Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Unraveling of Deformable Linear Objects Based on 2D Information about Their Crossing States

Hidefumi Wakamatsu, Akira Tsumaya, and Eiji Arai
Dept. of Materials & 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 unraveling deformable linear objects based on 2D information about their crossing states is proposed. In manipulation of a linear object, its raveling must be avoided. It takes much time to unravel it once it is raveled. Therefore, it is important to generate unraveling plans efficiently. First, an unraveling process of a linear object is represented a sequence of crossing state transitions. It can be generated on a computer if 3D information about the current crossing state is given. Second, the crossing sequence of a linear object, which corresponds to its 2D information, is categorized into two types: unravelable and not-unravelable. Third, a procedure to generate efficient unraveling processes based on unravelability of the crossing sequence is explained. Finally, examples of unraveling process generation with our developed system are demonstrated.**

***Index Terms*— linear objects, manipulation, planning, unraveling, crossing sequences**

I. Introduction

Deformable linear objects such as tubes, cords, cables, wires, and threads are used widely for fixing, fastening, wrapping, packing, suturing, and ligating of objects including themselves. In such manipulative tasks, knotting of linear objects is required. At the same time, their raveling must be avoided. If unexpected ravel occurs, it takes much time to unravel. For example, raveling of earphone/headphone cord of a portable audio player as shown in Fig.1 would puzzle you sometimes. So, efficient unraveling is important as well as avoidance of such raveling.

Knotting manipulation by robots has been studied. Inoue et al. reported tying a knot in a rope with a manipulator utilizing visual feedback[1]. Hopcroft et al. devised an abstract language to express various knotting manipulations and performed knot-tying tasks with a manipulator[2]. Matsuno et al. realized a task consisting of tying a cylinder with a rope using a dual manipulator system[3]. Morita et al. have been developing a system for knot planning from observation of human demonstrations[4]. Unknotting manipulation, *i.e.*, the inverse of knotting manipulation, has been also studied. We have realized automatic planning and execution of knotting/unknotting manipulation[5]. Ladd et al. developed an untangling planner for mathematical knots represented as closed piecewise linear curves[6].

Fig. 1. Raveled earphone cord and strap

Unraveling is equivalent to unknotting. However, the state of a raveled object can become more complex than that of a knotted object as shown in Fig.1. Moreover, it is difficult to recognize the state of a reveled object completely because it may twine itself. Therefore, recognition of the object state and manipulation planning are both important for unraveling. In this paper, we propose a planning method for unraveling a linear object when 3D information about the object state is unknown. First, an unknotting process of a linear object, which is equivalent to its unraveling process, is represented as a sequence of crossing state transitions. The object state is categorized according to three properties with respect to self-crossings of the object. State transitions are defined by introducing four basic operations. Then, possible unknotting processes can be generated if the current crossing state is completely identified. Second, the crossing sequence of a linear object, which corresponds to its 2D information, is considered. The crossing sequence can be categorized into two types: unravelable and not-unravelable. Third, a procedure to generate efficient unraveling processes based on unravelability of the crossing sequence is explained. An object with an unravelable crossing sequence can be unraveled by pulling its both endpoints. Finally, examples of unraveling process generation with our developed system are demonstrated.

II. Unknotting Process Generation

In this section, we briefly explain a method to generate possible processes for unknotting of a linear object, which is

equivalent to its unraveling. First, the state of a linear object can be topologically represented using three properties after projecting its shape on a projection plane. The first property is the *crossing sequence*. It is determined by numbering a crossing met first with tracing along the projected curve from one endpoint to the other. The i-th crossing is represented as symbol C_i. One endpoint where tracing starts is referred to as the left endpoint E_l and that where tracing ends as the right endpoint E_r. The second property is the *location* of a pair of points at each crossing, that is, which point is upper/lower. The upper point of i-th crossing is described as symbol C_i^u and the lower point of that as symbol C_i^l. The third property is the *helix* of each crossing. Let us define a crossing where the upper part overlaps first on the right side of the lower part and then overlaps on its left side as a *left-handed helical crossing*. Conversely, in a *right-handed helical crossing*, the upper part first overlaps on the left side of the lower part and then overlaps on its right side. The symbols C_i^- and C_i^+ represent left- and right-handed helical the i-th crossing, respectively. Moreover, let us describe a segment between two crossings C_i and C_j as ${}_i^p\mathrm{L}_j^q$, where p and q indicate whether the segment at each crossing is the upper ($p, q = u$) or lower part ($p, q = l$). Terminal segments adjacent to the left and right endpoints are described as L_i^p and ${}_j^q\mathrm{L}$, respectively.

Next, we introduce basic operations described in Fig.2, corresponding to state transitions. Crossing operations $\mathrm{CO_I}$, $\mathrm{CO_{II}}$, and $\mathrm{CO_{IV}}$ increases the number of crossings, while uncrossing operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_{IV}}$ decrease the number. Arranging operation $\mathrm{AO_{III}}$ does not change the number of crossings but permutes their sequence.

Each basic operation can be applied to specific subsequences of crossings. Let us investigate subsequences to which each operation is applicable. Operation $\mathrm{UO_I}$ is applicable to a subsequence represented as follows:

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots. \tag{1}$$

That is, two crossing points corresponding to one crossing C_i, should be adjacent to each other in applying $\mathrm{UO_I}$. Operation $\mathrm{UO_{II}}$ is applicable to subsequences described as follows:

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_j^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\mathrm{C}_j^{l/u}\text{-}\cdots, \tag{2}$$

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_j^{u/l}\text{-}\cdots\text{-}\mathrm{C}_j^{l/u}\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots. \tag{3}$$

That is, two upper crossing points C_i^u and C_j^u, should be adjacent to each other and the corresponding lower crossing points C_i^l and C_j^l, should also be adjacent to each other. Operation $\mathrm{UO_{IV}}$ is applicable to subsequences represented as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_i^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots, \tag{4}$$

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\mathrm{E}_r. \tag{5}$$

That is, a crossing adjacent to an endpoint can be deleted by operation $\mathrm{UO_{IV}}$. Operation $\mathrm{AO_{III}}$ is applicable to a subsequence represented as permutation of the following three subsequences: α, β, and γ, e.g., $\cdots$ -β-γ-α- $\cdots$:

$$\alpha:\ \cdots\text{-}\mathrm{C}_{i/j}^u\text{-}\mathrm{C}_{j/i}^u\text{-}\cdots, \tag{6}$$

$$\beta:\ \cdots\text{-}\mathrm{C}_{j/k}^{l/u}\text{-}\mathrm{C}_{k/j}^{u/l}\text{-}\cdots, \tag{7}$$

$$\gamma:\ \cdots\text{-}\mathrm{C}_{i/k}^{l}\text{-}\mathrm{C}_{k/i}^{l}\text{-}\cdots. \tag{8}$$

That is, three crossings consisting of three segments one of which overlaps the others can be permuted by operation $\mathrm{AO_{III}}$. Uncrossing operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_{IV}}$ and arranging operation $\mathrm{AO_{III}}$ are applicable to their specific crossing subsequences indicated above. Once the initial and the objective crossing states of a linear object are given, we can generate possible sequences of crossing state transitions, that, is, possible processes of unknotting manipulation by repeating detection of applicable subsequences of individual operations and deletion/permutation of relevant crossings.

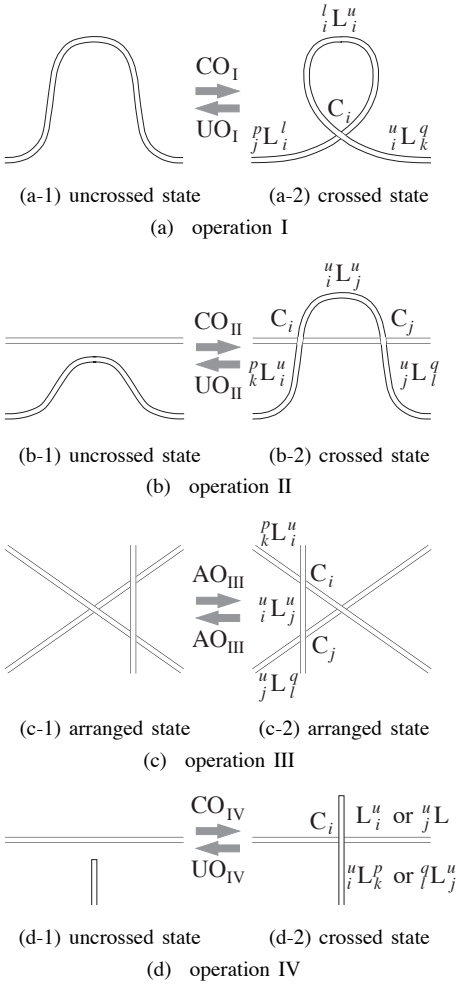


Fig. 2. Basic operations

III. Introduction of Unravelable Crossing Sequence

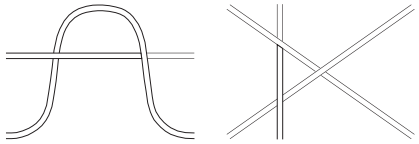
Once the current crossing state of a linear object is identified, we can unravel the object using the method proposed in the previous section. To identify the crossing state completely, the location at each crossing should be known. Morita recognized the state of a linear object with a 9-eye stereo camera[4] and Matsuno identified it utilizing variance of luminance at crossings[7]. Now, let us assume that only 2D information about the object state is available. It means that the location and the helix of any crossing can not be identified. Then, we can perform operation $\mathrm{UO_I}$ even if the location at crossing C_i shown in Fig.2-(a-2) is unknown. Operation $\mathrm{UO_{IV}}$ can also be realized regardless of the location at crossing C_i shown in Fig.2-(d-2). Contrary, whether operations $\mathrm{UO_{II}}$ and $\mathrm{AO_{III}}$ can be applied depends on the location of crossings. Fig.3 shows examples of crossings with a subsequence to which operations $\mathrm{UO_{II}}$ and $\mathrm{AO_{III}}$ are applicable but with locations to which they can not be applied. Any knot can be unknotted by applying operations $\mathrm{UO_{IV}}$ alone[8]. This implies that a raveled linear object can be unraveled by applying operations $\mathrm{UO_{IV}}$ alone regardless of the location at each crossing. Recall that we often search for an endpoint and manipulate it to unravel a self-entwined rope. However, such manipulation may be not efficient when the object is raveled intricately, *i.e.*, it has many crossings. In this section, we propose a method for generating efficient unraveling processes of a linear object based on its crossing sequence, *i.e.*, its 2D information.

First, we define a knot in which some crossings remain even if all possible operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{AO_{III}}$ are applied as a *tightenable knot*. For example, an overhand knot and a figure-of-eight knot are tightenable knots. Contrary, a knot which can be unknotted completely by applying operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and/or $\mathrm{AO_{III}}$ is defined as an *untightenable knot*. The untightenable knot is unknotted when its both endpoints are pulled away from each other[9]. We can check whether a knot is tightenable or untightenable from its crossing state description[5].

Fig.4-(a-1) illustrates the monochrome image of a knot with 3 crossings. Its crossing sequence is described as follows:

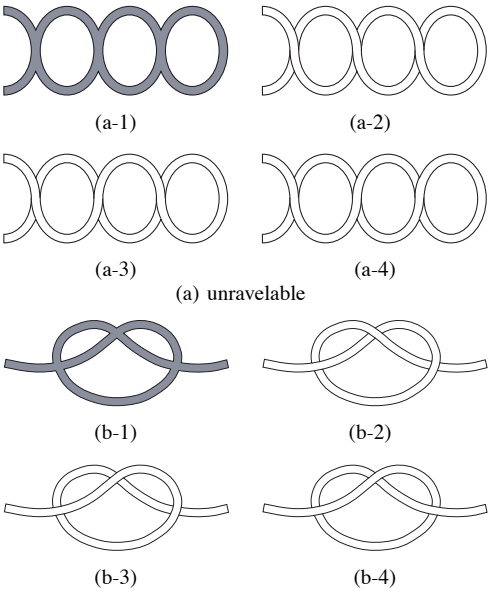
$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_3\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{E}_r. \tag{9}$$

Knots shown in Fig.4-(a-2) through (a-4) have the same crossing sequence. They include the subsequence $\cdots\text{-}\mathrm{C}_3\text{-}\mathrm{C}_3\text{-}\cdots$ to which operation $\mathrm{UO_I}$ can be applied. When crossing C_3 is deleted, it is found that crossing C_2 can also be deleted by application of operation $\mathrm{UO_I}$. After deletion of crossing C_2, we can delete crossing C_1 by applying operation $\mathrm{UO_I}$ once more. This means that knots shown in Fig.4-(a-2) through (a-4) are untightenable. Any knot with the crossing sequence described by eq.(9) can be unraveled by pulling its both endpoints regardless of the location at each crossing. In this paper, we define such crossing sequence as *unravelable crossing sequence*. An untighenable knot has an unravelable crossing sequence.



(a) with 2 crossings (b) with 3 crossings

Fig. 3. Crossings not applicable uncrossing operations



(a-1) (a-2) (a-3) (a-4)

(a) unravelable

(b-1) (b-2) (b-3) (b-4)

(b) not-unravelable

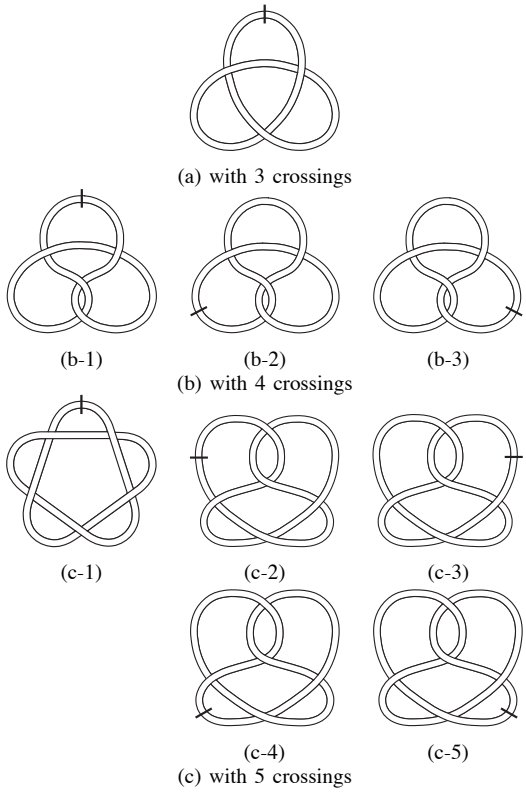
Fig. 4. Crossing sequences

A knot shown in Fig.4-(b-1) also has 3 crossings, sequence of which is as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{E}_r. \tag{10}$$

It is equivalent to that of knots shown in Fig.4-(b-2) through (b-4). Knots in Fig.4-(b-3) and (b-4) are both untightenable, but the knot in Fig.4-(b-2) corresponds to an overhand knot, that is, it is tightenable. This implies that a tightenable knot with the crossing sequence described by eq.(10) exists. Consequently, such crossing sequence is not unravelable. Note that knots in Fig.4-(b-3) and (b-4) can be unraveled, but they can not be distinguished from the knot in Fig.4-(b-2) when 3D information, *i.e.*, the location at each crossing is not given. Thus, we can categorize the crossing sequence of a knot into two types: unravelable and not-unravelable. The former can be unraveled by pulling its both endpoints regardless of the location at each crossing, while the latter may be tightened according to the location when its both endpoints are pulled.

Fig.5 shows looped prime knots in knot theory. We can



(a) with 3 crossings

(b-1) (b-2) (b-3)

(b) with 4 crossings

(c-1) (c-2) (c-3)

(c-4) (c-5)

(c) with 5 crossings

Fig. 5. Looped prime knots

not reduce the number of crossings of these knots even if any operation corresponding to Reidemister move[10] is applied. They are closely related to tightenable knots. Let us discuss the relationship between looped prime knots and unravelable crossing sequences. If the looped prime knot with 3 crossings is cut as shown in Fig.5-(a), its crossing state is described as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1^{l+}\text{-}\mathrm{C}_2^{u+}\text{-}\mathrm{C}_3^{l+}\text{-}\mathrm{C}_1^{u+}\text{-}\mathrm{C}_2^{l+}\text{-}\mathrm{C}_3^{u+}\text{-}\mathrm{E}_r. \tag{11}$$

If the crossing state of an unlooped linear object is described by eq.(11), it is equivalent to an overhand knot. If the object has 3 crossings but their sequence differs from eq.(11), it can be unknotted by applying operation $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and/or $\mathrm{AO_{III}}$. Consequently, a linear object with 3 crossings can be unraveled by pulling both endpoints if and only if it does not have a not-unravelable crossing sequence: E_l-C_1-C_2-C_3-C_1-C_2-C_3-E_r.

Fig.5-(b) shows the looped prime knot with 4 crossings. Cutting the knot as shown in Fig.5-(b-1) and tracing it counterclockwise from one endpoint, the crossing sequence is described as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_3\text{-}\mathrm{C}_2\text{-}\mathrm{C}_4\text{-}\mathrm{E}_r. \tag{12}$$

In the case of Fig.5-(b-2) and Fig.5-(b-3), the crossing sequence is described as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_3\text{-}\mathrm{E}_r. \tag{13}$$

A figure-of-eight knot has this crossing sequence. A knot with the crossing sequence described by eq.(12) or (13) may be tightened. This implies that a linear object with 4 crossings is unraveled if it does not have the above two sequences.

There are two types of the looped prime knot with 5 crossings as shown in Fig.5-(c). One type illustrated in Fig.5-(c-1) has the crossing sequence as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{E}_r. \tag{14}$$

This sequence corresponds to that of a double overhand knot. The other type illustrated in Fig.5-(c-2) through Fig.5-(c-5) has the following crossing sequences:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_3\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{E}_r, \tag{15}$$

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_5\text{-}\mathrm{C}_4\text{-}\mathrm{C}_3\text{-}\mathrm{E}_r, \tag{16}$$

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_5\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{E}_r, \tag{17}$$

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_5\text{-}\mathrm{C}_4\text{-}\mathrm{E}_r. \tag{18}$$

Then, a linear object with 5 crossings but without the crossing sequence described by eqs.(14) through (18) is unravelable. Thus, we can derive not-unravelable crossing sequences from looped prime knots in knot theory. If the crossing sequence of a linear object with n crossings does not include not-unravelable sequences with 3 through n crossings, it can be unraveled by pulling its both endpoint instead of applying n $\mathrm{UO_{IV}}$ operations.

IV. Procedure to Generate Efficient Unraveling Processes

In this section, we explain a procedure to generate unraveling processes. Let us assume that the monochrome image of a linear object shown in Fig.6-(a-1) is given. Its crossing sequence is described as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_3\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{E}_r. \tag{19}$$

The above sequence corresponds to a not-unravelable sequence with 5 crossings. It means that the object may be raveled and tightened if its both endpoints are pulled. Then, let us consider application of operation $\mathrm{UO_{IV}}$ so that the object does not include any not-unravelable sequence. If we apply operation $\mathrm{UO_{IV}}$ to the left terminal segment, the object state changes into the state shown in Fig.6-(a-2). Its crossing sequence is described as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{E}_r. \tag{20}$$

A set of closed regions surrounded by a linear object is defined as the *inner region*, and the other region in the projection plane as the *outer region*. Moreover, segments touch the outer region are referred to as *outer segments*, and segments do not touch as *inner segments*[5]. In Fig.6-(a-2), the left terminal segment is an inner segment. When one of terminal segments is inner, we can not pull both endpoints sufficiently without changing the crossing sequence. So, we

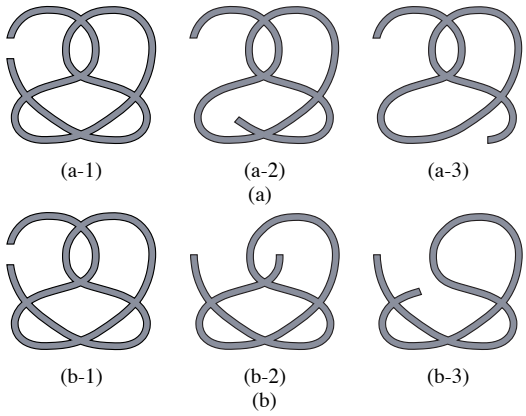


Fig. 6. Unraveling processes

apply another operation $\mathrm{UO_{IV}}$ to the left terminal segment. Then, the following sequence is derived:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{–}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{E}_r. \tag{21}$$

The above sequence is equivalent to the not-unravelable sequence with 3 crossings. This implies that additional $\mathrm{UO_{IV}}$ operations are required to unravel the object. Contrary, if we apply 2 consecutive $\mathrm{UO_{IV}}$ operations to the right terminal segment as shown in Fig.6-(b), the crossing sequence becomes as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_3\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{E}_r. \tag{22}$$

As this sequence differs from the not-unravelable sequence, the knot shown in Fig.6-(b-3), which is equivalent to that in Fig.4-(a-1), can be unraveled by pulling its both endpoints. Consequently, we can conclude that unraveling process shown in Fig.6-(b) is more efficient than that shown in Fig.6-(a). Thus, we can generate efficient unraveling processes of a linear object based on only its crossing sequence, *i.e.*, its 2D information. This indicates that we may unravel a linear object without a stereo camera.

Not-unravelable sequences can be extracted from the list of looped prime knots in knot theory[11]. Let us define the following subsequence as a *not-unravelable subsequence* with 3 crossings:

$$\begin{aligned}&\cdots\text{-}\mathrm{C}_i\text{-}\cdots\text{-}\mathrm{C}_j\text{-}\cdots\text{-}\mathrm{C}_k\text{-}\\ &\quad\cdots\text{-}\mathrm{C}_i\text{-}\cdots\text{-}\mathrm{C}_j\text{-}\cdots\text{-}\mathrm{C}_k\text{-}\cdots \quad (i<j<k).\end{aligned} \tag{23}$$

If the crossing state includes the above subsequence, the object has a part which may be tightened. The not-unravelable sequence described by eq.(11) is a kind of this subsequence. We can also define not-unravelable subsequences with n crossings referring to not-unravelable sequences. When such not-unravelable subsequence is detected from the crossing sequence, we delete a crossing included in the subsequence and nearest to one endpoint by applying operations $\mathrm{UO_{IV}}$ repeatedly. Let C_{li} and C_{rj} be i-th and j-th crossing met first when we trace an object from the left and the right endpoint, respectively. When the object has n crossings, we assume that only operation $\mathrm{UO_{IV}}$ is applied to delete $k(k=1,\cdots,n)$ crossings. Then, we check the number of remaining not-unravelable subsequences after deleting crossings $\mathrm{C}_{li}(i=k,k-1,\cdots,1,0)$ and $\mathrm{C}_{rj}(j=k-i)$. If the crossing sequence does not include any not-unravelable subsequence by $n-3$ crossings are deleted, the rest can be uncrossed by applying one pulling operation instead of some $\mathrm{UO_{IV}}$ operations. This implies that the object can be unraveled efficiently. For example, the crossing sequence described by eq.(19) is equivalent to the not-unravelable sequence with 5 crossings and includes three not-unravelable subsequences with 3 crossings:

$$\cdots\text{-}\mathrm{C}_1\text{-}\cdots\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\cdots\text{-}\mathrm{C}_1\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\cdots, \tag{24}$$

$$\cdots\text{-}\mathrm{C}_2\text{-}\cdots\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\cdots\text{-}\mathrm{C}_2\text{-}\cdots\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\cdots, \tag{25}$$

$$\cdots\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_3\text{-}\cdots\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\cdots. \tag{26}$$

In this case, crossings C_{l1}=C_1 and C_{r1}=C_5 can be deleted by operation $\mathrm{UO_{IV}}$. If crossing C_5 is uncrossed, all these subsequences are deleted. Then, the object becomes unravelable. Contrary, subsequences described by eqs.(25) and (26) remain even if crossing C_1 is deleted. Consequently, we select application of operation $\mathrm{UO_{IV}}$ to crossing C_5 as the first process for unraveling. After that, the object is completely unraveled by pulling its both endpoints. Thus, efficient unraveling processes of a linear object can be derived even if only its crossing sequence is identified.

V. Case Study

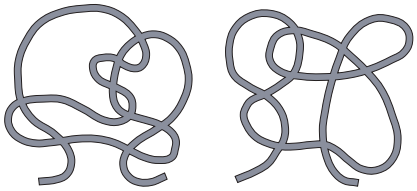
In this section, we discuss the effectiveness of our proposed method for efficient unraveling with some examples. Fig.7 shows two examples of a raveled object. They corresponds to not-unravelable sequences with 8 crossings. The crossing sequence in case-1 shown in Fig.7-(a) is described as follows:

$$\begin{aligned}&\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_5\text{-}\mathrm{C}_6\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\\ &\text{-}\mathrm{C}_7\text{-}\mathrm{C}_8\text{-}\mathrm{C}_6\text{-}\mathrm{C}_5\text{-}\mathrm{C}_4\text{-}\mathrm{C}_3\text{-}\mathrm{C}_8\text{-}\mathrm{C}_7\text{-}\mathrm{E}_r.\end{aligned} \tag{27}$$

In case-2, the crossing sequence shown in Fig.7-(b) is represented as follows:

$$\begin{aligned}&\mathrm{E}_l\text{-}\mathrm{C}_1\text{-}\mathrm{C}_2\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_2\text{-}\mathrm{C}_1\text{-}\mathrm{C}_5\text{-}\mathrm{C}_6\text{-}\\ &\text{-}\mathrm{C}_7\text{-}\mathrm{C}_3\text{-}\mathrm{C}_4\text{-}\mathrm{C}_8\text{-}\mathrm{C}_6\text{-}\mathrm{C}_7\text{-}\mathrm{C}_8\text{-}\mathrm{C}_5\text{-}\mathrm{E}_r.\end{aligned} \tag{28}$$

We developed a system to detect not-unravelable subsequences from a given crossing sequence. Using this system, it was found that the crossing sequence in case-1 include 49 not-unravelable subsequences and that in case-2 includes 19. TABLE I and TABLE II shows the number of remaining not-unravelable subsequences after deleting crossings C_{li} and C_{rj} in case-1 and case-2, respectively. As shown in TABLE I, when 3 crossings C_1, C_2, and C_7 or C_1, C_7, and C_8 are deleted, the crossing sequence in case-1 does



(a) (b)

Fig. 7. Examples of raveled objects

TABLE I

UNRAVELING PROCESS FOR CASE-1

deleted crossings	remaining not-unravelable subsequences
none	49
C_1	21
C_7	21
C_1, C_2	7
C_1, C_7	7
C_7, C_8	7
C_1, C_2, C_3	3
C_1, C_2, C_7	0
C_1, C_7, C_8	0
C_3, C_7, C_8	3

TABLE II

UNRAVELING PROCESS FOR CASE-2

deleted crossings	remaining not-unravelable subsequences
none	19
C_1	11
C_5	11
C_1, C_2	7
C_1, C_5	7
C_5, C_8	8
C_1, C_2, C_3	2
C_1, C_2, C_5	5
C_1, C_5, C_8	5
C_5, C_7, C_8	4
C_1, C_2, C_3, C_4	1
C_1, C_2, C_3, C_5	2
C_1, C_2, C_5, C_8	4
C_1, C_5, C_7, C_8	2
C_5, C_6, C_7, C_8	2
C_1, C_2, C_3, C_4, C_5	1
C_1, C_2, C_3, C_5, C_8	1
C_1, C_2, C_5, C_7, C_8	1
C_1, C_5, C_6, C_7, C_8	1
C_4, C_5, C_6, C_7, C_8	0

not include any not-unravelable subsequences. So, we can delete the rest crossings, *i.e.*, we can unravel the object at once pulling away its both endpoints. This indicates that we can perform unraveling with less operations than unraveling in which all 8 crossings are deleted by operation UO_{IV}. In case-2, 5 crossings C_4, C_5, C_6, C_7, and C_8 must be deleted to exclude not-unravelable subsequences from the crossing sequence as shown in TABLE II. After that, we can unravel the object with one pulling operation. This is a more efficient unraveling process than that consisting of 8 UO_{IV} operations. But, we have to delete more crossings in case-2 to change the crossing sequence into the unravelable one than in case-1. Thus, it is found that the effectiveness of our proposed method depends on the crossing sequences. Efficient unraveling processes are derived from some crossing sequences but they are not from others. However, we can determine whether 3D information is needed for its efficient unraveling from its crossing sequence, *i.e.*, 2D information.

VI. CONCLUSIONS

A planning method for unraveling deformable linear objects based on 2D information about their crossing states was proposed. First, an unknotting process of a linear object, which is equivalent to its unraveling process, is represented as a sequence of crossing state transitions. It can be generated on a computer if 3D information about the current crossing state is given. Second, the crossing sequence of a linear object, which corresponds to its 2D information, was categorized into two types: unravelable and not-unravelable. Third, a procedure to generate efficient unraveling processes based on unravelability of the crossing sequence was explained. An object with an unravelable crossing sequence can be unraveled by pulling its both endpoints. Finally, examples of unraveling process generation with our developed system were demonstrated. The crossing sequence is not sufficient information for deriving efficient unraveling processes, but it is useful for that.

REFERENCES

[1] H. Inoue and M. Inaba, "Hand-eye Coordination in Rope Handling", Robotics Research: The First International Symposium, MIT Press, pp.163–174, 1984.

[2] 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.

[3] 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.

[4] 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.

[5] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Manipulation Planning for Knotting/Unknotting and Tightly Tying of Deformable Linear Objects", Proc. of IEEE Int. Conf. Robotics and Automation, pp.2516–2521, 2005.

[6] A. M. Ladd and L. E. Kavraki, E, "Using Motion Planning for Knot Untangling", Int. J. of Robotics Research, Vol.23, No.7–8, pp.797–808, 2004.

[7] T. Matsuno, D. Tamaki, F. Arai, and T. Fukuda, "Rope Structure Recognition for Manipulation Using Topological Model and Knot Invariant", J. of SICE, Vol.41, No.4, pp.366–372, 2005, (in Japanese).

[8] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Planning of One-Handed Knotting/Raveling Manipulation of Linear Objects", Proc. of IEEE Int. Conf. Robotics and Automation, pp.1719–1725, 2004.

[9] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Manipulation Planning for Unraveling Linear Objects", Proc. of IEEE Int. Conf. Robotics and Automation, pp.–, 2006.

[10] C. C. Adams, "The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots", Henry Holt & Co., 1994.

[11] D. Rolfsen, "Knots and Links", Publish or Perish, Inc, 1976.