yarn center point for any given 2D slice. As such, for a given slice i, the trial point p_{trial} initialized by p_{i-1} is updated iteratively by applying the displacement indicated by the 2D vector field $\widetilde{Y}_H$ within the slice i

$$p_{\texttt{trial}} := p_{\texttt{trial}} + \frac{\widetilde{Y}_H(p_{\texttt{trial}})}{||\widetilde{Y}_H(p_{\texttt{trial}})||}. \tag{9}$$

until a certain stop criterion is verified. It is worth noting that the path may not be the most straightforward

(straight line). Also, after calibration on the dedicated images, it was found that following the path for 6 iterations was satisfactory (fixed parameter). Further increasing it or using an automatic criterion (such as $||\widetilde{Y}_H(p_{\texttt{trial}})||$) did not yield significant gain in accuracy or computational cost.

3.3.3. Point cloud clustering

First, the $\overline{Y}_D$ image is converted into a point cloud through the identification of 2D local maxima points across all slices. The minimum allowed distance between any two points within a slice is set to $\delta = 0.4$ mm, and the minimum peak intensity is set to 0.1. Subsequently, the initialization points p_0 (from one slice annotation input data) are added to the point cloud for proper labeling.

Then, the `DBSCAN` [39] algorithm is employed to cluster these points, utilizing the ε-neighborhood parameter set to 0.3 mm, with a requirement of a minimum of three points within a neighborhood to form a cluster. The value of ε-neighborhood is tuned on the calibration images, and it should not exceed the proximity criteria of $\delta = 0.4$ mm.

Only the clusters containing the initial points p_0 are retained as reliable. The points within these meaningful clusters are arranged based on their cross section order (along the corresponding yarn axial direction). Given the constraint of allowing only one point per slice within a centerline, when multiple points from the same cluster occupy the same slice, a replacement with their barycenter is then applied.

3.4. Quality evaluation

In the following subsections, two methods are presented for appraising the different models on the predicted centerlines. The first one is a "classical" performance metric that compares the known ground truth with the model predictions. The second one is novel and seeks to provide an estimation of the quality of the predicted yarns without a ground truth. The latter is meant to be used when applying the model to unseen and not annotated images.

3.4.1. Ground-truth based metric

The Hausdorff distance is used to compare two yarn centerlines P and Q. It is defined as

$$\mathcal{H}(P,Q) = \max\left(\max_{p\in P}\min_{q\in Q} d_\theta(p,q),\ \max_{q\in Q}\min_{p\in P} d_\theta(p,q)\right). \tag{10}$$

Here, $d_\theta(p,q)$ is a weighted Euclidean distance aligned according to the orientation field θ (see Section 3.1.2) between a pair of centerline points p and q. The weights are set so that distances along the major ellipse axes of the elliptical cross section is reduced by half.

Then, a yarn centerline P is considered as *correctly predicted* if $\mathcal{H}(P, P^*) < 0.5$ for corresponding ground truth yarn path P^*. The proposed ground-truth-based metric is the percentage of *correctly predicted* centerlines (respecting the Hausdorff condition) with respect to the total number of centerlines.

It is worth noting that this metric is very strict by construction. Indeed, one could encounter only one point in a given yarn path (composed of hundreds of points) that does not satisfy the distance criterion, yet the entire yarn would be considered *wrongly predicted*.

3.4.2. Quality evaluation without ground-truth

In real-world scenarios, obtaining the ground truth is often challenging, and visually evaluating thousands of predicted yarns can be a tedious task. To tackle this issue, we have devised a method for estimating the quality of the predicted centerlines. This approach allows us to filter-out (probably) poorly predicted results without using manual annotation and visual verification.

The proposed evaluation method relies on the grayscale values of the network output image. We noticed that a higher value in the voxels of the inferred distance transform indicates a higher probability of an existing centerline in those voxels. We propose to use this observation as an empirical benchmark to assess the post-processing (*e.g.*, tracking or clustering). Applying a common threshold to all voxels results in a binary quality mask, which ensures a quality above the chosen threshold for all the centerline voxels within the mask. The higher the threshold, the more likely the selected voxels are to be on a centerline path. The resulting quality mask does not use any prior knowledge on the textile architecture; therefore, the method can be easily applied for a material with different woven patterns. False positively predicted regions (at the voxel level) are typically associated with yarn collisions. Therefore, to enhance the robustness of the method, a collision removing algorithm for the quality mask is also exploited.

Let us first define the threshold mask M_T from the cross-correlation result $\overline{Y}_D$ (see Equation (8)), as

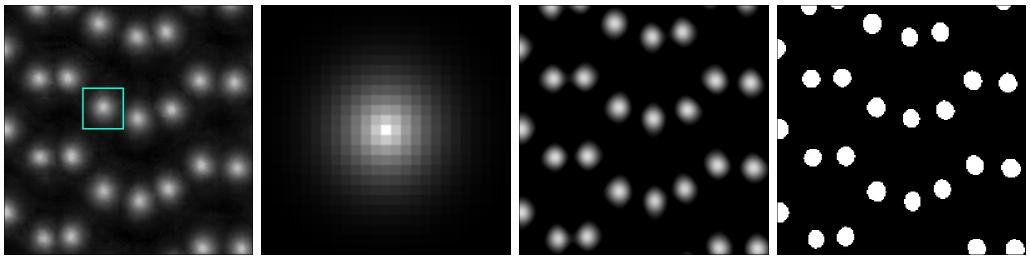
$$M_T(x, t_{||}, r) = \begin{cases} 1, & \text{if } \overline{Y}_D(x, r) > t_{||}, \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where r is the radius of the template-peak D_{ideal}, and $t_{||}$ is a brightness threshold parameter. An example of the image $\overline{Y}_D$ for corresponding warp inference $\widetilde{Y}_D$ are displayed respectively in Figure 11c and Figure 11a. The resulting M_T mask is demonstrated in Figure 11d. Then, a centerline, denoted by P, is assumed to be

positively estimated if all its points p are within the *quality mask*

$$M_{||}(p) = 1, \quad \forall p \in P \tag{12}$$

where $M_{||} \subseteq M_T$ is also a binary mask, which aims to remove regions of potential collision of the collinear yarn paths, as it is explained in Figure 12a. To compute mask $M_{||}$ starting from M_T, Algorithm 2 is proposed. An example of a set of yarn paths with collinear collision is illustrated in the Figure 12c.



(a) (b) (c) (d)

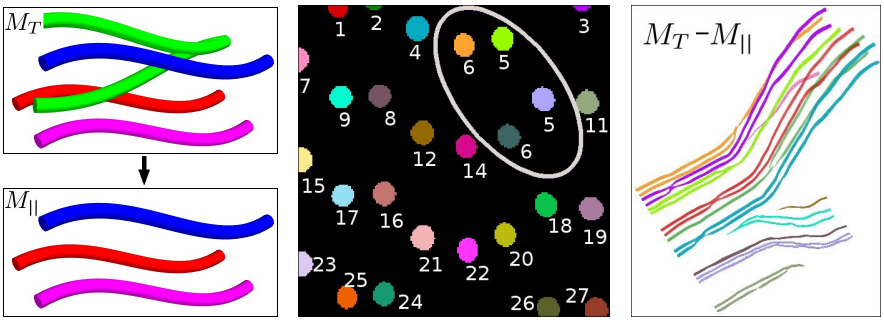
Figure 11: Creation of initial quality mask: (a) prediction $\widetilde{Y}_D$ with cross-correlation window dimensions indicated by the cyan rectangle, (b) template D_{ideal} (size of 25×25 pixels) used for the cross-correlation and (c) warp cross-correlation $\overline{Y}_D$ and (d) its corresponding mask M_T.

Algorithm 2: Collinear collisions detection

Data: Initial mask binary image M_T
Result: Quality mask image $M_{||}$
Remove small connected regions in M_T with a volume less than 10^4 voxels;
Compute 3D labels for M_T;
Assign a yarn region as "with collision" if the same 3D label appears more than once in a 2D slice;
Assign a yarn region as "complete" if it exists on both the first and the last slices;
Compute $M_{||}$ as a combination of yarns that are "complete" and "without collision";
return $M_{||}$

The parameter values, $r = 0.5$ mm and $t_{||} = 0.4$ were found using a grid search on the same set of five µ-CT images employed for the post-processing parameters setup (Section 3.3). The selection criterion was the maximization of the count of centerlines within $M_{||}$ (condition 12), while complying with the Hausdorff constraint. The determined values of r and $t_{||}$ are applied for quality estimation on the "unseen" ten images of the *test dataset*.

Centerlines, which are close to the ground-truth are called *actually positive*, otherwise *actually negative*. Similarly, the centerlines that meet the *quality criteria* (12), are designated as *positively estimated*, while the remaining ones as *negatively estimated*. A *positively estimated* centerline is a *true positive* (resp. *false positive*) if it is *actually positive* (resp. *negative*). Similarly, a *negatively estimated* centerline is a *true*



(a) (b) (c)

Figure 12: Touching yarn paths localization for warp direction: (a) schematic explanation of the mask $M_{||}$ derivation, (b) 2D (by colors) and 3D (by numbers) labeling of M_T mask, (c) an example of resulting collinear yarns with collisions.

negative (resp. *false negative*) if it is *actually negative* (resp. *positive*).

Here, the classical *precision* and *recall* metrics are used for the quality criteria evaluation. *Precision* is the ratio of *true positive* centerlines to the count of *positively estimated* ones, while *recall* is the ratio of *true positive* to the count of the *actually positive* instances. In the introduced terminology, the goal of the mentioned above grid search is the optimization of $(r,\ t_{||})$ by maximizing the *recall* with a constrained *precision*.

4. Results

Let us recall that the total 25 µ-CT images were split as follows: 10 for the neural network training, 5 for calibration of the post-processing parameters and quality evaluation method, and the remaining 10 "unseen" images for the final (non biased) results presented here.

The outcomes derived from different models are presented in Table 1. This ablation study includes variation of the target image channels (Y_D, Y_{DD} and Y_{HD}), model dimensionality (2D, 2.5D and 3D), dataset sampling parameters (N_{ch}, S_{ch}) and loss function (WMSE vs MSE).

The $\mathbb{DD}$ training (Y_{DD} target) is performed only once, with the WMSE loss function. In contrast, training of $\mathbb{HD}$ and $\mathbb{D}$ (Y_{HD} and Y_D targets) each undergo seven experiments with different configurations. Specifically, four experiments of $\mathbb{HD}$ and $\mathbb{D}$ models are related to the 3D U-Net, involving WMSE or MSE loss functions, while the remaining ten tests concern 2.5D U-Net models with WMSE. The latter is explored with five distinct combinations of N_{ch} and S_{ch} sampling parameters.

For example, the 2.5D model configured with $N_{ch} = 7$ and $S_{ch} = 21$ employs seven channels with a step of 21 voxels for selecting the next slice (*i.e.*, 20 voxels of space between two consecutive slices). This setup

represents a 2.5D sample depth of 127 voxels, which closely aligns with the corresponding size used for the 3D models (128^3 voxels). Similarly, models with (N_{ch}, S_{ch}) set to $(127, 1)$ and $(19, 7)$ yield a "complete" sample size of $127 \times 128 \times 128$, approximating the sample size of 3D models.

The model with the Y_{DD} target uses 826 samples, divided into 80% for the training and 20% for the validation, whereas Y_D and Y_{HD} models are applied to each sample twice (for warp and weft directions), resulting in a dataset approximately twice as large (1558 samples also distributed in 80% for the training and 20% for the validation). All the training samples are extracted from the larger non-overlapping sub-volumes (160^3 voxels), which are identical for each model.

The resulting accuracy in Table 1 is calculated for each of the ten test images as a percentage of *correctly predicted* centerlines, based on the Hausdorff distance (as described in Section 3.4.1). Subsequently, the minimum, maximum and mean values of the ten images are compiled.

The predicted centerlines are derived from the inference outcomes through the application of the *distance transform tracking*. The rationale behind opting for this method lies in its slightly superior accuracy when juxtaposed with alternative approaches such as *heat flow tracking* and `DBSCAN` clustering. Considering the post-processing of $\mathbb{HD}$ inference (model ID **14**), the accuracy stands at 96.8% for *distance transform tracking*, 95.3% for *heat flow tracking* and 92.3% for `DBSCAN` clustering. The model **14** is selected for the calibration and comparison of post-processing methods, as it is the most accurate model for which all three post-processing methods are applicable.

The number of training epochs presented in Table 1 also includes the last 20 epochs, referred to as "post early stopping" phase, during which the validation loss does not decrease. In addition, the inference time denotes the average duration per µ-CT image, considering both the warp and weft directions. The training and inference steps are performed on an NVIDIA Tesla P100 GPU with 16GB of VRAM.

In Figure 13a, an example of the voxel-wise inference evaluation is presented. Here, warp yarns prediction $\widetilde{Y}_{D1}$ (obtained by model ID **13**) is compared with the corresponding ground truth image (Y_{D1}). To highlight the prediction error, $\widetilde{Y}_{D1}$ is converted into three channels RGB image as follows

$$\widetilde{Y}_{\texttt{RGB}}(x) = \widetilde{Y}_{D1}(x) M_{\texttt{RGB}}(x), \tag{13a}$$

$$M_{\texttt{RGB}}(x, t_Y) = \begin{cases} [1,0,0], & \text{if } \widetilde{Y}_{D1}(x) - Y_{D1}(x) > t_Y, \\ [0,0,1], & \text{if } \widetilde{Y}_{D1}(x) - Y_{D1}(x) < -t_Y, \\ [0,1,0], & \text{otherwise.} \end{cases} \tag{13b}$$

Figure 13a shows $\widetilde{Y}_{D1}(x)$ in the `Green` channel ($M_{\texttt{RGB}} = [0,1,0]$), when inference is close to the ground truth, i.e. $|\widetilde{Y}_{D1}(x) - Y_{D1}(x)| \leq t_Y$. Since, $Y_{D1}(x) \in [0,1]$, the tolerance value of $t_Y = 0.1$ has been chosen. In cases where the difference (between ground truth and inference) is above the tolerance ($|\widetilde{Y}_{D1}(x) - Y_{D1}(x)| > t_Y$), the corresponding values are displayed in the `Red` or `Blue` channel, according to the error sign. As a result, blobs that appear almost entirely green indicate a near-perfect match between the inference and the annotated peaks, while the size of the red or blue spots reflects the magnitude of discrepancy between the inference and the ground truth peaks. It is important to note that, in some instances, the predicted peaks may better indicate the centers of their cross sections compared to the manual annotation. Therefore, a small distance between the ground truth and predicted peaks is still considered indicative of accurate prediction.

The distribution of the Hausdorff distance ($\mathcal{H}$), exemplified by the results of the model ID **14**, is depicted in Figures 13b and 13c. This distributions account for all yarns across 10 test images in both warp and weft directions, totaling 6950 yarns. The histograms illustrate that the majority of predicted centerlines exhibits Hausdorff distances to their ground truth counterparts, ranging from 0.1 to 0.3 mm (409 yarns are less than 0.1 mm). In order to enhance the clarity of the histogram visualization while maintaining its integrity, the centerlines with $\mathcal{H}$ distances surpassing 1 mm are clipped (3% of the total), highlighting a distinct second mode at this threshold (see Figure 13b). The count of centerlines with $\mathcal{H} \geq 1$mm for each CT image falls within the range of $[5, 65]$. In order to maintain the histogram resolution in the small error regions, log-scaled histograms is also used for the models ID **14** and ID **15**, respectively in Figure 13c and Figure 13d.

To illustrate the localized distribution of the Hausdorff distance between the predicted (model ID **14**) and the ground-truth centerlines, we perform a 2D interpolation representing the weft yarns direction, as depicted in Figure 14a for one of the ten images. In Figure 14a, all the centerlines with distance $\mathcal{H}$ above 1 mm are clipped to 1 mm to avoid the colors scaling.

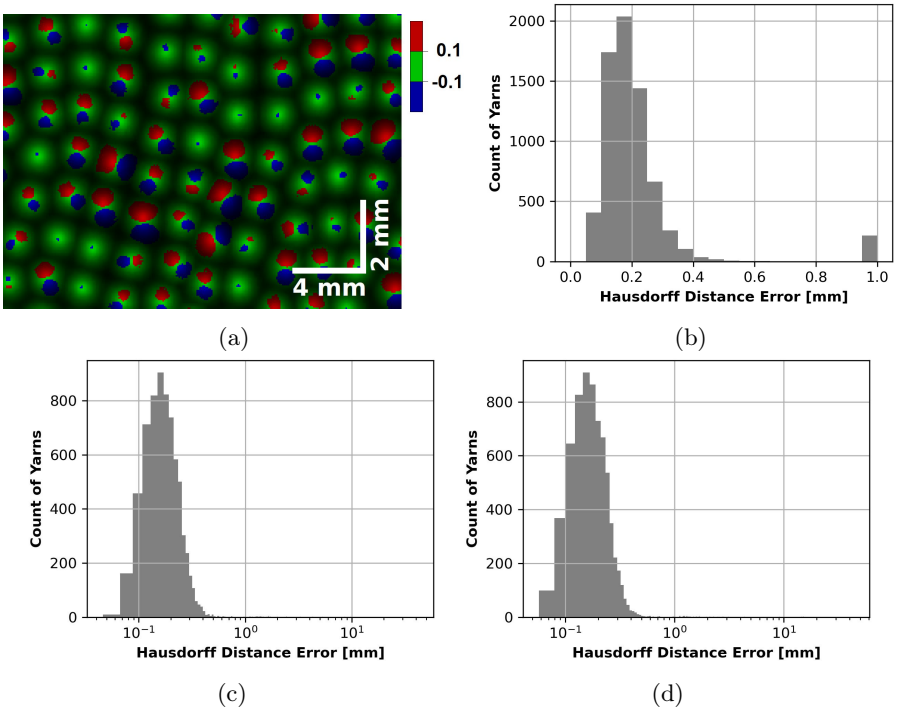
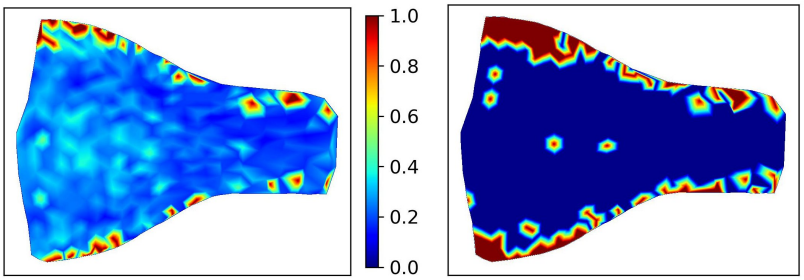


Figure 13: (a) Voxel-wise inference peak discrepancy ($\widetilde{Y}_{D1}$ evaluated against its ground truth) ; Distribution of Hausdorff distance between predicted and annotated centerlines for (b) model ID **14** (bin width 0.05 mm) and (c) its log-scaled representation and (d) for model ID **15** in log-scaled as well.

Similarly, the results of the *quality mask* evaluation (without using annotation data) are presented in Figure 14b. In this figure, the blue area corresponds to the predicted centerlines that lie entirely within the *quality mask*, while the red area visualizes the rest (non-confident) instances.

A comparison between Figure 14a and Figure 14b reveals that the estimated non-confident zone aligns with the actual poorly predicted regions, primarily situated at the boundary of the structure. The areas with similar colors in both figures represent *true estimation* (blue for *true positive* and red for *true negative*). Conversely, the *false estimated* regions exhibit distinct colors. The red regions in Figure 14b, which appear blue in Figure 14a, indicate *false negative* cases.



(a) (b)

Figure 14: Local accuracy evaluation (model ID **14**, one image, 577 weft yarns): (a) Interpolation of Hausdorff distance (in mm) between predicted and ground-truth centerlines; (b) *Quality mask* evaluation.

Precision and *recall* metrics for the *quality mask* evaluation, computed across all 10 test images, are presented in Table 2. Various post-processing methods are included in this assessment. The achieved *precision* is nearly 100% across all the tests, indicating a low likelihood of *false positive* occurrences in the quality estimation given by the *quality mask*. The *recall* value is comparable for *distance transform* and *heat flow tracking* due to the similarity of these methods. However, `DBSCAN` clustering yields a higher *recall* (over 93%), demonstrating that its centerlines are relatively close to the labeled instances used for *quality mask* derivation in Algorithm 2.

Summarizing the computational efficiency of the methods, it is noteworthy that the *distance transform tracking* takes approximately two minutes per µ-CT image and the *quality mask* evaluation requires about three minutes. These timings are evaluated using twelve parallel cores on an *Intel Xeon E5* CPU. All the algorithms are implemented in *Python*, and *ImageJ/Fiji* software [41] is widely used for image visualization.

Table 1: Models description and performance metrics (for ten "unseen" images)

Model						Times [h]		Accuracy [%]		
ID	**Target**	**Dim.**	**(N_{ch}, S_{ch})**	**Loss**	**Epochs**	**Training**	**Infer.**	**Min**	**Max**	**Mean**
1	Y_D	2D	-	WMSE	123	9	4	28.1	90.1	74.5
2	Y_{HD}	2D	-	WMSE	131	21	4	49.6	91.6	75.6
3	Y_D	2.5D	127, 1	WMSE	156	12	7	70.6	94.7	87.3
4	Y_D	2.5D	19, 7	WMSE	162	14	4	0	16.5	4.8
5	Y_D	2.5D	19, 3	WMSE	108	8	4	74.5	93.8	86.5
6	Y_D	2.5D	7, 21	WMSE	142	24	4	54.9	92.3	81.6
7	Y_{HD}	2.5D	127, 1	WMSE	119	21	7	68.4	93.2	86.3
8	Y_{HD}	2.5D	19, 7	WMSE	180	31	4	61.6	95.2	86.0
9	Y_{HD}	2.5D	19, 3	WMSE	119	21	4	64.3	94.6	85.0
10	Y_{HD}	2.5D	7, 21	WMSE	142	23	4	54.9	92.3	81.6
11	Y_D	3D	-	MSE	119	100	4	60.6	93.6	80.8
12	Y_{HD}	3D	-	MSE	193	183	4	57.7	93.8	81.1
13	Y_D	3D	-	WMSE	119	100	4	92.7	99.7	**97.1**
14	Y_{HD}	3D	-	WMSE	101	95	4	91.0	99.6	**96.8**
15	Y_{DD}	3D	-	WMSE	102	47	2	93.1	98.7	**96.6**

Table 2: *Quality mask* evaluation for different post-processing methods, obtained using 3D models (for ten "unseen" images)

Model ID	Post-processing	Precision [%]	Recall [%]
13	DT Tracking	100.0	89.19
13	`DBSCAN`	100.0	93.24
14	DT Tracking	99.97	90.17
14	HF Tracking	99.97	88.48
14	`DBSCAN`	99.97	93.48

5. Discussion

The conducted ablation study primarily analyzes the impact of dimensionality, dataset sampling, loss function and target channels nature on robustness, accuracy and computational expenses (see Table 1). The analysis of the *distance transform-based tracking* reveals a better accuracy compared to *heat flow-based tracking* and `DBSCAN` clustering.

The average accuracy for 2D and 2.5D models is 75%, whereas that of 3D models is largely superior at 95%. Moreover, the span accuracy (difference between min and max) decreases when 3D information is embedded into the model since, in average, it is about 52% and 28% for the 2D and 2.5D models respectively and falls to 18% for all the 3D models.

Furthermore, a positive correlation exists between accuracy and number of channels in 2.5D models (81.6%, 85.9% and 86.8% in average accuracy for respectively 7, 19 and 127 channels). However, the improvement is not linear. The accuracy rises by 4.3% when the number of channels is increased three-fold (from 7 to 19), whereas the gain is hardly 1% when the channels are increased by about seven times (from 19 to 127).

Concerning the loss functions, it is noteworthy that, for 3D models, the accuracy with WMSE grows by up to 16 *points* (from 81% to 97%) in average and over 30 *points* from the minimal value compared to MSE. Clearly, ensuring that the optimization process focuses on regions closest to the yarn centerline is beneficial.

So, the results show that adding 3D information and weighting the loss function as well has a significant influence on model performance. Nevertheless, this study reveals that it is possible to achieve quite good accuracy also with a 2.5D model, although its performance remains lower than that of a 3D model. Indeed, the best 2.5D model is the one with the highest number of channels (127 slices, ID **3**), akin to its 3D counterpart (ID **13**), since they use the same target images and loss function (WMSE) as well as an equivalent number of slices. The performance of this 2.5D model presents a drop of almost 10% for the mean accuracy (87.3% vs 97.1%) with about twice inference time (7 vs 4 hours), albeit around 10th of the training time (12 vs 100 hours). The explanation for this significant inference time could stem from the extensive processing required for 2.5D sampling in RAM and the subsequent transfer of data to GPU memory, while, during training, the time per epoch remains consistent across all 2.5D models since they originate from the same dataset at the GPU level.

Moreover, it is important to note that the training time for 3D models significantly exceeds those of 2.5D models, arising from 3D convolutions (from two days for ID **15** until more than a week ID **12**). The ID **5** model (2.5D with Y_D target) has the smallest training time (eight hours), that is twelve times faster

than the 3D model ID **14** with a close number of epochs (about 100). Globally, for all training, the number of epochs is within the range from 100 to 200, not emphasizing any particular dependency on the model characteristics.

Comparing models which differ only in the target image channels $\mathbb{D}/\mathbb{HD}$ and identical in the other setups (dimensionality, loss, N_{ch}, S_{ch}), the analysis does not demonstrate a significant difference in accuracy, excepted the 2.5D model ID **4** with a 0% min accuracy and hardly 5% in average. Our assumption is that this model must have fallen to a sub-optimal local minimum given the predetermined random seed (with the *determinism* TensorFlow option). To this end, an additional training was performed for this configuration without *determinism*. The second test yielded a mean accuracy of 82.2%, which is comparable to other 2.5D model results.

The training dataset of $\mathbb{DD}$ model (ID **15**) has two times fewer samples, consequently, its training time is about two times less (2 vs 4 days) than the other two 3D models with the same WMSE loss (*i.e.*, ID **13** and ID **14**) and with close number of epochs. For the same reason, its inference time is also two times less (2 vs 4 hours). Besides, having the most interesting training and inference times, the accuracy of the $\mathbb{DD}$ model is very close to the other two 3D models (only less than 1% lower). This model also exhibits the smallest accuracy span of 5.6%. These advantages make the $\mathbb{DD}$ model more appealing for practical applications compared to the considered alternatives among 3D models.

Evaluation results based on the *quality mask* approach indicate that `DBSCAN` achieves a recall rate of 93%, surpassing other methods and underscoring its robustness (see Table 2). This indicates that the quality measure is well-suited to the output of `DBSCAN` and incorporating the `DBSCAN` results into the quality measure can enhance the robustness of the evaluation. Remarkably, all three post-processing methods exhibit a precision close to 100%, affirming the resilience of the proposed *quality mask* approach. Furthermore, the calibrated parameters (r, $t_{||}$), derived exclusively from *distance transform tracking* results (based only on five dedicated volume images), demonstrate consistent performance across ten "unseen" images, even when employed with the other two post-processing methods. Notably, the *quality mask* parameters are fine-tuned exclusively using inferences of $\mathbb{D}$ and $\mathbb{HD}$ 3D models (ID **13** and ID **14**). For alternative model types, such as $\mathbb{DD}$ and 2.5D models, a recalibration of parameters may be necessary for providing an accurate assessment.

Alternatively, one can assess the quality of a yarn P (without ground-truth) by considering its minimal value of the cross-correlation result, *i.e.*, $\min_{p\in P} \overline{Y}_D(p)$, as a *quality index*. This approach provides a continuous quality index ranging from zero to one, in contrast to the binary mask value of Equation (12) that we used here. However, a drawback of such alternative is its potential insensitivity to the possible

collisions between collinear yarns. In perspective, a more sophisticated approach might prove advantageous for the collision excluding, incorporating not only the estimation of the distance to the centerline but also an assessment of the distance to the nearest boundary between yarns.

Both with and without ground-truth evaluations reveal that the accuracy near from structure boundaries tends to be comparatively lower than that of internal value regions, as illustrated in Figure 14. Besides, fine-tuning training focused on those boundary regions could enhance the performance of deep learning models; while, in the current sampling strategy, boundary samples, often featuring unique or rare crop of textile patterns, are utilized for training in the same manner as the internal ones.

6. Conclusions and perspectives

In this paper, we introduce a methodology for reconstructing the yarn centerlines from µ-CT data of textile composites. First, yarn path determination provide useful topology information in order to control whether the woven pattern is conformed to the expected one. In addition to the quality control, the other main goal is to provide a woven mesoscale description geometry of composite parts so as to improve the subsequent mechanical simulations. The proposed method yields high-precision results, validated on a large volume of annotated test data.

A wide range of deep learning models and post-processing methods is explored with the aim of optimizing performance on extensive datasets. This exploration involves training target images with various combinations of channels computed based on centerlines distance transform. While all combinations demonstrate comparable accuracy, the configuration that utilizes warp and weft distance transform images as a two-channel target stands out, achieving a significant twofold reduction in inference time.

Efficiency of 2D, 2.5D and 3D models are compared using the same training and inference dataset samples. The 3D models attain the highest mean accuracy up to 97%, whereas the best 2.5D model reaches about 87%. These results clearly demonstrate a significant performance advantage of 3D models over both 2.5D and 2D models. Furthermore, adapting the loss function to the specific characteristics of the target image notably enhances the accuracy of the results. The substantial variance observed in the accuracy of the models, characterized by the notable contrast between maximum and minimum values, underscores the critical role of extensive and diverse datasets in deriving accurate conclusions regarding models performance.

Various post-processing methods, aimed at converting inference results into centerlines, are proposed. The effectiveness of the simple post-processing algorithms is primarily achieved due to the high accuracy of neural network models. Additionally, a novel approach for the yarn quality evaluation, which operates

without relying on ground-truth data and utilizes a *quality mask*, is introduced. The high precision of this method is demonstrated through a comparison with the assessment based on the "unseen" annotated data.

While this work aims to provide input for mechanical modeling, it is important to acknowledge certain limitations. The current framework does not predict the detailed yarn cross sections. To obtain a realistic geometry of the textile structure, defining the yarn boundary is essential for a comprehensive mechanical model. However, identifying the yarn boundary directly from the input CT data is more challenging compared to centerline identification. Therefore, this sets the stage for future research works. At the current stage, the extracted centerline geometry can be used to generate a structure with uniform cross sections. A realistic deformed cross-sectional shape can be achieved, for example, through an expansion simulation, such as applying a thermal load on the structure until the desired yarn volume fraction is reached.

Another limitation is the applicability of 2.5D and 2D models to a non-interlock 3D woven fabrics with complex yarn entanglements. However, this limitation does not apply to the proposed 3D models, which can be easily adapted to predict 3D distance transforms in the original resolution with subsequent centerline tracking.

Looking ahead, there is room for significant enhancement in the proposed deep learning strategy through the enrichment of the training dataset. This could involve the integration of new types of textiles that encompass broader features such as diverse weaving patterns and yarn geometry.

In the field of post-processing advancements, a promising avenue for improvement lies in the development of a dedicated graph neural network [42] tailored to cluster the predicted point cloud of distance transform peaks into the centerline points. This approach aims to enhance the precision and efficiency of the post-processing stage, providing a more streamlined and accurate representation of yarn structures.

Furthermore, in the pursuit of refining the evaluation stage, deep learning techniques can be leveraged for the estimation of centerlines quality. This entails the development of a model capable of assessing the quality of the post-processing result without the necessity for a "manual" verification. Such an approach has the potential to significantly generalize the evaluation process, representing a notable advancement in assessment techniques and enhancing trustworthiness in the model accuracy.

CRediT authorship contribution statement

Y. Sinchuk: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft. **Y. Wielhorski:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing

- review & editing. **A. Mendoza:** Conceptualization, Formal analysis, Investigation, Methodology, Resources, Writing - review & editing. **S. Blusseau:** Conceptualization, Formal analysis, Investigation, Methodology, Resources, Writing - review & editing. **S. Velasco-Forero:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgement

This work was funded by the DGAC through the MERCY project.

References

[1] Y. Wielhorski, A. Mendoza, M. Rubino, S. Roux, Numerical modeling of 3D woven composite reinforcements: A review, Composites Part A: Applied Science and Manufacturing 154 (2022) 106729.

[2] S. V. Lomov, D. S. Ivanov, I. Verpoest, M. Zako, T. Kurashiki, H. Nakai, S. Hirosawa, Meso-FE modelling of textile composites: Road map, data flow and algorithms, Composites science and technology 67 (2007) 1870–1891.

[3] M. Sherburn, Geometric and mechanical modelling of textiles, Ph.D. thesis, University of Nottingham United Kingdom, 2007.

[4] Y. Mahadik, S. R. Hallett, Finite element modelling of tow geometry in 3D woven fabrics, Composites Part A: Applied Science and Manufacturing 41 (2010) 1192–1200.

[5] S. Green, M. Matveev, A. Long, D. Ivanov, S. Hallett, Mechanical modelling of 3D woven composites considering realistic unit cell geometry, Composite Structures 118 (2014) 284–293.

[6] N. Naouar, D. Vasiukov, C. H. Park, S. Lomov, P. Boisse, Meso-FE modelling of textile composites and X-ray tomography, Journal of Materials Science 55 (2020) 1–21.

[7] C. Hang, H. Cui, H. Liu, T. Suo, Micro/meso-scale damage analysis of a 2.5D woven composite including fiber undulation and in-situ effect, Composite Structures 256 (2021) 113067.

[8] Y. Sinchuk, Y. Pannier, M. Gueguen, M. Gigliotti, Image-based modeling of moisture-induced swelling and stress in 2D textile composite materials using a global-local approach, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 232 (2018) 1505–1519.

[9] V. Pidou-Brion, Y. Le Guilloux, Active yarn meshes for segmentation on X-ray computed tomography of textile composite materials at the mesoscopic scale, Composite Structures 281 (2022) 115084.

[10] B. Wintiba, D. Vasiukov, S. Panier, S. V. Lomov, K. E. M. Kamel, T. J. Massart, Automated reconstruction and conformal discretization of 3D woven composite CT scans with local fiber volume fraction control, Composite Structures 248 (2020) 112438.

[11] W. Huang, P. Causse, V. Brailovski, H. Hu, F. Trochu, Reconstruction of mesostructural material twin models of engineering textiles based on micro-CT aided geometric modeling, Composites Part A: Applied Science and Manufacturing 124 (2019) 105481.

[12] J. Bénézech, G. Couégnat, Variational segmentation of textile composite preforms from X-ray computed tomography, Composite Structures 230 (2019) 111496.

[13] G. Fourrier, A. Rassineux, F.-H. Leroy, M. Hirsekorn, C. Fagiano, E. Baranger, Automated conformal mesh generation chain for woven composites based on CT-scan images with low contrasts, Composite Structures (2023) 116673.

[14] Y. Sinchuk, O. Shishkina, M. Gueguen, L. Signor, C. Nadot-Martin, H. Trumel, W. Van Paepegem, X-ray CT based multi-layer unit cell modeling of carbon fiber-reinforced textile composites: Segmentation, meshing and elastic property homogenization, Composite Structures 298 (2022) 116003.

[15] A. Rassineux, Robust conformal adaptive meshing of complex textile composites unit cells, Composite Structures 279 (2022) 114740.

[16] I. Straumit, S. V. Lomov, M. Wevers, Quantification of the internal structure and automatic generation of voxel models of textile composites from X-ray computed tomography data, Composites Part A: Applied Science and Manufacturing 69 (2015) 150–158.

[17] Y. Liu, I. Straumit, D. Vasiukov, S. V. Lomov, S. Panier, Prediction of linear and non-linear behavior of 3d woven composite using mesoscopic voxel models reconstructed from X-ray micro-tomography, Composite structures 179 (2017) 568–579.

[18] Z. Lin, D. Wei, A. Gupta, X. Liu, D. Sun, H. Pfister, Structure-preserving instance segmentation via skeleton-aware distance transform, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2023, pp. 529–539.

[19] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[20] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19, Springer, 2016, pp. 424–432.

[21] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, UNet++: A Nested U-Net Architecture for Medical Image Segmentation, Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support 11045 (2018) 3–11.

[22] C. Stringer, T. Wang, M. Michaelos, M. Pachitariu, Cellpose: a generalist algorithm for cellular segmentation, Nature Methods 18 (2021) 100–106.

[23] U. Schmidt, M. Weigert, C. Broaddus, G. Myers, Cell detection with star-convex polygons, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20,

2018, Proceedings, Part II 11, Springer, 2018, pp. 265–273.

[24] J. P. C. Bertoldo, E. Decencière, D. Ryckelynck, H. Proudhon, A modular u-net for automated segmentation of x-ray tomography images in composite materials, Frontiers in Materials 8 (2021) 761229.

[25] M. A. Ali, Q. Guan, R. Umer, W. J. Cantwell, T. Zhang, Efficient processing of μCT images using deep learning tools for generating digital material twins of woven fabrics, Composites Science and Technology 217 (2022) 109091.

[26] K. Zheng, H. Chen, C. Wu, X. Zhang, Z. Ying, Z. Wang, Z. Wu, Z. Pan, B. Qiu, An improved dataset augmentation approach for deep learning-based XCT images segmentation in layered composite fabric, Composite Structures 317 (2023) 117052.

[27] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, Lecture Notes in Computer Science 11209 (2018) 432–448.

[28] Y. Pannier, P. Coupé, T. Garrigues, M. Gueguen, P. Carré, Automatic segmentation and fibre orientation estimation from low resolution X-ray computed tomography images of 3D woven composites, Composite Structures 318 (2023) 117087.

[29] A. Mendoza, R. Trullo, Y. Wielhorski, Descriptive modeling of textiles using FE simulations and deep learning, Composites Science and Technology 213 (2021) 108897.

[30] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[31] S. Blusseau, Y. Wielhorski, Z. Haddad, S. Velasco-Forero, Instance segmentation of 3D woven fabric from tomography images by deep learning and morphological pseudo-labeling, Composites Part B 247 (2022) 110333.

[32] A. Koptelov, A. Thompson, S. R. Hallett, B. El Said, A deep learning approach for predicting the architecture of 3d textile fabrics, Materials & Design 239 (2024) 112803.

[33] Y. Sinchuk, P. Kibleur, J. Aelterman, M. N. Boone, W. Van Paepegem, Geometrical and deep learning approaches for instance segmentation of cfrp fiber bundles in textile composites, Composite Structures 277 (2021) 114626.

[34] P. Soille, Morphological image analysis: principles and applications, Springer, 1999.

[35] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, Nature methods 17 (2020) 261–272.

[36] A. Mendoza, J. Schneider, E. Parra, E. Obert, S. Roux, Differentiating 3D textile composites: A novel field of application for Digital Volume Correlation, Composite Structures 208 (2019) 735–743.

[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: kdd, 1996, pp. 226–231.

[40] K. Briechle, U. D. Hanebeck, Template matching using fast normalized cross correlation, in: Optical Pattern Recognition XII, volume 4387, SPIE, 2001, pp. 95–102.

[41] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al., Fiji: an open-source platform for biological-image analysis, Nature methods 9 (2012) 676–682.

[42] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, J. M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Transactions on Graphics (tog) 38 (2019) 1–12.