SUPERVISED LEARNING FOR ROBOTIC DRAPING TASKS IN COMPOSITE PREFORMING

Hannah Dammers[1*], Mohamed Behery[2*], Max van Gemmeren[2], Boris Manin[1], Gerhard Lakemeyer[2], and Thomas Gries[1]

[1] RWTH Aachen University, Institut fuer Textiltechnik, Aachen, Germany,
Mail: hannah.dammers[at]ita.rwth-aachen.de,
Website: https://www.ita.rwth-aachen.de/go/id/jezh/

[2] RWTH Aachen University, Knowledge-Based Systems Group, Aachen, Germany
Mail: behery[at]kbsg.rwth-aachen.de
Website: https://www.kbsg.rwth-aachen.de/

Keywords: Preforming, Automation, Robotic Draping, Supervised Learning

ABSTRACT

Previous automation solutions for the production of composite parts are limited to the production of large series due to high investment cost and large floor space needed. For this reason, a significant proportion of these parts is still produced manually, resulting in high error rates, scrap, as well as slow production speeds and high labour cost. At the moment, the biggest challenge for automation is the high implementation efforts and the lack of robotics know-how in companies. Therefore, within this paper, we present an approach for the flexible automation of composite preforming using collaborative robots (cobots). The intention is to enable the cobot to work hand-in-hand with the human worker for maximum flexibility. In addition to the development of robotic draping tools, a supervised learning approach is adopted for cobot programming. Three-dimensional data in the form of point clouds is collected, pre-processed and augmented. Afterwards, scene flow approximation approaches are used to predict future motions, allowing the understanding of relationships between geometric features and motions. In this way, we make a first step towards automated robotic draping.

1 INTRODUCTION

The production of continuous fibre-reinforced composite parts with highly-complex geometries poses major challenges to companies because they have to fulfil high quality demands, e.g., in aviation or automotive applications. Accordingly, it is essential to avoid defects such as wrinkles, gaps, loops, or undulations when handling or draping (forming) the limp textile materials. On the one hand, the manual production of such composite parts is challenging, as handling and draping the limp, partly sticky textiles is difficult to learn. Composite producing companies rely on their employees who are skilled in dealing with the limp textiles and benefit from their sensorimotor skills that allow them to produce even highly complex composite parts. In addition, their cognitive abilities combined with low machine and tool costs enable fast product changeovers. However, human errors and the resulting waste of materials as well as slow production speeds and labour costs in high-wage countries have a negative impact. In addition, many workers suffer from wrist or back problems. [1] Despite this, around a third of all composite parts are currently produced in elaborate manual processes [2].

On the other hand, manually performed process steps can be reduced by automating the process. Large companies from aviation and automotive industries already use automated processes to produce composite parts. However, the used special machines, robot cells, or production lines are very expensive and require, in some cases, a disproportionate amount of floor space. The biggest challenge for the introduction of automation solutions such as robotics, especially for Small and Medium-sized Enterprises (SMEs), lies in the huge implementation efforts as well as a lack of know-how in robotics.

For these reasons, automating composite production is often perceived as a cost driver without direct value creation and is therefore not considered by SMEs. [3, 4]

Therefore, this paper presents an approach for the flexible automation of composite part production using collaborative robots (Cobots) in complex draping tasks. We present here both the robot draping tools and the robot learning algorithm used. For maximum flexibility, the intention is to enable the robot to collaborate with humans. We chose a supervised learning approach to learn the motions of the manual draping process. As no data is available for manual draping processes, our approach includes the collection, pre-processing, and augmentation of the data, as well as implementation, training and evaluation of the machine learning model. The data is collected as RGB-D video data and processed to point clouds to get a three-dimensional representation. Three-dimensional machine learning operations are used to predict the motions. In this way, the relationship between the process movements and the geometric features of the used mould can be revealed.

2 RELATED WORK

2.1 Automated Preforming

Preforming is the production of a three-dimensional, near-net-shape dry reinforcement structure. The preforming process consists of four steps: cutting the layers, handling them to the mould, forming (draping) them onto the mould, and joining the individual layers so that the resulting preform can be processed in further process steps, e.g., resin transfer moulding or vacuum-assisted resin infusion. Preforming processes are well suited for automation, as automated technologies already exist for the individual process steps. For example, cutting of textile plies can be automated by using CNC cutters or automated pick and place, and stacking can be realised by a range of available gripping technologies. [5, 6] Draping is the process step most difficult to automate, especially for complex geometries. To ensure the desired mechanical properties, the draping must be done without angular or positional deviation of the fibres. For this purpose, relative movements of the reinforcement fibres, so-called shearing, must be made possible in order to adapt the textiles to the desired geometry. However, the textile tends to wrinkle as soon as it is deformed in three planes simultaneously. [6, 7]

Various approaches to automated draping exist. However, with machines, draping textiles into complex three-dimensional geometries is often limited. In stamp forming, the textile is placed on a fixed lower mould, while a movable stamp serves as the upper mould. The textile ply is heated and remains in the mould under pressure until the binder solidifies. To improve pressure distribution, segmented or elastomer stamps can be used. [6] Another approach is the usage of a membrane as upper mould, which forms the textile ply by an applied vacuum. With both processes, flexibility is limited with regard to the production of different geometry variants and the achievable part complexity. If more flexibility is needed, robotic draping can be used. In this process, the degrees of freedom of a robot are used for draping with special end-effectors. In total, five operating principles can be distinguished: flexible materials, reconfigurable mechanisms, pixel-based systems, deformable membranes and ‘strike-out’ preforming. [8, 9] In summary, mostly part- and geometry-specific approaches are available for automated preforming, but their limit is the flexible automated production of small series. Therefore, we aim at a cost-effective and flexible automation of draping by using human-robot collaboration (HRC). Initial research approaches exist for this process, but no standardised end-effectors are available.

2.2 Human-robot collaboration

In fully automated preforming, robots and machines perform the same action repeatedly with low tolerance. Thus, this process is carried out correctly if the path planning and automation solution are designed correctly. In contrast, in hand lay-up, even the most experienced experts perform their work differently in each operation as they use their visual and tactile skills to achieve the desired result. Since the properties of reinforcement textiles always vary slightly and their behaviour is not precisely predictable, the working environment in composite preforming is highly variable. Consequently, there is no rigidly programmable robotic path that achieves a perfect result every time. [1, 6] It is therefore necessary to develop semi-automated preforming that does not require disproportionately high investment costs and floor space, and does not limit the flexibility with regard to the possible variety of variants and part complexity. Furthermore, the solution should not eliminate the advantages of human work such as sensory-motor skills. Accordingly, the collaboration of humans and robots in preforming is promising, since the strengths of humans (cognitive flexibility, feeling for materials, and dexterity) and robots (tirelessness, high speed, and precision) can be exploited in this scenario. [3, 4] For HRC, a shared workspace is needed that is subject to strict safety requirements. In order to avoid the usual robot safety precautions such as fences or light barriers, Cobots with limited payload and speed can be used. They are equipped with additional sensor technology allowing to stop in case of a collision with humans. Nevertheless, sharp tools can only be used with complex safety measures. [10]

First approaches exist for the implementation of HRC in preforming. An analysis of a manual preforming process in [3] shows that the introduction of HRC in preforming can be advantageous by means of efficiency. However, it shows that HRC will not work for all lot sizes and part complexities. In [4], five interaction modes for the handling and draping of reinforcement textiles are defined, concentrating on big plies for boating goods or aviation. Human factors within HRC preforming are investigated in [11], revealing that the cobot is considered as a technical assistant doing a good job in the lay-up process. Furthermore, there are ongoing research projects on the topic of HRC in composite preforming, e.g., RaCPro (Germany), DrapeBot (EU), or JARVIS4Pre (Austria).

2.3 Supervised learning for robot teaching

Despite the success of classical trajectory planning models, they remain unsuitable for the draping task due to the complexity of modelling the manipulation of limp textiles. Additionally, Learning from Demonstration (LfD) and Reinforcement Learning (RL) approaches often require a simulation for refining their policies which is challenging on its own, particularly given the dynamic properties of the textiles used in preforming. Therefore, we use supervised learning to learn the motions of manual draping. The most significant limitation of traditional Artificial Neural Networks (ANNs) is the inability to cope with computational complexity of high-dimensional data. Contrary to this, Convolutional Neural Networks (CNNs), a class of ANNs predominately used in pattern recognition, are using specialised layers that are applied to local regions of the input to cope with this limitation [12]. Therefore, CNNs can be used to process three-dimensional data, e.g., in the form of point clouds. A point cloud is a set of data points with three-dimensional cartesian coordinates that is able to include additional features such as RGB values or surface normal. By providing essential geometric, shape, and scale information, point clouds enhance the understanding of the captured environment. [13] Machine learning on point clouds has been recently used in numerous applications, for example robotics, autonomous driving, and virtual reality. However, applying convolutional operations on 3D point cloud data is challenging because of the point clouds' generally irregular, unstructured, and unordered nature (Figure 1). [13, 14]

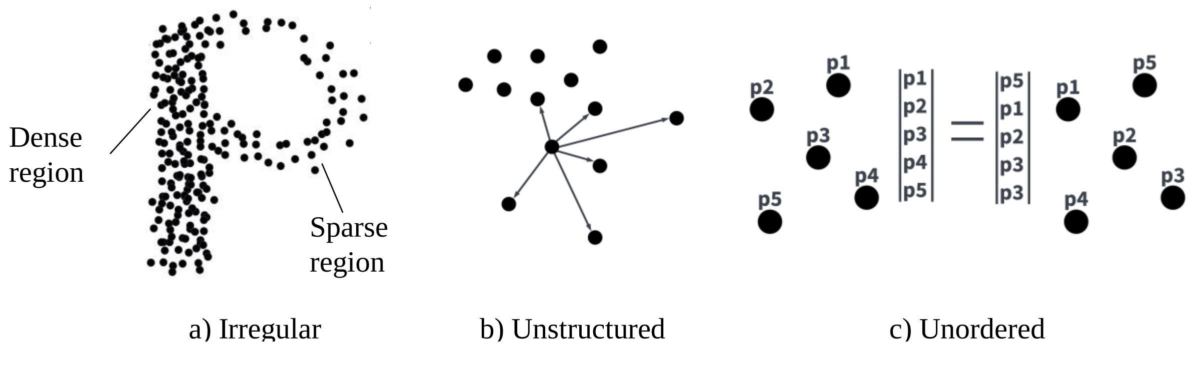


Figure 1: Challenges of point cloud data [13]

Applications for deep learning on point clouds include shape classification, segmentation and object detection. Early approaches that convert point clouds into a structured form can be broadly divided into voxel-based and multi-view based approaches. In contrast, contemporary methods can be applied directly to raw point clouds so that no explicit information loss is introduced. Point-based methods can be divided into pointwise multi-layer-perceptron (MLP), convolutional-based, and graph-based networks. [13, 14] The basis for point-based methods is PointNet [15] to directly apply deep learning on unstructured point clouds. This approach is further developed to PointNet++ [16] that extends PointNet for local region computations to capture fine geometric structures. ConvPoint [17] is a three-dimensional continuous convolutional approach that defines convolutional kernels on a continuous space and assigns the weights of adjacent points by determining their spatial distribution relative to a centre point. In [18], a graph-based network for point clouds is proposed, Dynamic Graph CNN (DGCNN). It constructs a graph in the feature space, which is dynamically modified.

With the advances in machine learning on point clouds, three-dimensional motion analysis with point clouds has emerged as a research area called scene flow estimation. The scene flow refers to the motion field of points in point clouds (Figure 2). Hence, it helps to understand the 3D motion of points in a dynamic environment. FlowNet3D [19] directly learns the scene flow with both, point-level features and motion features, from two sequential point clouds through the use of a flow embedding layer. The Bi-PointFlowNet [20] uses a hierarchical architecture and introduces the bidirectional flow embedding (BFE) layer that learns features along forward and backward directions. In [21], a pair of self-supervised losses is introduced to enable the network to be trained on unlabelled datasets.

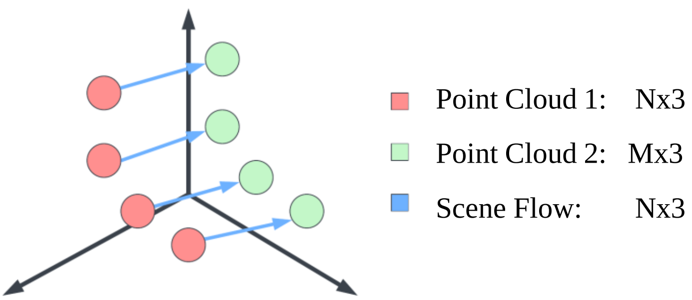


Figure 2: Scene Flow between two point clouds [19]

Predicting future points clouds from a set of consecutive point clouds is called point cloud prediction which is predominately researched for tasks in autonomous driving, such as path planning and collision avoidance. Point cloud prediction relies on past point clouds to predict the future state of a scene. Future point clouds can be estimated by applying scene flows to the previous frames or by directly generating a new set of future points. [22, 23] MoNet [22] is a novel motion-based neural network for point cloud prediction that extracts motion and content features using a motion- and content-encoder. Furthermore, a recurrent neural network called MotionRNN is proposed to capture temporal correlations between features. [23] provides a self-supervised Point Cloud Prediction network that converts the input point clouds into spherical coordinates and maps them to image coordinates. SPFNet [24], is a Long Short-Term Memory (LSTM) autoencoder model for sequential point cloud forecasting. In [25], a self-supervised point cloud prediction architecture is developed to extract temporal features and predict future frames with promising results.

3 APPROACH

In this section, we present the robot tools developed, the data collection approach, the pre-processing and augmentation of the data, and the machine learning models used.

3.1 Robot tools for human-robot preforming

In the development of robot draping tools, special emphasis is given to low cost and ease of manufacturing. Furthermore, as the tools shall be used in HRC, it is mandatory to guarantee safe tools without sharp corners or edges. The tool should also be flexibly usable for different geometries.

Therefore, three draping tools are developed (Figure 3 a)-c)). The squeegee is used to drape outside edges, while the silicone roller is used to drape inside edges. In addition, for draping larger textile areas, the silicone stamp can be used, which allows a roll-off motion. In this way, the robot can take on simple, repetitive draping tasks, while the human is playing to their strengths when draping the complex part areas. Three individual tools are used for data collection. Additionally, to avoid time-consuming tool change during the collaborative draping, we also combined them in one tool (Figure 3 d)).

a) Silicone roller b) Squegee c) Silicone stamp d) Combined draping tool

Figure 3: Robot draping tools for human-robot preforming

3.2 Data collection

To capture the draping process, the different robot draping tools are attached to an extension which facilitates the handling and provides space for a fiducial marker (ArUco marker). The ArUco markers are used to identify each tool and encode its pose. For the generation of the markers, OpenCV is used (marker size 6 6 x bits, dictionary size = 50). For data acquisition, three moulds with different geometries are selected (Figure 4, a)-c)). Another mould is used for testing (Figure 4, d)).

a) Propeller blade

b) Stringer c) Hemisphere d) Cymbal

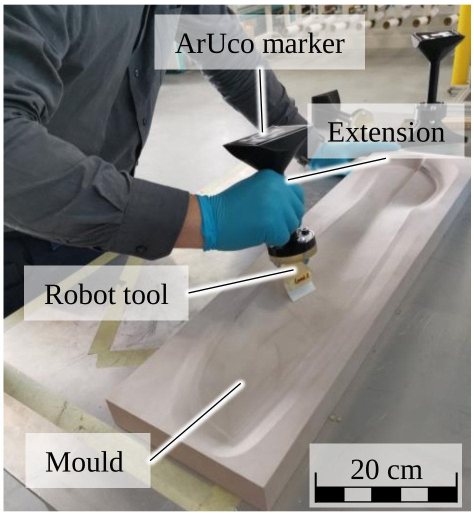


Figure 4: Moulds a) – c) for data acquisition, d) for testing (left); draping process (right)

As the focus of the data collection lies on the relation between the robot tool motion and the geometric features of the respective mould, the process is conducted without textiles. In this way, no material waste is created. Furthermore, the mould is not covered by textile, resulting in lower complexity of the process and consequently producing a higher amount of data is possible. An expert for draping is asked to carry out the process. By video capturing, the existing tacit knowledge is extracted and saved in the form of a human digital shadow [26]. A stereo camera (Intel RealSense Depth Camera D455) is used to record the process from bird's eye view as RGB-D video sequence. Each scene is recorded as two separate colour and depth streams, resulting in each frame consisting of a colour and a depth image. The Intel RealSense

Viewer is used for recording, while the videos are saved as bag-files. The default camera settings are used to achieve highest point coverage. In total, 568 GB of data are created, consisting of 190 RGB-D videos. For each data acquisition mould, 60 videos are recorded, while extra 10 videos are captured for the testing mould. With an average of 880 frames per video, 167.200 data points can be extracted.

3.3 Preprocessing

To assure a data format that machine learning models can efficiently analyse, the pre-processing of the raw data is mandatory. The pre-processing pipeline shown in Figure 5 allows the conversion of the captured RGB-D data into point cloud data to approach motion prediction.

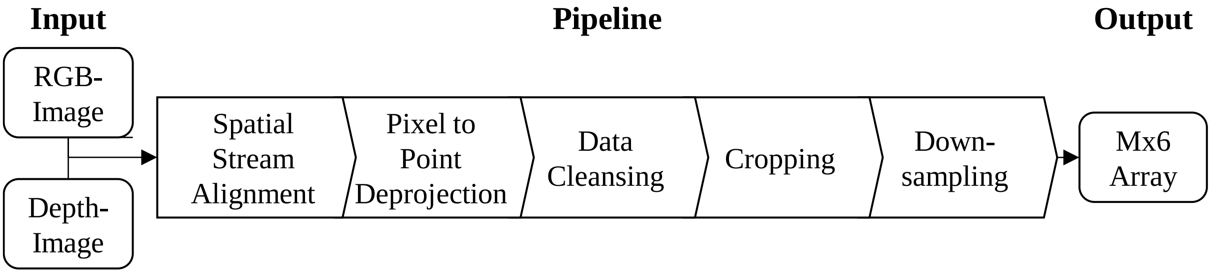
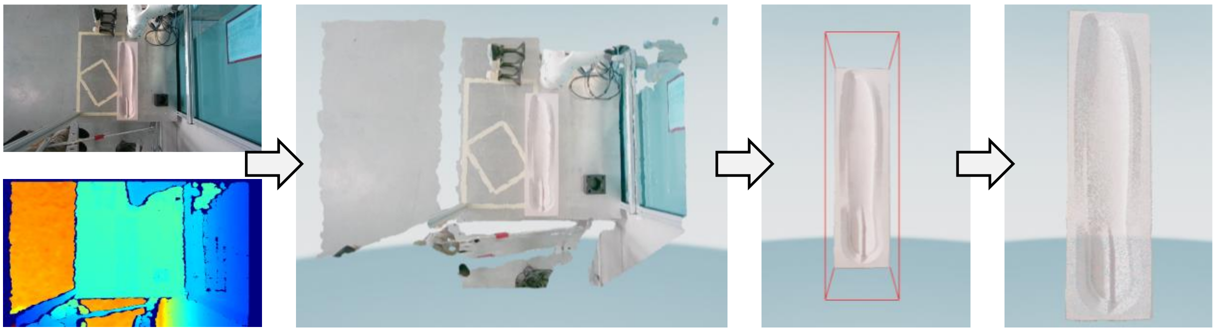


Figure 5: Pre-processing pipeline for conversion of RGB-D data into point cloud data

Within the first step, the spatial stream alignment, the two single RGB- and depth-images are aligned and form a merged RGB-D image. Afterwards, the resulting RGB-D image is converted to a point cloud by deprojecting its pixels to three-dimensional point coordinates. The deprojection of every pixel of the RGB-D image to point coordinates results in a point cloud that consists of XYZ coordinates and corresponding RGB values with a dimension of N x 6. The number of points N varies from frame to frame depending on the valid depth values within the frame. In step three, the data quality is increased by the application of data cleansing techniques that remove non-finite and duplicate points within the data. The collected images show the entire working area. However, the draping process is conducted within the immediate surrounding of the mould. Therefore, the point cloud is manually cropped using an axis-aligned bounding box that allows the reduction of computational complexity. After the cropping process, the dimension of the point cloud is reduced from N x 6 to N' x 6 with N' < N. The number of points N' per frame is estimated to be in the range of 20.000 to 50.000 points depending on the respective mould. Applying machine learning with this high amount of data points is computationally expensive and does not allow the training on multiple batches as the number of points is not fixed to one size. For this reason, within the last step, the data is downsampled using Farthest Point Sampling (FPS) [27]. As output data a M x 6 array results that includes XYZ coordinates and RGB values of the point cloud. The array is saved to the HDF5 dataset and annotated with mould and tool name. Figure 6 shows the RGB-D data in different steps of the pre-processing pipeline.



a) Aligned images b) Deprojected c) Cropped d) Downsampled

Figure 6: RGB-D data in different steps of pre-processing pipeline

3.4 Data augmentation

During data collection, the mould is placed in a fixed location to guarantee consistency and reproducibility. This results in a potential overfitting problem where the model learns absolute motions instead of relations between the tools' motions and mould. Therefore, data augmentation techniques are used to diversify the data and expose the model to more varied examples during training. By creating new, slightly modified versions of the original data, it is possible to improve the generalisation ability and avoid overfitting. Similar to data augmentation in the image domain, rotation, flipping, and scaling are chosen as transformations (Figure 7).

a) Original   b) Rotation   c) Flipping   d) Scaling

Figure 7: Data augmentation techniques applied to original data

To not change the dimension of the data, the transformations are applied pointwise. Thus, the transformations are applied by multiplying the corresponding transformation matrix with all points of the point cloud. In total, six augmentations are realisable by combining the augmentation techniques. With the angle θ, the two flipping values $a, b \in \{-1,1\}$, and a scaling factor $v \in \{0,1\}$, the rotation matrix $R_z(\theta)$, the flipping matrix *F(a,b)*, and the scaling matrix *S(v)* are defined as follows:

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad F(a,b) = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad S(v) = \begin{pmatrix} v & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & v \end{pmatrix} \quad (1)$$

Random parameters are generated, and the parameterised transformation is applied to the pre-processed data. To assure that no artificial motion is added to the scene, all frames corresponding to the same video are augmented with the same parameters.

3.5 Model architecture

We suggest a model architecture that enables learning on the pre-processed, augmented data. The model architecture used is based on contemporary point-based and scene flow operators, and is inspired by recent research on point cloud prediction by [25]. To learn the respective relations between the geometric structure of the mould and the motions, the past four frames of point clouds $x_{t-3}, \ldots, x_t$ serve as inputs for the models. By using two frames the velocity of objects in a scene can be determined, while at least three frames are obligatory to identify second-order dynamics such as acceleration. Further contextual information is gathered using the fourth frame. [25] Fundamentally, the network architecture consists on the one hand of a feature extractor layer that uses a point-based feature extractor directly applied to the input point cloud. Feature extractors proposed in PointNet ++ [16] and DGCNN [18] are used. On the other hand, a flow embedding layer from FlowNet3D [19] is built to learn the scene dynamics between two point clouds. First, from every frame the features are extracted pointwise with a feature extractor layer. Afterwards, the network computes the flow dynamics with a flow embedding layer. The process is repeated, a final feature extractor layer is used, and the extracted features are forwarded to the refinement layer that outputs a motion vector of the dimension $M \times 3$ for each point in the point cloud x_t. By adding the motion vectors to the point cloud, a new set of points is computed that can be used to compare the predicted point cloud to the original scene. To predict point clouds of more distant future, the prediction x_{t+1}can be used and applied to the model autoregressively.

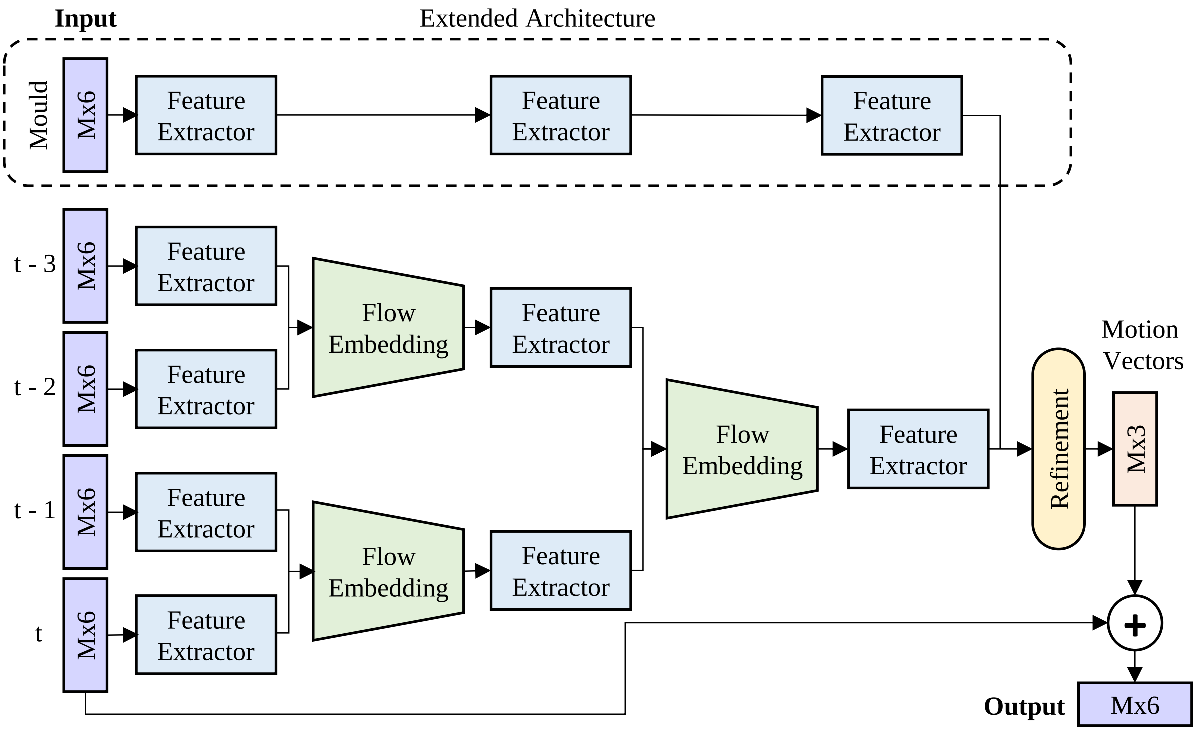


Figure 8: Model architecture based on [25]

To assure a better fit of Deng and Zhakor's approach [25] to the preforming use case, the model architecture is adjusted to accept input point clouds ($M \times 6$) including the three-dimensional point coordinates and the RGB information. Furthermore, the non-occluded point cloud representation of the mould is used as additional input which is intended to support the capturing of the mould's geometric features. The model architecture is varied (see Table 1) to evaluate the choice of the feature extractor and to investigate the effect of the additional mould input.

Model Name	Architecture	Feature Extractor	Downsampling
ConvPoint	Original	Continuous Convolution [17]	Yes
ConvPointM	Extended	Continuous Convolution [17]	Yes
EdgeConv	Original	Edge Convolution [18]	No
EdgeConvM	Extended	Edge Convolution [18]	No
PNPP	Original	PointNet++ [16]	Yes
PNPPM	Extended	PointNet++ [16]	Yes

Table 1: Model variations

4 EVALUATION

To evaluate our work, we train and test several state-of-the-art approaches for scene flow estimation [16–18] using our data. Additionally, we train and test these models by including the 3D models of the moulds during the training. This section summarizes the details as well as the quantitative results.

4.1 Experimental Setup

The models were trained using an Nvidia Tesla V100-SXM2-16GB and the date was split into training, test, and validation subsets using a 70-20-10 split. During the training, we use the loss function from [25] that combines the Earth Movers Distance (EMD) with Chamfer Distance (CD) proposed in [28]. These are widely used distance measures for point cloud data. EMD finds a bijection, φ , between two point clouds, P and Q, that minimizes the sum of distances between corresponding points

$$EMD(P,Q) = \min_{\varphi \in P \to Q} \sum_{p \in P} \|p - \varphi(p)\|_2 \tag{2}$$

CD calculates the sum of squared distances between corresponding nearest neighbours in P and Q.

$$L_{CD}(P,Q) = \sum_{p \in P} \min_{q \in Q} \left|\left|p - q\right|\right|_2^2 + \sum_{q \in Q} \min_{p \in P} \left|\left|p - q\right|\right|_2^2 \tag{3}$$

In our work, we follow [25], where the loss L is a weighted sum of EMD and CD

$$L(P,Q) \ = \alpha L_{CD}(P,Q) \ + \beta L_{EMD}(P,Q) \tag{4}$$

where α and β are hyperparameters for the weights of CD and EMD, respectively.

4.2 Results

As mentioned above, we compare the performance of model architecture mentioned in Section 3 with three different feature extractors, ConvPoint [17], EdgeConv [18], and PNPP [16]. Each one has a mould-variant (with “M” suffix) that includes the CAD model of the mould as input. As a baseline, we train the models using only a single mould (propeller blade shown in Figure 4. a). Additionally, we compare these to the identity function, i.e., predicting the same input. Results of the baseline comparison are shown in Table 2.

Model	Epochs	CD	EMD
ConvPoint	15	5.9*10-5	3.8*10-3
ConvPointM	7	3.4 *10-5	2.1 *10-3
EdgeConv	15	1.7 *10-5	1.4 *10-3
EdgeConvM	8	2.5 *10-5	1.9 *10-3
PNPP	15	3.0 *10-5	4.1 *10-3
PNPPM	14	3.7 *10-5	3.9 *10-3
Identity	N/A	1.4 *10-5	1.2 *10-3

Table 2: Results of baseline comparison

As seen in Table 2, using EdgeConv feature extractor yields the lowest error rates for both variants (with and without mould information). These results are confirmed by visual inspection (Figure 9) where we see more noise and artefacts in the predicted images of the M-models. Furthermore, EdgeConv models have less noise than the others. However, the M-models have less accuracy across all feature extractors. This could be due to the increased number of parameters involved in these models.

a) Ground Truth b) ConvPoint c) ConvPointM d) EdgeConv e) EdgeConvM f) PNPP g) PNPPM

Figure 9: Visual representation of results from experiments using the different models

To examine the effects of the data augmentation, we trained the PNPP model using the augmented and not augmented datasets and test it using both datasets. Table 3 shows the results of training and testing using the different combinations of the datasets. Training on the augmented dataset significantly reduces both CD and EMD when tested on the augmented or not augmented test set.

Training	Testing	CD	EMD
Not augmented	Not augmented	3.7*10-5	4.1*10-3
Augmented	Not augmented	2.1 *10-5	1.4 *10-3
Not augmented	Augmented	3.3 *10-5	4.0 *10-3
Augmented	Augmented	2.1 *10-5	1.3 *10-3

Table 3: Effects of training the PNPP model using the augmented dataset

Additionally, we examine how well the model generalizes to unknown moulds by training the different models on the entire dataset, containing data from different moulds and then test it on a previously unseen mould. Table 4 shows the results of training the regular ConvPoint, EdgeConv, and PNPP models using data from a single mould (Propeller blade) and using multiple moulds. Training with multiple moulds only increases the accuracy of the PNPP model but not the others. This could be attributed to using abstraction levels forming a hierarchy of local features, as opposed to the two other models that extract global features.

To sum up, the results show that EdgeConv model achieves the highest results compared to the other model variations. Additionally, the augmentation increases the accuracy overall, when testing over the augmented and not augmented datasets. Finally, the models show higher CD and EMD on average when trained on multiple moulds and tested on an unknown one, i.e., the models are unable to generalize well. This can be attributed to the limited data in terms of number of moulds; the data we collected contains only four moulds. This can be examined further either by collecting data from more moulds, or by using a sliding window approach in the training to learn local trajectories relative to the mould geometry under the window instead of a global trajectory for the whole mould.

Mould	Model	Epochs	CD	EMD
Propeller Blade	ConvPoint	15	1.4*10-4	1.7*10-2
	EdgeConv	15	2.9 *10-5	1.6 *10-3
	PNPP	15	5.6 *10-5	4.8 *10-3
Multiple Moulds	ConvPoint	15	4.9*10-4	9.6*10-3
	EdgeConv	8	3.4*10-5	2.1*10-3
	PNPP	15	4.0*10-5	3.4*10-3
N/A	Identity	N/A	2.5*10-5	1.2 *10-3

Table 4: Effects of training the models using the multiple moulds vs a single mould

5 CONCLUSION

This paper presents an approach for the flexible automation of composite preforming using collaborative robots. A robot arm is employed to perform draping. Supervised learning techniques are used which include the added challenge of data acquisition. We show how to transfer scene flow approximation approaches into composite preforming, making a step towards automated robotic draping. However, this work only shows how the different approaches estimate the motion of the robot tools (controlled by the human hand). This work is a step towards the automation of the draping process. The next step is to transfer these results to the robot. Thus, the results obtained here have to be used to generate a trajectory given the mould geometry. Additionally, the trajectory is to be transformed from the task space to the configuration space. For future work, the generalization to new moulds can be investigated. Finally, the findings can be transferred to the combined draping tool (Figure 3, d), and an automatic tool selection depending on the different geometric features should be integrated.

ACKNOWLEDGEMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2023 Internet of Production – 390621612. Simulations were performed with computing resources granted by RWTH Aachen University under project thes1391.

REFERENCES

[1] M. Elkington, D. Bloom, C. Ward, A. Chatzimichali, and K. Potter, “Hand layup: understanding the manual process,” *Advanced Manufacturing: Polymer & Composites Science*, vol. 1, no. 3, pp. 138–151, 2015, doi: 10.1080/20550340.2015.1114801.

[2] F. Reux, *JEC Observer: Current trends in the global composites industry*. Frankreich: JEC Group, 2021. Accessed: Oct. 13 2021. [Online]. Available: https://www.jeccomposites.com/product/jec-observer-current-trends-in-the-global-composites-industry-2020-2025/

[3] H. Dammers, M. Lennartz, T. Gries, and C. Greb, “Human-robot collaboration in composite preforming: chances and challenges,” Dallas, TX, 2021.

[4] C. Eitzinger, C. Frommel, S. Ghidoni, and E. Villagrossi, “System concept for human-robot collaborative draping,” Baden/Zürich, Schweitz, 2021.

[5] J. Fleischer, R. Teti, G. Lanza, P. Mativenga, H.-C. Möhring, and A. Caggiano, “Composite materials parts manufacturing,” *CIRP Annals*, vol. 67, no. 2, pp. 603–626, 2018, doi: 10.1016/j.cirp.2018.05.005.

[6] M. Elkington, C. Ward, and A. Sarkytbayev, “Automated composite draping: a review,” *SAMPE 2017*, 2017. [Online]. Available: https://research-information.bris.ac.uk/en/publications/automated-composite-draping-a-review

[7] D. H.-J. Lukaszewicz, C. Ward, and K. D. Potter, “The engineering aspects of automated prepreg

layup: History, present and future," *Composites Part B: Engineering*, vol. 43, no. 3, pp. 997–1009, 2012, doi: 10.1016/j.compositesb.2011.12.003.

[8] G. Schouterden, J. Cramer, E. Demeester, and K. Kellens, "Development of a membrane-shaped MR-based composite draping tool," *Procedia CIRP*, vol. 86, pp. 167–172, 2019, doi: 10.1016/j.procir.2020.01.048.

[9] H. Kunz *et al.*, "Novel form-flexible handling and joining tool for automated preforming," *Science and Engineering of Composite Materials*, vol. 22, no. 2, pp. 199–213, 2015, doi: 10.1515/secm-2013-0326.

[10] M. Giuliani, C. Lenz, T. Müller, M. Rickert, and A. Knoll, "Design Principles for Safety in Human-Robot Interaction," *Int J of Soc Robotics*, vol. 2, no. 3, pp. 253–274, 2010, doi: 10.1007/s12369-010-0052-0.

[11] H. Dammers, L. Vervier, L. Mittelviefhaus, P. Brauner, M. Ziefle, and T. Gries, "Usability of human-robot interaction within textile production: Insights into the acceptance of different collaboration types," in *Usability and User Experience*, 2022.

[12] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Nov. 2015. [Online]. Available: https://arxiv.org/pdf/1511.08458

[13] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Review: Deep Learning on 3D Point Clouds," *Remote Sensing*, vol. 12, no. 11, p. 1729, 2020, doi: 10.3390/rs12111729.

[14] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," Dec. 2019. [Online]. Available: https://arxiv.org/pdf/1912.12033

[15] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," Dec. 2016. [Online]. Available: https://arxiv.org/pdf/1612.00593

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," 2017.

[17] A. Boulch, "ConvPoint: Continuous Convolutions for Point Cloud Processing," Apr. 2019. [Online]. Available: https://arxiv.org/pdf/1904.02375

[18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," Jan. 2018. [Online]. Available: https://arxiv.org/pdf/1801.07829

[19] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning Scene Flow in 3D Point Clouds," Jun. 2018. [Online]. Available: https://arxiv.org/pdf/1806.01411

[20] W. Cheng and J. H. Ko, "Bi-PointFlowNet: Bidirectional Learning for Point Cloud Based Scene Flow Estimation," Jul. 2022. [Online]. Available: https://arxiv.org/pdf/2207.07522

[21] H. Mittal, B. Okorn, and D. Held, "Just Go with the Flow: Self-Supervised Scene Flow Estimation," Dec. 2019. [Online]. Available: https://arxiv.org/pdf/1912.00497

[22] F. Lu, G. Chen, Y. Liu, Z. Li, S. Qu, and T. Zou, "MoNet: Motion-based Point Cloud Prediction Network," Nov. 2020. [Online]. Available: https://arxiv.org/pdf/2011.10812

[23] B. Mersch, X. Chen, J. Behley, and C. Stachniss, "Self-supervised Point Cloud Prediction Using 3D Spatio-temporal Convolutional Networks," Sep. 2021. [Online]. Available: https://arxiv.org/pdf/2110.04076

[24] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris M. Kitani, and Nicholas Rhinehart, "Unsupervised Sequence Forecasting of 100,000 Points for Unsupervised Trajectory Forecasting," *ArXiv*, abs/2003.08376, 2020.

[25] D. Deng and A. Zakhor, "Temporal LiDAR Frame Prediction for Autonomous Driving," Dec. 2020. [Online]. Available: https://arxiv.org/pdf/2012.09409

[26] A. Mertens *et al.*, "Human Digital Shadow: Data-based Modeling of Users and Usage in the Internet of Production," in *2021 14th International Conference on Human System Interaction (HSI)*, Gdańsk, Poland, Jul. 2021 - Jul. 2021, pp. 1–8.

[27] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 6, no. 9, pp. 1305–1315, 1997, doi: 10.1109/83.623193.

[28] H. Fan, H. Su, and L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," Dec. 2016. [Online]. Available: https://arxiv.org/pdf/1612.00603