Re-configuring knots to simplify manipulation

Weifu Wang and Devin Balkcom

Dartmouth College Computer Science Department

Abstract. Humans often change the geometry of flexible objects during manipulation so that the goal is easier to accomplish with either simple motions or simple controls. This paper explores how to change the geometry of a knot to allow simpler tying or untying. The paper presents algorithms that modify the knot configuration to allow the knot to be arranged into the correct topological structure or untangled by moving the tip of the string along a straight line, with only a few re-grasps. The paper also presents proof-of-concept physical experiments in which robot arms arrange and untangle several knots.

1 Introduction

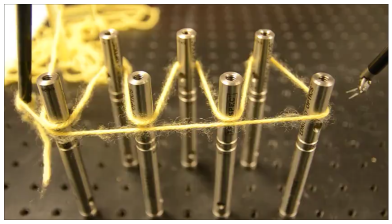
Humans stretch clothes while getting dressed, bend string during knitting and weaving, and change the shape of knots while tying, perhaps using techniques learned from others or based on their own experience. How can robots automatically determine how to change the geometry of flexible objects for easy manipulation? This paper explores algorithms for systematically discovering tricks and shortcuts for tying and untangling new and different knots.

The wide variety of knots, the diverse and complex geometries, and the flexible nature of string all make tying knots with robots challenging. Researchers have successfully tied some simple knots with arms [12, 13, 27, 28, 29, 31]; most attempts explore motions that trace some particular geometry.

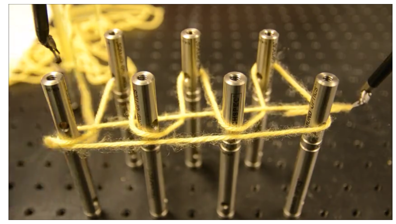
In this paper, we explore how to modify the geometry to make the tying or untying process simpler, mainly in the perspective of using fewer re-grasps. We present a general approach to changing knot geometry so as to allow tying and untangling using simple motions of a robot arm. For tying knots, the geometry is changed virtually – choosing a more convenient goal geometry than the input geometry. For detangling (untying loose knots), the re-configuration is physical, re-configuring the knot as a first step before pulling on a particular end.

We present an algorithm to change the geometry of a knot so that the knot can be arranged or untied by dragging the tip of the string along a straight line. Another algorithm shows that for any knot, there is a sufficient number of such straight line motions that can arrange or untangle the knot. Our algorithm takes the Gauss code, a text-based description of a knot, as input, and outputs the target geometry of the knot.

We also demonstrate a relation between re-grasps and the layout of the knots, and show why many re-grasps are usually necessary, unless the knot geometry is chosen carefully. We show that through manipulation of the knot geometry, we can reduce the use of re-grasps during knot arrangement. Re-grasping is a common practice in knot tying, especially when using a finite DOF robot arm mounted on a fixed base, because



(a) Arranging part of the knot without re-grasping.



(b) Completing the arrangement of a knot 7_1 with one re-grasp.

Fig. 1: Arranging a 7_1 knot with a Da Vinci robot arm.

the arm needs to arrange string both over and under other segments of the laid-out string. However, re-grasping is difficult, since it may require precise information about and control over the environment and the object being manipulated.

In physical experiments, we have arranged and untangled several different knots, including the *double-coin knot* and a knot known as 7_1 in the *standard knot table* [10], using different robot arms to show that the motions needed to tie or untie the knots in this work are in fact simple.

The paper is organized as follows. We first introduce some fundamental concepts about knots and show the importance of geometry and topology in knot tying in Section 2. Then in Section 3, we show the simple arrangement of knots through the manipulation of the knot geometry, using few re-grasps by dragging the tip of the string along a straight line. In Section 4, we show the simple knot untangling approach using the same knot manipulation scheme. Experiments are conducted to show the success in the manipulation of the geometry of knots in arrangement and untangling.

1.1 Related work

Even though changes of geometry are quite common when manipulating flexible objects, large-scale deformation away from the given goal geometry is often considered an undesirable error, instead of a means to simply the manipulation process to achieve the goal. One notable study about changing geometry to simplify manipulation is the work by Demaine *et al.* [11] on *kirigami*, in which paper is first folded into a particular shape, so that a single cut can be made such that when the paper is unfolded, a desired pattern or scene is created.

As a special case of flexible object manipulation, knot tying has been studied by roboticists as far back as the early 1980s [13] where a robot arm was used to tie knots using sensor feedback. More recently, researchers have attempted to use pairs of arms to tie knots [27, 29, 28]; there is also a rich body of work on machine suturing [15, 16, 14, 18]. Hopcroft *et al.* developed a graph-based language to design knot-tying motions and tested the approach using a robot arm [12]. Wakamatsu, Arai, and Hirai's tree search planner finds sequences of motions (selected from four primitives) to tie or untie a knot [31]. Untying knots has also been recently studied as a vision and learning problem [21].

Recent work on knots also includes the authors' work on fixture-based knot tying [7, 8, 34, 35, 33] that separates the knot tying process into *arrangement* and *tightening*, and provides the first bounds on the complexity of knot tying [32]. Apart from knot tying, other examples have been studied as a gateway to understanding string manipulation. Elastic rods have been used to model wires and string for manipulation [30].

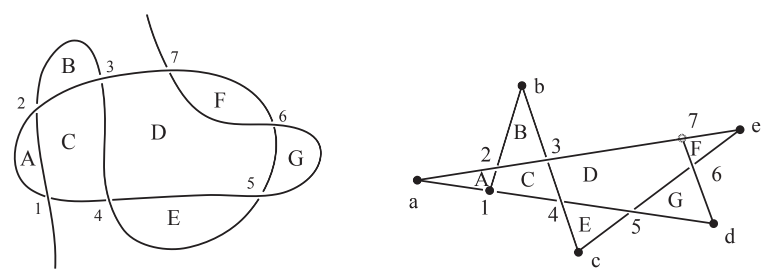
We use many terms from *knot theory* in this paper. A knot is described by its projection onto a plane, called its *knot diagram* [1, 2, 19, 26, 20]. *Physical knot theory* [17] studies the geometry of tight knots formed with thick string [3]. The tightness of a knot has been studied in applied mathematics and physics [24, 6, 22].

Friction plays an important role in tightening a knot, or in untying tight string. Analysis of the frictional forces is frequently a key component of analysis of rigid-body systems [5, 23, 9], and as string wraps around other segments of string with a certain thickness, the friction can be studied by applying the capstan equations [4]. In the present work, we consider only string that is loose enough that friction can be neglected from the analysis.

2 Knots and knot geometry

Knots are usually projected onto a plane for simple description and illustration. If no three points on the knot project to the same point, and no vertex projects to the same point as any other point on the knot [20], the projection is said to be *regular*, and the projected diagram is called a *knot diagram*.

On the drawing, broken lines are used to indicate where one part of the knot under-crosses the other part of the knot that is directly above the broken lines; such locations are called *crossings*. Each crossing is labeled with a unique number, indicating the order of appearance when tracing along the diagram. Figure 2a shows a shoelace knot diagram. In this work, when we say that we "remove" a crossing, we are referring to the result of manipulating one end of the string so that the projected diagram no longer contains the crossing.



(a) A knot diagram of a shoelace knot, with numbered crossings, and cells labeled with letters.

(b) A polygonal shoelace knot diagram.

Fig. 2: Shoelace knot diagrams, with Gauss code $1^+, 2^-, 3^-, 4^+, 5^-, 6^-, 7^+, 3^+, 2^+, 1^-, 4^-, 5^+, 6^+, 7^-$.

We can also use the labels of the crossings to describe the knot. One such description is the *Gauss code*: a sequence of labels for crossings indicating a walk along the diagram from a given starting point. We use numbers to label crossings, and a superscript "+" or "-" to indicate over- or under-crossing. For example, an overhand knot with Gauss code $G = \{1^+, 2^-, 3^+, 1^-, 2^+, 3^-\}$ has 3 crossings, so the length of G is 6 ($|G| = 6$). The following paragraphs give definitions of a few terms from [32].

Each crossing appears twice on a Gauss code for any knot. A sequence of one or more curves connecting two adjacent labels in the Gauss code is called a *c-path* (crossing path). The projected knot diagram separates the plane into several disconnected closed *cells*, labeled by capital letters in Figure 2a. Call the cell that extends to infinity the *exterior cell*, and all other cells *interior cells*.

A c-path is called an *exterior c-path* if it contacts the exterior cell. For example, Figure 2b has c-paths $(1,2)$ through a contacting interior cell A, $(2,3)$ through b contacting interior cell B, $(3,7)$ contacting interior cell D, $(7,6)$ through e contacting interior cell F, $(6,5)$ through d contacting interior cell G, $(5,4)$ through c contacting interior cell E, and $(4,1)$ contacting interior cell C. All of these exterior c-paths contacting the exterior cell that is the complement of the polygonal shape.

Let an *exterior crossing* be a crossing that is the common endpoint of two adjacent exterior c-paths; all the other crossings are *interior crossings*. If a connection between two crossings is not an exterior c-path, it is called an *interior c-path*.

Sometimes the number and the order of crossings can be different even for the same knot; *Reidemeister moves* [25] can be used to transform the crossings without changing the topology of the knot. Determining whether two different Gauss codes represent the same knot is one of the most challenging and fundamental problems in knot theory; we exclude Reidemeister moves in this work.

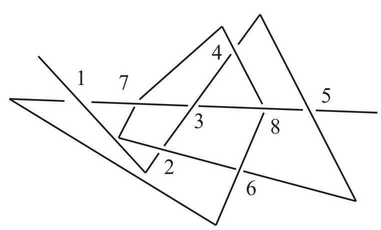
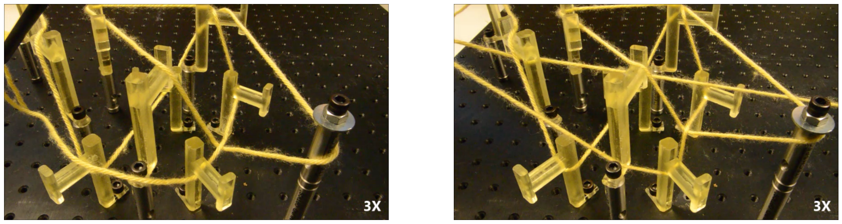


Fig. 3: The resulting configuration of a double-coin knot so that the last five crossings are on a straight line.

2.1 Knots and weaving

On a weaving loom, the warp is the set of strings that form the basic structure around which the weft (the string pulled by the shuttle) is woven. The approach we take in this work to knot tying is to find a simple substructure of the knot in such a way that the under-crossing always appears before its over-crossing so that this spiral-like structure is analogous to the warp. The rest of the knot, like the weft, is then arranged with respect to this warp to construct the more challenging crossings.



(a) Arranging the warp crossings of a double-coin knot.

(b) Completing the arrangement of a double-coin knot with one re-grasp.

Fig. 4: Arranging a double-coin with Da Vinci robot arm.

Let us consider the following example. For a double-coin knot with Gauss code $G = \{1^+, 2^-, 3^+, 4^-, 5^+, 6^-, 2^+, 7^-, 4^+, 8^-, 6^+, 1^-, 7^+, 3^-, 8^+, 5^-\}$, the last five crossings counting from the right open end are 5, 8, 3, 7, and 1. These five crossings can be formed by dragging the right open end along a straight line simulating the motion of a shuttle on a loom, provided that the other segments of string are arranged appropriately. An example of the rearranged polygonal configuration of a double-coin knot is shown in Figure 3. We implemented the knot arrangement using the proposed layout with a Da Vinci robot arm. Figure 4 shows the results of implementation.

3 Arranging knots

In this section, we introduce how to change the geometry of a knot to allow the division of knot arrangement into warp and weft stages. Let us refer to all the crossings formed by the warp stage as *warp crossings*, and all the crossings formed in the weft stage as *weft crossings*. Formally, a crossing i is a weft crossing if and only if after all the crossings to the right (left) of i^a have been removed, and the crossing to the left (right) of i^a has a different sign than a. Remember that each crossing appears twice on a Gauss code, once with a $^+$ superscript (over-crossing), and once with a $^-$ superscript (under-crossing).

We will show that for an arbitrary given knot, we can find patterns on its Gauss code, so that by arranging m alternating sequence of warp and weft crossings, the knot can be arranged with a upper bounded m re-grasps. Since re-grasping is not a trivial task, our approach gives a simpler knot arrangement method compared to knot arrangement methods where re-grasps have to be performed between the arrangement of over- and under-crossings. What is more, in our knot arrange method, weft crossings on the same sequence can be arranged by pulling string to follow a straight line.

3.1 Forming or removing crossings based on weaving sequence

We define a *minimal Gauss code* as a Gauss code that cannot be simplified by performing Reidemeister moves. Adding or removing of a crossing from a structure with minimal Gauss code through physical manipulation of the string can only be achieved if one of the two appearances of the crossing number is at one end of the Gauss code.

What happens if we remove crossings one-by-one from the open ends? Intuitively, we know that after a certain number of removals, the remaining crossing pattern is no longer knotted, because eventually the knot is untied if we remove all crossings. The knotting and unknotting process are symmetric, and it is easier to see the pattern when removing crossings from an existing sequence of crossings, so we choose the unknotting process for analysis.

We define a *weaving sequence* as a sequence of alternating over- and under-crossings that have to be formed or deleted in the given order indicated by the Gauss code. Consider the example of *unknotting* a double-coin knot, whose Gauss code is $G = \{1^+, 2^-, 3^+, 4^-, 5^+, 6^-, 2^+, 7^-, 4^+, 8^-, 6^+, 1^-, 7^+, 3^-, 8^+, 5^-\}$. Starting from the right end, crossing 5^- is adjacent to 8^+ in the Gauss code, and the crossings have *different* superscript signs. Therefore, we can identify these two crossings as part of a weaving sequence — a sequence of weft crossings, and remove crossing 5.

We continue to remove crossings that are part of the same weaving sequence from the right end, including crossings 8, 3, 7, and 1, in order. After we remove the last five crossings in the Gauss code, we have $G = \{2^-, 4^-, 6^-, 2^+, 4^+, 6^+\}$. Now, the next two crossings from the right have the same superscripts, so they are no longer part of a weaving sequence. We know that the five deleted crossings can be formed by a single weft (weaving) motion, dragging the string along a straight line. We continue searching for weaving sequences from right to left. In this example, there are none, and the remaining structure consists only of warp crossings.

The following algorithm, which takes the Gauss code of the knot as input, finds weaving sequences for an arbitrary knot. With a single pass through the Gauss code, the algorithm outputs a sufficient number of m weaving sequences that can be used to form the knot; m is also a sufficient number of re-grasps to tie the knot with a fixed-base arm.

Algorithm 1: WEAVE

1. Select either left or right end of the Gauss code.
2. Delete crossings from the selected end.
3. If the crossing to be removed has a different sign from the next crossing to removed, then the two crossings belong to the same weaving sequence. Register a new weaving sequence if the current crossing to be removed is not already on a weaving sequence. If the crossing to be removed has the same sign as the next crossing to be removed, then terminate the current weaving sequence; if the crossing to be removed has the same label as the next crossing to be removed (for example, $i^- j^- j^+$ where j^- and j^+ have the same label), then compare the sign to the first crossing with a different label (compare j^+ with i^- in the given example).
4. Repeat steps 2 and 3 until only one crossing is left, attach the last crossing to the on-going pattern.

For a Gauss code with k crossings where $|G| = 2k$, we can find $O(k^2)$ different sequence of labels that are the results of removing the crossings at the beginning or the end of the Gauss code. However, since a weaving sequence can only be formed by weaving with one end of the string, we only need to check the sequence of labels that are the results of removing crossings from solely the left or right end. The total length of such a sequence of labels is $2k$.

The algorithm only checks if the current crossing has a different sign from the adjacent one. This approach may overlook some structures that are unknotted but still contain adjacent crossings that have different signs, such as $\{1^+, 2^-, 3^+, 4^-, 4^+, 3^-, 2^+, 1^-\}$. This structure is unknotted, but still contains one weaving sequence.

Knots such as the overhand knot contains only one weaving sequence associated with the last two crossings, while the first crossing can be formed by a type I Reidemeister move. Similarly, the *figure eight knot* with Gauss code $G = \{1^+, 2^-, 3^+, 4^-, 2^+, 1^-, 4^+, 3^-\}$, also contains only one weaving sequence associated the last three crossings where the first crossing is achieved by a type I Reidemeister move. The double-coin knot shown earlier contains one weaving sequence, with the crossings 2, 4 and 6 forming the initial unknotted structure.

Lemma 1. *If a knot can be arranged by following a single weaving sequence, then a motion that removes all weft crossings unties the knot.*

Proof. If we remove all the crossings on the weaving sequence and there is only one weaving sequence, the remaining crossings are all warp crossings by definition. Then, if we continue to remove crossings, every crossing they remove will have the same sign as the the next crossing to remove. Without loss of generality, let us assume the first crossing we will remove is an over-crossing. Then, since all remaining crossings are warp crossings, whenever we are trying to remove a crossing i^a, a is $+$ until all crossings are removed. Then, all these warp crossings can be arranged on two layers. One plane contains only the over-crossings, while the other plane contains all the under-crossings, with finitely many vertical line segments connecting two planes. This structure has the topology of a circle when the ends of the string are connected to each other, an *unknot*.

The lemma shows that even though there are knots of many crossings, but they are in fact simple knots. A single motion can untie the knot. We believe that the number of crossings may not be the best way to illustrate how complex a knot is.

3.2 Aligning crossings on a straight line for simple manipulation

A straight-line motion is easy to achieve even for simple robotic devices. This section will show that weft crossings can always be aligned on a single straight line, without changing the knot topology.

Theorem 1. *In a weaving sequence, each crossing label appears only once.*

Proof. A weaving sequence contains an alternating over- and under-crossing pattern, and all the crossings on the weaving sequence are adjacent to each other in the Gauss code. If the same label j appears twice, let crossings i^- and k^- be the two crossings in the Gauss code adjacent to j^+, and let s^+ and t^+ be the two crossings adjacent to j^-. The crossings i, k, s and t have the corresponding signs because they are adjacent crossings to j, and they are on the same weaving sequence. Without loss of generality, let j^- be closer to the open end. After the deletion of the crossing j^-, crossings i and k are now adjacent in the Gauss code, and they have the same sign, so they cannot be on the same weaving sequence.

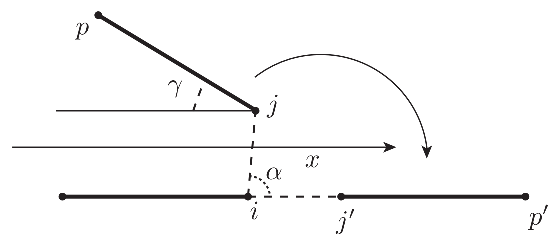
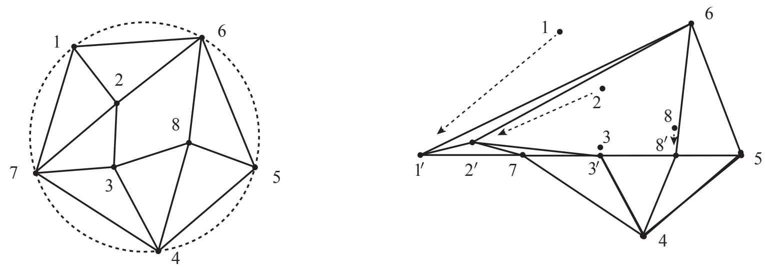


Fig. 5: Rotating extreme segments to align all weaving crossings on a straight line.



(a) A polygonal knot diagram for a double-coin knot, computed using methods proposed in [32].

(b) The rearranged configuration for the double-coin knot, by rotating extreme segments using the proposed method indicated in the Algorithm 2.

Fig. 6: Reconfiguration of a double-coin knot to align weft crossings onto a straight line.

Since no crossing appears twice on a weaving sequence, we can move the crossings so that they appear on a straight line. Therefore, a single straight line motion of the string can form multiple crossings at once. Define an *extreme segment* of the weaving sequence as the segment between two exterior crossings.

The algorithm below shows how to arrange extreme segments on a straight line if they are on the the same weaving sequence. The input to the algorithm is the locations of all the crossings, which may be computed using the technique implied by the proof of Theorem 10 in [32]. The algorithm computes the geometry of the knot during arrangement, and computes the placement of the fixtures that are used in our experiments. Because the geometry computed are straight line configurations supported by the fixtures, no vision feedback is needed in our approach.

Algorithm 2: ALIGN

1. For each given extreme segment, connect a line between the two exterior crossings (or an interior to an exterior crossing), and move each of the crossings on the extreme segment to its projection on the connected line;
2. Let crossing i $((x_i, y_i))$ and crossing j $((x_j, y_j))$ be the two adjacent exterior crossings on two adjacent extreme segments, with k connections between them; without loss of generality, let $y_j < y_i$; let p be the other end point on the extreme segment with j as an end point;

3. Rotate all the points above extreme segment between p and j (because $y_j < y_i$) around j then around i in the stated order, so that (x'_j, y'_j) is the new location of crossing j where $y'_j = y_i$, as shown in Figure 5; The angle rotated around crossing i can be calculated as the acute angle α between the x axis and the vector ij, and the rotation angle around crossing j is $\beta = \pi - \alpha - \gamma$, where γ is the angle between pj and the x axis;
4. Along the line $x = (x_i + x'_i)/2$, find k points above (or under) the line $y = y_i$ if $x'_i \geq x_i$ ($x'_i < x_i$) with equal distance. For the endpoints of the k pairs of connection between two extreme segments, connect to k points along the line $x = (x_i + x'_i)/2$ in order based on the distance of the end point on segment pj to the exterior crossing j, such that no additional intersection is introduced;
5. Adjust the z coordinates of all crossings so that the crossings on the rearranged weaving sequence lie on a straight line in three dimensions;

The result of applying the process to a double-coin knot is shown in Figure 6.

In our previous work [32], we have shown that an arbitrary (polygonal) knot with k crossings can be laid out based on its Gauss code using no more than $3k-2$ line segments. We use this projected configuration to compute where each crossing should be placed so the crossings on a weaving sequence is on a straight line in our experiments.

3.3 Re-grasping and weaving

This section analyzes the number of re-grasps needed to arrange each of the two types of crossings. If we do not choose the geometry of the final knot configuration carefully, the number of re-grasps needed to arrange a knot maybe be as large as the number of weft crossings, plus one re-grasp for each sequence of warp crossings.

We show that the output of Algorithm 1 also gives a sufficient bound for the number of re-grasps needed to arrange a given knot, if each sequence of weft crossings are aligned on a straight line following Algorithm 2.

We can use a fixed-base robot arm to lay out the warp crossings without re-grasping:

Lemma 2. *No re-grasp is needed to arrange a collection of warp crossings if for each warp crossing, its under-crossing appears before its over-crossing.*

Proof. The structure is unknotted and belongs to two layers. If for each layer, we trace along the configuration with a robot arm, then the two layers form the corresponding crossings. Therefore, no re-grasping is needed.

Sometimes, changing the geometry of the knot can reduce the number of degrees of freedom that a fixed-base arm must have to tie the knot using a particular number of re-grasps. Notice that an elephant-trunk arm with infinite degrees of freedom can arrange any knot without re-grasping.

The next lemma shows that changing the geometry of the knot is in fact sometimes necessary to optimize this tradeoff between degrees of freedom of the arm and the number of re-grasps required: some knot geometries are quite difficult for any finite-DOF arm. For example, a sequence of crossings of the form i^a, j^b, k^a, where a and b have opposite signs. A 4 DOF robot arm needs at least one re-grasp to arrange this sequence of the crossings, if j^a has already been arranged. Such sequence of crossings are the basic structures of the weaving sequence, so traditional knot tying approach of complex knot usually involves many re-grasps.

A weaving sequence contains a sequence of consecutive over- and under-crossings. However, if we imagine the robot end-effector as the shuttle on the loom, it does not need to re-grasp every time the string switches between over- and under-crossing, if all the crossings on this weaving sequence is on a straight line. However, for a weaving sequence, one re-grasp is still needed.

Lemma 3. *To arrange a weaving sequence with a fixed-base robot arm grasping the ends of the string, at least one re-grasp is needed.*

Proof. Let the two ends of a knot be S_1 and S_2, and let S_1 be fixed to the ground. Let the base of the robot arm be B, and the end effector be E. The robot arm needs to grasp S_2 with E during the arrangement of the knot. Let us assume no re-grasp is performed during the arrangement of a weaving sequence w. At the end of the arrangement of the weaving sequence, let the configuration of the string be fixed in space. Let curve c_1 be the current configuration of the robot arm, connecting from B to E. Let curve c_2 be the curve of a different robot arm configuration connecting from B to E, where the entire robot arm is outside the convex hull of the knot. In both configurations, the E is attached to S_2. Then, the two curves belong to two different homotopy classes with respect to the string. The curve connecting c_2 to S_2 and then to S_1 has the correct knot topology. Therefore, at least one re-grasp is needed to arrange a weaving sequence.

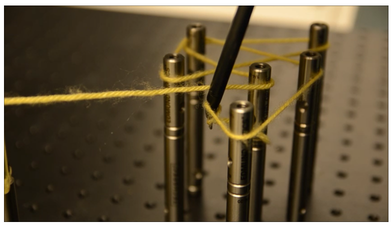
The number m output by Algorithm 2 gives a sufficient number of re-grasps needed to tie a given knot. For many knots, including the double-coin knot, the number is 1. Since these knots are in a different topology class from a topological loop when one of their end points are grasped by a robot arm and the other end point is attached to the ground, at least one re-grasp is needed, so this number is also a lower bound.

3.4 Knot weaving with Da Vinci

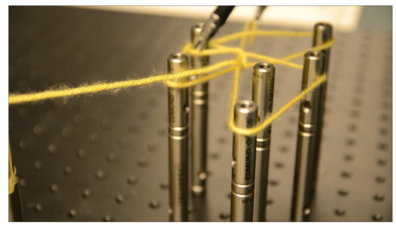
We conducted experiments with a Da Vinci surgical robot, which has two symmetric high precision arms. We computed knot layouts and built fixtures to support string arranged at various heights, allowing weaving to be implemented with a single translation. Based on the computed knot configuration, such as the example shown in Figure 3, the robot manipulator follows predefined paths without vision feedback.

In all the experiments conducted, during the layout of the warp crossings, the string is supported by fixtures. Each fixture is either straight rods or upside down "L" shape fixtures on top of rods. These fixtures are placed at various locations to support warp crossings to form the computed geometry, so that the weft crossings are all aligned. The location of the fixtures are also computed automatically by placing a fixture on the inside of each turn of the string, so that the string wraps around them follows the shortest path in the homotopy class [34]. Because the fixtures are simply rods placed at various computed locations, we consider the approach simple and general.

Figure 4a shows the layout of the warp crossings of a double-coin knot, arranged without re-grasps. After layout, the effector of the second arm grasps the tip of the string and uses a pure translation to complete the knot, as shown in Figure 4b and in the multimedia attachments. In the attached video, the author had to manually intervene to loosen the string wrap around the fixtures, due to high tension along the string. The need for human intervention is the combined results of the use of yarn, the manipulator of Da Vinci robot cannot fully close, and the high friction coefficient between yarn and the printed “L” shaped fixture. One of the future work is to study how to automatically use robots to loosen string during manipulation.



(a) Arranging warp crossings of a double-coin knot at the same height.



(b) Completing the arrangement of a double-coin knot with a single re-grasp.

Fig. 7: Arranging a double-coin by laying out the warp crossings on the same height.

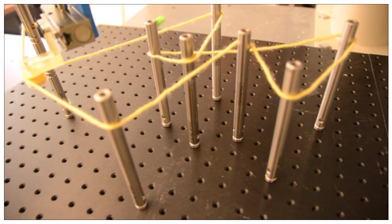
The preliminary arrangement of the string requires many support structures laid out in the workspace of the robot. For simplicity, we programmed the robot to just arrange the warp crossings of the string at the same heights around simple fixtures. Figures 1a and 1b show the arrangement of a 7_1 knot. Figures 7a and 7b shows the arrangement of a double-coin knot.

Our approach is able to arrange a double-coin knot with only a simple re-grasp. There has not been any other successful attempt to arrange a double-coin knot with a robot arm. However, by following the manipulation sequence suggested in previous work such as [28], five re-grasps are needed to arrange the double-coin knot. Even with state of the art robotic manipulation strategies, a re-grasp takes a long time to execute. Even though we did not compare the execution time or our approach to other knot tying approaches, we believe when a robot arm following two paths of similar lengths, the fewer re-grasps are performed, the shorter the execution time.

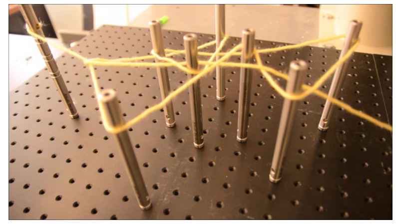
Even though weaving sequence can be found in many knots, different knots containing the same number of crossings can have different number weaving sequences. For example, the double coin knot contains 8 crossings, and is labeled 8_{18} on the standard knot table. This knot, as shown above, can be arranged with a single re-grasp. However, the knot labeled 8_4 contains two weaving sequences, even though it contains the same number of 8 crossings. Following our knot arrangement strategy, arranging the double-coin knot will be simpler compared to knot 8_4. We are not able to determine how the number of weaving sequences on a knot is related to the number of crossings.

3.5 Robot-human collaboration

When we arrange the segments of string that are not part of a weaving sequence at the same height, the robot arm weaves around arranged segments of string. Even though the

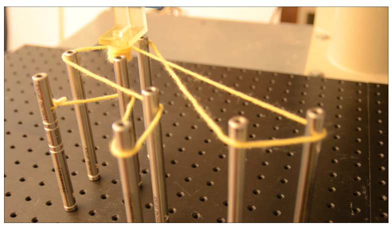


(a) Arranging warp crossings of a 8_{10} knot at the same height.

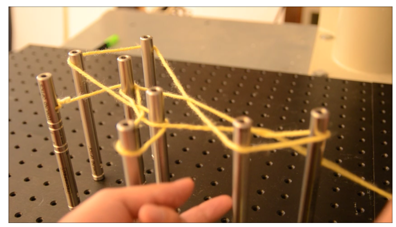


(b) Completing the arrangement of a 8_{10} knot by a human weaving the string.

Fig. 8: Arranging a 8_{10} knot by robot and human collaborating together.



(a) Arranging warp crossings of a 9_{31} knot at the same height.



(b) Completing the arrangement of a 9_{31} knot by a human weaving the string.

Fig. 9: Arranging a 9_{31} knot by robot and human collaborating together.

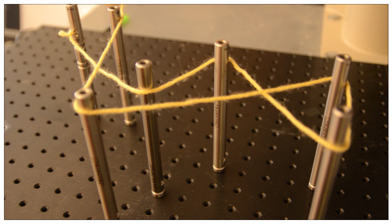
locations of the string segments are known, the motion still may not be easy to perform for a robot. Humans, however, can arrange the weaving sequence easily with re-grasps.

For example, a double-coin knot can be tied using the robot to lay out the structure, and allowing the human to finish the weft crossings. We used this technique to tie figure-eight knots, and knots 7_1, 8_2, 8_5, 8_{10}, 9_{31}, 9_{32} from the standard knot table. Applying Algorithm 2, all the listed knots can be tied with one re-grasp. Figures 8, 9, and 10 show the examples of a human collaborating with an Adept Cobra industrial arm to tie knot 8_{10}, knot 9_{31} and knot 9_{32}.

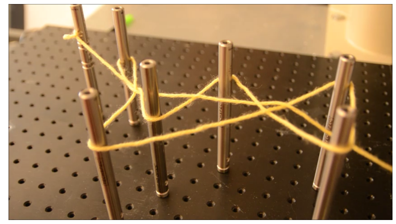
4 Untangling knots

In the previous approach, we changed geometry of knots to simplify the knot arrangement, based on the identification of wrap and weft crossings on the knots, and we identify the weft crossings by processing the Gauss code.

The same change of geometry can also be used to untangle knots. In this work, we will focus on untangling knots from a loose configuration rather than untying knots from a tight configuration. We untangle the knot by changing the geometry of the knot and pulling the string several times along straight lines. With each pull of the string, we remove all consecutive weft crossings that extend to the current end of the string.



(a) Arranging warp crossings of a 9_{32} knot at the same height.



(b) Completing the arrangement of a 9_{32} knot by a human weaving the string.

Fig. 10: Arranging a 9_{32} knot by robot and human collaborating together.

However, non-consecutive weft crossings may not be able to be removed in a single motion.

Given two different sequences of weft crossings, in order for them to be aligned, other crossings need to be relocated. In order to change the knot configuration of the knot to align the second consecutive sequence of weft crossings on a straight line, the crossings on the first consecutive sequence of weft crossings need to be relocated, which may break the alignment. Therefore, without knowing which specific knot we are trying to untangle and detailed analysis of the specific knot, the best we can do with each grasp is to align a sequence of consecutive weft crossings, and remove them by moving the string along a straight line.

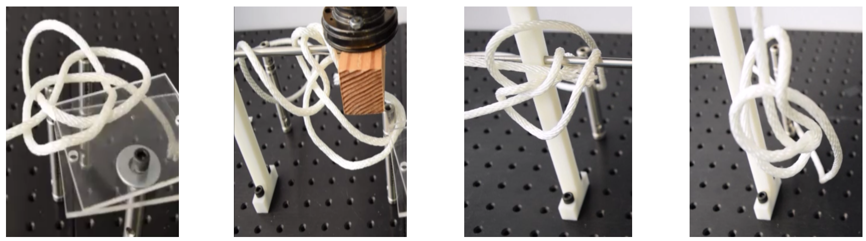
Even when a knot is in a loose configuration, the untangling of knots usually have to overcome friction. After we have identified all the weft crossings, and attempt to use a single motion to untangle them, friction may prevent the untangling, such as shown in Figure 11 where we attempt to remove the last five crossings by pulling the string direction without changing the geometry. Therefore, a pulling motion of string that involves least friction is desirable. It appears that pulling string along a straight line can keep friction relatively low.

Fig. 11: Friction prevents the untying of a double-coin knot.

The process of manipulating the geometry can be described as follows. We first choose one end of the string to untangle the knot. Along the chosen end of the string, we determine a side that is closer to the boundary, left or right. Starting from the chosen end of the string, we identify all the cells on the chosen side to the string in sequence, until the last consecutive weft crossing, and then we find the largest inscribed circle in each cell. Using the same algorithm we presented in the previous section, we will delete crossings from the chosen end, and record how the crossings change, either from under-crossing to over-crossing, or from over-crossing to under-crossing.

Given the x-y plane on which the knot diagram is projected, we place a vector parallel to the z axis at each center of the largest inscribed circle we have identified. The direction of the vector is positive if the crossings associated with the cell change from under-crossing to over-crossing, otherwise negative.

We then use a robot arm manipulating a rod to follow these vectors parallel to the z axis to the points above and below the $z = 0$ plane, and connect between these points by following linear motions with the end effector. After tracing all the vectors, we have aligned all the weft crossings. Figures 12 and 13 show the change of the geometry.



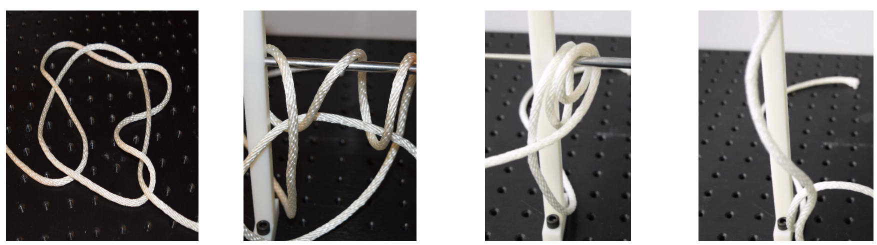
(a) The initial configuration of double-coin knot before untangling.

(b) Aligning several crossings on a straight line for untangling.

(c) Pulling string along straight line to untangle.

(d) Knot is fully untangled after removing the rod.

Fig. 12: Untangling a double-coin knot.



(a) The initial configuration of knot 7_1 before untangling.

(b) Aligning several crossings on a straight line for untangling.

(c) Pulling string along straight line to untangle.

(d) Knot is fully untangled after removing the rod.

Fig. 13: Untangling knot 7_1.

We then identify the last weft crossing we have aligned, and the warp crossing adjacent to it, and let the robot grasp any point between the two crossings. The robot arm then pulls the string along the direction parallel to the vector point along the rod we used to align the crossings. After all the consecutive weft crossings are removed and the rod is removed, the knot will be untangled. Even though we have only demonstrated the untangling of loose knots, the principle can be applied to tight knots, if we can identify the crossings and thread a needle through those enclosed cells.

5 Conclusions and future work

This work shows that some knots can be tied or untied with simple motions by changing the geometry of the knots. We discussed an algorithm for changing the geometry of the knots, and another algorithm for discovering different knot tying or untying phases using the Gauss code descriptor for a knot. We also showed practical implementations using simple robot arms, and also as a collaboration between a robot arm and a human.

For future work, we would like to better understand how motions can be designed to mechanically simplify tying knots and untying even tightened knots. We are also interested to know if by changing the geometry of the goal, we can manipulate other flexible objects, such as cloth, using simple motions.

We are particularly interested in knots like the shoelace and sheepshank; humans tie these knots by pulling loops through loops. We can identify these structures from the Gauss code, as Type II Reidemeister moves: for each adjacent appearance, crossings i and j have the same sign, and the sign is different from the crossings adjacent to the ij (or ji) sequence.

We would like to thank Dmitry Berenson and Gregory Fisher for letting us use the Da Vinci robot for some of the experiments. This work is supported by NSF grant IIS-1217447.

References

[1] C.C. Adams. *The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots*. American Mathematical Society, 2004.
[2] J. W. Alexander. Topological invariants of knots and links. *Trans. Amer. Math. Soc.*, 20:275–306, 1923.
[3] Ted Ashton, Jason Cantarella, Michael Piatek, and Eric Rawdon. Knot tightening by constrained gradient descent. *Experimental Mathematics*, 20(1):57–90, 2011.
[4] Stephen W. Attaway. The mechanics of friction in rope rescue. *International Technical Rescue Symposium*, 1999.
[5] Devin J. Balkcom, Jeffrey C. Trinkle, and E. J. Gottlieb. Computing wrench cones for planar contact tasks. In *ICRA*, pages 869–875. IEEE, 2002.
[6] J. Baranska, S. Przybyl, and P. Pieranski. Curvature and torsion of the tight closed trefoil knot. *The European Physical Journal B - Condensed Matter and Complex Systems*, 66(4):547–556, 2008.
[7] Matthew P. Bell. Flexible Object Manipulation. Technical Report TR2010-663, Dartmouth College, Computer Science, Hanover, NH, February 2010.
[8] Matthew P. Bell, Weifu Wang (co-first author), Jordan Kunzika, and Devin Balkcom. Knot-tying with four-piece fixtures. *International Journal of Robotics Research (IJRR)*, vol 33, no. 11:1481–1489, Sep, 2014.
[9] Stephen Berard, Kevin Egan, and Jeffrey C. Trinkle. Contact modes and complementary cones. In *ICRA*, pages 5280–5286, 2004.
[10] G. Burde. Knoten. *Jahrbuch Ueberblicke Mathematik*, pages 131–147, 1978.
[11] Erik D. Demaine, Martin L. Demaine, Andrea Hawksley, Hiro Ito, Po-Ru Loh, Shelly Manber, and Omari Stephens. Making polygons by simple folds and one straight cut. In *Revised Papers from the China-Japan Joint Conference on Computational Geometry, Graphs and Applications (CGGA 2010)*, Lecture Notes in Computer Science, pages 27–43, Dalian, China, November 3–6 2010.
[12] John E. Hopcroft, Joseph K. Kearney, and Dean B. Krafft. A case study of flexible object manipulation. *International Journal of Robotic Research*, 10(1):41–50, 1991.
[13] H. Inoue and M. Inaba. Hand-eye coordination in rope handling. *Robotics Research: The first International Symposium*, pages 163–174, 1985.
[14] H Kang and J.T. Wen. Robotic knot tying in minimally invasive surgeries. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[15] Hyosig Kang and John T. Wen. Endobot: a robotic assistant in minimally invasive surgeries. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 2031–2036, 2001.

[16] Hyosig Kang and J.T. Wen. Robotic assistants aid surgeons during minimally invasive procedures. *Engineering in Medicine and Biology Magazine, IEEE*, 20(1):94–104, Jan 2001.

[17] L.H. Kauffman. *Knots and Physics*. K & E series on knots and everything. World Scientific, 1991.

[18] Makoto Kudo, Yasuo Nasu, Kazuhisa Mitobe, and Branislav Borovac. Multi-arm robot control system for manipulation of flexible materials in sewing operation. *Mechatronics*, 10(3):371 – 402, 2000.

[19] W.B.R. Lickorish. *An Introduction to Knot Theory*. Graduate Texts in Mathematics. Springer New York, 1997.

[20] Charles Livingston. *Knot theory*. The carus mathematical monographs, Volume Twenty Four. The mathematical Association of America, Washington D.C., 1993.

[21] Wen Hao Lui and Ashutosh Saxena. Tangled: Learning to untangle ropes with RGB-D perception. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 837–844, 2013.

[22] J.H.Maddocks M. Carlen, B. Laurie and J. Smutny. Biarcs, global radius of curvature, and the computation of ideal knot shapes. *Physical and numerical models in knot theory*, 36 of Ser. Knots Everything:75–108, 2005.

[23] Matthew T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, Cambridge, MA, August 2001.

[24] Eric J. Rawdon. Approximating the thickness of a knot. *Ideal knots*, 19 of Ser. Knots Everything:143–150, 1998.

[25] Kurt Reidemeister. Elementare begrndung der knotentheorie. *Abhandlungen aus dem Mathematischen Seminar der Universitt Hamburg*, 5(1):24–32, 1927.

[26] D. Rolfsen. *Knots and Links*. AMS/Chelsea Publication Series. AMS Chelsea Pub., 1976.

[27] Mitul Saha and Pekka Isto. Motion planning for robotic manipulation of deformable linear objects. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2478–2484, May 2006.

[28] Mitul Saha and Pekka Isto. Manipulation planning for deformable linear objects. *IEEE Transaction on Robotics*, 23(6):1141–1150, December 2007.

[29] Mitul Saha, Pekka Isto, and J.-C. Latombe. Motion planning for robotic knot tying. In *Proc. International Symposium on Experimental Robotics*, July 2006.

[30] Zoe McCarthy Timothy W Bretl. Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *International Journal of Robotics Research (IJRR)*, June 2013.

[31] Hidefumi Wakamatsu, Eiji Arai, and Shinichi Hirai. Knotting/unknotting manipulation of deformable linear objects. *International Journal of Robotics Research*, 25:371–395, 2006.

[32] Weifu Wang and Devin Balkcom. Grasping and folding knots. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 3647–3654, 2016.

[33] Weifu Wang and Devin Balkcom. Towards tying knots precisely. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 3639–3646, 2016.

[34] Weifu Wang, Matthew P. Bell, and Devin J. Balkcom. Towards arranging and tightening knots and unknots with fixtures. In *International Workshop on the Algorithmic Foundations of Robotics, WAFR 2014*, pages 677–694, 2014.

[35] Weifu Wang, Matthew P. Bell, and Devin J. Balkcom. Towards arranging and tightening knots and unknots with fixtures. *IEEE T. Automation Science and Engineering*, 12(4):1318–1331, 2015.