Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Planning of One-Handed Knotting/Raveling Manipulation of Linear Objects

Hidefumi Wakamatsu, Akira Tsumaya, and Eiji Arai
Dept. of Manufacturing Science, Osaka Univ.
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
{wakamatu, tsumaya, arai}@mapse.eng.osaka-u.ac.jp

Shinichi Hirai
Dept. of Robotics, Ritsumeikan Univ.
1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan
hirai@se.ritsumei.ac.jp

Abstract - A planning method for linear object manipulation including knotting/unknotting by one hand is proposed. Firstly, topological states of a linear object are represented as finite permutations of crossing points. Secondly, transitions among topological states are defined. Then, we can generate possible sequences of state transitions, that is, possible manipulation processes from the initial state to a given objective state. Thirdly, a method for determination of grasping points and their moving direction is proposed in order to realize derived manipulation processes. Furthermore, a planning method for one-handed manipulation is proposed. Knotting by one hand is possible as any manipulation processes can be realized by iteration of one-handed operations. Finally, it is demonstrated that our developed system based on the above method can generate manipulation plans for raveling out of an overhand knot.

Index Terms - Linear Object, Manipulation, Knotting, Planning

I. Introduction

The majority of manipulative tasks, including grasping and assembly, are performed through mechanical contact. A manipulation process of rigid objects can be represented as a sequence of finite contact states. Therefore, planning methods for rigid object manipulation have been studied by using contact state graphs[1][2]. However, systematic approach to the planning of deformable object manipulation has not been established yet. We have proposed a qualitative representation method of thin object manipulation considering the contact state of the object and applied it to manipulation planning[3].

Deformable linear objects such as tubes, cords, and wires are used widely ; data transmission, object transportation, fixing or packing of objects, and so on. Such manipulative tasks include knotting. On the other hand, self-entwining of linear objects should be avoided during their manipulative processes. Therefore, it is important for linear object manipulation to analyze knotting or entwining. Hopcroft et al. have devised a grammar of knots to express various knotting manipulation[4]. Leaf has described deformed shape of threads in a fabric geometrically[5]. Phillips et al. have simulated knot tying of a thread using a particle-based model of the thread[6]. Matsuno et al. realized a task of tying a cylinder with a rope by a dual manipulator system identifying the rigidity of the rope from visual information[7]. Morita et al. have been developing a system for knot planning from observation of human demonstrations[8].

To make a bowknot, for example, we manipulate a linear object dexterously by using several fingers of both hands for bending, twisting, holding, and/or binding. However, how to tie a bowknot of us depends on our physical makeup and experience, so it is not unique. We can generate manipulation plans suitable for hardwares with unlike physical makeup of human if manipulation processes for knotting/unknotting a linear object can be modeled. Therefore, in this paper, we propose a method for automatic planning and execution of linear object manipulation including knotting/unknotting, especially, by one hand.

Firstly, a qualitative representation of the crossing state of a linear object in three-dimensional space is proposed. Secondly, transitions among those states are defined by introducing four kinds of basic operations. Then, a manipulation process of a linear object can be represented as a sequence of crossing state transitions. Thirdly, a procedure to determine grasping points and their moving direction for realization of manipulation processes is explained. Furthermore, it is shown that any manipulation processes can be realized by one hand and a planning method for one-handed manipulation is proposed. Finally, we demonstrate raveling out of an overhand knot performed by a vision-guided manipulator system to show the usefulness of our approach.

II. Representation of Crossing States

First, we propose a method to represent the state of linear objects in three-dimensional space.

Let us project the 3D shape of a linear object on a 2D projection plane. Then, the projected 2D curve may cross with itself. Note that how to cross of the 2D curve depends on the projection plane.

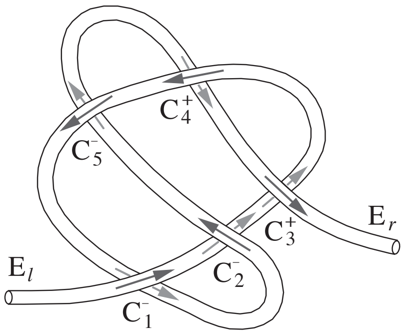


Fig. 1 Example of knotted linear object

Next, let us number crossing points of the object along it. Fig.1 shows an example of a linear object. It has 5 crossing points and their sequence is E_l-C_1-C_2-C_3-C_4-C_5-C_1-C_2-C_5-C_4-C_3-E_r, where E_l, E_r, and C_1 through C_5 represent the left endpoint, the right endpoint, and crossing points, respectively. Then, we can define the state of the object as a sequence of its crossing points. At each crossing point, the upper part C_i^u and the lower part C_i^l can be also defined. Furthermore, we can distinguish two types of crossing; one is the crossing so that the upper part overlaps from the left side of the lower part to its right side and the other is the opposite crossing. Let us define the former as the right hand helix crossing C_i^+ and the latter as the left hand helix crossing C_i^-. Then, the crossing state of the object shown in Fig.1 can be described as E_l-C_1^{u-}-C_2^{l-}-C_3^{l+}-C_4^{u+}-C_5^{u-}-C_1^{l-}-C_2^{u-}-C_5^{l-}-C_4^{l+}-C_3^{u+}-E_r.

Thus, we can represent the state of linear objects, especially knotted ones as finite crossing states regardless of their length, thickness, or other physical properties.

III. Definition of State Changing Operations

Next, let us consider transitions among crossing states defined in the previous section. In order to change the crossing state of a linear object, some operations must be performed on the object. Therefore, a state transition corresponds to an operation that changes the number of crossing points or permutates their sequence. In this paper, four basic operations are prepared as shown in Fig.2. Operation type-I, type-II, and type-III are equivalent to Reidemeister move type-I, type-II and type-III in the knot theory[9], respectively. Type-IV operation is needed because a linear object has endpoints in general while knot theory does not focus on the endpoints of the object. By type-I, type-II, and type-IV operations, the number of crossing points is increased or decreased. Type-III operation does not change the number of crossing points but change their sequence. Furthermore, let us define operations to increase crossing points as crossing operations CO_{I}, CO_{II}, and CO_{IV}, operations to decrease them as uncrossing operations UO_{I}, UO_{II}, and UO_{IV}, and an operation keeping the number of them as an arranging operation AO_{III}.

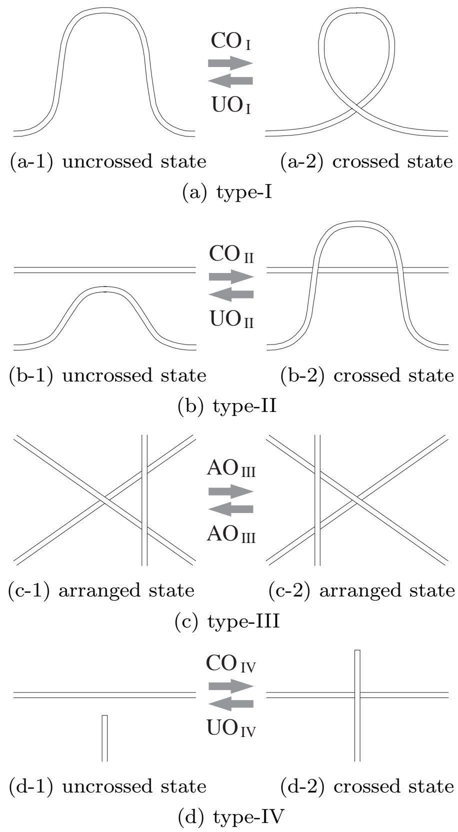


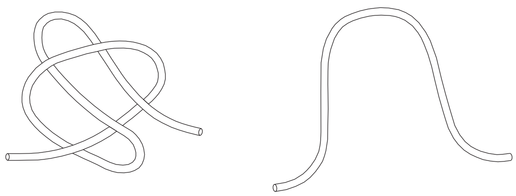
Fig. 2 Basic operations

The number of possible crossing states after a crossing operation can be very larger than that after a uncrossing operation. Therefore, in this paper, we define that a state transition is caused by a uncrossing operation alone. Then, a manipulation process can be represented as a sequence of uncrossing operations.

Fig.3 shows an example of a required manipulation. The initial state in Fig.3(a) can be represented as E_l-C_1^{u-}-C_2^{l-}-C_3^{l+}-C_4^{u+}-C_5^{u-}-C_1^{l-}-C_2^{u-}-C_5^{l-}-C_4^{l+}-C_3^{u+}-E_r and the objective state in Fig.3(b) can be represented as E_l-E_r. Assuming that only uncrossing operations can be used, that is, without AO_{III}, 14 crossing states and 32 state transitions are derived as shown in Fig.4. If AO_{III} is included, we can derive 21 crossing states and 69 state transitions.

Thus, we can generate possible manipulation processes of a linear object as a sequence of crossing state transitions when the initial state, the objective state, and several intermediate states are given.

For knotting manipulation, we search for possible sequences of uncrossing operations where the crossing state is changed from the objective one to the initial one at first. Next, by following found sequences backward, knotting



(a) initial state (b) objective state

Fig. 3 Example of required manipulation

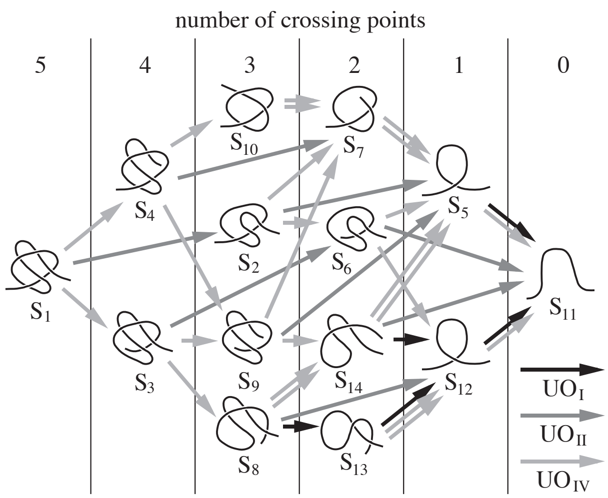


Fig. 4 Result of manipulation process planning

manipulation processes can be derived.

IV. Determination of Grasping Points and Their Moving Direction

In this section, we explain a procedure to determine grasping points and their moving direction in order to realize a derived sequence of state transitions. We assume that manipulators grasp not a crossing point but a segment between two neighboring crossing points for unknotting manipulation because two parts are overlapped at each crossing point. Let us define a segment whose adjacent crossing points are removed after uncrossing operations as a target segment.

Then, the following three type motions can realize state transitions; Type-A is translation/rotation of the whole upper/lower part. In this case, each part is regarded as a rigid body, and motions for uncrossing can be derived by considering qualitative geometry of a crossing region. Type-B type is the motion of a target segment by grasping it directly. We can also select feasible unit motions with respect to each crossing region. Type-C is the motion of a target segment by grasping its adjoining segments.

Furthermore, we define the approach direction of manipulators with respect to the projection plane; from the front side or the back side. Realizability of each operation depends on this direction. For example, in Fig.2(d-2), UO_{IV} can not be realized when the terminal segment is grasped from the back side. Then, 17 grasping patterns that can realize each basic operation are derived as shown in Fig.5, where a circle with dot, a circle with cross, and a open circle represent a point to be grasped from the front side, the back side, and whichever side, respectively. Fig.5(g)(h), Fig.5(k), and Fig.5(p)(q) indicates the opposite of a crossed state shown in Fig.2(b-2), Fig.2(c-2), and Fig.2(d-2), respectively.

Next, let us consider moving direction of a grasping point to realize each operation. We define four unit motions; translation parallel to the central axis of an object, translation perpendicular to the axis, rotation around the axis, and rotation around a line perpendicular to the axis. Then, by selecting feasible combinations of grasping points and unit motions, basic operations can be realized.

Thus, we can derive finite sequences of crossing state transitions of a linear object and feasible combinations of grasping points and unit motions with respect to each sequence, that is, rough manipulation plans.

V. Planning of One Handed Manipulation

In this section, a planning method for one-handed manipulation of a linear object is proposed. A crossing state graph illustrated in Fig.4 includes sequences consisting of type-IV operations alone. Any manipulation processes can be achieved by iteration of type-IV operations. Therefore, in this paper, we focus on type-IV operation as an unit of manipulation. Let us define a grasping point and the approach direction of a manipulator for type-IV operation as shown in Fig.6. Fig.6(a) shows them for CO_{IV} and Fig.6(b) shows them for UO_{IV}. For crossing operation, we define that a manipulator can grasp a point to be crossed on a linear object. Fig.6(a-2) and (b-2) indicates the opposite crossing of the case illustrated in Fig.6(a-1) and (b-1), respectively. Let us define the crossing shown in Fig.6(a-1) and (b-1) as the up-end crossing and that shown in Fig.6(a-2) and (b-2) as the down-end crossing. In both crossing, the upper part is selected as the grasping point. Furthermore, the manipulator can approach to the object from the front side of the projection plane in both cases. It means that type-IV operation can be realized by one side approach of one manipulator. Therefore, we can manipulate a linear object by one hand without turning over the whole/partial part of it when it is lied on a table.

As a uncrossing operation is an operation to decrease crossing points, the position of a crossing point to be uncrossed is known. However, for a crossing operation, the position of two points to be crossed, which exist somewhere on segments, must be determined. In the objective state, a knotted object with n crossing points has $2n+1$ segments and $2n$ dividing points, that is, upper and lower points of

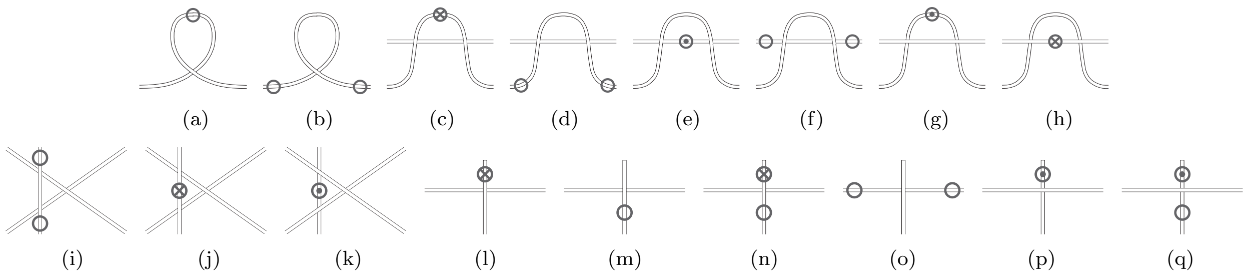


Fig. 5 Grasping patterns

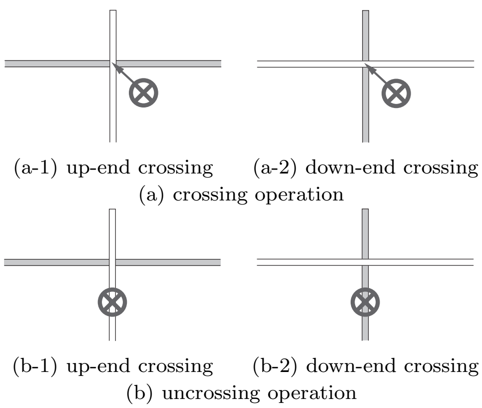


Fig. 6 Grasping point for type-IV operation

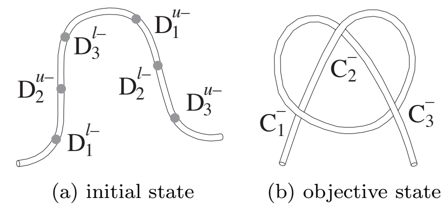


Fig. 7 Example of knotting manipulation

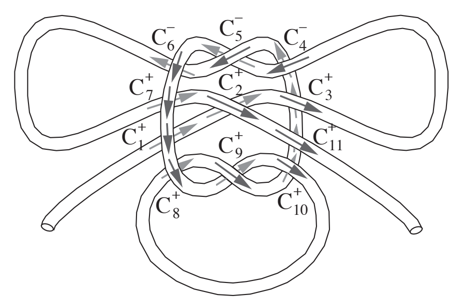


Fig. 8 Bowknot

crosses. Therefore, we divide the object into $2n+1$ segments in the initial state. Let D_i^* be a dividing point where superscript $*$ and subscript i are equivalent to those of a crossing point created by crossing it. Note that the actual length of each segment by which the object shape becomes stable depends on physical properties of the object.

Fig.7 shows an example of knotting manipulation. The initial and the objective state can be described as E_l-D_1^{l-}-D_2^{u-}-D_3^{l-}-D_1^{u-}-D_2^{l-}-D_3^{u-}-E_r and E_l-C_1^{l-}-C_2^{u-}-C_3^{l-}-C_1^{u-}-C_2^{l-}-C_3^{u-}-E_r, respectively. Let $CO_{IV}(i)$ be a type-IV crossing operation to create i-th crossing point. Then, for example, the initial state is changed into the state E_l-C_1^{l-}-D_2^{u-}-D_3^{l-}-C_1^{u-}-D_2^{l-}-D_3^{u-}-E_r by $CO_{IV}(1)$. In order to realize this crossing operation, a manipulator grasps point D_1^{u-} and crosses it on point D_1^{l-} so that these two points create a left hand helix crossing. If the position of these two points and the tangent at them are given, we can generate possible trajectories of the manipulator to create crossing point C_1^-. Thus, we can determine a grasping point and the motion of a manipulator for both a CO_{IV} and a UO_{IV}.

Fig.8 shows a bowknot. It has 11 crossing points and its crossing state is described as E_l- C_1^{l+}- C_2^{l+}- C_3^{u+}- C_4^{u-}- C_5^{l-}- C_6^{u-}- C_7^{u+}- C_1^{u+}- C_8^{u+}- C_9^{l+}- C_{10}^{u+}- C_8^{l+}- C_9^{u+}- C_{10}^{l+}- C_{11}^{l+}- C_3^{l+}- C_4^{l-}- C_5^{u-}- C_6^{l-}- C_7^{l+}- C_2^{u+}- C_{11}^{u+}- E_r. Then, a crossing state graph for tying this bowknot can be generated as shown in Fig.9. It includes 44 crossing states and 75 type-IV crossing operations. In general, we tie a bowknot in the following order; $S_0 \rightarrow S_2 \rightarrow S_7 \rightarrow S_{11} \rightarrow S_{13} \rightarrow S_{20} \rightarrow S_{26} \rightarrow S_{31} \rightarrow S_{38}$(by CO_{II}) $\rightarrow S_{41} \rightarrow S_{43}$. However, it is not unique. Fig.9 indicates that another procedures to tie the bowknot exist. Furthermore, they can be executed by one manipulator approaching from the front side of the projection plane. It implies that a hardware with simple mechanism can make complex knots.

Thus, we can plan one-handed manipulation of a linear object regardless of its physical properties.

VI. Case Study

We demonstrate the effectiveness of our proposed method in this paper. Fig.10 shows an overview of our developed system consisting of a PC, a 6 DOF manipulator, and a CCD camera. We try planning and executing one-handed

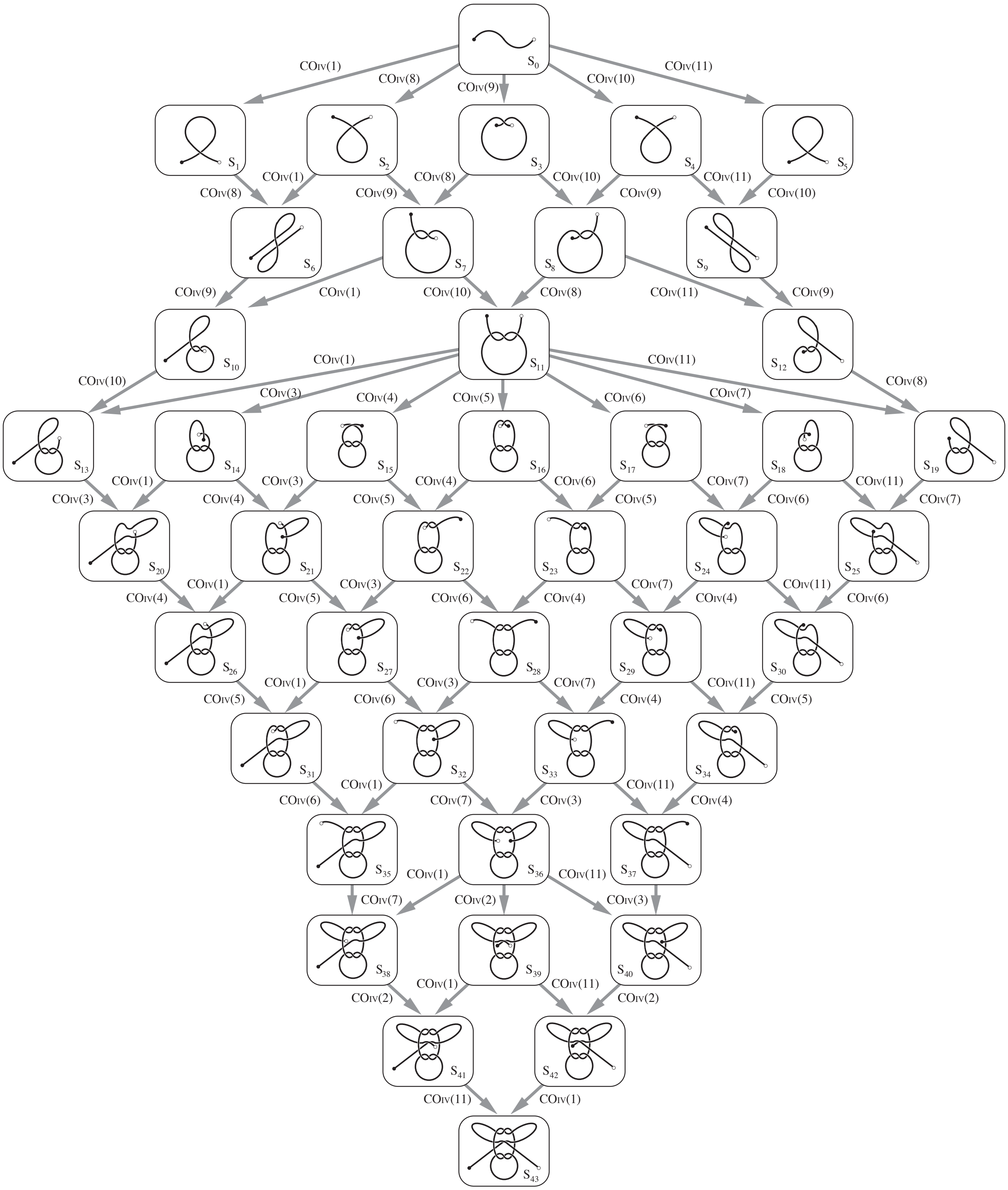


Fig. 9 Crossing state graph for tying bowknot

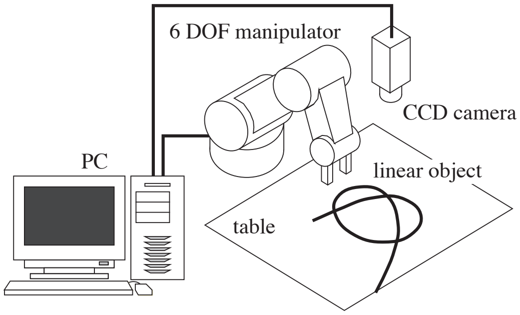


Fig. 10 Overview of developed system

unknotting manipulation with this system. A linear object, made from rubber but whose physical properties are uncertain, is laid on a table and its shape is captured by the camera fixed above the table. The table corresponds to the projection plane.

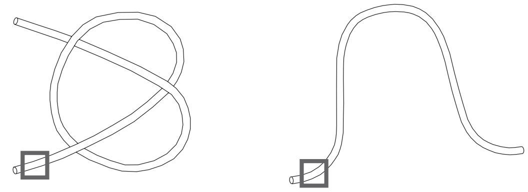
Fig.11 shows a required manipulation. It corresponds to raveling out of an overhand knot. The initial state shown in Fig.11(a) can be represented as E_l-C_1^{u-}-C_2^{l-}-C_3^{u-}-C_1^{l-}-C_2^{u-}-C_3^{l-}-E_r and the objective state shown in Fig.11(b) can be represented as E_l-E_r. Assumptions of this case study are as follows :

- The left endpoint of the object is fixed during manipulation; an open square in Fig.11 indicates the position of a fixure.
- The manipulator can not move the fixed endpoint and its adjacent segment of the object.
- The manipulator releases the object whenever one uncrossing operation is finished.

Then, one manipulation plan as shown in Fig.12 is generated. It consists of three UO_{IV} and corresponds to a sequence; $S_{10} \rightarrow S_7 \rightarrow S_5 \rightarrow S_{11}$ in Fig.4.

Next, the system recognizes the current crossing state of the object from a gray-scale image. The position of individual crossing points can be identified by analyzing the image. In this experiment, information about which part is top at each crossing point is given for simplicity. However, it can be obtained automatically using a stereo camera. We regard the position of each grasping point as be the midpoint of each segment. Direction of the axes for four unit motions can be calculated from the tangent at a grasping point. As adequate moving distance for a state transition is unknown, the system checks whether its crossing state is changed or not after moving the object. Thus, the manipulator can grasp, move, and release the object according to the generated qualitative plan. Fig.13 shows the result of this manipulation.

Thus, we conclude that our proposed method is useful for automatic planning and execution of linear object manipulation.



(a) initial state (b) objective state

Fig. 11 Required manipulation – raveling out of overhand knot

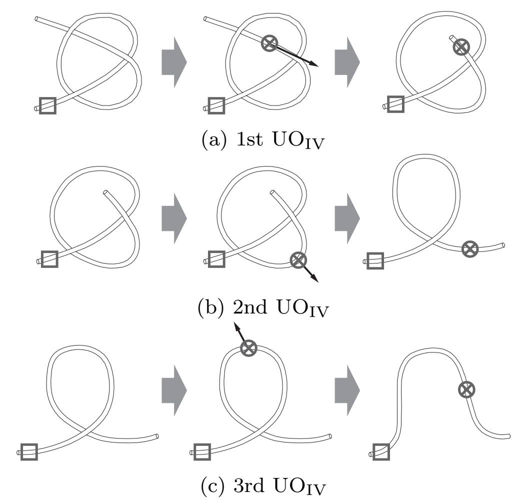


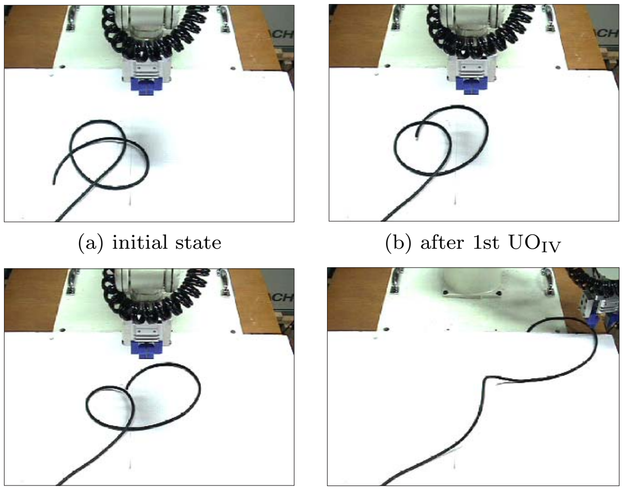
Fig. 12 Generated manipulation plan

VII. Toward Detailed Planning

We can plan linear object manipulation qualitatively by applying our method proposed in the previous sections. It is not enough to determine grasping points of manipulators and their trajectories in detail. Quantitative analysis should be performed in order to check whether generated plans can be realized practically or not considering physical properties of a linear object such as rigidity. We had developed an analytical method to model the stable shape of a deformable linear object[10]. Fig.14 shows the computed shape of an overhand knot. From this result, we can estimate the position of dividing points D_i^* for crossing operations. Therefore, the manipulation strategy can be derived automatically by combining a qualitative planning proposed in this paper with the quantitative analysis.

VIII. Conclusions

In this paper, a rough planning method for linear object manipulation including knotting/unknotting was proposed. Especially, it was shown that any knotting/unknotting manipulation can be realized by one hand and that our proposed method can be applied to such one-handed manipulation.



(a) initial state (b) after 1st UO_{IV}

(c) after 2nd UO_{IV} (d) after 3rd UO_{IV}

Fig. 13 Result of manipulation

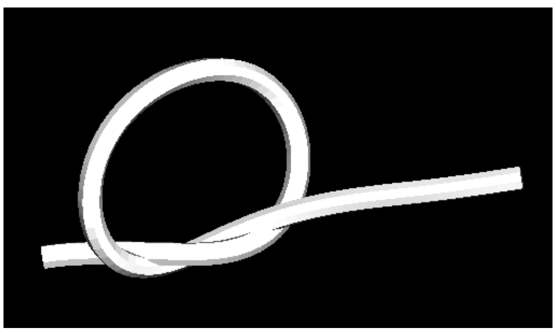


Fig. 14 Computed shape of overhand knot

Firstly, a representation of topological states of a linear object was proposed. Its topological states can be represented as finite crossing states including the number of crossing points and how to cross at each crossing point. Secondly, transitions among those states were defined by introducing four basic operations. A state transition corresponds to a basic operation that changes the number of crossing points or permutates their sequence. Then, possible sequences of crossing state transitions, that is, possible manipulation processes can be generated once the initial state and the objective state are given. Thirdly, a method for determination of grasping points and their moving direction was proposed in order to realize derived manipulation processes. Furthermore, a planning method for one-handed manipulation is proposed because it was found that any manipulation processes can be realized by one hand. Finally, it was demonstrated that our proposed method can be applied to planning and execution of linear object manipulation by one manipulator.

References

[1] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, *Automatic Synthesis of Fine Motion Strategies for Robots*, Int. J. of Robotics Research, Vol.3, No.1, pp.3–24, 1984.

[2] R. S. Desai and R. A. Volz, *Identification and Verification of Termination Conditions in Fine Motion in Presence of Sensor Errors and Geometric Uncertainties*, Proc. of IEEE Int. Conf. Robotics and Automation, pp.800–807, 1989.

[3] H. Wakamatsu, A. Tsumaya, K. Shirase, and E. Arai, *Representation of Deformable Thin Object Manipulation Based on Contact State*, Proc. of the 5th Japan-France Congress & 3rd Asia-Europe Congress on Mechatronics, pp.519–524, 2001.

[4] J. E. Hopcroft, J. K. Kearney, and D. B. Krafft, *A Case Study of Flexible Object Manipulation*, Int. J. of Robotics Research, Vol.10, No.1, pp.41–50, 1991.

[5] B. G. A. V. Leaf, *Models of the Plain-Knitted Loop*, Journal of the Textile Institute Transaction, pp.49–58, 1960.

[6] J. Phillips, A. Ladd, and L. E. Kavraki, *Simulated Knot Tying*, Proc. of IEEE Int. Conf. Robotics and Automation, pp.841–846, 2002.

[7] T. Matsuno, T. Fukuda, and F. Arai, *Flexible Rope Manipulation by Dual Manipulator System Using Vision Sensor*, Proc. of International Conference on Advanced Intelligent Mechatronics, pp.677–682, 2001.

[8] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, *Knot Planning from Observation*, Proc. of IEEE Int. Conf. Robotics and Automation, pp.3887–3892, 2003.

[9] C. C. Adams, *The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots*, W. H. FREEMAN AND COMPANY, 1994.

[10] H. Wakamatsu, S. Hirai, and K. Iwata, *Modeling of Linear Objects Considering Bend, Twist, and Extensional Deformation*, Proc. of IEEE Int. Conf. Robotics and Automation, pp.433–438, 1995.