

Robotic Perception-motion Synergy for Novel Rope Wrapping Tasks

Zhaoyuan Ma and Jing Xiao¹

Abstract—This paper introduces a novel and general method to address the problem of using a general-purpose robot manipulator with a parallel gripper to wrap a deformable linear object (DLO), called a rope, around a rigid object, called a rod, autonomously. Such a robotic wrapping task has broad potential applications in automotive, electromechanical industries construction manufacturing, etc., but has hardly been studied. Our method does not require prior knowledge of the physical and geometrical properties of the objects but enables the robot to use real-time RGB-D perception to determine the wrapping state and feedback control to achieve high-quality results. As such, it provides the robot manipulator with the general capabilities to handle wrapping tasks of different rods or ropes. We tested our method on 6 combinations of 3 different ropes and 2 rods. The result shows that the wrapping quality improved and converged within 5 wraps for all test cases.

Index Terms—Perception for Grasping and Manipulation, Perception-Action Coupling, Reactive and Sensor-Based Planning, Bimanual Manipulation, Sensor-based Control

I. INTRODUCTION

Manipulating flexible wire, cable, rope, or other DLOs has a wide range of applications, such as catheter inserting [1], surgical suturing [2], automotive [3], aerospace [4], electromechanical industries [5], and so forth. The deformable property results in high-dimensional state space for modeling DLO, which makes manipulating DLO challenging.

Existing research for handling DLOs is focused on robot motion planning for basic tasks such as tying/untying knots [6], [7], [8], forming a given shape [9], contact-based cable routing [10], [11], inserting string and rope in a hole [12], and winding [13], [14]. While most of the research about DLO considers quasistatic manipulation, some also address dynamic manipulation [7], [14], [15].

Solving a DLO manipulation task usually contains three key steps: perception, modeling, and motion planning. Computer vision is often used to perceive a DLO's state. Some researchers attach AR tags along a wire harness as sampling points to detect deformation [3]. More generally, classic algorithms, such as uniform thresholding and Canny edge detector, are applied to extract cables in a controlled environment [10]. With the development of neural networks, deep learning is used to extract features of a DLO [9], [16]. Tactile servoing is another approach for DLO manipulation. She *et al.* design a gripper with GelSight for cable following and cable insertion tasks [17].

Common DLO modeling is done topologically or geometrically. Topology is helpful to describe the spatial relation of a DLO fragment in a knotting task, such as presented

in Wakamatsu's work [8]. Different geometric methods are broadly used for other types of tasks, for example, thin plate splines [2], parameterized curve with minimum energy-based scheme [4], multi-link system [7], truncated Fourier series model [11]. A bi-directional long short-term memory (LSTM) is also used to model the structure of a chain-like mass-spring system [9]. Alternatively, it is possible to bypass the modeling step with deep learning to create low-level joint control directly from input sensor data, as suggested by Suzuki *et al.* [6], who use a convolutional auto-encoder (CAE) and LSTM structure. The system uses RGB images and proximity sensor information as input to generate robot joint angles directly.

Once the model is ready, a task-oriented motion planning method is introduced to solve the problem, such as learning from demonstration [2], model predictive path integral (MPPI) control [9], planning based-on angular contact mobility index (ACMI) [10], hierarchical action primitives [11].

There are several papers investigating DLO wrapping tasks from dynamic or quasistatic aspects. Lee *et al.* focused on testing the performance of simulating a high volume of possible contacts [18]. Göbert *et al.* [13] designed a customized end-effector to attach a winding filament to a cyclical mold with a rotation axis and planned the path of the mold. Ito *et al.* studied using one whipping motion to wind a whip onto a target object with dynamic manipulation [14].

However, there is a lack of study to enable an off-the-shelf, general-purpose robot manipulator to perform general wrapping manipulation of a DLO around another object.

In this paper, we address the open problem of enabling a general-purpose robot manipulator with a simple parallel gripper to autonomously wrap a DLO around the object based on synergizing real-time perception and robot motion planning and control without requiring prior physical and geometrical information of both the DLO and the object. We are interested in providing a general robotic wrapping capability that can be applied to DLOs of varied materials, including different kinds of ropes, flexible cables, fibers, and so on. Specifically, the paper presents a novel approach for general-purpose robot wrapping operations with the following characteristics:

- It uses real-time perception of the objects and wrapping state to determine and adapt the robot motion for accomplishing and improving wrapping operations to achieve high-quality results.
- It re-uses and adjusts the canonical motion of making a single wrap of the DLO around the object to be flexible to the length of the DLO (thus the length of the coil as

¹The authors are with Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA {zma3, jxiao2}@wpi.edu

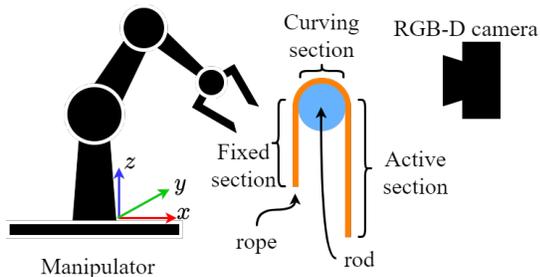


Fig. 1. The side view of the task setup. The system is composed of a manipulator, a rope over a rod, and an RGB-D camera.

the wrapping result) and to enable constant check and improvement of wrapping quality with feedback control.

- Hence, it is able to achieve high-quality wrapping without requiring prior knowledge of the physical and geometrical properties of the DLO and the other object.

This paper is organized as follows. Section II defines the problem and setup. Sections III to V introduces our approach in detail. Section VI describes our experiments and presents the results. Section VII concludes the paper.

II. TASK DESCRIPTION

We define the task we study as wrapping a DLO, called the “rope”, around a given rigid object, called the “rod”. A coordinate system is set up at the base of the manipulator. The positive directions along the x , y , and z axes are defined as “front”, “left”, and “up” respectively from the manipulator’s perspective. The rod is set in front of the manipulator. The manipulator has a gripper installed as the end-effector. In the initial state, the rope rides over the rod. We divide the rope into three sections: the fixed section, the curving section, and the active section. The fixed section is attached to a fixture or held by the manipulator without moving during the wrapping. The curving section has already been wrapped around the rod. The active section has sufficient length to create a few more wraps. The system could only obtain information through an RGB-D camera set to face the rod and the manipulator. Fig. 1 shows the side view of the setup. The goal of the task is for the manipulator to wrap the rope around the rod to create a helix that is tight in both radial and axial directions. Neither the dimensions nor the materials of the rod and the rope are known to the robot system.

III. APPROACH

We set up a real environment to realize the task, as shown in Fig. 2. A dual-arm manipulator (YuMi IRB 14000, ABB) is placed on a tabletop for the task. The fingers of the manipulator have been modified to have a 2.5mm wide 30mm long slot between two fingers when the gripper is fully closed. This allows the rope to slide in between while keeping the tension of the rope. A support structure is mounted in front of the manipulator to install the rod. An RGB-D camera (RealSense Depth Camera D415, Intel) is placed to face the manipulator. The manipulator and the camera are connected to a desktop computer as the controller.

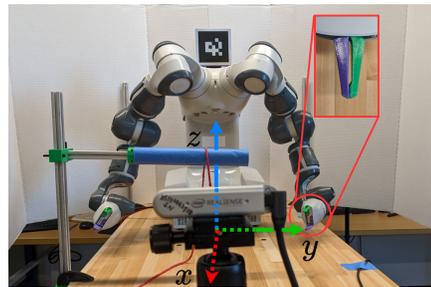


Fig. 2. The front view of the experiment setup. In order from closest to farthest are the camera, the rope (red), the rod (blue), and the YuMi manipulator. The world coordinate system is set on the tabletop.

Our system first conducts **rod estimation**: using collected RGB images and point cloud data to estimate the position, orientation, and dimensions of the rod with respect to the manipulator automatically. This step is detailed in Section IV-A.

Next, our system processes **rope estimation** by using RGB images and the rod estimation result to obtain the rope’s color and diameter information. This step is described in Section IV-B.

Subsequently, our system conducts a single wrap of the rope around the rod. It involves the following major perception and motion procedures:

- 1) **Grasp point selection**: using RGB images to search for a grasp point along the rope.
- 2) **Motion adjustment for wrapping**: generating the manipulator’s wrapping motion path based on adjustable parameters.
- 3) **Auxiliary motion generation to facilitate wrapping**: generating picking and releasing motions for the manipulator to perform before and after the wrapping motion respectively to complete the whole process.
- 4) **Real-time perception and feedback control**: using RGB images to estimate the wrapping result of the motion and adjust the parameters of the motion path generator in procedure 2).

Those procedures are described in Sections V-A to V-D in detail. Our system repeats the above four-step perception-motion synergy to wrap the rope around the rod while improving the wrapping quality until it can produce a tight helix wrap satisfactorily.

IV. ROD AND ROPE ESTIMATION

In this section, we present our approach for estimating the rod and the rope as a preprocessing step for a wrapping task.

A. Rod estimation

We developed a process to identify the rod’s dimensions, position, and orientation. Note that the rod’s axis does not need to be parallel to the yz -plane (Fig. 2). The process starts with localizing the camera with the fiducial marker [19] that was attached to the robot. The RGB-D camera provides a colored depth map of the workspace (Fig. 3a). Let P denote the set of all the 3D points in the map, and let Q

be the set of all 2D pixels with color information in the map. Each data point within the frame can be represented as $(p_x, p_y, p_z, q_x, q_y, c)$, where $(p_x, p_y, p_z) \in P$ is the position in the camera coordinate system, $(q_x, q_y, c) \in Q$ contains the pixel location on the image plane and the color c . For any given (p_x, p_y, p_z) , there is a unique (q_x, q_y) corresponding to it, and vice versa.

The system extracts the points between the robot and the camera, and above the tabletop from the point cloud data. The system downsamples the extracted point cloud to further reduce the number of data points and then applies DBSCAN [20] to create clusters according to the points' position. An example result is shown as Fig. 3b. Knowing the rod is the closest object to the camera, the system keeps the cluster with the shortest distance to remove the remaining robot parts. A 3D bounding box is created around this cluster (Fig. 3c). All points within the 3D bounding box are collected. White pixels in Fig. 3d indicate the selected data points on the 2D image plane. The system uses K-mean [21] to classify the data points into 2 clusters according to their hue, assuming that the rod has a more uniform color and occupies a larger portion of the image. Subsequently, the system retains the larger cluster, which contains the data points belonging to the rod. A maximum inscribed rectangle is used to fit the kept points, as shown in Fig. 3e.

Finally, the points within the rectangle are treated as points on the rod, with $Q' \subset Q$ denoting the pixels and $P' \subset P$ being the corresponding 3D points. P' is used to estimate the radius r_{rod} and the length l_{rod} of the rod. Our system creates a half-cylinder surface template based on the estimation and employs ICP [22] to match it to P' (Fig. 3f). At this point, the estimates of the rod's center position $(x_{rod}, y_{rod}, z_{rod})$, the rod's orientation, and r_{rod} are obtained.

B. Rope estimation

The point cloud only captures small fragments of the rope due to its limited 3D resolution (see Fig. 3b). Therefore, the system uses RGB images for rope-related processes. Our system estimates the color (hue range) and the diameter d of the rope using Q' (highlighted by the red rectangle in Fig. 4a). It extracts the hue channel of Q' to create a histogram. Otsu's method [23] is applied to the histogram to find a threshold that can separate Q' into the rope and the rod. Then a Gaussian function $N(\mu, \sigma)$ is used to approximate the normalized rope's hue histogram. The hue range of the rope is chosen as $[\mu - 3\sigma, \mu + 3\sigma]$.

The hue threshold found above is also applied to Q' to create a binary mask of the rope. The result is shown in Fig. 4b. A minimum area rectangle is generated to enclose the selected area, as in Fig. 4c. The width (short edge) of the rectangle is taken as d .

V. PERCEPTION-MOTION SYNERGY FOR WRAPPING

In this section, we introduce the perception and motion procedures that iterate to generate and improve wraps.

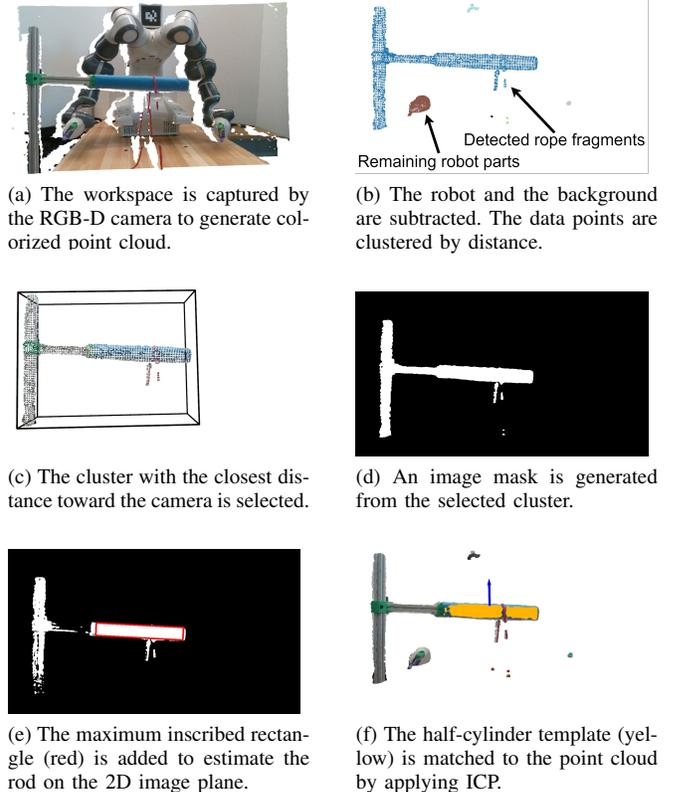


Fig. 3. Key steps to estimate the rod's dimension and pose.

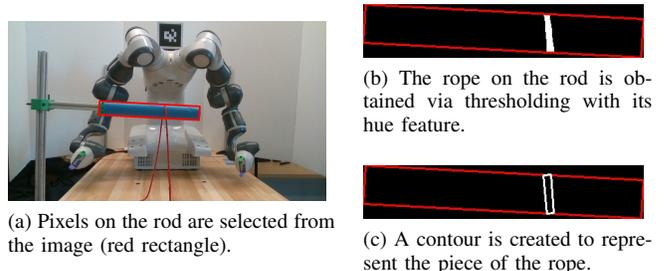


Fig. 4. Estimating the rope's width and color.

A. Grasp point selection

Wrapping a rope typically requires grasping both the fixed section and the active section and moving the active one around the rod. Finding the grasp points on the two sections is performed with an unwrapped rope. For each additional wrapping motion, only the grasp point on the active section needs to be updated. This step needs an input l_{gp} (in millimeters). It measures from the grasp point up to the lower edge of the rod, along the designated rope section. For the fixed section, l_{gp} is set as a constant. For the active section, the l_{gp} selection is given in Section V-B.

Grasp point selection starts with extracting the two sections of the rope from the image. A sub-image of Fig. 4a is created as shown in Fig. 5a, by extending the bounding box of the rod at both sides and downward. The system employs Ariadne+'s [16] pre-trained DeepLabV3+ [24] to create a binary mask M_1 from the sub-image. Compared

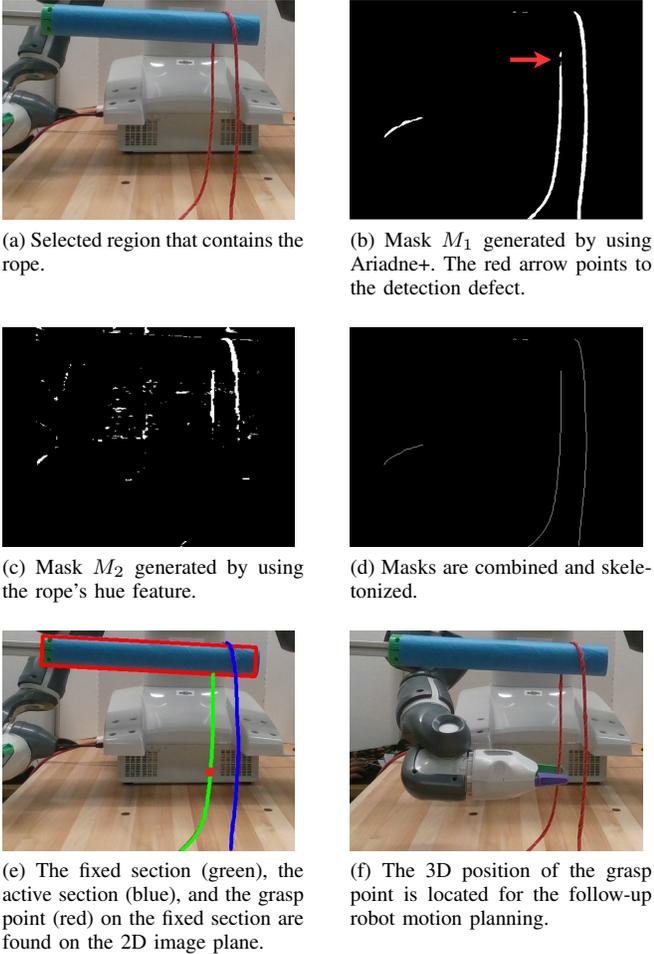


Fig. 5. Example of grasp point selection for the fixed section.

to traditional computer vision methods, this deep learning approach suggests possible rope areas with less noise. However, we observed that extraction defects may happen. For example, Fig. 5b shows an incomplete detection on the fixed section that is near the rod. Therefore, our system applies the rope's hue range as the threshold to the sub-image to create another binary mask M_2 , as shown in Fig. 5c. M_2 is used to connect detected rope segments in M_1 as much as possible. For a white pixel (q_x, q_y) in M_1 , if M_2 has a set of vertically connected white pixels that across at the same position (q_x, q_y) , all the pixels in this set are added to M_1 . The updated M_1 is then skeletonized [25] to reduce detected objects to 1-pixel width, as shown in Fig. 5d. Finally, our system searches for all lines from the bottom of the skeletonized mask, extracts the two longest lines, and calculates their center. The one near the right gripper is taken as the fixed section (highlighted with green), and the one near the left gripper is taken as the active section (highlighted with blue), see Fig. 5e.

After the two sections of the rope are detected, our system converts l_{gp} to the measurement in pixels and finds the corresponding point along the designated rope section. In order to find the corresponding point on the rope, our system

assumes that (i) the tangent point of a section is on the rod, and its position can be obtained by using the rod's estimation result, and (ii) for two points on a section, they have the same x value, and the distance on the 2D image and the distance in 3D are uniformly scaled. The assumptions hold because the two rope sections are kept vertical by an extra motion of the manipulator, as described in Section V-C, in addition to gravity. Hence, both fixed and active sections within the sub-image are roughly parallel to the yz -plane and tangent to the rod, as in Fig. 1. In Fig. 5e, the found point in the image plane is represented by a red dot. The position information of the pixel is converted back to the world coordinate system and used to move the robot's gripper, as shown in Fig. 5f.

B. Motion adjustment for wrapping

The wrapping motion is the most crucial as it determines the quality of the generated wrapping helix. It is performed in the presence of the estimation error or uncertainty of the rod's pose, dimension, the grasp points along the rope, and the unknown physical properties of the rope.

Our approach is to create a spiral curve with just a few parameters for the gripper to follow and then adjust the parameter values based on perception feedback. The goal is to achieve a resulting wrapping motion that can overcome the estimation errors and achieve a high-quality wrap in spite of the unknown physical properties of the rope.

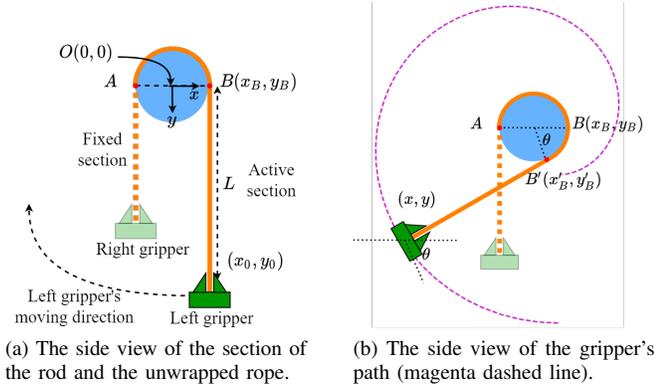
To design the canonical spiral curve for the gripper, we assume a rope hangs over the rod naturally due to gravity, creating contact with the rod on the upper half of the cylinder and leaving the surface of the rod tangentially at points A and B , as shown in Fig. 6a. Let O be the center of the rod's cross-section where the rope lies. For the convenience of deriving the spiral function, we define a new 2D coordinate system with the origin at O and the x and y axes as indicated in Fig. 6a on the cross-section. The 3D coordinate of O can be obtained from its relation to the rod's center.

The fixed section starts from point A , extends downwardly, and is held by the robot's right gripper. The active section is extended from point B to the grasp point (x_0, y_0) , with the length $l_{gp} = L$, as shown in Fig. 6a. We define $R = r_{rod} + \epsilon$, where ϵ is a variable for correcting the estimation error in r_{rod} . R is considered in its entirety. Its adjustment is discussed in Section V-D. We define $L = 2\pi R + L'$, where L' is a safe distance that considers the gripper's size. Once the left gripper starts to make a wrap, the tangential contact point B moves to B' . Note that the wrapping angle $\theta \in [0, 2\pi]$ is the angle $\angle B'OB$. As an additional $\widehat{BB'} = \theta R$ of the rod is covered by the rope, the distance between the gripper to the tangential point B' is reduced to $L - \theta R$, as in Fig. 6b.

Now we consider the following spiral curve with respect to the rod coordinate system:

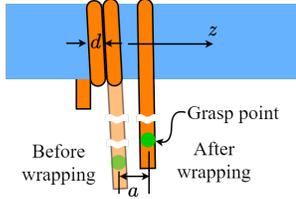
$$\begin{cases} x = R \cos \theta - (2\pi R + L' - \theta R) \sin \theta \\ y = R \sin \theta + (2\pi R + L' - \theta R) \cos \theta \\ z = a\theta/2\pi \end{cases} \quad (1)$$

where a is the displacement of the gripper along the rod's axial direction for one wrap (Fig. 6c), which can be decided



(a) The side view of the section of the rod and the unwrapped rope.

(b) The side view of the gripper's path (magenta dashed line).



(c) The front view of the rope's diameter and the wraps' advance. The translucent orange section indicates the active section before the wrapping. The opaque one on the right is after wrapping.

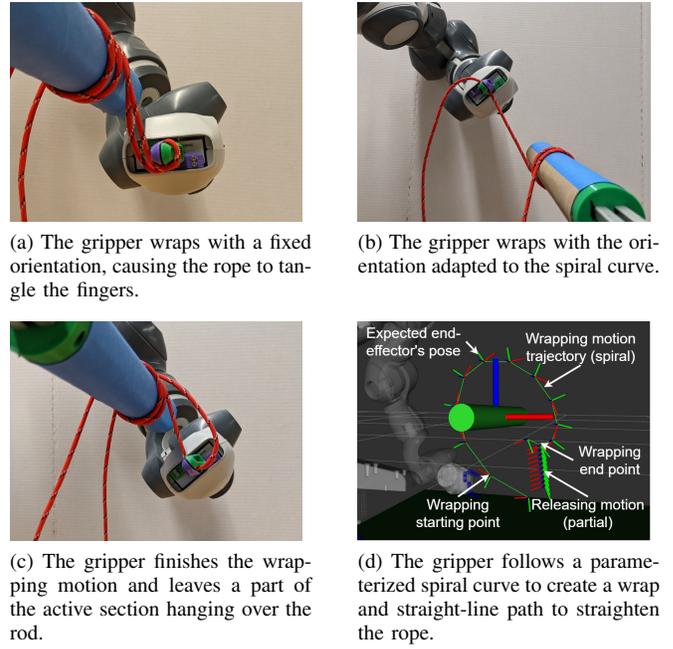
Fig. 6. Spiral curve for the robot's gripper.

through feedback (see Section V-D). The 2D projection of this spiral path for the gripper is shown as the magenta dashed curve in Fig. 6b. Our system searches for the safe distance $L' \in [L'_{\min}, L'_{\max}]$ so that the wrapping path has a feasible inverse kinematics (IK) solution.

Note that the goal of the wrapping is to create wraps that are tight along the radial and axial directions of the rod, such that the wrap along the radial direction is as close to the radius of the rod as possible, and the distance along the axial direction from the center of two adjacent wraps equal to the diameter of the rope.

During the wrapping process, when the wrapping angle changes to θ , the gripper rotates to the same angle θ about the rod's axis simultaneously, as indicated by its orientation in Fig. 6b. Without this rotation, the rope tends to entangle the fingers and hinder the gripper's opening motion at the end of the wrap, as shown in Fig. 7a. The results of wrapping with the change of the gripper orientation are shown in Fig. 7b ($\theta = 120^\circ$) and Fig. 7c ($\theta = 330^\circ$). At this stage, the position and orientation of the robot gripper along the spiral path have been determined with respect to the rod coordinate system, which can be converted to the world coordinate system.

In practice, our system discards the first and the last sampled points ($\theta = 0$ and 2π) and takes samples along the spiral curve to generate the path of the wrapping motion. The wrapping motion is connected with auxiliary motions to prepare the rope for wrapping and to release it after wrapping, as detailed in Section V-C. Fig. 7d illustrates the connection of the wrapping motion to the releasing motion.



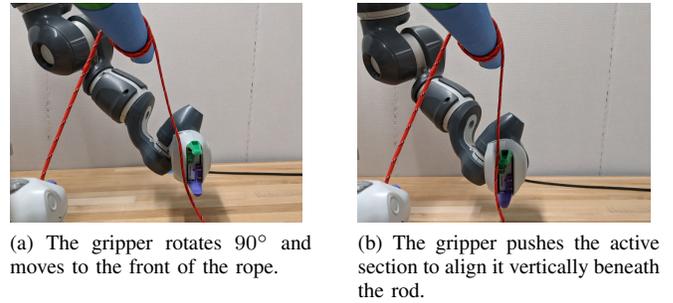
(a) The gripper wraps with a fixed orientation, causing the rope to tangle the fingers.

(b) The gripper wraps with the orientation adapted to the spiral curve.

(c) The gripper finishes the wrapping motion and leaves a part of the active section hanging over the rod.

(d) The gripper follows a parameterized spiral curve to create a wrap and straight-line path to straighten the rope.

Fig. 7. The wrapping motion.



(a) The gripper rotates 90° and moves to the front of the rope.

(b) The gripper pushes the active section to align it vertically beneath the rod.

Fig. 8. The auxiliary motions.

C. Generation of auxiliary motions to facilitate wrapping

Rope picking and releasing motions are defined based on the spiral wrapping motion. Picking is a point-to-point motion that contains three key poses: 1) the entry pose, where the gripper moves to the point with the opening facing toward the rope at a distance. 2) the grasp pose, where the rope is squeezed in between the fingers of the gripper, and 3) the connection pose, where the gripper moves to the starting point of the spiral path. The grasp pose is derived as in Section V-A with L being the input value for l_{gp} . The entry pose is obtained by offsetting the grasp pose to move the gripper away from the rope.

The release motion is designed to straighten the rope and to move the gripper away from the rope to avoid occlusion for image processing after the wrapping motion is done. Straightening the rope happens after the gripper arrives at the last sampled point of the spiral (see Fig. 7c and 7d). The gripper keeps its current orientation and moves downward to follow a straight-line path, sliding along the rope to flip the active section to the front. Then the gripper releases

the rope and withdraws to be away from the rope. At this point, if the remaining length of the active section is longer than the rod's height, part of the section may land on the table, preventing the rope drops vertically from the rod. An additional rope alignment step is added. The left gripper rotates 90° to increase the contact area with the rope. Then it moves to the front of the active section, as shown in Fig. 8a and pushes the rope towards the manipulator to ensure that the rope is roughly vertical beneath the rod (Fig. 8b). Finally, the gripper moves away from the rope to leave space for the next grasp point selection step.

D. Real-time Perception and feedback control

Following the completion of one wrap, our system takes an image of the result and checks for the tightness along the radial direction (refer to as the height) and the axial direction (refer to as the advance) of the wrap, which is then used for feedback control to improve the next wrap.

It starts by checking the height of the last wrapped part of the rope. At this step, a rectangle that contains the rod and some areas below is extracted from the image to include pieces of rope that could possibly hang under the rod, as shown in Fig. 9a. The rope is extracted from the image by using the rope's hue range as the threshold. Assuming that the rope segments in this binary image share the same width d as the detected rope's diameter, our system searches the image from right to left, row by row. For each row, it removes the first d white pixels and keeps the next set of connected white pixels with a total number less than d . The remaining white pixels belong to the rope segment created by the last wrap. The segment is skeletonized to find the valley point along it, as shown in Fig. 9b. The pixel distance from this point to the bottom of the rod is taken as the height feedback of the last wrap, noted as h . This feedback is used to estimate the radius R . The feedback controller updates the R by:

$$R_{n+1} = R_n - K_{PR} \cdot q_r, \text{ where } q_r = h - t_R \quad (2)$$

where K_{PR} is the proportional coefficient, t_R is a threshold. The stop condition is when $h \leq t_R$.

The RGB image is also used to check the advance. Pixels of the rod's area (Fig. 9c) are selected and processed to extract the rope. Like height-checking, the system uses the threshold to create a binary image and scans each row to find the first set of connected white pixels. If the number of pixels is greater than $1.5d$, the system assumes that the last two wraps are in close contact. Otherwise, the set of connected black pixels adjacent to the left of the white pixel set is considered the gap area. The rope area created by the last wrap S_r and the gap between the last two wraps S_g are measured (Fig. 9d). The feedback controller updates a by:

$$a_{n+1} = a_n - K_{Pa} \cdot q_a, \text{ where } q_a = S_g / (S_g + S_r) \quad (3)$$

where K_{Pa} is the proportional coefficient. There are two stop conditions for learning the advance: 1) $q_a = 0$, 2) for two consecutive wraps n and $n + 1$, $|q_{an} - q_{a(n+1)}| < t_a$, where t_a is a threshold. When the wrapping output meets either condition, the system stops updating the advance.

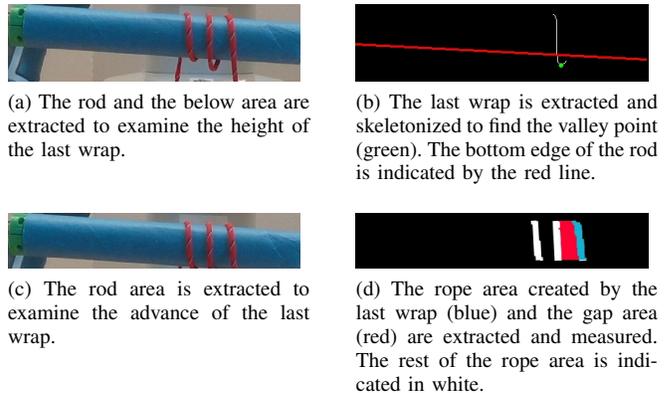


Fig. 9. Image processing to examine the quality of the last wrap.

VI. EXPERIMENTS AND RESULTS

In this section, we describe our experiments to implement and test our approach described in Sections III to V and present the results. We then discuss the performance and potential improvement of the wrapping algorithm.

A. Experiment setup

The PC of our system is equipped with an Intel Core i5-7500, 16GB RAM, and an NVIDIA GeForce GTX 1050 Ti, running Ubuntu 20.04 with ROS Noetic. The YuMi was configured to run in manual mode, with 100% speed, controlled by a modified KTH's YuMi package [26]. To achieve the required gripper motion during the wrapping as mentioned in Section V-B, Axis 6 of the left arm was set to 216° at the beginning of wrapping and -144° when finished. Trac-IK [27] was used to solve the IK for the remaining 6 axes of the left arm and the 7 axes of the right arm for expected gripper poses. The solver's solving timeout and the error toleration were set to 50ms and 1mm, respectively.

The system parameters were set as follows. For the fixed section grasp point selection, $l_{gp} = 13\text{mm}$. The safe distance was $L' \in [20, 60]\text{mm}$. The system's feedback control coefficients were given as $K_{PR} = 0.001$, and $K_{Pa} = 0.04$. The thresholds were set to $t_R = 1.5d$ and $t_a = 5\%$. The system's advance feedback started with the initial $a_0 = 20\text{mm}$. For a rod with an estimated radius r_{rod} , the initial $R_0 = 1.5r_{rod}$. In practice, with a larger R , some waypoints of the spiral path may lie outside the manipulator's workspace, which prevents the manipulator from executing the path. If this happens, R is reduced by a small value, which is set to 5mm in the experiments, to generate a new spiral path. This process repeats until the manipulator is able to follow the path to get height feedback for the first time. Then the system uses feedback control to tune R .

To validate the capability of handling objects with unknown attributes, we tested our system on two cylindrical rods and three ropes, namely Rod1, Rod2, Rope1, Rope2, and Rope3. Two rods share the same length of $l = 280\text{mm}$ but have different radii. Table I shows the ground truth, the mean and the standard deviation (SD) of the estimated radius for each rod. We ran the estimation 10 times per rod.

TABLE I

THE GROUND TRUTH AND THE ESTIMATION OF r_{rod} (OVER 10 TIMES)

	Ground truth	Estimation mean(SD)	Range
Rod1	21mm	18.5(± 1.8)mm	17.4 ~ 20.7mm
Rod2	17mm	14.3(± 1.1)mm	13.6 ~ 14.6mm

TABLE II

THE SYSTEM SELECTED PARAMETER VALUES FOR RADIAL TIGHTNESS

	Estimated radius r_{rod}	R	L'
Rod1	20.7mm	21.0mm	60.0mm
Rod2	14.6mm	16.9mm	60.0mm

Rope1 is complaint and light yarn (Softex Chunky Solid Yarn, Bernat). Rope2 is complaint and heavier yarn (Chenille Home Yarn, Loops & Threads). Rope3 is stiff paracord (1/8 in. x 50 ft. Assorted Color Paracord, Everbilt). We tested our system on the 6 combinations of the rods and ropes.

Note that the active section length that the auxiliary motion can handle is limited and varies depending on the rope length. If the remaining rope is not sufficient for more wraps. We enable the manipulator to continue practicing wrapping by unraveling the rope manually and letting the manipulator repeat the whole process.

B. Wrap quality

We used one estimated radius r_{rod} for each rod to examine the wrapping algorithm. The system uses the method mentioned in Section V-B to obtain R and L' . We find that the system-selected values of the two parameters are rope-independent. The values are shown in Table II.

The robot took 5 trials to reach the stop condition for learning the advance in the axial direction for {Rod1, Rope3}, {Rod2, Rope1}, and {Rod2, Rope3}. For {Rod1, Rope1} and {Rod2, Rope2}, it took 3 trials. It achieved the axial tightness in the first trial for {Rod1, Rope2}. The result can be found in Table III. These images were taken by the RGB-D camera and were also used for the feedback control

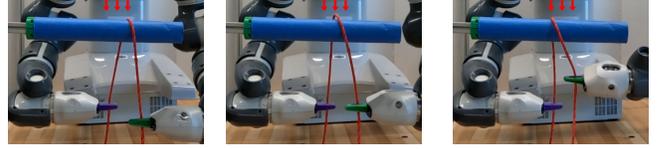


Fig. 10. The fixed end of the Rope3 is shifted during creating the first wrap because of its stiffness. The red arrows are used as references.

process. Note that since Rope3 has a higher stiffness, making the first wrap tends to push the fixed section to create a larger gap compared to the following wraps (see Fig. 10). This only happens to Rope3 when no previous wrap is made. Therefore, we always kept one pre-wrap on the rod that was not taken into account when testing Rope3 on both rods.

For all cases, the tightness along the radial direction was met. For cases with Rope1 and Rope2, the axial tightness was also met. For cases with Rope3, the final result was comparable with the wrapping result by a person manually (see the last column of Table III).

The attached video first demonstrates the feedback control process described in Section V-D, referred to as the training session. It next shows how the system performs after it is trained with the adjusted parameter values to verify the stability, referred to as the testing session.

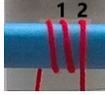
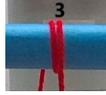
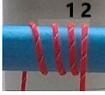
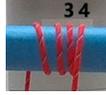
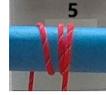
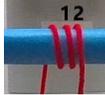
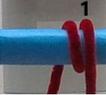
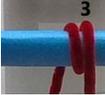
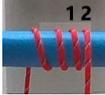
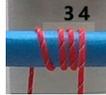
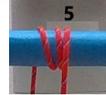
C. Time efficiency analysis

We report the time consumption of our method as the computation time and the total execution time. The former includes all the calculation steps that are described in both Sections IV and V. The latter includes the computation time for Section V and execution of the picking, wrapping, and releasing motions. All measurements are repeated 10 times.

Our system spends 13.197(± 0.402)s for the rod estimation and 1.042(± 0.053)s for the rope estimation at the beginning of a wrapping task. These are the pre-processing steps before wrapping and only need to be done once for each combination. The average time spent for the major procedures of creating one wrap is as follows: grasp point selection takes 4.447(± 0.082)s, wrapping motion adjustment and auxiliary

TABLE III

WRAPPING QUALITY OF EACH CASE. THE NUMBERS ABOVE THE WRAPS INDICATE THE TRIAL NUMBER. a_i IS THE CORRESPONDING ADVANCE THAT IS USED TO ACHIEVE THAT WRAP.

	Rope1		Rope2			Rope3			Human wrapping
Rod1	 $a_1 = 20.0$ $a_2 = 11.2$	 $a_3 = -2.7$ (complete)	 $a_1 = 20.0$ (complete)			 $a_1 = 20.0$ $a_2 = 2.4$	 $a_3 = -11.8$ $a_4 = -18.1$	 $a_5 = -21.0$ (complete)	 Wrapping performed by hands as a comparison.
Rod2	 $a_1 = 20.0$ $a_2 = 10.3$	 $a_3 = 9.7$ $a_4 = 6.0$ (complete)	 $a_1 = 20.0$	 $a_2 = 17.3$	 $a_3 = 17.0$ (complete)	 $a_1 = 20.0$ $a_2 = 0.1$	 $a_3 = -7.3$ $a_4 = -7.4$	 $a_5 = -11.2$ (complete)	

motion generation take $2.302(\pm 0.050)$ s, real-time perception and feedback control takes $0.312(\pm 0.016)$ s.

We measured the total execution time of one wrap for {Rod1, Rope1}. The mean is $63.933(\pm 0.611)$ s, which is an order of magnitude higher than the computation time for motion planning. Therefore, computation is much faster than the physical wrapping motion.

D. Discussion

We have observed that the radial feedback meets the stop condition from the first wrap for all test cases. Due to the camera's perspective and the methods mentioned in Section IV-A, the estimated r_{rod} is always smaller than the ground truth. With R being less or equal to the ground truth, the gripper slides slightly down the active section of the rope during the wrapping to compensate for the insufficient rope length from the gripper to the rod. This motion keeps the tension of the section and results in radial tightness. This suggests that to wrap over a solid of revolution, an R that is smaller than the ground truth can be tolerated and often helpful for radial tightness.

VII. CONCLUSIONS

In this paper, we present a novel method of perception-motion synergy to solve the problem of using a general-purpose robot manipulator with a parallel gripper to wrap ropes around a rigid rod. This method is based on a parameterized canonical motion and does not require prior knowledge of the rope or the rod. It uses RGB-D images to estimate the state of the rope and rod, evaluates the quality of each wrap, and generates feedback from the evaluation to improve motion planning. We tested our method with 6 combinations of ropes and rods. The result shows that our general method applies to different ropes and rods very well. However, our current method estimates wrapping quality based on colors and cannot handle two wraps that cross each other. In addition, the perception step only happens before and after a wrapping motion; thus, information during the wrapping motion is not perceived, which could be informative to wrapping motion adjustment. In the next step, we will expand our method to using other types of rods, for instance, solids of revolution other than cylinders, such as a cone, and to incorporate force control and tactile sensing to achieve more accurate and robust perception. We will also improve the efficiency of motion planning.

REFERENCES

- [1] J. Jayender, R. Patel, and S. Nikumb, "Robot-assisted catheter insertion using hybrid impedance control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 607–612.
- [2] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4111–4117.
- [3] X. Jiang, K.-m. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in car production line," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 490–495.
- [4] A. Shah, L. Blumberg, and J. Shah, "Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1823–1838, 2018.
- [5] J. Franke and A. Dobroschke, "Robot-based winding-process for flexible coil production," in *Proc. Elect. Manufact. Tech. Conf.*, 2009, pp. 157–163.
- [6] K. Suzuki, M. Kanamura, Y. Suga, H. Mori, and T. Ogata, "In-air knotting of rope using dual-arm robot based on deep learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6724–6731.
- [7] 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.
- [8] H. Wakamatsu, E. Arai, and S. Hirai, "Knotting/unknottting manipulation of deformable linear objects," *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 371–395, 2006.
- [9] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2372–2379, 2020.
- [10] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 16–23, 2020.
- [11] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, H. Wang, and D. Navarro-Alarcon, "Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5222–5229, 2022.
- [12] W. Wang, D. Berenson, and D. Balkcom, "An online method for tight-tolerance insertion tasks for string and rope," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2488–2495.
- [13] A. Göbert, A. Deetman, A. Rossi, O. Weyhe, and P. Eversmann, "3dwoodwind: robotic winding processes for material-efficient lightweight veneer components," *Constr. Robot.*, vol. 6, no. 1, pp. 39–55, 2022.
- [14] Koichiro Ito, Y. Yamakawa, and M. Ishikawa, "Winding manipulator based on high-speed visual feedback control," in *2017 IEEE Conf. Control Technol. Appl. (CCTA)*, 2017, pp. 474–480.
- [15] H. Zhang, J. Ichnowski, D. Seita, J. Wang, H. Huang, and K. Goldberg, "Robots of the lost arc: Self-supervised learning to dynamically manipulate fixed-endpoint cables," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 4560–4567.
- [16] A. Caporali, R. Zanella, D. De Gregorio, and G. Palli, "Ariadne+: Deep Learning-based Augmented Framework for the Instance Segmentation of Wires," *IEEE Trans. Ind. Inform.*, pp. 1–1, 2022.
- [17] Yu She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *Int. J. Robot. Res.*, vol. 40, no. 12-14, pp. 1385–1401, 2021.
- [18] J. Lee, M. Lee, J. Yoon, and D. Lee, "A parallelized iterative algorithm for real-time simulation of long flexible cable manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 12040–12046.
- [19] ar_track_alvar. [Online]. Available: http://wiki.ros.org/ar_track_alvar
- [20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226–231.
- [21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Stat. Probab., Vol. 1: Stat.*, vol. 5.1, pp. 281–298, 1967.
- [22] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.
- [23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proc. Conf. Eur. Comput. Vis.*, 2018, pp. 833–851.
- [25] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [26] kth-ros-pkg/yumi. [Online]. Available: <https://github.com/kth-ros-pkg/yumi>
- [27] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, 2015, pp. 928–935.