IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Fabric Dynamic Motion Modeling and Collision Avoidance With Oriented Bounding Box

Letian Li, *Graduate Student Member, IEEE*, Fuyuki Tokuda, *Member, IEEE*, Akira Seino, *Member, IEEE*, Akinari Kobayashi, *Member, IEEE*, Norman C. Tien, *Senior Member, IEEE*, and Kazuhiro Kosuge, *Life Fellow, IEEE*

***Abstract*—Avoiding collision between the fabric and the obstacle is critical to transport fabric piece in the garment factory. If the fabric collides with the sharp-edged obstacle, it can be scratched or contaminated, resulting in poor product quality and increased waste. However, when we consider the fabric model, we find that current fabric models are not accurate enough for this real-world application. It is almost impossible to model the dynamic motion of the fabric with high accuracy, because its motion and deformation are affected by many hard-to-estimate factors. In this paper, instead of using an accurate fabric model, we propose a new fabric motion modeling method using the proposed OBB-Transformer, which models the fabric motion as a time series of oriented bounding boxes (OBBs). Using OBB-Transformer, a dynamic collision avoidance method is designed to plan a robot trajectory connecting the start point and the goal point without collision between the fabric and the obstacle. The performance of the fabric dynamic motion modeling is compared between the proposed and conventional methods. Then, the collision avoidance of a piece of fabric using the proposed method is demonstrated on a real robot system in both 2D and 3D scenarios.**

***Index Terms*—Collision avoidance, deep learning methods, motion and path planning, deformable object modeling.**

I. Introduction

COMPARED to other manufacturing industries, the garment industry has lagged behind in the use of robots to automate production processes [1]. One such process is transporting fabric between adjacent production processes. For example, T-shirt panels are manually transported before and after being printed by a semi-automatic printing machine [2].

Received 14 March 2025; accepted 19 July 2025. Date of publication 4 August 2025; date of current version 12 August 2025. This article was recommended for publication by Associate Editor X. Xiao and Editor A. Bera upon evaluation of the reviewers' comments. This work was supported in part by the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust and in part by the Innovation and Technology Commission of the HKSAR Government through the InnoHK initiative. *(Corresponding author: Letian Li.)*

Letian Li, Fuyuki Tokuda, Akira Seino, Akinari Kobayashi, and Kazuhiro Kosuge are with the JC STEM Lab of Robotics for Soft Materials, Department of Electrical and Electronic Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong SAR, China, and also with the Centre for Transformative Garment Production, Hong Kong SAR, China (e-mail: letian.li@connect.hku.hk; kosuge@hku.hk).

Norman C. Tien is with the Centre for Transformative Garment Production, Hong Kong SAR, China, and also with the Department of Electrical and Electronic Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong SAR, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3595026, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3595026

A major challenge in automating fabric transport is the confined space of garment factories [3], [4], which are crowded with machines, racks, and other equipment. Simply programming a robot to repeat a fixed motion is insufficient because the fabric deforms during transportation. The deformation can cause collisions between the fabric and surrounding objects. Such collisions can result in bruises, holes, or stains on the fabric [5], which degrades product quality and increases waste.

To solve this problem, we propose a fabric dynamic motion modeling method for fabric collision avoidance in a confined space. There are two problems we need to solve:

- *How to model the fabric state:* The nonlinearity of the deformation, the infinite dimension of the state space, the sim-to-real gaps, etc. make it difficult to model the fabric state accurately. The fabric motion and deformation are easily influenced by many hard-to-estimate factors, such as airflows caused by the robot motion and the environment, and non-uniform properties of the fabric [6]. Even if the robot trajectory is the same, the fabric deformation may vary.
- *How to plan the robot motion to avoid collision between the fabric and the obstacle:* We need to check for possible collisions between the fabric and the obstacle along the entire robot trajectory from the start point to the goal point. Unlike rigid objects, the fabric state is affected by hard-to-estimate factors and evolves along the robot trajectory. Configuration space-based collision avoidance methods cannot be applied to the collision avoidance problem considered in this paper.

There are many methods for modeling deformable objects, including mass-spring system [7], finite element analysis [8], position-based dynamics [9], [10], graph dynamics [11], [12], mesh dynamics [13], and others [14], [15]. Some literature reviews can be found in [16], [17].

Collision avoidance of deformable objects is reported in some specific tasks. The manipulation of a rope by a dual manipulator system was discussed in [18]. [19] dealt with a belt assembly task using a humanoid robot. Narrow passage path planning of a piece of fabric was implemented in [20].

Some research considers the dynamic manipulation of deformable objects, including adjusting a rope to pass through the target point [6], knotting a rope with high-speed motion [21], hitting two targets with a soft rod in a single dynamic motion [22], and cloth unfolding using self-supervised learning framework

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

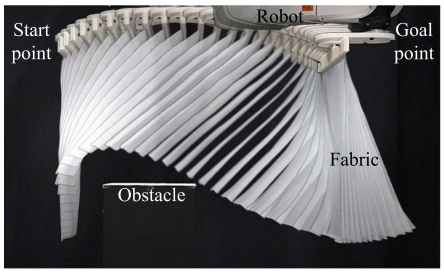


Fig. 1. The robotic fabric transport task. Modeling and predicting fabric states is necessary to find a collision-free robot trajectory.

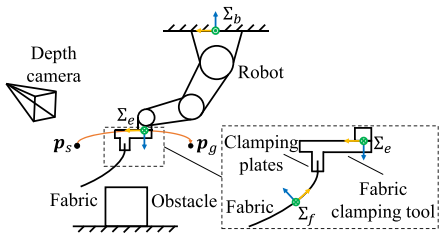


Fig. 2. The coordinate systems used in this paper. The yellow, green, and blue arrows refer to the x, y, and z axes in each coordinate system Σ. The symbol $\odot$ means that the axis is out of the plane and the symbol $\otimes$ means the opposite. The orange line with an arrow is the robot endpoint trajectory from the start point $\boldsymbol{p}_s$ to the goal point $\boldsymbol{p}_g$. Specific meanings of the symbols are given in Section II.

FlingBot [23]. However, highly dynamic motions of deformable objects remain under-explored [16].

Collision avoidance [24], [25] is a fundamental problem in robotics. It ensures that robots and manipulated objects avoid collisions while moving from the start point to the goal point. The trajectory generated by the collision avoidance method is always collision-free. To the best of our knowledge, dynamic fabric collision avoidance as shown in Fig. 1 has not been reported due to the challenges of fabric motion modeling.

In this paper, we propose a fabric motion modeling method based on OBB-Transformer, which represents the fabric motion as a time series of oriented bounding boxes (OBBs). We apply this method to the collision avoidance task, where OBB-Transformer iteratively predicts the future fabric OBBs based on the initial fabric OBBs and a given robot trajectory.

Since the current fabric state depends on its motion history along the robot trajectory, collision-free motion planning for fabric remains a complex challenge. To address this challenge, we propose a simple yet effective method that selects the shortest-path collision-free trajectory from a set of trajectory candidates. Each trajectory candidate is generated by connecting two trajectory segments at a point selected from a set of points uniformly sampled within the robot working space.

To simplify the discussion, we first consider the collision avoidance of a rectangular piece of fabric in the 2D scenario, where the robot motion is restricted to the 2D vertical plane as shown in Fig. 2 . We then extend this to the 3D scenario, which is shown in Section V.

The contributions of this paper are summarized as follows:

- The OBB-based fabric dynamic motion modeling method, OBB-Transformer, is proposed and shown to be effective for fabric dynamic collision avoidance.

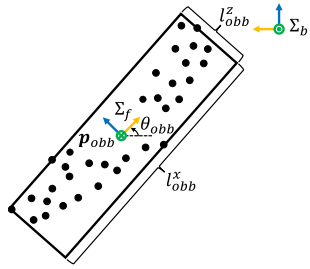


Fig. 3. The fabric OBB model. The fabric OBB state $\boldsymbol{S}$ consists of the OBB center position $\boldsymbol{p}_{obb}$, the OBB size $\boldsymbol{l}_{obb}$, and the OBB orientation in the robot working space θ_{obb}. The black points represent the fabric point cloud $\boldsymbol{P}_f$, which is completely covered by the fabric OBB in black lines.

- Modeling fabric states using OBBs allows us to account for hard-to-estimate factors such as small perturbations in airflow, non-uniform fabric properties, and point cloud noises. These environmental uncertainties are embedded in the proposed OBB-Transformer, which is trained using the fabric OBB dataset collected in the same environment. Note that we do not consider hard-to-estimate factors that suddenly change during the robot motion, such as sudden air blowing.
- The proposed OBB-Transformer shows superior performance in modeling fabric motion compared to existing methods, including the mass-spring-damper 3D mesh model and point cloud-based networks.
- 2D and 3D collision avoidance experiments show that the proposed motion planning method can achieve collision-free fabric transport in a confined space. With a small amount of new data, the proposed OBB-Transformer can be used for fabrics of different materials and sizes.

The rest of this paper is organized as follows. Section II gives the problem formulation and the outline of the proposed method. Section III describes how to model the fabric state with OBB and the fabric motion as a time series of OBBs using the proposed OBB-Transformer. Section IV introduces a robot motion planning method for fabric collision avoidance using OBB-Transformer. Section V presents the experimental results. Section VI concludes the paper.

II. Problem Formulation and Outline of the Proposed Method

A. Coordinate Systems

Let us define coordinate systems as shown in Fig. 2. The robot base coordinate system Σ_b is attached to the center of the robot base. Its z axis points upwards and its x-y plane is horizontal. In this paper, Σ_b serves as the world coordinate system and vectors are represented with respect to Σ_b.

The robot endpoint coordinate system Σ_e is attached to the robot endpoint. We assume that the robot endpoint trajectory is restricted in an x-z plane parallel to the x-z plane of Σ_b. The orientation of Σ_e is assumed to be fixed during the motion. Its y and z axes are opposite to those of Σ_b.

The fabric coordinate system Σ_f is defined based on the fabric point cloud $\boldsymbol{P}_f$, which will be described in detail in Section III. Its y axis is opposite to that of Σ_b.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. Problem Formulation

We consider the problem of finding a robot endpoint trajectory such that the fabric clamped by the fabric clamping tool does not collide with the obstacle along the entire trajectory. When the robot is stationary, a piece of fabric is naturally hung. It is almost flat and perpendicular to the x-y plane of Σ_b. The robot endpoint moves from the start point $\boldsymbol{p}_s$ to the goal point $\boldsymbol{p}_g$ in the x-z plane parallel to that of Σ_b. For simplicity, we will refer to this x-z plane as the robot working space. Our task is to find the shortest-path robot endpoint trajectory that avoids collision between the fabric clamped by the robot end-effector and the obstacle. This can reduce the energy consumption of the robot and improve the operational safety of the garment factory.

C. Method Outline

First, multiple raw point clouds $\boldsymbol{P}_r$ are captured by the depth camera during the robot motion with the fabric clamped by the fabric clamping tool. For each $\boldsymbol{P}_r$, the fabric point cloud $\boldsymbol{P}_f$ is extracted. Then, for each $\boldsymbol{P}_f$, the fabric OBB state is calculated.

Second, OBB-Transformer models the fabric dynamic motion as a time series of OBBs. It is used to predict a time series of future fabric OBBs given the initial fabric OBBs and an entire robot endpoint trajectory. OBB-Transformer is trained using the dataset collected by moving the robot along random robot endpoint trajectories.

Third, multiple robot endpoint trajectory candidates are generated. The proposed OBB-Transformer is used to predict a time series of future fabric OBBs for each trajectory candidate. The shortest-path collision-free robot endpoint trajectory is selected as the planned trajectory.

III. Fabric State and Motion Modeling

A. Fabric State Modeling Using OBB

Detailed modeling of the fabric states does not work for dynamic motion planning of the fabric because many hard-to-estimate factors affect it during the robot motion. OBBs are used to approximate the fabric motion for collision check between the fabric and the obstacle without too much computation. This compensates for modeling errors caused by hard-to-estimate factors. The proposed fabric OBB model is shown in Fig. 3. It is calculated with the following steps:

1) $\boldsymbol{P}_r$ is downsampled with voxel grid and then denoised with radius outlier removal.
2) $\boldsymbol{P}_f$ is extracted from the downsampled and denoised $\boldsymbol{P}_r$ by removing the fabric clamping tool and robot point clouds based on the robot endpoint position, the fabric clamping tool size, and the fabric size.
3) The fabric OBB orientation in the robot working space θ_{obb} is calculated as

$$\theta_{obb} = \arctan\left(\frac{z_f^1}{x_f^1}\right) \tag{1}$$

where $(x_f^1, y_f^1, z_f^1)^\top$ is the first eigenvector of the principal component analysis (PCA) result of $\boldsymbol{P}_f$. We select the eigenvector so that $z_f^1 > 0$.

4) The x, y, and z axes of Σ_f are defined as

$$\boldsymbol{e}_f^x = (\cos\theta_{obb}, 0, \sin\theta_{obb})^\top, \tag{2a}$$

$$\boldsymbol{e}_f^y = (0, -1, 0)^\top, \tag{2b}$$

$$\boldsymbol{e}_f^z = (\sin\theta_{obb}, 0, -\cos\theta_{obb})^\top. \tag{2c}$$

5) The fabric OBB size $\boldsymbol{l}_{obb} = (l_{obb}^x, l_{obb}^y, l_{obb}^z)^\top$ is the length of the $\boldsymbol{P}_f$ projection on each axis of Σ_f separately, which is

$$l_{obb}^x = \max_{i=1}^{N}\left(\boldsymbol{e}_f^{x\top}\boldsymbol{p}_f^i\right) - \min_{i=1}^{N}\left(\boldsymbol{e}_f^{x\top}\boldsymbol{p}_f^i\right), \tag{3a}$$

$$l_{obb}^y = \max_{i=1}^{N}\left(\boldsymbol{e}_f^{y\top}\boldsymbol{p}_f^i\right) - \min_{i=1}^{N}\left(\boldsymbol{e}_f^{y\top}\boldsymbol{p}_f^i\right), \tag{3b}$$

$$l_{obb}^z = \max_{i=1}^{N}\left(\boldsymbol{e}_f^{z\top}\boldsymbol{p}_f^i\right) - \min_{i=1}^{N}\left(\boldsymbol{e}_f^{z\top}\boldsymbol{p}_f^i\right) \tag{3c}$$

where $\boldsymbol{p}_f^i$ is the position of each point in $\boldsymbol{P}_f$ and N is the number of points in $\boldsymbol{P}_f$.

6) The fabric OBB center position $\boldsymbol{p}_{obb}$ is derived as

$$\boldsymbol{p}_{obb} = \left(\boldsymbol{e}_f^x, \boldsymbol{e}_f^y, \boldsymbol{e}_f^z\right)(d_x, d_y, d_z)^\top \tag{4}$$

where

$$d_x = \frac{1}{2}\left\{\max_{i=1}^{N}\left(\boldsymbol{e}_f^{x\top}\boldsymbol{p}_f^i\right) + \min_{i=1}^{N}\left(\boldsymbol{e}_f^{x\top}\boldsymbol{p}_f^i\right)\right\}, \tag{5a}$$

$$d_y = \frac{1}{2}\left\{\max_{i=1}^{N}\left(\boldsymbol{e}_f^{y\top}\boldsymbol{p}_f^i\right) + \min_{i=1}^{N}\left(\boldsymbol{e}_f^{y\top}\boldsymbol{p}_f^i\right)\right\}, \tag{5b}$$

$$d_z = \frac{1}{2}\left\{\max_{i=1}^{N}\left(\boldsymbol{e}_f^{z\top}\boldsymbol{p}_f^i\right) + \min_{i=1}^{N}\left(\boldsymbol{e}_f^{z\top}\boldsymbol{p}_f^i\right)\right\}. \tag{5c}$$

$\boldsymbol{p}_{obb}$ is defined as the origin of Σ_f.

7) The fabric OBB state $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \left(\boldsymbol{p}_{obb}{}^\top, \boldsymbol{l}_{obb}{}^\top, \theta_{obb}\right)^\top \tag{6}$$

where $\boldsymbol{S} \in \mathbb{R}^7$.

B. Fabric Motion Modeling Using OBB-Transformer

Bounding box is widely used to represent objects such as vehicles and pedestrians in autonomous driving [26], [27]. In this paper, we propose two networks for modeling and predicting the fabric motion as the motion of OBBs: OBB-LSTM and OBB-Transformer. Long-short term memory and Transformer are known to be effective for modeling and predicting sequential data with time dependencies [28], [29]. We will show that OBB-Transformer outperforms OBB-LSTM in Section V, so we select OBB-Transformer for the collision-free motion planning.

Fig. 4 shows the OBB-LSTM architecture. The symbol $\oplus$ represents the element-wise addition. FC represents the fully connected layer. The inputs are sequences of robot endpoint positions $\{\boldsymbol{p}\} = \{\boldsymbol{p}_{t-L+1}, \ldots, \boldsymbol{p}_{t-1}, \boldsymbol{p}_t\}$, robot endpoint velocities $\{\boldsymbol{v}\} = \{\boldsymbol{v}_{t-L+1}, \ldots, \boldsymbol{v}_{t-1}, \boldsymbol{v}_t\}$, and fabric OBB states $\{\boldsymbol{S}\} = \{\boldsymbol{S}_{t-L+1}, \ldots, \boldsymbol{S}_{t-1}, \boldsymbol{S}_t\}$ where $\boldsymbol{p}_t, \boldsymbol{v}_t \in \mathbb{R}^3$ and L is the number of input and output time steps. The output is a sequence of predicted fabric OBB states $\{\bar{\boldsymbol{S}}\} = \{\bar{\boldsymbol{S}}_{t+1}, \bar{\boldsymbol{S}}_{t+2}, \ldots, \bar{\boldsymbol{S}}_{t+L}\}$. The decoder predicts variations of the fabric OBB state $\{\Delta\bar{\boldsymbol{S}}\} = \{\Delta\bar{\boldsymbol{S}}_t, \Delta\bar{\boldsymbol{S}}_{t+1}, \ldots, \Delta\bar{\boldsymbol{S}}_{t+L-1}\}$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

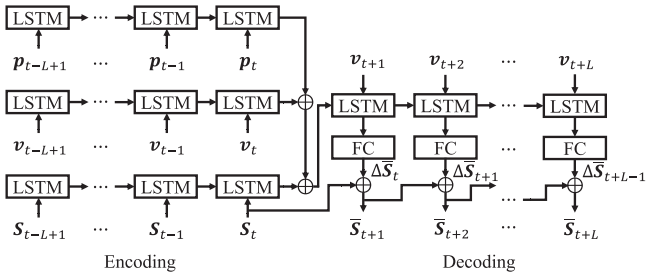


Fig. 4. The OBB-LSTM architecture.

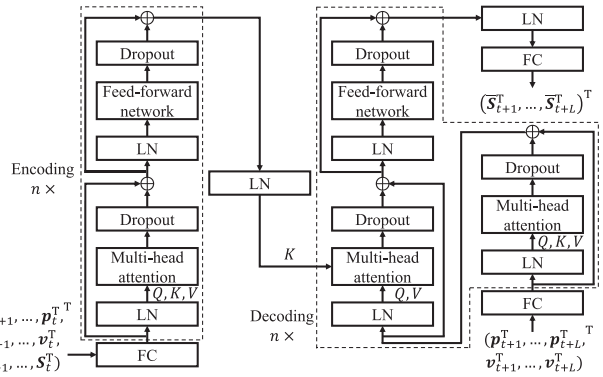


Fig. 5. The OBB-Transformer architecture.

from $\{\boldsymbol{S}_t, \bar{\boldsymbol{S}}_{t+1}, \ldots, \bar{\boldsymbol{S}}_{t+L-1}\}$, which means $\{\Delta\bar{\boldsymbol{S}}\} = \{\bar{\boldsymbol{S}}_{t+1} - \boldsymbol{S}_t, \bar{\boldsymbol{S}}_{t+2} - \bar{\boldsymbol{S}}_{t+1}, \ldots, \bar{\boldsymbol{S}}_{t+L} - \bar{\boldsymbol{S}}_{t+L-1}\}$.

Fig. 5 shows the OBB-Transformer architecture. LN represents the layer normalization. Q, K, and V are the queries, keys, and values in the attention mechanism. n is the number of encoding and decoding parts. Other symbols are the same as in OBB-LSTM. The inputs are concatenated $\{\boldsymbol{p}\}$, $\{\boldsymbol{v}\}$, and $\{\boldsymbol{S}\}$. The outputs are concatenated $\{\bar{\boldsymbol{S}}\}$. The decoder predicts $\{\bar{\boldsymbol{S}}\}$ directly. The embedding and positional encoding layers are removed here because the inputs and outputs are aligned and sequential.

IV. Robot Motion Planning for Fabric Collision Avoidance

A. Design of Robot Endpoint Trajectory

We want to find a smooth trajectory for the robot endpoint, to which the fabric clamping tool is attached, that avoids collision between the fabric and the obstacle. There are many trajectory generation methods to which modeling fabric motion using OBB-Transformer can be applied, such as rapidly-exploring random tree [30] and reinforcement learning [31]. In this paper, we design a simple yet effective trajectory generation scheme for a given obstacle using fifth-order polynomials [32] to show the effectiveness of the proposed OBB-Transformer. The robot endpoint trajectory is generated by combining two fifth-order polynomials connected at the control point $\boldsymbol{p}_c$. The first polynomial generates the trajectory from $\boldsymbol{p}_s$ to $\boldsymbol{p}_c$, whose boundary conditions are

$$\boldsymbol{p}_s = (p_s^x, p_s^y, p_s^z)^\top, \boldsymbol{v}_s = \boldsymbol{0}, \boldsymbol{a}_s = \boldsymbol{0}, \tag{7a}$$

$$\boldsymbol{p}_c = (p_c^x, p_c^y, p_c^z)^\top, \boldsymbol{v}_c = (v_c^x, 0, 0)^\top, \boldsymbol{a}_c = \boldsymbol{0}, \tag{7b}$$

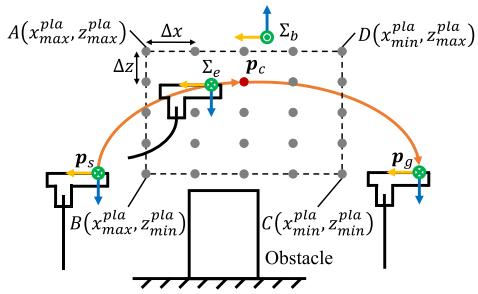


Fig. 6. The robot motion planning scheme. The robot endpoint trajectory combines two fifth-order polynomials connected at the control point $\boldsymbol{p}_c$. Two orange lines with arrows are polynomials from the start point $\boldsymbol{p}_s$ to the goal point $\boldsymbol{p}_g$. The red point is the corresponding $\boldsymbol{p}_c$ of the orange trajectory. The search area $ABCD$ is discretized as candidates of $\boldsymbol{p}_c$, which are shown as gray and red points. Δx and Δz are the discretization resolutions in the x and z axes separately. Some $\boldsymbol{p}_c$ candidates around the fabric clamping tool are not shown for clarity.

$$t_s = 0, t_c = \frac{2\left(p_c^x - p_s^x\right)}{v_c^x} \tag{7c}$$

where $v_c^x < 0$. The second polynomial generates the trajectory from $\boldsymbol{p}_c$ to $\boldsymbol{p}_g$, whose boundary conditions are

$$\boldsymbol{p}_c = (p_c^x, p_c^y, p_c^z)^\top, \boldsymbol{v}_c = (v_c^x, 0, 0)^\top, \boldsymbol{a}_c = \boldsymbol{0}, \tag{8a}$$

$$\boldsymbol{p}_g = \left(p_g^x, p_g^y, p_g^z\right)^\top, \boldsymbol{v}_g = \boldsymbol{0}, \boldsymbol{a}_g = \boldsymbol{0}, \tag{8b}$$

$$t_c = \frac{2\left(p_c^x - p_s^x\right)}{v_c^x}, t_g = \frac{2\left(p_g^x - p_s^x\right)}{v_c^x}. \tag{8c}$$

As shown in Fig. 6, the robot endpoint trajectory is in the robot working space with the following constraints

$$p_s^x > p_c^x > p_g^x, \tag{9a}$$

$$p_s^y = p_c^y = p_g^y, \tag{9b}$$

$$p_c^z \geq p_s^z = p_g^z. \tag{9c}$$

For given $\boldsymbol{p}_c$ and v_c^x, the robot endpoint trajectory is uniquely defined. We define v_c^x as a constant, so that each $\boldsymbol{p}_c$ has a unique corresponding trajectory.

B. Collision-Free Trajectory Planning

As shown in Fig. 6, the search area $ABCD$ in the robot working space is discretized to define candidates of $\boldsymbol{p}_c$. The related constraints are

$$p_s^x > x_{max}^{pla} > x_{min}^{pla} > p_g^x, \tag{10a}$$

$$z_{max}^{pla} > z_{min}^{pla} = p_s^z. \tag{10b}$$

The discretization resolutions Δx and Δz should be selected according to the tradeoff between computation time and path length. This issue is common to many collision avoidance methods; however, discussing the optimal setting is beyond the scope of this paper, so we just select it empirically.

We select robot endpoint trajectories for which all predicted OBB vertices' trajectories do not intersect the obstacle in the robot working space. For each selected robot endpoint trajectory, we use Shapely [33] to calculate the minimum distance between all predicted OBB vertices' trajectories and the obstacle, which is denoted as d_{min}. We select the robot endpoint trajectory with the shortest path whose predicted d_{min} is larger than a threshold

TABLE I
FABRIC MECHANICAL PROPERTIES

Fabric No.	A	B	C	D
Photo				
Material Textile	Polyester Woven	TC Woven	Cotton Woven	Polyester Knitted
T (mm)	0.08	0.36	0.32	0.44
ρ (kg/m^2)	0.05	0.16	0.13	0.15
M (mN·mm)	3.45	4.50	5.81	2.16

T: Thickness, ρ: Area Density, M: Bending Stiffness
TC: Polyester and Cotton

d^*_{min} as the planned robot endpoint trajectory. How to select the value of d^*_{min} will be shown in Section V.

V. EXPERIMENTS

A. Experimental System

The experimental system consists of an industrial robot (DENSO VS-087 6-axis articulated robot), a depth camera (Photoneo MotionCam-3D M+), and a robot control PC (Intel Core i9-10900 CPU, 64 GB RAM, and NVIDIA GeForce RTX 3090 GPU). A 3D-printed fabric clamping tool with two clamping plates is attached to the industrial robot as the end-effector. The depth camera captures $\boldsymbol{P}_r$ in real time at 13Hz. A rectangular fabric A (see Table I for fabric mechanical properties) with size M (W 280.0 mm $\times$ H 415.0 mm, including the clamped part) is clamped by the robot end-effector. The real-time operating system (TenAsys INtime 6) is used to control the entire experimental system.

To evaluate the collision avoidance performance of the proposed method, two rectangular obstacles with different dimensions are prepared. The dimension of Obstacle A is 200.0 mm $\times$ 200.0 mm $\times$ 950.0 mm, and the dimension of Obstacle B is 100.0 mm $\times$ 100.0 mm $\times$ 950.0 mm. The obstacle can be replaced with other objects, such as sewing machines and racks, depending on the real-world application.

B. Network Training

Based on $\boldsymbol{P}_r$, the fabric OBB is calculated using the VoxelGrid, StatisticalOutlierRemoval, PassThrough, and PCA classes in PCL 1.12.1 [34]. The OBB-LSTM and OBB-Transformer are implemented in PyTorch v1.12.1 [35]. By preliminary experiments $L = 5$ is selected for overall performance. For OBB-LSTM, the hidden state size is 8192. For OBB-Transformer, the number of heads is 8, the hidden state size is 512, the dropout ratio is 0.1, the number of encoding and decoding layers is 2, and the feed-forward network hidden state size is 2048. The batch size is 512. The optimizer is Adam [36]. The loss function is mean squared error (MSE). The maximum number of training epochs is 2000 and the early-stopping conditions are set.

The search area $ABCD$ is defined as $A = (250.0, 450.0, -920.0)$ mm, $B = (250.0, 450.0, -1000.0)$ mm, $C = (-150.0, 450.0, -1000.0)$ mm, and $D = (-150.0, 450.0, -920.0)$ mm. $\boldsymbol{p}_s = (350.0, 450.0, -1000.0)$ mm, $\boldsymbol{p}_g = (-250.0, 450.0, -1000.0)$ mm, and $v_c^x = -800.0$ mm/s are selected based on the sensing range and the sampling rate of the depth camera. For training dataset collection, $\boldsymbol{p}_c$ is randomly sampled 100 times within $ABCD$. The average number of OBBs per trajectory is 27. The total number of OBBs for training is 2693.

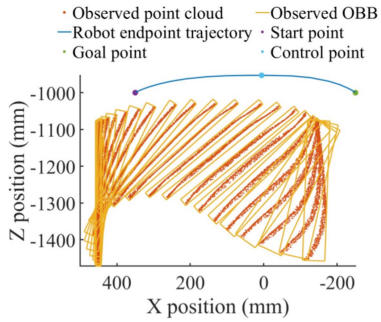


Fig. 7. An example of the fabric state modeling using OBB.

C. Experimental Results

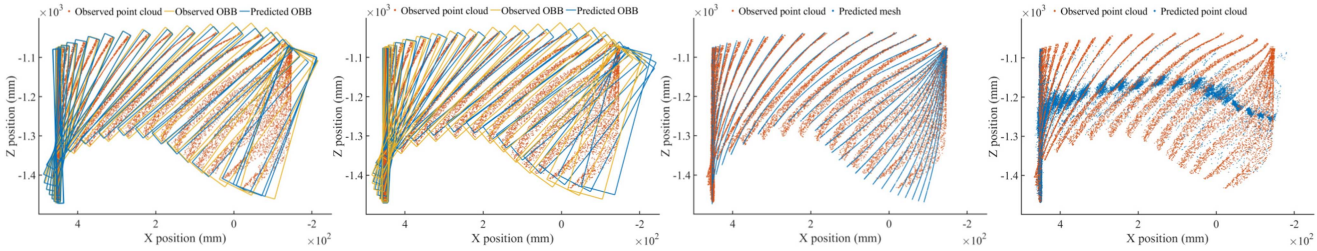
1) Fabric State Modeling Using OBB: Fig. 7 shows an example of the fabric state modeling using OBB. It can be seen that the fabric OBB models $\boldsymbol{P}_f$ and its motion approximates that of the fabric during the robot motion. Note that the depth camera cannot capture the entire fabric around the end of the robot endpoint trajectory due to the sensing limitations, so the partially observed $\boldsymbol{P}_f$ that appear around the end of the robot endpoint trajectory are excluded.

2) Comparison of Fabric Motion Modeling Performance: We compared the observed and predicted fabric motions using the proposed OBB-Transformer, OBB-LSTM, and conventional fabric modeling methods, including the mass-spring-damper 3D mesh model, PointRNN, and PointLSTM [37]. The mass-spring-damper 3D mesh model is from the cloth simulation function in Blender 4.0 [38]. PointRNN and PointLSTM predict the motion of fabric point clouds. In this paper, only the result of PointLSTM is reported because the result of PointRNN is similar to that of PointLSTM.

Fig. 8 shows the comparison of the observed and predicted fabric motions using the above methods. The mass-spring-damper 3D mesh model in Fig. 8(c) showed similar fabric motion along the trajectory, but the error between the predicted and observed fabric states was large around the end of the trajectory, even after extensive parameter tuning. The final observed point cloud and the predicted mesh did not completely overlap. PointLSTM successfully tracked the motion of the centroid of the observed point clouds in Fig. 8(d), but it did not preserve the shape of the observed point cloud well, especially the lower part.

The comparison between OBB-Transformer in Fig. 8(a) and OBB-LSTM in Fig. 8(b) is summarized in Table II. It shows the average displacement error (ADE), final displacement error (FDE), average intersection over union (AIOU), final intersection over union (FIOU) [27], average point cloud coverage rate (APCCR), and final point cloud coverage rate (FPCCR) of the predicted OBBs. The point cloud coverage rate is the percentage of the observed points within the corresponding predicted OBB. OBB-Transformer achieved lower ADE and FDE while

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) OBB-Transformer (b) OBB-LSTM (c) Mass-spring-damper 3D mesh model (d) PointLSTM

Fig. 8. Comparison of the observed and predicted fabric motions using (a) OBB-Transformer, (b) OBB-LSTM, (c) mass-spring-damper 3D mesh model, and (d) PointLSTM. OBB-Transformer and OBB-LSTM predict fabric motion using OBBs. Mass-spring-damper 3D mesh model predicts fabric motion using meshes. PointLSTM predicts fabric motion using point clouds.

TABLE II
COMPARISON OF OBB-TRANSFORMER AND OBB-LSTM

	OBB-LSTM	OBB-Transformer
ADE (mm, ↓)	16.92	**10.22**
FDE (mm, ↓)	19.45	**13.25**
AIOU (↑)	0.58	**0.66**
FIOU (↑)	0.54	**0.64**
APCCR (↑)	0.75	**0.86**
FPCCR (↑)	0.57	**0.67**

ADE/FDE: Average/Final Displacement Error
AIOU/FIOU: Average/Final Intersection Over Union
APCCR/FPCCR: Average/Final Point Cloud Coverage Rate
↓/↑: The smaller/larger value is better.

TABLE III
COORDINATES OF THE OBSTACLE AND THE CONTROL POINT $\boldsymbol{p}_c$ FOR EACH CASE IN FIG. 11

	$\boldsymbol{p}_c$ (mm)	x^{obs}_{min} (mm)	x^{obs}_{max} (mm)	z^{obs}_{max} (mm)
Case 1	(50.0, 450.0, −940.0)	174.6	382.1	−1348.3
Case 2	(−150.0, 450.0, −940.0)	148.7	358.0	−1347.9
Case 3	(−50.0, 450.0, −980.0)	198.1	298.9	−1348.0
Case 4	(−150.0, 450.0, −960.0)	156.6	256.2	−1348.5

obtaining higher AIOU, FIOU, APCCR, and FPCCR compared to OBB-LSTM, demonstrating better performance in modeling fabric dynamic motion.

Based on the above comparisons, OBB-Transformer is used to model the fabric motion for collision avoidance experiments in the following section. Note that the prediction error can be reduced by introducing the reactive motion planning scheme, which replans the collision-free trajectory at each sampling time. We leave this as future work due to the limitations of the currently available hardware.

3) Collision Avoidance Experiments: $\Delta x = 100.0$ mm and $\Delta z = 20.0$ mm are selected to define the $\boldsymbol{p}_c$ candidates. For a given obstacle located between $\boldsymbol{p}_s$ and $\boldsymbol{p}_g$, the depth camera is used to detect the obstacle. The obstacle top surface point cloud is extracted using the PassThrough and SACSegmentation classes in PCL 1.12.1 [34]. As shown in Fig. 9, the coordinates x^{obs}_{min}, x^{obs}_{max}, and z^{obs}_{max} are calculated from the extracted obstacle top surface point cloud to define the obstacle in the robot working space. Based on the detected obstacle, we predicted the dynamic motion of the fabric along the trajectory generated for each $\boldsymbol{p}_c$ using OBB-Transformer and calculated the predicted d_{min}.

First, we show how to select the collision check threshold d^*_{min} for selecting robot endpoint trajectories. Given Obstacle A with $x^{obs}_{min} = 175.5$ mm, $x^{obs}_{max} = 372.7$ mm, and $z^{obs}_{max} = -1348.1$ mm, OBB-Transformer predicted 16 out of 25 trajectories to be collision-free (the predicted d_{min} was positive). For each predicted collision-free trajectory, we performed 20 experiments.

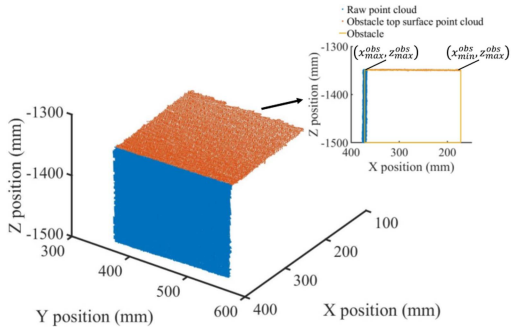


Fig. 9. An example of the obstacle detection.

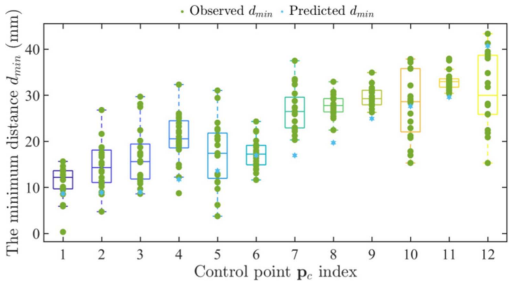
For 12 out of 16 predicted collision-free trajectories, no collisions between the fabric and the obstacle were observed during the 20 trials. For the remaining four trajectories, sometimes the fabric collided with the obstacle in the latter part of the trajectory, where the prediction error accumulated and became relatively large. As mentioned before, the prediction error can be reduced by introducing the reactivate motion planning scheme.

d^*_{min} is selected to mitigate the prediction error based on 12 collision-free trajectories. Fig. 10 shows the corresponding observed and predicted d_{min}. The variations of the observed d_{min} are not surprising because the fabric motion and deformation are influenced by hard-to-estimate factors and the prediction error accumulates and becomes relatively large in the latter part of the trajectory. We select $d^*_{min} = 15.6$ mm, which is twice as large as the maximum standard deviation of the observed d_{min} data shown in Fig. 10 to satisfy the two-sigma rule.

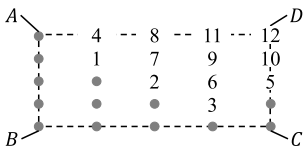
Next, we show that the planned robot endpoint trajectories can avoid collision between the fabric and the obstacle with the selected d^*_{min}. Fig. 11 shows fabric motions along the planned collision-free trajectories using the real robot with the fabric. We planned trajectories for Obstacle A and B placed at two different positions. For each predicted collision-free trajectory, we performed 10 experiments.

No collisions between the fabric and the obstacle were observed during the 10 trials for each collision-free trajectory.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

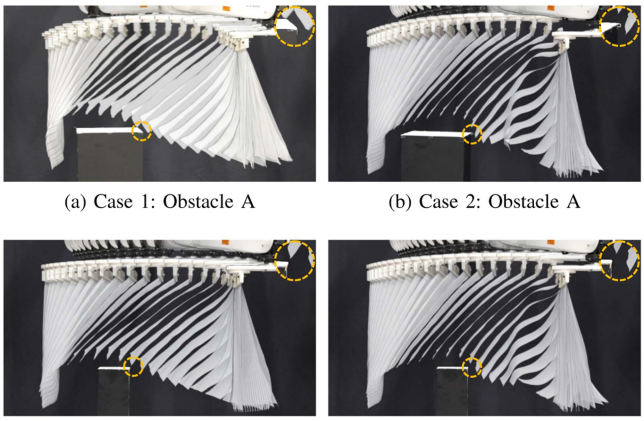


(a) Minimum distance d_{min} distribution



(b) Control point $\boldsymbol{p}_c$ position

Fig. 10. The observed and predicted minimum distance d_{min}. The observed d_{min} are green dots, which are summarized in the box plot. The predicted d_{min} are cyan hexagons. The positions of the control point $\boldsymbol{p}_c$ are labeled with the corresponding indices.



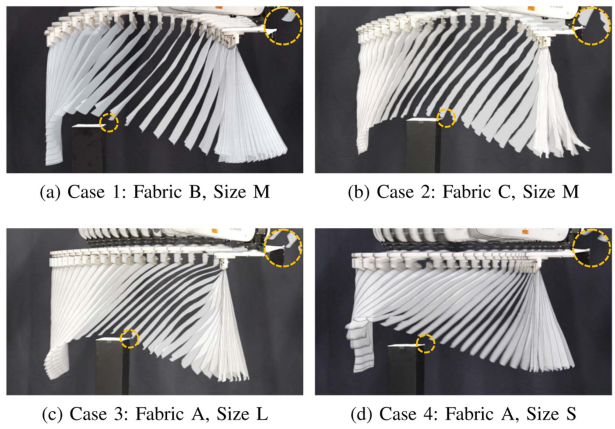
(a) Case 1: Obstacle A (b) Case 2: Obstacle A

(c) Case 3: Obstacle B (d) Case 4: Obstacle B

Fig. 11. Fabric motions along the planned collision-free robot endpoint trajectories with different obstacle settings. Fabric motion that is closest to the obstacle is enlarged in the yellow circle. The coordinates of the obstacle and the control point $\boldsymbol{p}_c$ for each case are in Table III. More fabric motions are in the supplementary video.

Compared to the naive solution, which lifts the fabric up high enough, moves it horizontally, and then puts it down, the proposed method has two advantages:

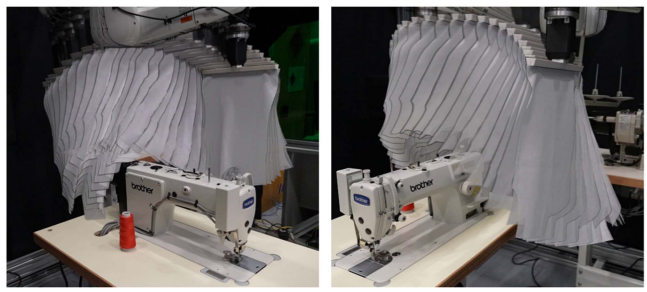
- The fabric travel path is shorter. The maximum path length of the proposed method shown in Fig. 11 is 632.0 mm (Case 2), while the minimum path length of the naive solution is 900.0 mm (150.0 mm $\times$ 2 + 600.0 mm). For the same cycle time to transport the fabric, the average velocity of the robot end-effector along the planned path is lower than that of the naive solution.
- The maximum height of the collision-free path is reduced. The maximum height of the path of the proposed method shown in Fig. 11 is -940.0 mm (Case 1 and Case 2), while the minimum height of the naive solution is -850.0 mm, 90.0 mm higher than the proposed method.



(a) Case 1: Fabric B, Size M (b) Case 2: Fabric C, Size M

(c) Case 3: Fabric A, Size L (d) Case 4: Fabric A, Size S

Fig. 12. Fabric motions along the planned collision-free robot endpoint trajectories with different types of fabrics. Fabric motion that is closest to the obstacle is enlarged in the yellow circle. The mechanical properties of fabrics are in Table I. More fabric motions are in the supplementary video.



(a) Front view (b) Back view

Fig. 13. T-shirt front panel motion along the planned 3D collision-free robot endpoint trajectory. More T-shirt front panel motions and corresponding point clouds are in the supplementary video.

Finally, we show that the proposed collision avoidance method can be applied to other types of fabrics. Fig. 12 shows fabric motions along the planned collision-free trajectories using the real robot with fabrics of different materials and sizes. The fabric mechanical properties are shown in Table I. The size L is W 280.0 mm $\times$ H 440.0 mm and the size S is W 280.0 mm $\times$ H 390.0 mm. For each predicted collision-free trajectory, we performed 10 experiments.

With a small amount of new data (20 trials), we retrained OBB-Transformer for a new piece of fabric. The generalization of the proposed method is limited and requires a small amount of new data, since the initial fabric OBB states do not contain information about the fabric material. No collisions between the fabric and the obstacle were observed during the 10 trials for each collision-free trajectory.

4) T-Shirt Front Panel 3D Transport Experiment: We extend the proposed collision avoidance method to the transport task of a T-shirt front panel in 3D space. The T-shirt front panel is made of fabric D (see Table I for the mechanical properties of the fabric). Its size is W 335.0 mm $\times$ H 450.0 mm. We use PALGRIP [39] as the end-effector to grasp the T-shirt front panel. Due to the confined space, the robot must cross over the sewing machine to deliver the T-shirt front panel for sewing. For each predicted collision-free trajectory, we performed 10 experiments.

Fig. 13 shows an example of the T-shirt front panel motion along the shortest 3D collision-free trajectory using the real robot. The trajectory parameters are $\boldsymbol{p}_s = (450.0, 250.0, -880.0)$ mm, $\boldsymbol{p}_c = (200.0, 350.0, -760.0)$ mm, and $\boldsymbol{p}_g = (-150.0, 450.0, -880.0)$ mm. No collisions between the T-shirt front panel and the sewing machine were observed during the 10 trials.

VI. Conclusion

In this paper, we have proposed a fabric dynamic motion modeling method and its application to the collision-free motion planning for the robotic fabric transport task. The fabric state is modeled as an OBB, and the fabric motion is modeled and predicted as a time series of OBBs using the proposed OBB-Transformer. We have also proposed a simple yet effective motion planning method using fifth-order polynomials to plan a collision-free trajectory using OBB-Transformer. Experimental results on the real robot system show that the planned collision-free trajectories work well for the fabric in both 2D and 3D scenarios. The proposed method can be used for different types of fabrics by retraining the network with a small amount of additional training data of the new fabric. However, since the initial observation of the fabric does not include information about the fabric material, the generalization to different types of fabrics is limited. Future work includes 3D reactive motion planning and the garment transport task.

References

[1] Reuters, "The 5 biggest users of industrial robots," 2019. Accessed: Oct. 23, 2024. [Online]. Available: https://www.reuters.com/graphics/HEALTH-CORONAVIRUS/JAPAN-FACTORIES/rlgvdonnjpo/

[2] KTK, "KTK - Screen printing machines | all over the world!" 2019. Accessed: Oct. 23, 2024. [Online]. Available: https://www.youtube.com/watch?v=Kho19hFn5U0&ab_channel=KTK

[3] M. B. Sarder, S. N. Imrhan, and N. Mandahawi, "Ergonomic workplace evaluation of an Asian garment-factory," *J. Hum. Ergol.*, vol. 35, no. 1–2, pp. 45–51, 2006.

[4] R. K. Seidu, E. A. Ofori, B. Eghan, G. K. Fobiri, A. O. Afriyie, and R. Acquaye, "A systematic review of work-related health problems of factory workers in the textile and fashion industry," *J. Occup. Health*, vol. 66, no. 1, 2024, Art. no. uiae007.

[5] The Hong Kong Cotton Spinners Association, *Textile Handbook*, 2000, pp. 6-125–6-142.

[6] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy for goal-conditioned dynamic manipulation of deformable objects," in *Proc. Robot. Sci. Syst.*, 2022.

[7] N. Mouazé and L. Birglen, "Deformation modeling of compliant robotic fingers grasping soft objects," *J. Mech. Robot.*, vol. 13, no. 1, Feb. 2021, Art. no. 011009.

[8] A. C. Dometios and C. S. Tzafestas, "Interaction control of a robotic manipulator with the surface of deformable object," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1321–1340, Apr. 2023.

[9] F. Liu, E. Su, J. Lu, M. Li, and M. C. Yip, "Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics," *IEEE Robot. Autom. Lett.*, vol. 8, no. 7, pp. 3964–3971, Jul. 2023.

[10] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 417–436, Feb. 2023.

[11] Y. Huang, C. Xia, X. Wang, and B. Liang, "Learning graph dynamics with external contact for deformable linear objects shape control," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3892–3899, Jun. 2023.

[12] B. Yang, C. Sui, F. Zhong, and Y. -H. Liu, "Modal-graph 3D shape servoing of deformable objects with raw point clouds," *Int. J. Robot. Res.*, vol. 42, no. 14, pp. 1213–1244, Dec. 2023.

[13] Z. Huang, X. Lin, and D. Held, "Mesh-based dynamics with occlusion reasoning for cloth manipulation," in *Proc. Robot. Sci. Syst.*, 2022.

[14] H. Shehawy, D. Pareyson, V. Caruso, S. D. Bernardi, A. M. Zanchettin, and P. Rocco, "Flattening and folding towels with a single-arm robot based on reinforcement learning," *Rob. Auton. Syst.*, vol. 169, Nov. 2023, Art. no. 104506.

[15] R. Hoque et al., "VisuoSpatial foresight for physical sequential fabric manipulation," *Auton. Robot.*, vol. 46, no. 1, pp. 175–199, Jan. 2022.

[16] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Sci. Robot.*, vol. 6, no. 54, May 2021, Art. no. eabd8803.

[17] J. Zhu et al., "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 67–77, Sep. 2022.

[18] P. Mitrano, D. McConachie, and D. Berenson, "Learning where to trust unreliable models in an unstructured world for deformable object manipulation," *Sci. Robot.*, vol. 6, no. 54, May 2021, Art. no. eabd8170.

[19] Y. Qin, A. Escande, F. Kanehiro, and E. Yoshida, "Dual-arm mobile manipulation planning of a long deformable object in industrial installation," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 3039–3046, May 2023.

[20] J. Huang, X. Chu, X. Ma, and K. W. S. Au, "Deformable object manipulation with constraints using path set planning and tracking," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4671–4690, Dec. 2023.

[21] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 49–54.

[22] S. Zimmermann, R. Poranne, and S. Coros, "Dynamic manipulation of deformable objects with implicit integration," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4209–4216, Apr. 2021.

[23] H. Ha and S. Song, "FlingBot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Proc. Conf. Robot. Learn.*, 2022, pp. 24–33.

[24] H. Choset et al. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT Press, 2005.

[25] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[26] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5632–5640.

[27] S. Saadatnejad, Y. Z. Ju, and A. Alahi, "Pedestrian 3D bounding box prediction," in *Proc. 10th Symp. Eur. Assoc. Res. Transp.*, 2022.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[30] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 995–1001.

[31] X. Li, H. Liu, and M. Dong, "A general framework of motion planning for redundant robot manipulator based on deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5253–5263, Aug. 2022.

[32] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*. Cambridge, MA, USA: MIT Press, 1981.

[33] S. Gillies et al., "Shapely," 2023. [Online]. Available: https://github.com/shapely/shapely

[34] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.

[35] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[37] H. Fan and Y. Yang, "PointRNN: Point recurrent neural network for moving point cloud processing," 2019, *arXiv:1910.08287*.

[38] Blender online community, "Blender - A 3D modelling and rendering package," 2018. [Online]. Available: https://www.blender.org

[39] A. Seino, F. Tokuda, A. Kobayashi, and K. Kosuge, "Passive actuator-less gripper for pick-and-place of a piece of fabric," *IEEE/ASME Trans. Mechatron.*, early access, Mar. 6, 2025, doi: 10.1109/TMECH.2025.3543557.