HAL open science

Instance segmentation of 3D woven fabric from tomography images by Deep Learning and morphological pseudo-labeling

Samy Blusseau, Yanneck Wielhorski, Zyad Haddad, Santiago Velasco-Forero

To cite this version:

Samy Blusseau, Yanneck Wielhorski, Zyad Haddad, Santiago Velasco-Forero. Instance segmentation of 3D woven fabric from tomography images by Deep Learning and morphological pseudo-labeling. 2021. hal-03345132v1

HAL Id: hal-03345132

https://hal.science/hal-03345132v1

Preprint submitted on 16 Sep 2021 (v1), last revised 13 Oct 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Instance segmentation of 3D woven fabric from tomography images by Mathematical Morphology and Deep Learning methods

Samy Blusseau[a,*], Yanneck Wielhorski[b], Zyad Haddad[a] and Santiago Velasco-Forero[a]

[a] *Center for Mathematical Morphology, MINES ParisTech, PSL Research University, 35, rue Saint Honoré - F-77305 , Fontainebleau, 77305, Cedex, France*

[b] *Safran Aircraft Engines, Rond Point Réné Ravaud-Réau, Moissy-Cramayel, 77550, France*

ARTICLE INFO

Keywords:
Deep learning
Instance segmentation
3D angle-interlock fabric
X-ray computed tomography

Abstract

In the field of composite materials, mesoscale modeling based on X-ray computed tomography are very widespread nowadays. This descriptive method requires image processing to identify the different objects within the material. In the present study, two different instance segmentation approaches are proposed: 1) a method based on Mathematical Morphology and 2) a Deep Learning one. Both methods are applied to determine the yarns paths and their envelopes. We succeed in both tasks on a dry 3D ply-to-ply angle-interlock fabric at low compaction level. In absence of manual labelling of the yarns envelopes, we manage to train a Deep Convolutional Neural Network (DCNN) on the pseudo-labeling provided by the morphological method, improving the latter and showing the potential of deep learning for image segmentation in this context, when yarns cross-sections are labelled by distance functions. At a higher level of compaction, we also manage to recover the yarn paths thanks to deep learning.

1. Introduction

The growing interest of composite materials in many industrial fields is mainly motivated by their good mechanical properties combined with a quite low density. There exists a great variety of composite materials according to the kind of fibrous reinforcement and polymer employed for the impregnation which is then cured until final consolidation. Before the impregnation and the curing steps, the weaving and forming processes are also crucial steps of the global composite material manufacturing. Indeed, while the weaving could lead to miscellaneous topological mistakes, forming could also lead to undesired phenomena such as waviness, wrinkles or locking [13, 33, 34, 25]. Such phenomena are very linked to the type of reinforcement and could have a great influence on the mechanical properties and on the lifetime (service time). It is important to add that such composite reinforcements could be described at two different scales, hence increasing the complexity of these entangled materials: the microscale relative to the fibers and the mesoscale corresponding to the yarn's scale. Thus, the need for effective computational tools to model textiles and predict the physical properties of these entangled materials, is obvious. Besides, they allow to massively study sensitive parameters such as woven patterns, yarn morphologies and fiber volume fractions. Overall, two main families of textile modelings could be distinguished:

samy.blusseau@mines-paristech.fr (S. Blusseau); yanneck.wielhorski@safrangroup.com (Y. Wielhorski); zyad.haddad@mines-paristech.fr (Z. Haddad); santiago.velasco@mines-paristech.fr (S. Velasco-Forero)
ORCID(s):

1. the predictive approaches, which are purely numerical methods aiming to predict the morphology of the textile in a given state, and
2. the descriptive approaches, which are μ-CT image-based modeling methods where the virtual model is deduced from a volume of images of existing material combined with segmentation methods.

In such approaches, the micro and sub-mesoscale ("intermediate scale") modeling could be achieved in limited proportion compared to the whole material to avoid high computational costs. Hence, the common scale is the mesoscale for which the main issue is to determine accurately the yarn shape (its trajectory and envelope) during the forming.

In a **predictive approach**, the modeling of woven fabric at the fiber scale seems unrealistic because of the great quantity of entities entailing a large number of degrees of freedom. For this reason, a method based on virtual fibers (1 virtual fiber = a bundle of real fibers) produced a good description of micro-mechanisms by performing sub-mesoscale simulations [59, 51, 38, 32, 20, 22, 19, 15, 21]. Thus, the wide literature on woven fabric modeling leads to meso and sub-mesoscale textile generating softwares developed for two decades such as WiseTex [31, 55], DFMA [56, 59], TexGen [51] and Multifil [20, 21]. Hence, the "as-molded" (as-manufactured) state could be computed through simulations of the mesoscale forming process with the yarn mechanical behaviour [14].

The **descriptive approaches** mainly rely on high-resolution X-ray computed tomography (μ-CT) which is widely used nowadays in non-destructive control fields. This great tool was used for design and process modeling of aerospace composites [42] as it provides spatial information allowing to follow the yarn (fiber) life from its manufacturing to its forming process. For instance, yarn deformations induced by the manufacturing process were measured thanks to the Digital Image Correlation (DIC) [36]. Similarly, Digital Volume Correlation (DVC) has proven to be a suitable technique for establishing a novel differentiating tool based on full field measurements of 3D woven composites. This approach provided a quantifiable description of woven by assessing metric differences (yarn deformations) and topological differences (missing yarns) with respect to a reference one [35].

The segmentation method leading to a "digital material twin" can be classified as a region-based segmentation since the main goal is to distinctly separate different classes of objects inside the images. However, nowadays, automated procedures for computer tomography data transfer into finite elements models are missing in the literature [5]. Different mesoscale segmentation approaches are described in the literature, depending on the labeling of images. They range from a simple bimodal segmentation [3] (single global thresholding value for the whole volume to separate only resin/porous media from yarns), to multimodal segmentation using clustering [57, 30] or optimization algorithms [7]. Moreover, many studies have shown that segmentation requires a trade-off between quite high enough resolution (*e.g.*, from 1μm [40, 41] to 25μm [3, 4, 7]) and contrast [18] to highlight the orientation and/or shape of yarns. Regarding the analysis of μ-CT tomographies, we shall distinguish between three categories of image processing tasks. First, the

object detection consists in locating the yarns in images, without finding their boundaries. In this case, possible outputs may be the yarns centers or their bounding boxes. Second, the **semantic segmentation** produces a partition of the pixels (or voxels) of an image into several classes (*e.g.*, yarn, resin, porosity). This is a more complex and more accurate task, as it requires to find the object boundaries. Finally, the **instance segmentation** goes a step further, as different objects of the same class are labelled differently (*e.g.*, $yarn_1$, $yarn_2$, etc.). To achieve these tasks, many suitable methods are widely presented in the literature, as reviewed next. The principal directions and the degree of microstructural anisotropy of yarns can be measured through image local descriptors like the structure tensor [40, 53, 30, 39, 57, 58]). This was successfully applied to separate the warp and weft yarns. However, this approach is efficient for suitable contrast and resolution. In some cases, for binder identification, additional information related to the voxel orientation is required [58]. A post-processing using mathematical morphology operations (closing, convex hull) is then necessary to close and fill the yarn cross-sections as the results obtained with these methods remain a binary mask.Overall, the degree of anisotropy and the average grey value have proven to be suitable to separate entities like the weft and warp yarn groups as well as the resin/air voxels. However, they do not directly give a complete identification of each tow (shape and label) requiring a full separation. Indeed, some issues remain when the yarn is in contact with its neighbour, especially for high compacted states leading to high fiber volume fraction. Even at high resolution, yarns compressed against each other look merged together. In such a case, defining proper contours of yarns accurately remains a complex task.

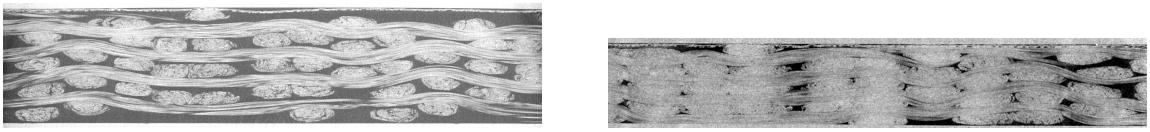
The idea is then to apply clustering algorithms to identify all the subdomains. The two-parameters K-means clustering algorithm has been studied and shown issues to emphasize yarn clusters [53]. Furthermore, a supervised scheme seems to be the most efficient for clustering classification [1]. So, many works have applied a method based on a machine learning clustering Gaussian Mixture Model (GMM) followed by a 3D mapping-smoothing algorithm ([57, 30, 58]). These studies showed that the complete yarn separations still require additional final steps (*e.g.*, alphaShape function and element orientation [30, 58] or manual user intervention on first binary image [57]). Bénézech and Couégnat [7] proposed an image-based variational segmentation relying on an iterative global-local approach driven by a heuristic algorithm. The goal is to optimize the voxelized geometric model with respect to the μ-CT, through three terms: a) the gray levels, b) the local directions and, c) the interpenetrations between yarns (regularization term). Additionally, yarn paths were described manually by a given number of points, then interpolated using B-splines, whereas it is not necessary for the yarn sections in the slices. Indeed, it is an expansion method since they initialize the model with small ellipses which grow up just after a few number of iterations. Sinchuk et al. [52] developed another variational approach consisting in finding an approximation of the initial image by minimizing the Mumford–Shah functional, also defined by three terms: a) the distance between the initial image and its approximation, b) an approximation of the smoothness within sub-regions and c) the total length of the edges.

Over the past few years, deep learning (DL) approaches, mainly Deep Neural Networks (DNNs) have yielded a new generation of image segmentation models with remarkable performance improvements on popular benchmarks resulting in what many regard as a paradigm shift in the field [47, 23]. Most of these benchmarks include thousands or even millions of training data, composed of images with their respective ground-truth annotation [29]. For our application the annotations could be an "a priori" shape of each phase, in a given slice of the image stack, targeted by the segmentation (*e.g.*, yarn path and/or sections, porosities). The creation of initial datasets is an essential point. Indeed, it could be a very time-consuming procedure and operator dependent, as labelling is usually provided by manual annotations [24, 2, 52]. This novel semantic segmentation technique was proposed to separate weft, warp and resin phases inside 2D [2, 52] and 3D [2] reinforcements. The authors explored convolutional neural networks like U-Net [52] and Residual architectures [2]. In both cases only the in-plane slices were used for the training, leading to some issues when the neural networks were applied to the 3D fabric. Moreover, U-net was also applied recently [6] to perform a semantic segmentation of a rod-shaped SiC–SiC composite, identifying four material phases: fibers, matrix, pores, and environmental barrier coating (EBC). Finally, in [37] a Mask R-CNN network was trained on synthetic but realistic images generated by a U-net from simplistic labelling. By this two-steps approach, the authors achieve instance segmentation of yarns represented as finite element meshes without the need of manual labelling. The present paper also deals with the yarn segmentation problem as an **instance segmentation**. Two approaches are considered here: morphological and deep learning methods. The studied textile is a 3D ply-to-ply interlock dry reinforcement. CT-images of the same sample at two different compaction states are available and shown in section 2. The main ideas of our morphological and deep learning segmentation approaches are detailed in section 3. Then, the study focuses on the detection of yarn paths for both samples and the assessment of results are presented in section 4. Finally, to predict the yarn envelopes without any prior manual annotation, the pseudo-labels provided by the morphological approach are used as a dataset to train a deep learning model. Yarn section detections are shown in section 5, along with discussions.

2. Material

2.1. Tomographic volume of 3D woven fabrics

The 3D woven fabric studied in this work is a ply-to-ply angle-interlock, composed of carbon fibre yarns. Two samples from this woven pattern were studied at two different compaction steps: an *uncompacted state* (Figure 1a), and a half-thickness sample (*i.e.*, 50% thickness of the initial uncompacted state), called *compacted state* (Figure 1b). Each sample was scanned with a GE Phoenix-Xray tomograph at a resolution of 20μm. The image size is $1698 \times 1814 \times 402$ voxels for the *uncompacted state* (Figure 1a) and $1725 \times 1545 \times 255$ voxels for the *compacted state* (Figure 1b). In these dry preforms the background is significantly darker than warp and weft yarn gray levels.



(a) Initial woven pattern (b) Compacted woven pattern

Figure 1: Images from high-resolution X-ray computed tomography of a 3D woven at two different compaction levels.

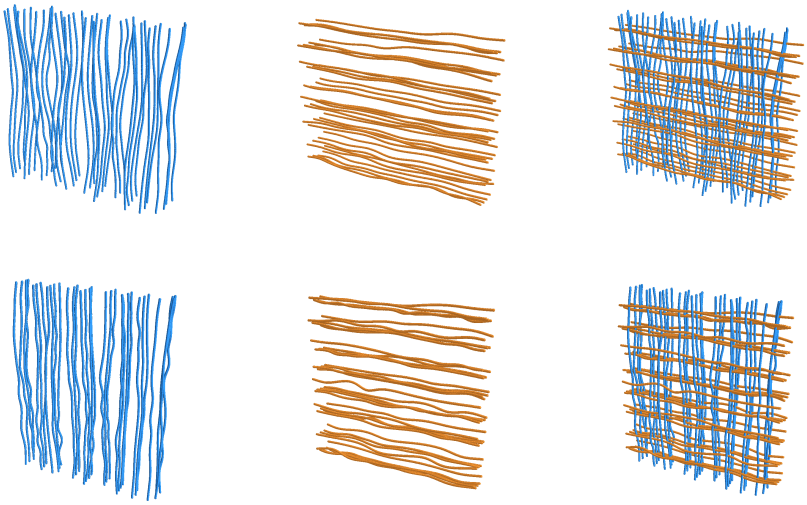


Figure 2: Ground truth yarn paths of the uncompacted dry preform (first row) and the compacted one (second row). Blue stands for warp paths and orange for weft.

2.2. Data labelling

For both tomographies described in the previous section, the yarn path (*i.e.*, the centers of the yarns) were available thanks to a manual labelling. Indeed, each warp and weft yarn path had been created by first clicking the center point of each yarn every 30 slices. The complete path was then interpolated by B-spline functions, as shown in Figure 2. This labelling was used for supervised learning as well as for quality assessment in the learning-free morphological approach. Note that for the *uncompacted state* (Figure 1a), 39 warp and 36 weft yarns were annotated. Concerning the second sample, *compacted state* (Figure 1b), annotations are available for 32 warp and 36 weft yarns. Concerning the contours of the yarn cross-sections, no labeling was available. Hence, we used the output of our morphological segmentation (Section 5.1) as labeling to train a deep learning model.

3. Methods

3.1. General concepts

3.1.1. Image segmentation goals

In this paper, we address two image processing issues. On the one hand, we aim at **reconstructing the yarn paths**, that is to say their centerlines. This boils down to detecting the location of yarns in each slice and, from these locations, inferring as many paths as there are yarns in the tomography. On the other hand, the second target is to segment the yarns themselves by detecting their individual **contours**. This kind of tasks is known as **instance segmentation**, since we do not only seek which pixels belong to a yarn, but also to identify that there are several *instances* of yarns in each slice, and finally in the whole tomography.

3.1.2. Classical image processing and Mathematical Morphology

Image processing researchers and engineers design and implement algorithms to process images automatically. These algorithms are meant either to improve the quality of images or retrieve information from them, or both. Among many others, two tasks are often addressed by image processing: object detection and image segmentation. The former consists in finding whether, and where, a given object appears in an image. The latter consists in splitting the image into meaningful regions, also called *segments*, each segment corresponding ideally to one object.

Before the recent and impressive success of deep convolutional neural networks (see section 3.1.3), most image processing specialists would try to define relevant features for the targeted task. For example, in the case of object detection, they would model the object by a set of features that characterize it: shape, size, colors, texture and so on. These features would be translated into *filters* applied to images to enhance the sought information. This very broad class of approaches is what we call here *classical* image processing, as opposed to Deep Learning based image processing, where the relevant features (hence, the filters) are learnt automatically by a neural network.

Mathematical Morphology (MM) [50] is one of the mathematical theories applied to classical image processing. Its particularity is to define non-linear filters based on two kinds of elementary operators: dilations and erosions. A dilation usually spreads the bright pixels of an image over local neighbourhoods, while an erosion spreads the dark pixels. Iterating these operators thus produces the diffusion of local maxima and minima of the image, allowing its topological analysis. The *watershed transform* [8, 10, 46] is built on this principle and is one of the most important contributions of Mathematical Morphology to image segmentation. Its main idea is to see an image as a topological surface. The pixels' coordinates stand for locations on the ground and their values for the elevation of the ground. Then segments provided by the watershed transform are the drainage basins of this surface, and the contours of objects should be the watershed lines. In this paper morphological methods will be presented, not only to be compared to, but also to be combined with deep learning.

3.1.3. Deep Learning for image processing

The first main step in the Deep Learning method is to train a dataset for the learning phase of the algorithm. Let $\{(x_k, y_k)\}_{1\leq k\leq n}$ be a training dataset of pairs, where each x_k is a data-sample from an input space $\mathcal{X}$ and y_k its corresponding label from the output space $\mathcal{Y}$. In our case, the x_ks are the slices of a tomography, like those of Figures 1a and 1b, and the y_ks are images of the same size where the yarn paths or the strands contours are highlighted. Learning from these data consists in using the examples $\{(x_k, y_k)\}_{1\leq k\leq n}$ to build a parametric map $\phi : \mathcal{X} \mapsto \mathcal{Y}$ that accurately predicts the label y_{n+1} of any new data sample x_{n+1}, that is: $y_{n+1} \approx \phi(x_{n+1})$. A DNN, in its simplest form, is a *compositional* map that may be written as $\phi_{f;\theta} := f^{(d)}(g^{(d)} \cdots f^{(1)}(g^{(1)}(\boldsymbol{X})))$ where each $f^{(l)}$ is a nonlinear function called *activation function* and each $g^{(l)}$ is usually an affine application defined by its weight matrix $\mathbf{W}^{(l)}$ and bias vector $b^{(l)}$: $g^{(l)}(\boldsymbol{Y}) = \mathbf{W}^{(l)}\boldsymbol{Y} + b^{(l)}$. Here θ denotes the set of affine parameters $[\mathbf{W}^{(1)}, b^{(1)}, \ldots, \mathbf{W}^{(d)}, b^{(d)}]$ and $f = [f^{(1)} \ldots f^{(d)}]$ the set of activation functions. The composed application $f^{(l)} \circ g^{(l)}$ is what we call a *layer* of the network. The term *deep* in DNN refers to neural networks with many layers, usually $d > 2$. As we can see from the equation above, the output of a layer becomes the input of the following one, except for the input and output layers, which are merely the input and output of the whole network. When the input of a layer is an image (or a stack of images) the linear part of the layer's affine transformation is actually a *convolution*, and therefore the matrix $\mathbf{W}^{(l)}$ is fully determined by a small number of parameters, called the *convolution kernel*. Such a layer is called a *convolutional layer*, and neural networks including such layers are referred to as *convolutional neural networks* (CNNs) [28].

CNNs can model complex non-linear relationships and have shown their goodness in different kinds of problems such as automatic speech recognition, image recognition, natural language processing, among others [17]. However, one of the fundamental points is the selection of an adequate *architecture* [16], *i.e.* the structure of composition between layers: number of layers, dimension of each layer's output, size of the convolution kernels, type of activation functions. In the training process, we would usually tune the parameters θ so as to minimise the difference between the labels (ideal maps y_k) and the CNN outputs (estimated maps $\hat{y}_k = \phi_\theta(x_k)$) at any training instance x_k, such that the difference goes to zero as the number of samples N increases. For that, we define a loss function $l(y_k, \hat{y}_k)$ that represents the difference between the labels and the CNN output. In an ideal case, we would like to find θ^* such that:

$$\theta^* = \arg\min_{\theta} E_{\forall x_k \in \mathcal{X}, y_k \in \mathcal{Y}} \left[loss(y_k, \hat{y}(x_k)) \right], \tag{1}$$

where $E_{\forall x_k \in \mathcal{X}, y_k \in \mathcal{Y}}$ denotes the expected value over all possible pairs of (x, y), which is impossible to calculate in most of scenarios . However, we can compute an approximation, called *empirical risk* [54], by averaging the loss function

(a) (b) (c)

Figure 3: First step of the morphological processing (either for the detection of yarn paths or the segmentation of yarns cross-sections). (a) Input slice. (b) Thresholded image. (c) Result after suppression of horizontal strands.

on a large set of training examples $(x_k, y_k)_{1 \leq k \leq N}$,

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=0}^{N} loss(y_k, \hat{y}(x_k)). \tag{2}$$

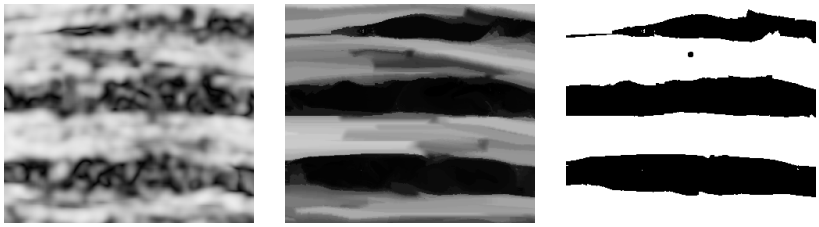
This minimization is usually done via *stochastic gradient descent* (SGD) [27]. SGD starts from a certain initial θ and then iteratively updates each parameter by moving it in the direction of the negative gradient with respect to the loss function. The computation of gradient with respect to the loss function is done via a direct application of the chain rule in networks, called *back-propagation* [48]. The term *stochastic* in SGD indicates that a random small number of training samples, called a *batch* is used in the gradient calculation. This reduces considerably the computational cost of the gradient evaluation. Additionally, by the law of large numbers, this stochastic gradient should be close to the full sample one, though with some random fluctuations. A pass of the whole training set is called an *epoch*. Usually, after each epoch, the error on a validation dataset is evaluated and when it stabilizes the training is complete.

3.2. Morphological method

3.2.1. Detection of yarn centers

We first address the task of finding the yarn path, or roughly speaking the center, of each strand in each slice of a tomography. This section describes a slice-by-slice method based on Mathematical Morphology (see Section 3.1.2) and designed for this purpose, to be compared to the deep learning approach of Section 3.3.3. The next section extends the present method to the segmentation of yarns cross-sections. The following algorithmic pipeline combines some of the newest methods in Mathematical Morphology as well as more classical ones. It counts two main steps and is applied to each slice independently (except for one regularization step).

Step 1 The first step consists in thresholding the slice and suppressing the horizontal yarns, as shown in Figure 3. The threshold is set automatically by Otsu's method [43]. The suppression of the horizontal components relies on the anisotropy of the structure tensors of the grayscale image (Figure 4a). More precisely, an adaptive opening of this anisotropy image is performed, as described in [12] (Figure 4b). This morphological filtering yields a mask of the



(a) (b) (c)

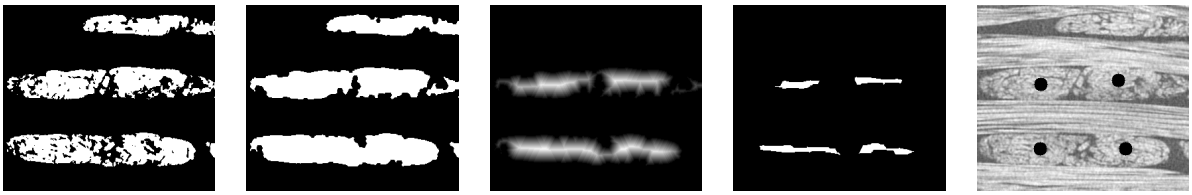
Figure 4: More details on the first step illustrated in Figure 3. (a) Anisotropy image corresponding to the input slice of figure 3. (b) Adaptive opening of the anisotropy image, as described in [12]. (c) Binary mask of the horizontal strands obtained by thresholding the opened image. The complementary version of this mask is applied to figure 3b to get figure 3c.

horizontal, elongated components (Figure 4c), which is subtracted to the original thresholded image to obtain a result similar to Figure 3c. The actual result of this process is a slightly noisier image. Regularized masks like Figure 3c, are produced by applying a median filter along the depth direction of the tomography, that is, across slices.

Step 2 The second step aims at recovering the barycenters of the yarns from a binary image like Figure 3c. This is achieved through two sub-steps, which both rely on classical morphological processing [49, 11]. First, filling small holes and removing small noise components, to get denser shapes like in Figure 5b. Second, separating touching yarns. In the first one, a *fill-holes* filter is applied, then a closing, a fill-holes again, and an area opening to remove small components. The resulting connected components are then of two types: those with "normal" size for a yarn, and those too large to be a unique yarn. In the first case, the yarn paths of the corresponding yarns are simply the barycenter of the connected component. In the second case, the *distance function* of the large connected component is computed (Figure 5c). The distance function of a component assigns to each of its pixels the distance to the background, that is, to the complementary of the component in the image. When a large component is composed of several touching yarns, a bottleneck usually appears near the edge between two yarns. The distance function is therefore lower in this area than in the center of a yarn. Hence, the barycenters of touching yarns can usually be recovered by selecting the pixels where the maxima of the distance function are achieved (Figure 5d). For the moment this was applied only to the uncompacted woven pattern presented earlier, because the first step of the method, which is crucial, does not work as well on compacted patterns.

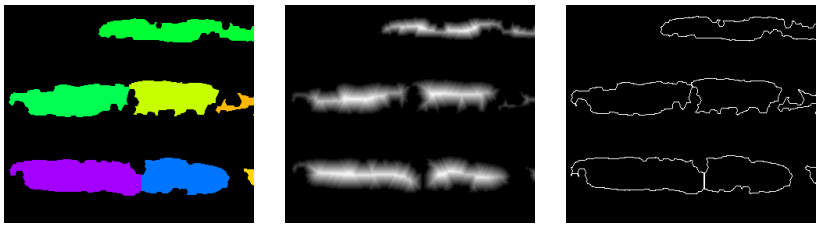
3.2.2. Segmentation of yarn sections

For the segmentation of yarns, our morphological approach is very similar to the one described in section 3.2.1 for the detection of yarn paths, as only the final steps differ. The first step is unchanged, and produces an output like figure 3c, from figure 3a as input. The second step also starts the same way, in order to get a binary mask of yarns like figure 5c.However, we are now interested in recovering the outlines of the yarns from that binary mask. When dealing with



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

Figure 5: Second step of the morphological detection of yarn paths. (a) Result of the previous step, as in Figure 3c, showing touching yarns. (b) Result of the morphological processing (fill-holes and closings) for the same crop. (c) The distance functions of the connected components that are too large to be single yarns. (d) Significant maxima of the distance function. (e) Barycenters of the regional maxima found in (d), as detections of the corresponding yarns. Note that the center of the remaining yarn, not marked in (e), is simply recovered as the barycenter of the corresponding component in (b).



(a) (b) (c)

Figure 6: Second step of the morphological segmentation of yarns cross-sections, for figure 3a as input. (a) Labelling of yarns after morphological processing (fill-holes and closings) of figure 3c. (b) Distance functions associated with each yarn. Brighter pixels indicate larger distances to the background. (c) Computed outlines of yarns with the morphological approach.

components with "normal" size for a yarn, the corresponding outlines are simply the pixels at distance one from the background (that is, pixels with value one in the distance function). Otherwise, connected components that are too large to be unique yarns are split apart by applying the watershed algorithm [9, 10] on their reversed distance functions, using as markers those detected by the method described in section 3.2.1. The overall result is a set of labelled components, each one corresponding to one yarn (Figure 6a). A convenient way to encode simultaneously the outline of the yarns and their ownership (in the case of touching strands), is to represent these components by their distance functions, as in figure 6b. Indeed, recall that the distance function of a strand assigns to each pixel the distance to the background, that is, the complementary of the strand in the image. Hence, by selecting the pixels at distance one, we recover the outline of the strand, as shown in figure 6c. At the same time, the distance function increases as we progress inside the strand until its centre, characterized by a local maximum of the distance function. Therefore, the distance function provides the outlines of objects as well as the objects they belong to. This is why we use this representation to train a deep neural network to segment strands, in section 3.3.4.

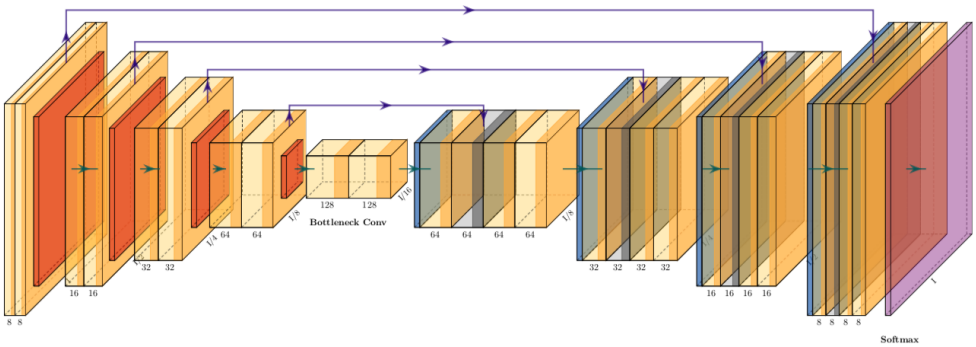


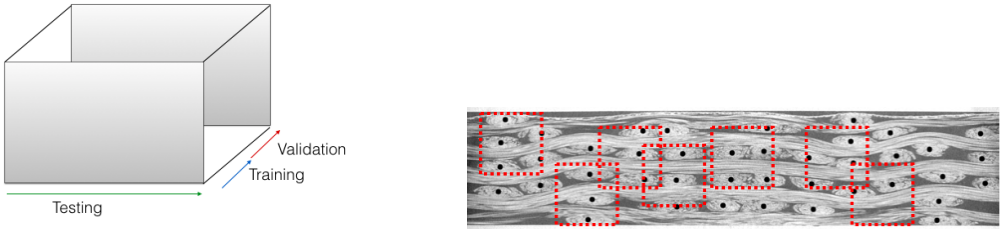
Figure 7: Architecture of U-net used for the yarn paths detection. The number of filters per layer is indicated.

3.3. Deep Learning method

Like in the morphological case, we first address the task of detecting the yarn paths and subsequently extend the approach to the segmentation of the strand section.

3.3.1. Architecture

For the different tasks presented here, a Deep Convolutional Neural Network (DCNN) architecture, called U-net [47], which was originally designed for biomedical image segmentation was used. The U-net architecture was originally designed for biomedical image segmentation and has become very popular for its impressive performances. A U-net maps an input image to an output image expected to show a segmentation of the input. As shown in figure 7, the computations performed by a U-net follow two main steps. In the first step, it keeps extracting and combining features from the input image while reducing the spatial resolution by pooling filters at each layer. This is called the *contracting path*. In the second step, called *expansive path*, it increases back the resolution using the accumulated features to produce a new image of the same size as the input, where only the desired objects should be highlighted. More precisely, for the yarn paths detection task, The contracting path consists of the repeated application of two 3×3 convolutions with a rectified linear unit (ReLU) activation and a 2×2 max pooling operation for downsampling. At each downsampling step we have doubled the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. In total our architecture has 490,993 parameters. For the segmentation of the cross-sections, the architecture was almost the same except that the first layer contains four filters instead of eight, and the last layer was a 1×1 convolution instead of a softmax layer as in figure 7.



(a) (b)

Figure 8: Illustration of the training protocol in the deep learning approach. (a) Partition of a tomography into training, validation and test slices. (b) Set of random patches superimposed on a slice.

3.3.2. Training Protocol

One of the challenges we faced to apply the deep learning approach is that we only dispose of two 3D images with their corresponding ground-truth, the yarns centers (see section 2). Hence, our training protocol is designed to avoid model over-fitting. First, we propose to extract squared 2D patches with a random center position (see some examples in figure 8b). The goal of random crops is to both prevent the selection bias and the redundancy of the data. We argue that allowing the overlap is essential for the network to learn features shared between patches. Second, we use one direction (warp or weft) during training and leave the other one for testing, as illustrated in figure 8a. The direction used for training is itself split into training and validation slices. More specifically, the *training set* is composed of 10000 patches randomly selected from two thirds of the slices in one direction, and the *validation set* is composed of 1000 patches randomly selected from the remaining third in the same direction. When not specified otherwise, training and test are run on images of the same level of compaction, but a mix can be done as we will see in section 4.3.2. During training, at the end of each epoch, the current state of the model is evaluated on the training set to give an idea of how well the model is *learning*, then the evaluation on the validation set tells how well the model is *generalizing*. As we have explained in section 3.1.3, the training is stopped when the error on the validation set does not reduce after some epochs [44]. Additionally, we run the training process on a single NVIDIA Titan XP GPU using ADAM optimizer [26], with a mini-batch size of 64 and a learning rate of 0.001. In the sequel, we applied this procedure to both compacted and uncompacted woven patterns simultaneously for the yarn paths detection (Section 3.3.3), and to the uncompacted pattern only for the segmentation of cross-sections (Section 3.3.4). Obviously, we note that after training, the deep learning model can be applied to any tomography slice.

3.3.3. Detection of yarns centers

Ground truth and labels: For this task, we had the necessary ground truth for supervised learning, namely the coordinates of the yarn paths in each slice (see section 2.2). These coordinates were translated into binary images where the yarn paths were marked by white squares, as shown in Figure 9. The neural network was thus trained to map a 2D

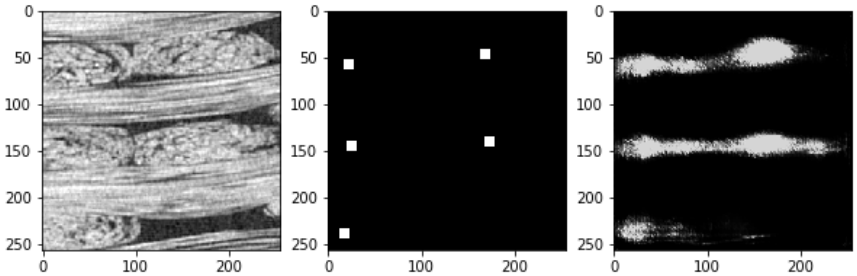


Figure 9: Left: Input patch. **Centre:** Ground truth markers. **Right:** Output by the U-net in an early learning stage.

input image showing strands, to an output image where the yarn paths were marked in white on a black background. **Loss function:** A performance measure commonly used for evaluating segmentation masks is the Jaccard index , also called the intersection-over-union (IoU) score [45]. In the case of a binary ground truth image y and its gray-scale estimation $\hat{y}$, this can be written

$$J(y, \hat{y}) = \frac{\epsilon + \sum_i y[i] \cdot \hat{y}[i]}{\epsilon + \sum_i^N (y[i] + \hat{y}[i] - y[i] \cdot \hat{y}[i])}, \tag{3}$$

where $y[i]$ and $\hat{y}[i]$ denote the values of images y and $\hat{y}$ at pixel i, N is the number of pixels in each image, and ϵ is a very small number preventing division by zero. Accordingly, the corresponding loss function to be employed in empirical risk minimization (2) is $loss(y, \hat{y}) := 1 - J(y, \hat{y})$.

3.3.4. Segmentation of yarns sections

The protocol followed here is very similar to the one in the detection of yarn paths. The training and validation sets are generated the same way. **Ground truth:** Contrary to the case of yarn paths, no ground truth was available for the outline of yarns. Indeed, such labelling is much harder and time consuming. Hence, in principle, supervised learning was not possible. However, during the tests on yarn paths we noticed that in an early learning stage the U-net would learn to roughly locate strands by extended markers more similar to the strand shapes than the small ground truth markers (see right hand image in figure 9). This observation raised our hope that the designed model was capable to learn the actual shapes of strands. We did not have any ground truth labelling of the strands outlines but we made the hypothesis that, if given an approximate segmentation of strands as ground truth, the U-net could learn to perform accurate segmentations provided the approximate labelling was unbiased. We chose to use as approximate labelling, that we will call *pseudo-labelling*, the output of the morphological pipeline described in section 3.2.2. Hence, the label images used to train the network here were the distance functions of yarns, as shown in figure 6b. **Loss function:** Here

the loss function is not the Jaccard index anymore but the mean square error (MSE):

$$loss(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y[i] - \hat{y}[i])^2 \tag{4}$$

where N is the number of pixels in the input y and output $\hat{y}$. Here y is an input slice and $\hat{y}$ the distance function used as a label (Figure 6b).

3.4. Post-processing: identification of yarn paths from independent detected points

The aim of this section is to determine the path of each yarn from the point clouds given by either the morphological or the deep learning detections of yarns centers (Sections 3.2.1 and 3.3.3). Indeed, note that since the predictions of the yarn centers or sections are made in each slice of the tomography independently, the labels of yarns have to be reconstructed further. As we will see in section 4 (see for example figures 11 and 12), most detected points are close to the actual yarn paths but some false positives and negatives detections also occur. Hence, we propose a post-processing of the output point clouds in order to recover the yarns paths and remove false positives. This procedure counts two main steps:

1. **tracking**: identifying candidate clusters of points to be yarn paths, and
2. **parametric modeling**: fitting each cluster with a parametric function.

1. Tracking: This first step consists in building *tracks*, that is to say sequences of detected points that are likely to belong to the same yarn. A track is initialized by one detected point in one slice, and extended forward and backward by points from subsequent and previous slices, complying with a smoothness criterion. To illustrate this step, let's detail one forward extension. Consider an incomplete track $[p_0, \dots, p_n]$, where each point p_i is identified by its coordinates (x_i, y_i, z_i), the last coordinate z_i denoting the slice to which the point belongs and (x_i, y_i) its 2D coordinates within the slice. Suppose we want to extend the track to a point from a subsequent slice s indexed by $z^* > z_n$. We denote by $p^* = (x^*, y^*, z^*)$ the closest point to p_n in s and define $dX = |x_n - x^*|$, $dY = |y_n - y^*|$, $dZ = |z_n - z^*|$. Then p^* is appended to the track (that is, $p_{n+1} \leftarrow p^*$) if $\sqrt{dX^2 + dY^2} \leq \min(\alpha dZ, \rho_{\max})$ for some parameters $\alpha > 0$ and $\rho_{\max} > 0$. This means p^* needs to fall within the intersection of a cone pointing on p_n and of slope α, and a cylinder of central axis z_n and radius $\rho_{\max}$. If this condition is not fulfilled by any point in s, then we look further for one in a subsequent slice. The range of subsequent slices to search (step, furthest slice) may vary. If no extension is found in this range, then the forward tracking is over. The backward tracking works in the same way. By changing the initial point and the range of subsequent (or previous) slices, we manage to build many tracks and with high probability the desired yarn paths are among them.

2. Parametric modeling: Among the set of tracks built in the previous step, some may be redundant (i.e. there might be several tracks for the same yarn) and others may be wrong, in the sense that they might not follow a yarn path. The goal of this step is to keep only accurate, mutually different tracks, using as criteria their fit to a parametric model which is expected to suit yarns paths. Visualizing the tomography should tell what model seems realistic. In the images considered here, the yarn path vary like sinusoids in one dimension (height) and are almost constant in the other (width). Therefore we solve for each track $[p_0, \dots, p_n]$ two least squares problems:

$$(a_0, \dots, a_K, b_1, \dots, b_K)^* = \arg \min_{(a_0,\dots,a_K,b_1,\dots,b_K)} \sum_{i=0}^{n} \left(y_i - a_0 - \sum_{k=1}^{K} \left[a_k \cos(\frac{2k\pi}{T} z_i) + b_k \sin(\frac{2k\pi}{T} z_i) \right] \right)^2$$

and

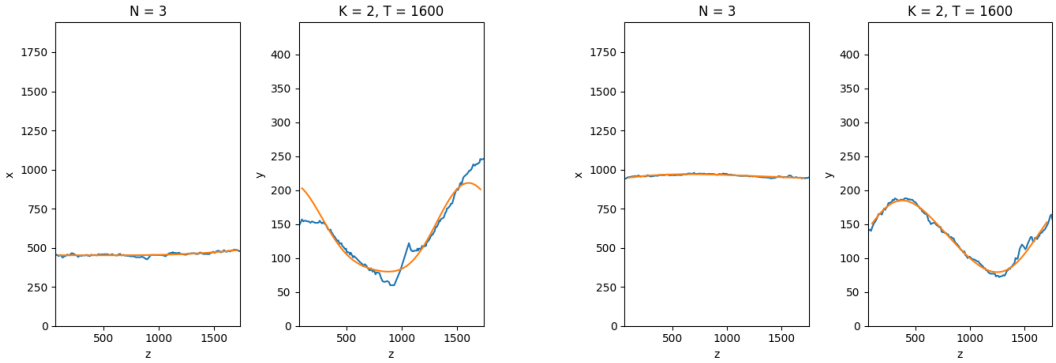
$$(c_0, \dots, c_N)^* = \arg \min_{(c_0,\dots,c_N)} \sum_{i=0}^{n} \left(x_i - \sum_{k=0}^{N} c_k z_i^k \right)^2,$$

where the period T, the number K of harmonics and the degree N of the polynomial are parameters to set. Only tracks achieving a small mean squared error for these two problems are kept. Besides, comparing the best fits of two tracks allows us to tell whether they correspond to the same yarn path or not. For instance, figure 10 shows two examples of yarn tracks (in blue) and their parametric fits (in orange). The polynomial's degree is $N = 3$, the number of harmonics is $K = 2$ and the period is $T = 1600$. Note that the left hand track (Figure 10a) is discarded because of its poor fit to the sinusoidal model in the height dimension, whereas the right hand track is kept (Figure 10b). In the latter, the parametric model is somehow a smoothed version of the track built on the initial noisy output. If one iteration of this process does not provide an adequate parameterization of all the yarn paths, it is possible to remove from the initial point cloud the points associated with the found fibers, and repeat the tracking and parametric modeling on the residual point cloud. As we will see in section 4, the initial point clouds output either by the morphological or deep learning method are more or less noisy. When the point cloud is accurate enough, the model based post-processing may converge in one or two iterations. When it is noisier, this may take a few more iterations, including a user interaction to select the correctly found yarn paths before computing the residual point cloud.

4. Results on the identification of neutral fibres

4.1. Quality assessment

For each of the methods presented earlier on the detection of yarns centers (Sections 3.2.1 and 3.3.3), the output is a set of coordinates representing the points detected as part of the neutral fibers in the tomography. We will call this set the *output point cloud*. We compare the output point cloud to the ground truth neutral fibres marked as another set of points, referred to as the *ground truth point cloud*. A ground truth point is considered as detected if at least one output point is close enough to it. Proximity is measured by the Euclidean distance in three dimensions. The threshold



(a) (b)

Figure 10: Two examples of tracks (in blue) and their parametric fits (in orange).

defining detection is 25 pixels. Finally, as the performance of a method is given by its ability to detect just the neutral fibres and no other object, we consider two quantities, *precision* and *recall*. The recall is the proportion of ground truth points that are detected by the methods, while precision is the proportion of true positives among the output point cloud. Both are numbers between zero and one, and the closer to one the better. From these two measures a unique score called F1 score is computed as follows $\text{F1} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$. Again, the closer F1 to one, the better.

4.2. Results on the uncompacted dry preform

4.2.1. Morphological approach

Table 1 sums up the results of the morphological approach in terms of recall and precision. Figure 11 is a 3D visualization of the detected points and their filtered version after post-processing (tracking and parameterization). Although the raw results are already very satisfactory, the post-processing allows to remove most false positives, as the improvement of precision shows. Additionally, it offers an identification of entire neutral fibres along the volume (hence the yarns paths), instead of independent points. Note that in both warp and weft directions, some neutral fibres were accurately reconstructed near the edges of the tomography although they were not labelled in the ground truth. They were discarded in the computation of precision, not to undermine it unfairly.

4.2.2. Deep learning approach

The results of the deep learning approach are presented in the same manner as in section 4.2.1. Recall and precision before and after post-processing are reported in Table 2, and the corresponding point clouds are shown in Figure 12. Comments on these results are very similar to those on the morphological approach (see section 4.2.1). Again, post-processing improves mostly the precision and allows the identification of distinct neutral fibres throughout the tomography. As previously, the detected neutral fibres that were not in the ground truth (although they do appear in the tomography) were not included in the recall and precision computation.

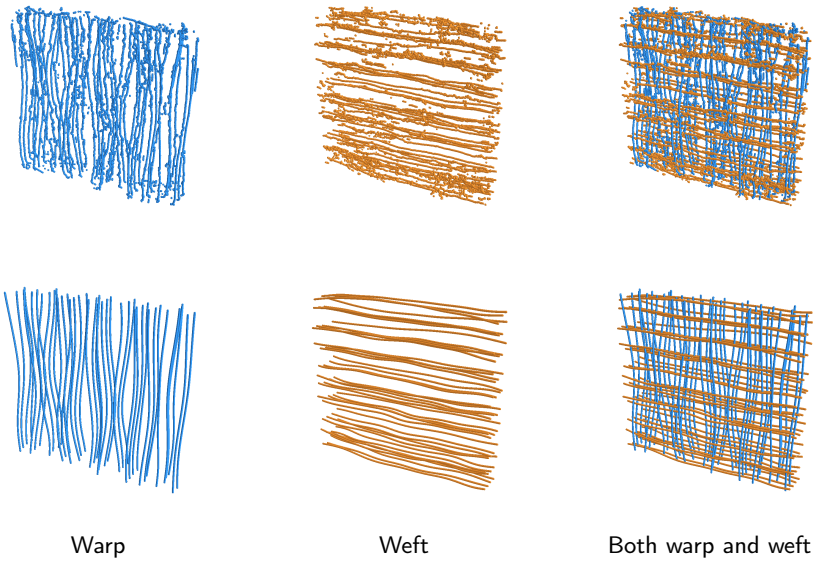


Figure 11: Points detected on the uncompacted dry preform by the morphological method (first row), and result after post-processing (second row). The latter can be compared to the corresponding ground-truth in the first row of Figure 2.

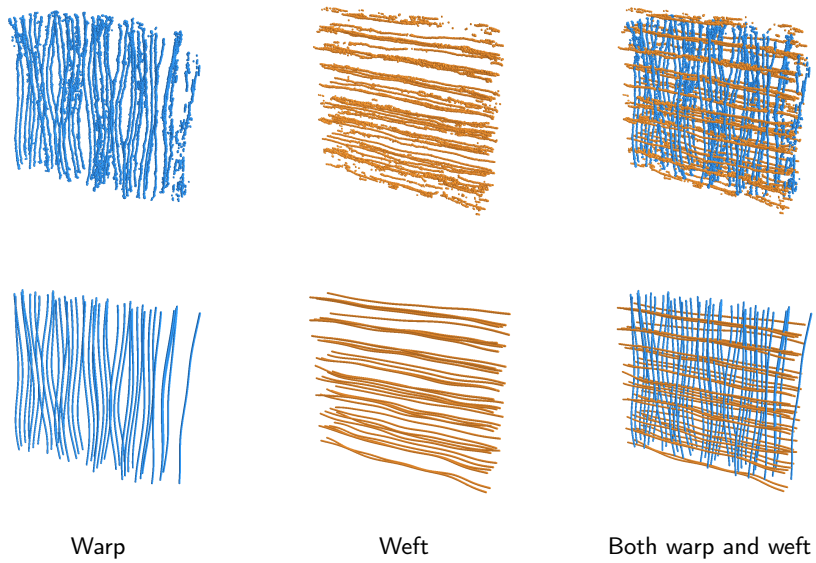


Figure 12: Points detected on the uncompacted dry preform by the deep learning method (first row), and result after post-processing (second row). The latter may be compared to the corresponding ground-truth in the first row of Figure 2.

4.3. Results on the compacted dry preform

4.3.1. Morphological approach

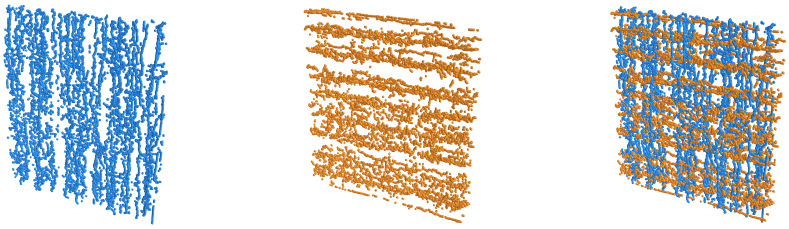
On the tomography of the compacted dry preform, the morphological approach does not perform as well as in the uncompacted case. This is due to the suppression of transverse yarns (as illustrated in the uncompacted case in Figure 3), which is not very accurate here. As we can see in Figure 13, the output point clouds are a lot noisier. This leads

Table 1
Results of the **morphological approach** on the uncompacted dry preform, before and after post-processing.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.983	.984	.979	.990
Precision	.916	.984	.879	.982
F1 score	.948	.984	.926	.987

Table 2
Results of the **deep learning approach** on the uncompacted dry preform, before and after post-processing.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.980	.996	.979	.987
Precision	.959	.996	.962	.987
F1 score	.969	.996	.971	.987



Warp Weft Both warp and weft.

Figure 13: Points detected on the compacted dry preform by the morphological method. The quality of these point clouds are too low to automatically recover the neutral fibres with our post-processing method.

to low precision levels, as shown in Figure 4. From these raw point clouds, the post-processing procedure described in Figure 3.4 needs significantly more iterations and user interactions to recover parametric neutral fibres. Therefore, we choose not to present the results of this post-processing step as it does not really correspond to a semi-automatic method.

4.3.2. Deep learning approach

In a first attempt to apply our deep learning approach to compacted dry preform tomography, we used as training data images from the same tomography but orthogonal direction with respect to the test direction (that is, we sampled patches from the weft direction and tested on warp or the other way around). The results in terms of precision and recall are presented in Table 3. They are significantly better than those of the morphological approach, but still not good enough to recover good performance after post-processing. In particular, the drop in recall is explained by the fact that the post-processing removes correctly detected points if they are not sufficient to recover the whole yarn path.

Therefore, we tried to add images from the non-compacted preform tomography, but still the orthogonal direction, to the training data. This improved significantly the quality of the raw output point clouds, as shown in Table 5. From the visualization of figure 14, top row, it appears that it is still quite noisier than for the initial uncompacted state. The post-processing step needed therefore a few more iterations and user interactions than with the uncompacted pattern, but the process still converged quickly. The final results after pre-processing, shown in figure 5 still need to be improved in the weft direction, where we still get a drop in recall for the reasons explained earlier.

Table 3
Results of the deep learning approach, before and after post-processing, on the compacted dry preform. Training: on the compacted pattern, orthogonal direction.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.949	.839	.854	.710
Precision	.851	.927	.782	.857
F1 score	.897	.881	.817	.777

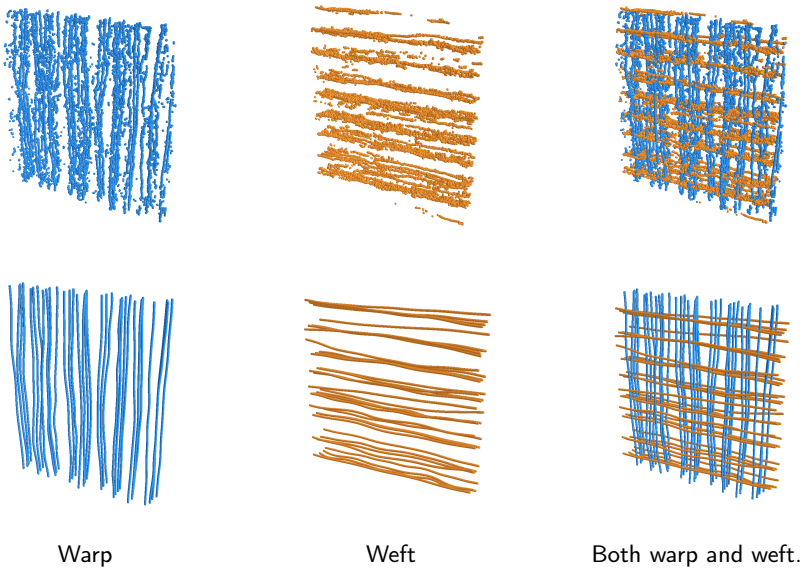


Figure 14: Points detected on the compacted dry preform by the deep learning method (first row), and result after post-processing (second row). The latter may be compared to the corresponding ground-truth in the second row of figure 2.

Table 4
Results of the **morphological approach** on the compacted dry preform. Because of the law precision levels, the proposed post-processing is inefficient and no results are reported here.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.903	-	.825	-
Precision	.752	-	.679	-
F1 score	.821	-	.745	-

Table 5
Results of the **deep learning approach**, before and after post-processing, on the compacted dry preform.
Training: on both compacted and uncompacted patterns, orthogonal direction.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.980	.983	.895	.799
Precision	.901	.985	.824	.829
F1 score	.939	.984	.858	.814

5. Results on the segmentation of yarn cross-sections

In this section we present the results obtained by both morphological and deep-learning approaches in the segmentation of yarns cross-sections. As explained earlier, so far the morphological segmentation was successfully achieved only on the uncompacted pattern, and the deep learning model could be trained only with the pseudo labels provided by the latter method. Hence, the segmentation results presented here only concern the uncompacted pattern.

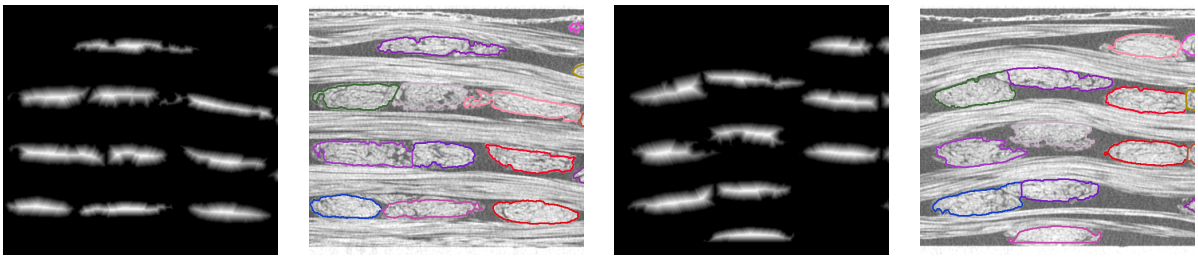


Figure 15: Instance segmentation of yarns by the morphological pipeline (crops from two different warp slices). The coloured lines are the found yarns outlines, corresponding to the distance functions shown on the left. Each color corresponds to a unique yarn throughout the tomography.

Regarding their quality assessment, it can only be made based on visual inspection, as we lack a real ground truth for this task. All the segmentation results are made available online, as indicated in the next subsections. Note that both approaches provide high resolution contours, contrary to [37] where the output is a finite element mesh.

5.1. Results of the morphological approach

Figure 15 shows the results of the morphological segmentation of yarns cross-sections for two slices. Note that in the images where the found contours are highlighted, each color corresponds to a unique yarn throughout the whole tomography, yielding a 3D instance segmentation. This was made possible by the parameterization of neutral fibres described in Section 3.4. We used the parameterized neutral fibres obtained from the morphological detections not only to ensure 3D consistency in the instance segmentation, but also as markers in the watershed transform allowing us to distinguish between touching yarns. For a visualization of the results on all slices, we provide the two following videos: https://drive.google.com/file/d/15rnxb1E4naZGX62geFAEdJ8929YnFxul/view?usp=sharing (warp) and https://drive.google.com/file/d/1bTSWTsSUDLg1I0IRj792h4xqxpM9P-_M/view?usp=sharing (weft). For a frame by frame examination, the corresponding sequences of slices can be found here: https://cloud.mines-paristech.fr/index.php/s/mFxfUt2NDYwLKI5. It appears that the labelling of contours (and the associated distance functions) is satisfactory in most cases, but the edges between touching strands are somewhat approximate and unstable.

5.2. Results of the deep learning approach

As described in Section 3.3.4, the labels used to train a U-net to segment yarns were the distance maps produced by the morphological method. We observe that the model does learn to compute an output similar to a distance function. A simple thresholding allows recovery of the connected components associated to the yarns, and therefore most contours except those separating touching strands. The latter are computed thanks to a watershed transform on the inverse of the output image, with the parameterized yarn paths of section 3.4 (those extracted by deep learning) as inner markers. In many cases the model seems to better approximate the shapes of touching

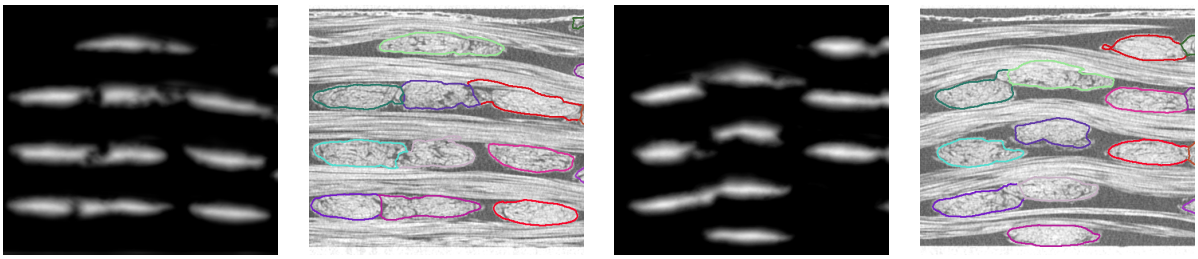


Figure 16: Instance segmentation of yarns by deep learning, shown on crops of two warp slices (after training on weft slices). The coloured lines are the found yarns outlines, corresponding to the deep U-net's output shown on the left, which look very similar to distance functions. Each color corresponds to a unique yarn throughout the tomography. For comparison, the input slices are the same as in Figure 15.

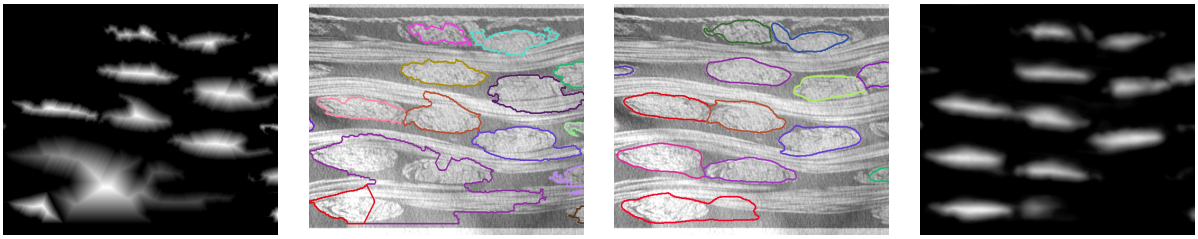


Figure 17: Comparison of morphological and deep learning results on a low quality warp slice of the uncompacted preform (the U-net was trained on weft slices). The two leftmost crops are the output of the morphological pipeline and the corresponding yarns contours. The two rightmost images show the output of the deep neural network and the corresponding contours, on the same crop. We can see the better robustness to artifacts of the DNN compared to the morphological pipeline.

strands, and is more robust to artifacts. Two examples of results are shown in Figures 16 and 17, and the whole results are presented in two videos available at the two following urls: https://drive.google.com/file/d/10d_eEWhBbpz8Z1Q4uoTLOAZq6t-4j9kz/view?usp=sharing (warp) and https://drive.google.com/file/d/1Sm_W6eaO4PYupmgtu-q0-17YbRNS8C06/view?usp=sharing (weft). For a frame by frame examination, the corresponding sequences of slices can be found here: https://cloud.mines-paristech.fr/index.php/s/mFxfUt2NDYwLKI5.

6. Discussion and conclusion

In this paper we have presented two image processing frameworks aiming at extracting the yarns paths and their envelopes from μ-CT tomographies of a dry 3D ply-to-ply angle-interlock fabric, producing an instance segmentation of yarns. One method is based on mathematical morphology and the other on deep learning. Regarding the extraction of paths, both morphological and deep learning approaches performed very well on a non-compacted pattern, but only the deep CNN output allowed to reach good results on a compacted pattern, after a training phase on mixed data (compacted *and* uncompacted images). This seems to advocate in favour of deep learning in this very hard case, even

though we suggest that a morphological (or other kind of) pre-processing of the data before training may help improve the deep learning performance.

Regarding the segmentation of yarns, only a qualitative assessment can be made as no manual labelling was available. Based on visual inspection, it seems that the morphological approach produces very satisfactory results on the uncompacted pattern although the accuracy should be improved where yarns are in contact. On this same task, we used the morphological output to train a deep learning model. From this experiment we can draw two important conclusions. First, a deep architecture such as U-net can learn to approximate the distance function of the yarns cross sections, which indicates that the distance function is a suitable labelling for such a supervised learning algorithm on this task. Second, on a visual basis, the U-net seems to perform at least as well as the morphological algorithm it learnt from. In particular, it behaves better at the edges between touching yarns and in slices affected by artifacts. This shows that deep learning may be used successfully even without a manual labelling of data, provided we can produce an automatic good quality pseudo-labelling.

In future works, we shall focus on the difficult issue of highly compacted fabrics. Indeed, the detection of yarns centers can still be improved and the segmentation of yarns cross sections is not satisfactory yet. To achieve a good quality pseudo-labelling, several strategies may be explored, such as pre-processing to regularize the data. More *a priori* information on the shape of yarns and the expected woven pattern, can also be exploited. Furthermore, the deep learning models we trained may be better adapted to compacted patterns through data augmentation strategies.

References

[1] Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Machine Learning 6, 37–66.

[2] Ali, M., Guan, Q., Umer, R., Cantwell, W.J., Zhang, T., 2020. Deep learning based semantic segmentation of μCT images for creating digital material twins of fibrous reinforcements. Composite Part-A: Applied Sciences and Manufacturing 139, 106–131.

[3] Ali, M., Umer, R., Khan, K., Bickerton, S., Cantwell, W., 2018. Non-destructive evaluation of through-thickness permeability in 3d woven fabrics for composite fan blade applications. Aerospace Science and Technology 82, 520–533.

[4] Ali, M., Umer, R., Khan, K., Cantwell, W., 2019. In-plane virtual permeability characterization of 3d woven fabrics using a hybrid experimental and numerical approach. Composites Science and Technology 173, 99–109.

[5] Auenhammer, R.M., Mikkelsen, L.P., Asp, L.E., Blinzler, B.J., 2021. Automated x-ray computer tomography segmentation method for finite element analysis of non-crimp fabric reinforced composites. Composite Structures 256, 113136.

[6] Badran, A., Marshall, D., Legault, Z., Makovetsky, R., Provencher, B., Piché, N., Marsh, M., 2020. Automated segmentation of computed tomography images of fiber-reinforced composites by deep learning. Journal of Materials Science 55, 16273–16289.

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

[8] Beucher, S., Lantuejoul, C., 1979. Use of watersheds in contour detection. int, in: Workshop on Image Processing, CCETT/IRISA, Rennes, France.

[9] Beucher, S., Meyer, F., 1990. Morphological segmentation. Academic Press.

[10] Beucher, S., Meyer, F., 1993. The morphological approach to segmentation: the watershed transformation. Marcel Dekker, Inc.. chapter 12. pp. 433–481.

[11] Bloch, I., Heijmans, H.J., Ronse, C., 2007. Mathematical Morphology. Springer Netherlands. chapter 14. pp. 857–944.

[12] Blusseau, S., Velasco-Forero, S., Angulo, J., Bloch, I., 2018. Tropical and morphological operators for signal processing on graphs, in: 25th IEEE International Conference on Image Processing (ICIP), pp. 1198–1202.

[13] Boisse, P., Zouari, B., Gasser, A., 2005. A mesoscopic approach for the simulation of woven fibre composite forming. Composites Science and Technology 65, 429–436.

[14] Charmetant, A., Vidal-Sallé, E., Boisse, P., 2011. Hyperelastic modelling for mesoscopic analyses of composite reinforcements. Composites Science and Technology 71, 1623–1631.

[15] Daelemans, L., Faes, J., Allaoui, S., Hivet, G., Dierick, M., Van Hoorebeke, L., Van Paepegem, W., 2016. Finite element simulation of the woven geometry and mechanical behaviour of a 3d woven dry fabric under tensile and shear loading using the digital element method. Composites Science and Technology 137, 177–187.

[16] Deng, L., 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing 3.

[17] Deng, L., Yu, D., 2014. Deep learning: methods and applications. Foundations and trends in signal processing 7, 197–387.

[18] Djukic, L., Herszberg, I., Walsh, W., Schoeppner, G., Gangadhara Prusty, B., Kelly, D., 2009. Contrast enhancement in visualisation of woven composite tow architecture using a microct scanner. part 1: Fabric coating and resin additives. Composites Part-A: Applied Sciences and Manufacturing 40, 553–565.

[19] Drach, A., Drach, B., Tsukrov, I., 2014. Processing of fiber architecture data for finite element modeling of 3d woven composites. Advances in Engineering Software 72, 18–27.

[20] Durville, D., 2010. Simulation of the mechanical behaviour of woven fabrics at the scale of fibers. International Journal of Material Forming 3, S1241–S1251.

[21] Durville, D., Baydoun, I., Moustacas, H., Périé, G., Wielhorski, Y., 2018. Determining the initial configuration and characterizing the mechanical properties of 3d angle-interlock fabrics using finite element simulation. International Journal of Solids and Structures 154, 97–103.

[22] Green, S., Long, A., El Said, B., Hallett, S., 2014. Numerical modelling of 3d woven preform deformations. Composite Structures 108, 747–756.

[23] He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-cnn, in: ICCV, pp. 2961–2969.

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

[25] Iwata, A., Inoue, T., Naouar, N., Boisse, P., Lomov, S.V., 2019. Coupled meso-macro simulation of woven fabric local deformation during draping. Composites Part A 118, 267–280.

[26] Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Bengio, Y., LeCun, Y. (Eds.), ICLR.

[27] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.

[28] LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361.

[29] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European conference on computer vision, Springer. pp. 740–755.

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

[31] Lomov, S., Gusakov, A., Huysmans, G., Prodromou, A., Verpoest, I., 2000. Textile geometry preprocessor for meso-mechanical models of woven composites. Composites Science and Technology 60, 2083–2095.

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

[33] Mathieu, S., Boisse, P., Hamila, N., Bouillon, F., 2014. Locking and stability of 3d woven composite reinforcements. Key Engineering Materials 611-612, 292–299.

[34] Mathieu, S., Hamila, N., Dupé, F., Descamps, C., Boisse, P., 2016. Stability of 3d textile composite reinforcement simulations: Solutions to spurious transverse modes. Applied Composite Materials 23, 739–760.

[35] Mendoza, A., Schneider, J., Parra, E., Obert, E., Roux, S., 2019a. Differentiating 3d textile composites: A novel field of application for digital volume correlation. Composite Structures 208, 735–743.

[36] Mendoza, A., Schneider, J., Parra, E., Roux, S., 2019b. Measuring yarn deformations induced by the manufacturing process of woven composites. Composites Part-A: Applied Sciences and Manufacturing 120, 127–139.

[37] Mendoza, A., Trullo, R., Wielhorski, Y., 2021. Descriptive modeling of textiles using fe simulations and deep learning. Composites Science and Technology , 108897.

[38] Miao, Y., Zhou, E., Wang, Y., Cheeseman, B.A., 2008. Mechanics of textile composites: Micro-geometry. Composites Science and Technology 68, 1671–1678.

[39] Naouar, N., Vasiukov, D., Park, C.H., Lomov, S.V., Boisse, P., 2020. Meso-fe modelling of textile composites and x-ray tomography. Journal of Materials Science 55, 16969–16989.

[40] Naouar, N., Vidal-Sallé, E., Schneider, J., Maire, E., Boisse, P., 2014. Meso-scale fe analyses of textile composite reinforcement deformation based on x-ray computed tomography. Composite Structures 116, 165–176.

[41] Naouar, N., Vidal-Sallé, E., Schneider, J., Maire, E., Boisse, P., 2015. 3d composite reinforcement meso fe analyses based on x-ray computed tomography. Composite Structures 132, 1094–1104.

[42] Naresh, K., Khan, K.A., Umer, R., Cantwell, W.J., 2020. The use of x-ray computed tomography for design and process modeling of aerospace composites: A review. Materials and Design 190, 108553.

[43] Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics 9, 62–66.

[44] Prechelt, L., 1998. Early stopping-but when?, in: Neural Networks: Tricks of the trade. Springer, pp. 55–69.

[45] Rahman, M.A., Wang, Y., 2016. Optimizing intersection-over-union in deep neural networks for image segmentation, in: International symposium on visual computing, Springer. pp. 234–244.

[46] Roerdink, J., Meijster, A., 2000. The watershed transform: Definitions, algorithms and parallelization strategies. Fundamenta Informaticae 41, 187–228.

[47] Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.

[48] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1985. Learning internal representations by error propagation. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.

[49] Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press, London.

[50] Serra, J., 1983. Image analysis and mathematical morphology. Academic Press, Inc.

[51] Sherburn, M., 2007. Geometric and Mechanical Modelling of Textiles. Ph.D. thesis. University of Nottingham.

[52] Sinchuk, Y., Kibleur, P., Aelterman, J., Boone, M.N., Paepegem, W.V., 2020. Variational and deep learning segmentation of very-low-contrast x-ray computed tomography images of carbon/epoxy woven composites. Materials 13.

[53] Straumit, I., Lomov, S.V., Wevers, M., 2015. Quantification of the internal structure and automatic generation of voxel models of textile composites from x-ray computed tomography data. Composite Part-A: Applied Sciences and Manufacturing 69, 150–158.

[54] Vapnik, V., 1992. Principles of risk minimization for learning theory, in: Advances in neural information processing systems, pp. 831–838.

[55] Verpoest, I., Lomov, S., 2005. Virtual textile composites software wisetex: integration with micro-mechanical, permeability and structural analysis. Composites Science and Technology 65, 2563–2574.

[56] Wang, Y., Sun, X., 2001. Digital-element simulation of textile processes. Composites Science and Technology 61, 311–319.

[57] Wijaya, W., Ali, M.A., Umer, R., Khan, K.A., Kelly, P.A., Bickerton, S., 2019. An automatic methodology to ct-scans of 2d woven textile fabrics to structured finite element and voxel meshes. Composites Part-A: Applied Sciences and Manufacturing 125, 105561.

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

[59] Zhou, G., Sun, X., Wang, Y., 2004. Multi-chain digital element analysis in textile mechanics. Composites Science and Technology 64, 239–244.