Knot-tying with four-piece fixtures

Matthew P. Bell* Weifu Wang* Jordan Kunzika Devin Balkcom

Abstract

We present a class of fixtures that can be disassembled into four pieces to extract the loosely-tied knot. We prove that a fixture can be designed for *any* particular knot such that the knot can be extracted using only simple pure translations of the four fixture sections. We explore some of the issues raised by our experimental work with these fixtures, which show that simple knots can be tied extremely quickly (less than half a second) and reliably (99% repeatability) using four-piece fixtures.

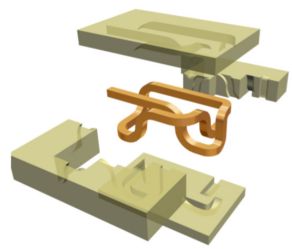


Figure 1: A four-piece fixture for the overhand knot.

1 Introduction

Autonomous knot tying machines have often used complex manipulators and extensive sensing. For example, in work dating back almost three decades, Inoue and Inaba developed a system using a 6+1 DOF arm with stereo machine vision [13].

In contrast, this paper demonstrates that knot tying is possible using fixtures with very minimal actuation, and with no sensing. Fixtures are static devices that manipulate an object into a desired configuration when the object is pushed against or through the fixture, allowing a complex manipulation task to be achieved with a simple (typically one DOF) control. This paper extends work first presented in [2] and [1].

Knot-tying is a hard problem; we have only begun to understand the behavior of knots in fixtures. The main theoretical contribution of this work is a provably complete algorithm that can (in theory) be used to design a four-piece fixture that can arrange any number of strings into *any* desired knot described by a mathematical knot diagram; this fixture can be disassembled to fully expose the tied string using only four simple translations of the pieces.

In practice, physical string manipulation challenges intrude when attempting to tie complex knots. However, we show empirically that knots as complicated as the "Harness Bend" can be tied reliably and extremely quickly (in all cases, within less than half of a second) using this novel approach to the design of knot-tying devices.

In previous work [2], we presented one-piece fixtures, which allow wire, fishing line or other similar materials with high stiffness to be pushed through the fixture and either pulled out the entrance hole or through an exit hole. When extracted, a knot has been tied. However, one-piece fixtures are limited because they rely on the ability to push the string through a winding tube; we have only successfully designed one-piece fixtures to tie simple knots such as *overhand knots* and *square knots*.

For more complex knots, or for materials that cannot easily be pushed, both insertion and extraction become more challenging. In current work, we show how four-piece fixtures , such as the one shown in Figure 1, can be designed. The assembled fixture contains a single tube for the knot shape, simplifying insertion. The fixture can be disassembled by simple straight-line translation of each of the four components of the fixture, allowing easy extraction of the loosely-tied knot. A single-degree-of-freedom

*Matthew P. Bell and Weifu Wang contributed equally in this work.

mechanism can achieve this disassembly, simplifying extraction further.

Formally, a mathematical knot is an embedding of a topological circle in R^3. We might observe that the "knots" tied by our fixtures are not mathematical knots in the truest sense, since the ends of mathematical knots are "glued" together to enforce a particular topology. However, simply connecting the correct ends of the string(s) together after extraction from the fixture would create a structure with the desired topology. Our fixtures are able to tie several strings together, technically, a mathematical *link.* Four-piece fixtures focus on the problem of achieving the correct topology of a knot or link in string; geometric considerations (such as tightening a complex knot or fake knot) are left for future work.

1.1 Related work

Fixtures present a way to manipulate or grasp flexible materials with little or no sensing, and with simplified motions. Lu and Akella developed a method of folding cardboard cartons using fixtures as manipulators [16, 17]. They view the unfolded carton blank as a robot with a series of revolute joints connecting the segments, and develop a motion planning algorithm that generates folding sequences for such robots, provided that there are no kinematic loops. From these sequences, it is possible to generate fixtures that fold the carton blank into a box.

Caging is a similar problem that involves adding constraints to a system in an uncertain configuration in order to bring it into a known configuration. In early work in this area, Rimon and Blake developed an algorithm to to determine caging grasps for complex 2D objects [21].

Linear deformable objects (LDOs) and string have been analyzed extensively, and many different models have been developed. Pai used Cosserat rods to simulate thin strands, such as sutures, and developed a fast simulator capable of supporting real-time interaction with a virtual suture [18]. Wakamatsu and Hirai developed a detailed model of linear object deformation based on differential geometry, and applied it to path planning with an LDO [31]. Wakamatsu *et al.* also developed a 2D model of LDOs in contact with obstacles, with extensions to support dynamic deformation as a result of external forces and moments, as well as geometrical constraints. [32]. Recently, in the robotics community, work by Bretl and McCarthy has shown that a Pontryagin-type formulation can be used to allow motion planning for elastic rods (*e.g.* stiff string or wire) over equilibrium configurations [3].

Work on general deformable objects is also relevant. Rodríguez, Lien, and Amato worked on motion planning in an environment where every object is deformable [22]. They focus on maintaining constant volume for all deformable objects, and use a tree-based planner to explore the system. This type of planning can also be applied to grasping problems in deformable environments.

Work on snake robots considers the problem of "active" control of flexible linear objects, whereas we consider passive control. See Henning, Hickman, and Choset's work on motion planning for serpentine robots for an example and a discussion of related work in this area [11], as well as Degani *et al.* 's work on manipulators that traverse tubes [8].

While our system is not the first to tie knots autonomously, our system is considerably less complex than others. Knot tying was first explored by Inoue and Inaba using a 6+1 DOF robot arm with stereo machine vision [13]. The arm is able to successfully insert a rope into a ring, and then tie the rope into a knot around the ring. Hopcroft *et al.* developed a graph-based language meant for programming knot tying motions at a fairly high level, and tested it by tying knots with a robot arm [12].

Phillips, Ladd and Kavraki created a simulator which models realistic rope using a spline formed from linear springs (with support for collisions), with suturing explored as a possible application [19]. Taylor gives a survey of medical robots in general, with some mention of suturing systems [29]. One such suturing system is the EndoBot, developed by Kang and Wen [14]. Their system includes algorithms for autonomously tying knots while suturing, and they have modified a shuttle needle device for use by the robot. Other examples of suturing work include [33, 23, 10].

Saha, Isto, and Latombe developed a string model and motion planning algorithms for tying both real and simulated knots [25, 26, 24]. They analyze a knot, and construct a sequence of motions for tying it using two robot arms. Wakamatsu, Arai and Hirai developed a very detailed description of the theory involved in tying and untying knots with robots [30]. They defined a set of four basic operations for transitioning between states, and used tree search to build a planner for finding a sequence of motions to tie a knot. They also showed that a SCARA robot (three translational DOF and one rotational DOF) is sufficient to tie an LDO into a knot.

There are also patents for devices to assist in knot tying, from tying fishing line [5] and shoelaces [4], to suturing [27]. Each of these devices is carefully designed to tie a particular, and fairly simple, type of knot; the goal of our work is to develop an approach to designing fixtures that is in some sense general (although the final instantiation of each fixture is suited only to a single knot type). The stitches tied by the ubiquitous sewing machine are also a type of knot: typically, a loop of string is pushed through cloth, and a device underneath the cloth loops a second piece of string through and around the first, which is then pulled tight. This motion is carefully designed for each type of stitch.

Separating multi-piece knot fixtures is very similar to the problem of extracting cast or injection-molded parts from their molds. Typically, only 2-moldability is considered, in which there are two mold pieces that are separated with one translation. Early approaches only considered the three principal axes as parting directions; however, they allowed complex (non-planar) parting surfaces [20, 6]. Later approaches allowed for multiple mold pieces. Chen gave criteria for identifying parting faces, which are used with a reverse glue operation to produce a set of mold pieces [7]. Khardekar *et al.* developed an algorithm running on graphics hardware for computing a feasible parting direction for two mold pieces, with real-time highlighting of undercuts [15].

2 Four-piece fixture design

The fixtures presented contain simple tubes for the string, and can be disassembled to extract the "tied" knot. How many pieces should a fixture be disassembled into? We will show in this section that for *any* given knot type (with any number of strings to be tied together), a four-piece fixture can always be designed that allows extraction through simple translations of each of the four fixture pieces.

In order to formally study algorithmic design of four-piece fixtures, we will need some definitions and a model of the string. Let us initially model a knot using an infinitely thin smooth closed curve in R^3. (We will relax the restriction that the string be infinitely thin in the paragraphs below, and we will cut the closed curve as well.)

In mathematical knot theory, two knots are *equivalent* if there is a smooth deformation between the curves that does not cause self-intersection; this allows classification of knots into types based on the equivalence classes. We use the standard definition of the *knot diagram*: a planar projection of a knot curve in general position, such that there are a finite number of crossings on the projected curve, and tangent vectors at these crossings are in different directions.

Given a knot diagram, we can construct a planar directed graph from the diagram by placing a graph node at each crossing, a graph node on each strand between crossings (Figure 2(b)). We may also choose an arbitrary location away from crossings to cut the loop, and add graph nodes at the endpoints. Each graph node is denoted as either an *endpoint* node or a *segment* node.

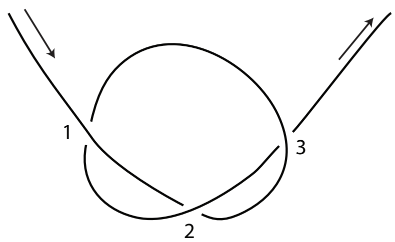
We can now describe the knot using a *Gauss code*, which consists of an ordered list of junctions, each of which is denoted as an over-crossing (o) or an under-crossing (u), generated by following the original curve in space continuously and the corresponding points on the planar projection. As an example, the overhand knot shown in Figure 2(a) has the Gauss code `1o, 2u, 3o, 1u, 2o, 3u`.

We are now ready to state the main result.

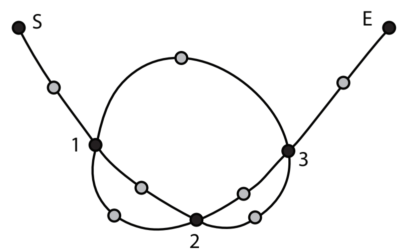
Theorem 1. *Given any (mathematical) knot or link consisting of one or more strands of string, and described by a Gauss code, a fixture can be constructed that loosely arranges string into a knot with the same Gauss code, provided that the endpoints of the string are connected together outside of the fixture. Furthermore, this fixture can be cut into four pieces in such a way that all four pieces can be removed by pure translation without interfering with the string, in the sense that if the initial string configuration were treated as a rigid body, no interpenetration between the string and fixture occurs during fixture separation.*

The proof is constructive – we will show that starting with a desired Gauss code, there are well-defined steps (some of which are themselves known complete algorithms) that create a fixture with the desired properties.

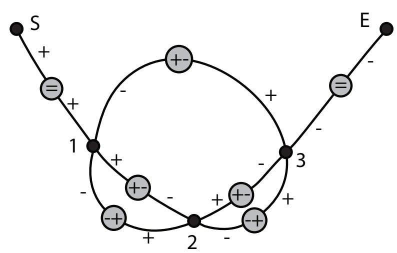
We begin by creating a knot diagram based on the Gauss code, and adding graph nodes as described before.



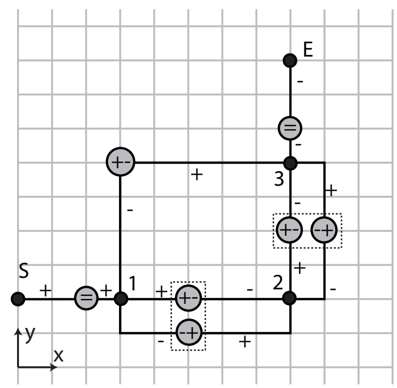
(a) Knot diagram



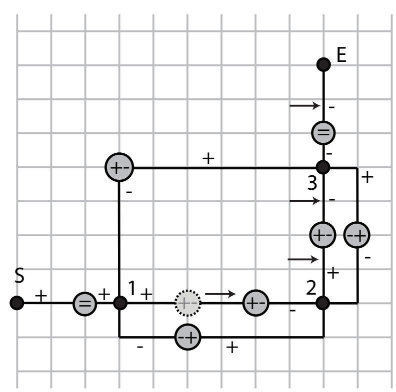
(b) Knot diagram interpreted as graph (black nodes are crossing nodes, gray nodes are segment nodes)



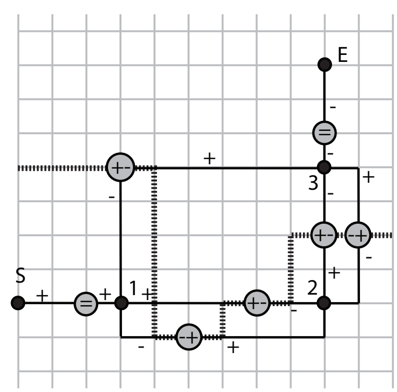
(c) Knot graph with edges assigned to layers, and labels on segment nodes



(d) Orthogonal version of knot graph (occlusions indicated with dashed boxes)



(e) Graph after expansion to remove occlusion



(f) Location of vertical cut through middle layer

Figure 2: Development of orthogonal structure for overhand knot

Next, we separate the graph into "upper" and "lower" layers by walking through the graph using the Gauss code. Each edge is adjacent to exactly one crossing or endpoint and belongs to either the "upper" layer or the "lower" layer. If an edge is adjacent to an over-crossing, that edge is in the upper layer; if it is adjacent to an under-crossing, it is in the lower layer. Each graph edge is labeled with '+' or '-' to indicate its layer (Figure 2(c)). For convenience, the start and end segments (adjacent to S and E in the figure) receive the same label as the next segments in from the start and end.

The segment nodes are further labeled based on whether the adjacent edges are ascending from the lower layer to the upper layer ('- +'), descending ('+ -'), or remaining in the same layer ('='). We can uniquely identify segment nodes by using the adjacent crossing node labels; for example, the node that is moved in Figure 2(e) is 1(+ -)2.

This graph is then converted into an orthogonal graph (Figure 2(d)). Tamassia showed that this can be done while preserving faces (the regions between graph edges) for four-planar graphs [28]; the knot graph indeed has vertices of no more than degree four by construction (each crossing is caused by intersection of at most two curves on the knot diagram, or can be separated into multiple crossings, so that crossing nodes are of no more than degree four). Tamassia presented a complete $O(n^2 \log n)$ algorithm for constructing an orthogonalization with the minimum number of bends; this bound was subsequently refined by Garg and Tamassia to $O(n^{\frac{7}{4}}\sqrt{\log n})$ [9].

To construct the fixture design, we now add a third dimension to the previously identified orthogonal grid, with two possible coordinates: top or bottom. Edges are placed at the appropriate coordinate based on their label. At ascending or descending segment nodes, we add a shaft edge from the top layer to the bottom layer. Now, all planar edges are located in either the top or bottom layers, and the middle layer contains only out-of-plane shaft edges. This is similar to a two-layer circuit board, with the shaft edges equivalent to vias.

Two pieces of the four-piece knot fixture act as caps for the top and bottom layers of the 3D orthogonal graph. These pieces can be removed by translating them up or down, respectively. The remaining two pieces slide together horizontally to form the middle layer. In order for these pieces to be separable, the middle layer must be arranged such that there is some projection direction in which no shaft edges occlude each other (in the example of Figure 2(d), the dashed boxes indicate the occluded nodes in both the x and y directions).

A simple complete algorithm for expanding the graph on the plane to remove occlusion, while retaining the same Gauss code, is as follows:

1. Pick a projection direction (x or y) with the fewest occlusions. With n shaft edges, this will always result in no more than $n/2$ occlusions. Without loss of generality, let the projection direction be along the y-axis.

2. For each pair of occluding edges (e_i, e_j) with $|x_j - x_i| < 2$:

    (a) Take all vertices with $x_v > x_j$, and set $x'_v = x_v + (2 - |x_j - x_i|)$. This expands the graph at x_j, shifting everything to the right of x_j one or two grid units (as needed) to the right. This shift preserves the graph regions (and the knot).

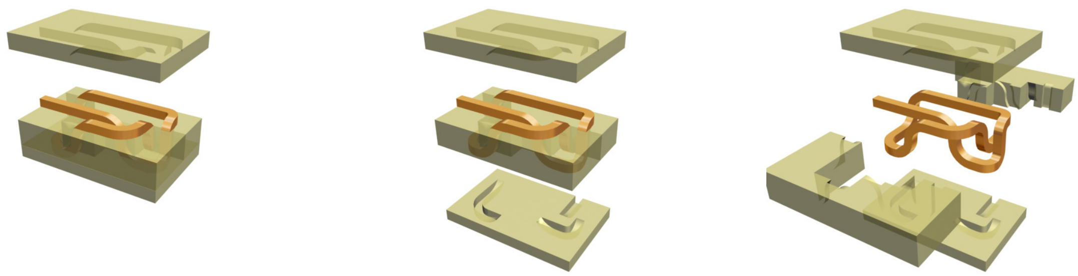
    (b) Shift e_j to the right ($x'_j = x_j + (2 - |x_j - x_i|)$).

    (c) If e_j was connected to an edge e_y parallel to the y-axis, keep e_y in place while shifting e_j, and add a new edge parallel to the x-axis between e_y and e_j (this new edge should be in the same layer as e_y).

In the example shown in Figure 2(e), only one node (1(+ -)2) needs to be moved to remove the occlusion in the y direction. The apparent occlusion between 2(+ -)3 and 3(=)E is not actually an issue, as the 3(=)E node remains in the same layer, and thus no shaft edge is present. We only need to consider removing occlusion between shafts.

At worst, the graph will expand by $n/2$ units in the x-axis during this algorithm. At step 2a, we can insert a check to see if it is possible to simply shift e_j without having to shift the entire graph right of x_j, which may reduce the amount of expansion if the graph was not initially compact.

This algorithm is correct and works for all graphs. Since each time everything to the right of x_j is moved, the movement will not create any new occlusions. Since every movement deletes one occlusion, the algorithm will remove occlusion for all graphs.



(a) Top (b) Bottom (c) Middle

Figure 3: Sequence of steps for taking apart a four-piece fixture

Once there is no occlusion along the projection direction, it is trivial to slice the middle layer to allow the two pieces to slide free, fully exposing the knot and allowing it to tighten. Given a list of shaft edges E_v, sorted by x coordinate, the slice between shaft edges e_k and e_{k+1} is given by the set of coordinates $\{(x_k, y_k), (\frac{x_k+x_{k+1}}{2}, y_k), (\frac{x_k+x_{k+1}}{2}, y_{k+1}), (x_{k+1}, y_{k+1})\}$. Figure 2(f) shows the location of the cut for the overhand knot.

Since this procedure of orthogonalizing the graph, splitting it into top and bottom pieces, expanding the orthogonalization, and slicing the middle layer of the fixture yields a four-piece fixture, we have proven Theorem 1.

Building the knot on an orthogonal grid with extra space between shaft edges guarantees that there will be enough room to round off the corners and also to allow string of non-zero thickness when building the fixture. Corners both in the plane and transitioning from the plane to the shaft direction can be replaced with arcs with grid unit radius. Thus, the maximum curvature is bounded exactly based on the choice of grid size. For a grid with spacing r, the maximum curvature is $\kappa = 1/r$.

Figure 3 shows an example fixture for simple overhand knot. The orange tube represents the shape of the string inside the fixture, and is not actually part of the fixture.

3 Experiments with four-piece fixtures

We can separate the task of tying knots using four-piece fixtures into *insertion* and *extraction* steps. Before insertion, we attach a rigid plastic cap attached to the leading end of the string (called a *slider*); pressurized air forces this cap through the fixture tube, pulling the string through the fixture. We designed the slider with a dome head, cylindrical body, and a flat back. This slider has a large surface area so as to be strongly impelled by the pressurized air; the cylindrical shape prevents flipping in the tube.

We designed fixtures for 7 knots ranging in complexity from the simple overhand knot to a *Carrick Bend* knot, and printed the fixtures on an OBJET Eden 250 Rapid Prototyper. Although the shape of the tube and the locations of the cuts are computed using the algorithm described above, the 3D model of the fixture was created by hand using SolidWorks.

Most of these knots tie two pieces of string together. Human-tied versions of the knots (because our fixtures arrange string in the desired topology, but do not tighten them), as well as the fixture tube shapes (different colors represent different strings), and the success rate over 100 insertion trials per fixture are shown in Table 1.

Figure 4(a) shows an example of a loose square knot arranged by a fixture. Insertion of string into each fixture is very fast (in all cases less than .5 seconds), since the string is driven by high-speed air pressurized at 80 psi.

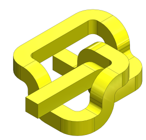
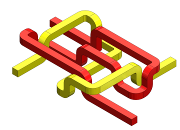
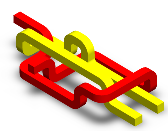
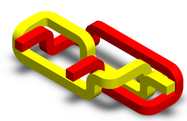
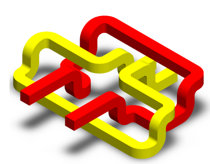
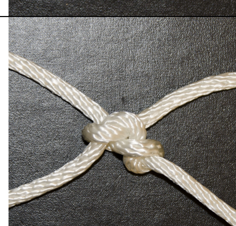
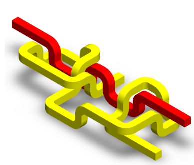
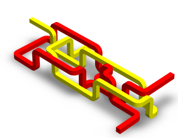
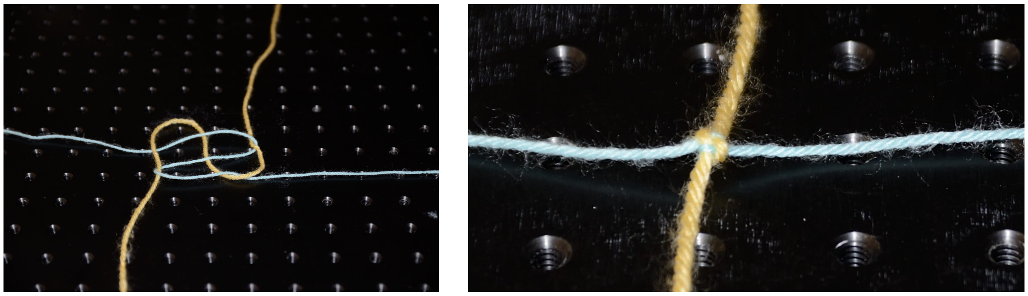
Knot type	Knot example	Tube shape	Success Rate
Overhand Knot			99%
Square Knot			99%
Bowline Knot			100%
Sheet Bend			100%
Strop Bend			100%
Harness Bend			95%
Carrick Bend			81%

Table 1: Knot example, designed tube shape, and corresponding success rate of 100 trials for each knot fixture. Different string colors represent different strands of string.



(a) Square knot loosely tied by fixture. (b) Square knot tightened by (human) pulling on string ends.

Figure 4: Square knot tied by fixture.

As seen in the table, most knot fixtures demonstrate a very high success rate, with the exception of the Carrick Bend, the most complicated knot considered. Fixtures other than the Carrick Bend have tube cross-section lengths between 7 and 8 mm, and overall fixture dimensions of less than 15 cm by 8 cm by 5 cm. Due to the otherwise large dimensions of the Carrick-bend fixture (>20 cm along one side, the maximum envelope size of the 3D printer), we chose a cross-section length of 4 mm for the Carrick Bend.

We believe that limited success with the Carrick Bend is due to the smaller cross-section of the tube (causing increased turbulence and pressure drop along the tube), to a smaller slider, and to increased air loss along fixture seams in longer tubes. In fact, the success rate with the Carrick Bend was initially close to zero due to these problems; the 81% shown in the figure is for a modified fixture with eight additional air inputs along the top and bottom sections of the tube.

The Carrick Bend certainly seems to be at the limit of complexity of knots that can be tied with our current approach using pressurized air for insertion. The simpler Harness Bend exhibits some similar difficulties along the more complex of its two tubes.

Extraction of the knot is simple using a four-piece fixture. We designed a mechanism composed of four-bar linkages with a single degree of actuation, shown in (Figures 5 and 6). The design is inspired by the mechanism used to open tackle boxes that fishermen store fishing equipment in. (Figures 5 and 6). Using a simple Dynamixel servo motor, opening the fixture takes about 1.5 seconds. Sealing the fixture with the current mechanism design, so that pressurized air can be used to insert string remains a challenge.

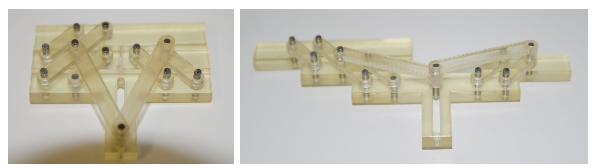


Figure 5: The opening mechanism for the four-piece fixture.

3.1 Limitations of four-piece fixtures

For even more complex knots, such as the River Knot shown in Figure 7, the approach described above is infeasible using a practically sized fixture.

There are at least two major difficulties: insertion, and tightening of the knot after extraction. We believe that the challenge of insertion may be solved in a straightforward manner, by changing the mechanism for actuating the string in the fixture. We will discuss the issue of tightening in the following section of the paper.

Pressurized air applied to body of the slider tends to pull the string along the tube. There is a nice additional effect of the pressurized air, which is to “float” the string along the tube, providing an air cushion that separates the string from the walls of the tube. However, as the tubes

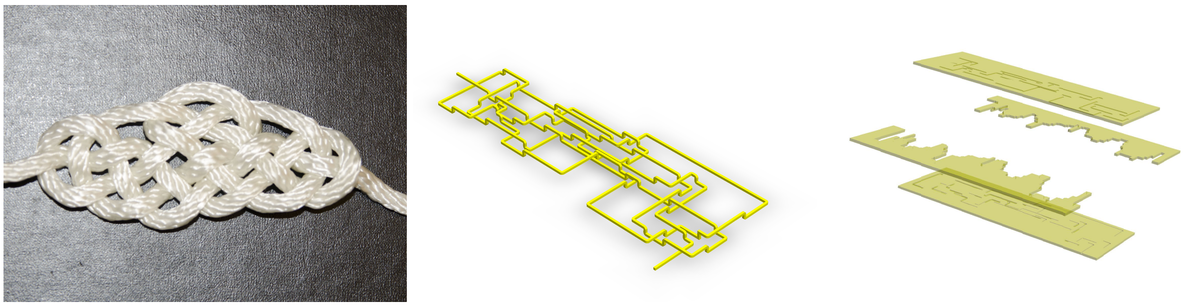


Figure 7: Human-tied river knot, and (unimplemented) four-piece fixture design.

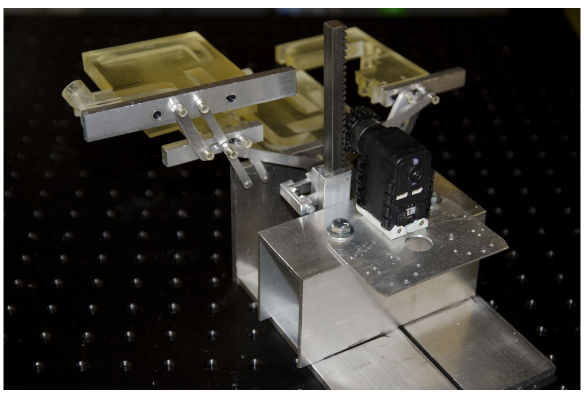


Figure 6: Machined complete fixture opener with fixture components attached.

get longer with greater numbers of turns, this beneficial effect diminishes.

We have therefore experimentally begun to explore the capabilities needed to pull string through a fixture tube. We have built a test structure, one module of which is shown in Figure 8; additional modules can be attached to lengthen the snake-like tube. This structure is not specific to any particular knot, but is rather used to test how many sharp bends a string can be impelled through using pressurized air. We consider three cases: forcing the string through the tube with pressurized air (and a slider), pulling the string through the tube, and pulling the string along a tube with ball-bearing rollers attached at turns.

Each modular section of the test structure contains a tube of length of about 47 cm. The tube has square cross-sections with edge length 7.5 millimeters. Each modular fixture contains twelve 90 degree turns.

With pressurized air, the string attached to a slider travelled through 150 cm on average over ten trials, with a maximum travel distance of 165 cm.

Pulling string without pressurized air was much less successful, demonstrating the strong positive benefit of the air cushion generated by air flow. Friction prevented motion of the string completely after about 25 cm; further force breaks the string.

Most of the friction force occurs at turns. We therefore installed rollers at each turn to reduce the friction and further test the pulling mechanism. The force required to pull the string through a tube of 190 cm is roughly equal to that required to lift a weight of 40 grams.

In spite of the promising results of using rollers to reduce friction of pulled string, the increased complexity of the fixture design is a disadvantage, and would presumably lead to larger fixture sizes. It is for this reason that our initial explorations make use of pressurized air.

4 Conclusions and Future Work

This paper presented a few approaches to designing fixtures for tying knots. Central among the contributions is a proof that any knot can be laid out in such a way that a four-piece fixture can be designed allowing easy extraction.

We have only begun to understand the behavior of flexible materials. While we know how to tie knots, we do not

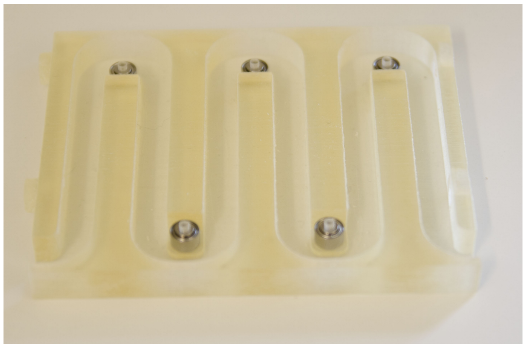


Figure 8: The modular test structure with rollers (silver ball bearings) installed at every turn.

have a good way to simulate the process, or to analyze fixture types to understand why some techniques work better than others. We also do not have a good way to design fixtures for knots, such as *hitch knots* that are attached to objects. A related challenge is how to design the form-factor of the fixture so as to have a small profile for particular applications (e.g., suturing) in which string is tied to objects.

This paper addressed only the problem of laying out a knot, without regards to the geometry of the tightened knot. An exciting future direction is understanding how knots should either laid-out for best tightening, or how knots may be tightened consistently from arbitrary loose geometries. This study would also allow extension of current ideas to *unknots*, such as the shoelace knot and bow ties, which are held together *only* by friction or other forces, and not by their topology. (Mathematically, unknots are topologically circles.)

Since the diameter of tubes in in a fixture is essentially fixed by the diameter of the string, the size of the fixture increases with the complexity of the knot. Large fixtures can be an engineering challenge. Better techniques for reducing friction and impelling the string through the fixture would perhaps allow somewhat smaller tube diameter.

Another future direction is the development of modular or more generic fixtures that easily allow tying of multiple types of knots.

5 Acknowledgments

Discussions with Yuliy Baryshnikov were instrumental in the development of four-piece fixtures. We also would like to thank Yu-Han Lyu and George Boateng for helpful discussion and feedback. This work was supported by NSF grant IIS-1217447.

References

[1] Matthew P. Bell. Flexible Object Manipulation. Technical Report TR2010-663, Dartmouth College, Computer Science, Hanover, NH, February 2010.

[2] Matthew P. Bell and Devin J. Balkcom. Knot tying with single piece fixtures. In *ICRA*, pages 379–384. IEEE, 2008.

[3] T. Bretl and Z. McCarthy. Mechanics and quasi-static manipulation of planar elastic kinematic chains. *Robotics, IEEE Transactions on*, 29(1):1–14, 2013.

[4] James Burns and Andrew Fung. Shoelace knot assisting device. US Patent No. 7044508, May 2006.

[5] Mark Champion. Knot tying device. US Patent No. 6817634, November 2004.

[6] Y. H. Chen. Determining parting direction based on minimum bounding box and fuzzy logics. *International Journal of Machine Tools and Manufacture*, 37(9):1189 – 1199, 1997.

[7] Yong Chen and David W. Rosen. A reverse glue approach to automated construction of multi-piece molds. *Journal of Computing and Information Science in Engineering*, 3(3):219–230, 2003.

[8] Amir Degani, Siyuan Feng, Howie Choset, and Matthew Mason. Minimalistic, dynamic, tube climbing robot. In *IEEE International Conference on Robotics and Automation*, pages 1100–1101, Anchorage, AK, May 2010.

[9] Ashim Garg and Roberto Tamassia. A new minimum cost flow algorithm with applications to graph

drawing. In Stephen C. North, editor, *Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 201–216. Springer, 1996.

[10] Minoru Hashimoto, Tomoaki Ichikawa, Shigeru Inui, and Yosuke Horiba. Dynamic manipulation of a string using a manipulator-parameter identification and control based on a rigid body link model. In *ICRA*, pages 4815–4820. IEEE, 2004.

[11] Wade Henning, Frank Hickman, and Howie Choset. Motion planning for serpentine robots. In *ASCE Space and Robotics*, Albuquerque, NM, 1998.

[12] John E. Hopcroft, Joseph K. Kearney, and Dean B. Krafft. A case study of flexible object manipulation. *Int. J. Rob. Res.*, 10(1):41–50, January 1991.

[13] H. Inoue and M. Inaba. Hand-Eye Coordination in Rope Handling. *Robotics Research: The first International Symposium*, pages 163–174, 1985.

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

[15] Rahul Khardekar, Greg Burton, and Sara McMains. Finding feasible mold parting directions using graphics hardware. *Computer-Aided Design*, 38(4):327 – 341, 2006. Symposium on Solid and Physical Modeling 2005.

[16] Liang Lu and S. Akella. Folding cartons with fixtures: a motion planning approach. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1570–1576 vol.2, 1999.

[17] Liang Lu and S. Akella. Folding cartons with fixtures: a motion planning approach. In *IEEE trans. Robot. Autom.*, volume 16(4), pages 346–356, August, 2000.

[18] Dinesh K. Pai. Strands: Interactive Simulation of Thin Solids using Cosserat Models. *Comput. Graph. Forum*, 21(3):347–352, 2002.

[19] Jeff Phillips, Andrew M. Ladd, and Lydia E. Kavraki. Simulated knot tying. In *Proc. IEEE International Conference on Robotics and Automation*, pages 841–846, May 2002.

[20] B. Ravi and M. N. Srinivasan. Decision criteria for computer-aided parting surface design. *Computer-Aided Design*, 22(1):11–18, 1990.

[21] Elon Rimon and Andrew Blake. Caging 2D bodies by 1-parameter two-fingered gripping systems. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1458–1464, Minneapolis, MN, April 1996.

[22] Samuel Rodríguez, Jyh-Ming Lien, and Nancy M. Amato. Planning motion in completely deformable environments. In *IEEE International Conference on Robotics and Automation*, pages 2466–2471, Orlando, FL, May 2006.

[23] David Rojas, Sayra Cristancho, Claudia Rueda, Lawrence E. M. Grierson, Alex Monclou, and Adam Dubrowski. The validation of an instrumented simulator for the assessment of performance and outcome of knot tying skill: A pilot study. In James D. Westwood, Susan W. Westwood, Li Felländer-Tsai, Randy S. Haluck, Helene M. Hoffman, Richard A. Robb, Steven Senger, and Kirby G. Vosburgh, editors, *MMVR*, volume 163 of *Studies in Health Technology and Informatics*, pages 517–523. IOS Press, 2011.

[24] M. Saha and P. Isto. Manipulation planning for deformable linear objects. *Robotics, IEEE Transactions on*, 23(6):1141–1150, 2007.

[25] 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.

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

[27] Wamis Singhatat. Intracorporeal knot tier. US Patent No. 6716224, April 2004.

[28] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, June 1987.

[29] R.H. Taylor and D. Stoianovici. Medical robotics in computer-integrated surgery. *Robotics and Automation, IEEE Transactions on*, 19(5):765–781, 2003.

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

[31] Hidefumi Wakamatsu and Shinichi Hirai. Static modeling of linear object deformation based on differential geometry. *International Journal of Robotics Research*, 23(3):293–311, March 2004.

[32] Hidefumi Wakamatsu, Tatsuya Yamasaki, Akira Tsumaya, Eiji Arai, and Shinichi Hirai. Dynamic modeling of linear object deformation considering contact with obstacles. In *Proc. of 9th International Conference on Control, Automation, Robotics, and Vision*, Singapore, December 2006.

[33] Huijuan Wang, Shuxin Wang, Jienan Ding, and Haifeng Luo. Suturing and tying knots assisted by a surgical robot system in laryngeal mis. *Robotica*, 28(2):241–252, 2010.