3D Thread Tracking for Robotic Assistance in Tele-surgery

Nicolas Padoy, Gregory D. Hager
The Johns Hopkins University, Baltimore, MD, USA
{padoy, hager}@jhu.edu

***Abstract*— Remote tele-manipulation tasks can be both long and exhausting. The operative workload can however be reduced through contextual systems, in which routine or dexterous actions are performed automatically. In this paper, we investigate this idea in tele-surgery by proposing *automatic scissors*, namely the possibility for a surgeon to invoke a third robotic arm to come and automatically cut the thread that he/she is holding. In particular, we address the problem of tracking deformable 3-dimensional (3D) curvilinear objects from stereo images. We propose an approach based on discrete Markov random field (MRF) optimization to track, in 3D, a thread modeled by a non-uniform rational B-spline (NURBS). We evaluate its accuracy off-line on synthetic and real data and illustrate its use for an *automatic scissors* command within an assistance system based on the da Vinci tele-surgical robot.**

I. INTRODUCTION

Many dexterous tasks involve the manipulation of deformable 3-dimensional (3D) curvilinear objects. Examples of such objects are sutures in surgery, catheters in interventional radiology and wires in maintenance tasks. Developing robotic assistance systems for such tasks requires the accurate localization and tracking of the curvilinear structures present in the scene. Contrary to the tracking of contours, which has been much addressed in the computer vision community [1], [2], [3], [4], [5], the tracking of purely curvilinear structures has received less attention. This is especially the case for open curves in a 3D setting. Related work in this area mainly comes from the medical imaging community, where vessels [6] or catheters [7], [8] need to be localized in angiographic images. In these cases, the displacements and deformations are constrained by the human anatomy. The free manipulation of an object like a thread is, however, less constrained and larger deformations can occur. Moreover, as opposed to contours, purely curvilinear objects do not have any stable side and are usually more flexible.

In this paper, we focus on tracking a thread in a surgical setting. Thread tracking can potentially be used for robotic assistance during an operation or for skill evaluation during the training of an operator. Typical surgical tasks involving thread manipulation are suturing and knot tying. Even though efforts towards robotic knot tying exist [9], [10], none of these approaches tracks the thread. This is indeed a very difficult problem, since the thread can deform in different directions with high speed and also undergo multiple occlusions. Our objective in this work is to track a thread to develop a simpler assistance primitive, namely *automatic scissors* during tele-operation. Supposing that the thread is held between two instruments, by recovering the deformation of the thread while the instruments are moved, a third robotic instrument equipped with scissors can automatically come and cut it. Such a command could for instance be triggered by voice command.

Fig. 1. System setup showing the four robotic arms of a da Vinci tele-surgical robot. Three arms hold instruments and the fourth arm (center) holds a stereo endoscopic camera.

Our approach is inspired by the recent method proposed in [8], which uses discrete optimization to track in 2-dimensional (2D) fluoroscopic images a catheter modeled by a B-spline. We are interested in a few properties of this method, namely that it is derivative-free, less sensitive to initialization and has a large capture range compared to gradient-based approaches.

We extend the method to 3D tracking from stereo and compare two discrete approximations of the energy for this problem. A stereo projection cost is used as external energy to recover the depth and the 3D curve is modeled with a non-uniform rational B-spline (NURBS). The projective invariance property of NURBS allows us to compute the 2D projected curves from the 3D curve (and vice-versa) by considering solely the control points. Also, tracking the extremities of the curve accurately proves to be challenging under a stereo system with small baseline, even though a length constancy assumption is used. For this reason, we introduce information about the positions of the instruments holding the thread in the energy function.

The tracking is first evaluated off-line with synthetic data where the ground truth is available and also qualitatively with real data. In both cases, the thread is undergoing large deformations. Finally, the approach has been implemented on a real system based on a non-commercial version of the da Vinci robot from Intuitive Surgical^{TM}, in order to demonstrate an *automatic scissors* command.

* Preprint, to appear in the IEEE/RSJ proceedings of IROS 2011 *

The remainder of this paper is as follows: section II describes the thread modeling and tracking, sec. III the experiments and sec. IV discusses and concludes this work.

II. Methods

A. Setup

The scene, containing the thread and the instruments, is observed with a stereo camera system, as shown in figure 1. The two cameras are assumed to be calibrated. In the following, we denote their projection matrices by $P_i \in \mathbb{R}^{3\times4}, i \in \{1,2\}$. The last row of these matrices is denoted by $P_i^{(3)}$. When the thread is manipulated by the robotic instruments, we also assume that approximate 3D positions of each instrument are available in the camera frame at every time step.

B. Thread modeling

We model the thread in 3D using non-uniform rational B-splines [11], for their projective invariance property. This modeling allows us to work conveniently with parameterizations of either the 3D thread or of its 2D projections. A NURBS curve $\mathcal{C}(\mathcal{Q},\mathcal{W},u)$ of degree d is defined as a linear combination of a set of control points $\mathcal{Q} = \{Q_k\}_{k\in\{1,K\}}$ with weights $\mathcal{W} = \{w_k\}$:

$$\mathcal{C}(\mathcal{Q},\mathcal{W},u) = \sum_{k=1}^{K} R_{k,d}(u)Q_k, \quad u \in [0,1], \tag{1}$$

where u is the curve parameter and $R_{k,d}$ are the rational basis functions [11]:

$$R_{k,d}(u) = \frac{N_{k,d}(u)w_k}{\sum_{i=1}^{K} N_{i,d}(u)w_i}, \quad u \in [0,1]. \tag{2}$$

The functions $N_{k,d}$ are the usual spline basis functions. If $\mathcal{C}(\mathcal{Q},\mathcal{W},u)$ represents a 3D curve, with $\mathcal{Q}$ a set of 3D points in homogeneous coordinates expressed as $Q_k^{3D} = [q_k\ 1]^{\mathrm{T}} \in \mathbb{R}^4$, the projective invariance property of NURBS is expressed as follows for $i \in \{1,2\}$:

$$\begin{cases} P_i(\mathcal{C}(\mathcal{Q},\mathcal{W},u)) &= \mathcal{C}(P_i(\mathcal{Q}),\mathcal{V}_i,u) \\ v_{i,k} &= w_k.P_i^{(3)} \begin{bmatrix} q_k \\ 1 \end{bmatrix}. \end{cases} \tag{3}$$

In other words, the projected 2D curves are the curves defined by the projected control points (see Fig. 2) and the appropriate weights $\mathcal{V}_i = \{v_{i,k}\}$. In the following, we name $\mathcal{C}^{3D}$ the 3D curve that models the thread with control point set $\mathcal{Q}^{3D}$. Its projection on the two images are called $\mathcal{C}_1^{2D}$ and $\mathcal{C}_2^{2D}$ with control point sets $\mathcal{Q}_1^{2D}$ and $\mathcal{Q}_2^{2D}$. We also use the notation $\mathcal{C}(u)$, omitting the control points and the weights for better readability.

C. Tracking approach

Tracking the thread implies recovering, at each time step, the optimal parameters of the curve $\mathcal{C}^{3D}$ so that its projections $\{\mathcal{C}_i^{2D}\}$ match the thread visible in the stereo input images I_1 and I_2. In order to deal with noisy images and to resolve ambiguities, additional regularization constraints are also enforced. In curve tracking, the computation of the maximum a-posteriori estimate of the parameters based on the information from the input images is usually reformulated as an energy minimization problem [8], as done below.

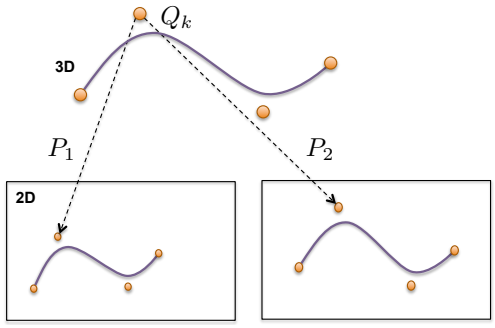


Fig. 2. Projection of the 3D NURBS curve on two images from the control points.

1) Energy: The energy is defined as a sum of an external energy term, also called data term, driving the curve to its position observed by the images, and of an internal energy term providing curve regularization:

$$E = E_{ext} + \lambda E_{int}. \tag{4}$$

$\lambda > 0$ is a parameter weighting the influence of the two terms. We define the external energy as a symmetric stereo projection error:

$$E_{ext} = \frac{1}{2}\sum_{i=1}^{2}\int_0^1 \tilde{I}_i(P_i(\mathcal{C}^{3D}(\mathcal{Q},\mathcal{W},u)))\mathrm{d}u \tag{5}$$

$$= \frac{1}{2}\sum_{i=1}^{2}\int_0^1 \tilde{I}_i(\mathcal{C}_i^{2D}(u))\mathrm{d}u \tag{6}$$

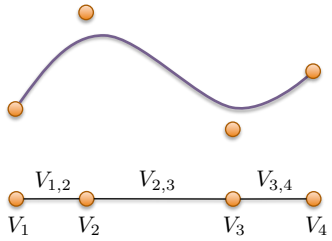
where $\{\tilde{I}_i\}$ are cost images penalizing projected curve points if they do not lie on the object. $\{\tilde{I}_i\}$ are obtained from the input stereo images and defined in section II-E. The internal energy maintains desired curve properties using curve derivatives, such as constant length using the initial curve at time 0 as reference:

$$E_{int} = \int_0^1 \left(1 - \frac{\|\mathcal{C}^{3D'}(u)\|}{\|\mathcal{C}^{3D'}_{ref}(u)\|}\right)^2 \mathrm{d}u. \tag{7}$$

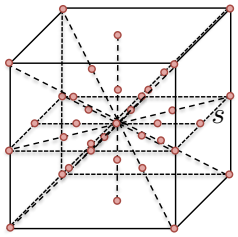
Since the spline modeling already provides curve smoothness, additional smoothness terms are usually not necessary.

2) Parameterization: The energy E needs to be optimized with respect to the set of control points $\mathcal{Q}^{3D}$ containing $3K$ parameters. Since the NURBS spline representation is redundant, we fix the weights w_k of the 3D curve to 1. But by updating the weights $v_{i,k}$ of the 2D curves according to eq. 3, one obtains a convenient parameterization of the 2D projections.

One should note that it is also possible to use a 2D-based parameterization, as done e.g. in [4] for contours. E would then have to be optimized over $4K$ parameters, namely $\mathcal{Q}_1^{2D}$ and $\mathcal{Q}_2^{2D}$. An additional energy term would be needed to enforce the stereo constraint. We have implemented this parameterization and noticed that even though the 2D spline



(a) NURBS curve with four control points and corresponding MRF chain below.



(b) Exemplary discrete label set L. The spheres indicate the allowed 3D displacements from the center.

Fig. 3. Modeling of the control point optimization using a discrete MRF formulation.

tracking may be accurate, additional 3D constraints need to be added to enforce temporal smoothness over the 3D control points. In the work below, we therefore focus on the 3D parameterization only.

D. Discrete optimization

We use a discrete optimization scheme [8] to optimize the continuous energy presented above. Each control point is associated with a discrete random variable that describes its space of allowed 3D local displacements. These displacements are then computed using MRF modeling and optimization.

1) MRF modeling: Let (G, E) be a graph with a finite set of nodes G and set of edges E. Let also L be a discrete set of labels representing the search space. $x \in L$ represents a unique 3D displacement, as illustrated in fig. 3(b). The nodes G correspond to the control points and the edges connect pairs of nodes to model their inter-dependencies. If we assume dependencies only between pairs of neighboring control points when evaluating the energy, the graph is a chain, as illustrated in fig. 3(a). With this interpretation, tracking the curve is formulated as finding a label assignment

$$\begin{array}{rcl} G & \longrightarrow & L \\ p \in G & \longmapsto & l_p \end{array}, \tag{8}$$

associating each control point with a 3D displacement, such that the energy E is minimized. A first order Markov random field (MRF) [12] solves such labeling tasks by modeling and approximating the energy as a sum of unary and pairwise potentials:

$$E_{mrf} = \sum_{p \in G} V_p(l_p) + \lambda_1 \sum_{(p,q) \in E} V_{pq}(l_p, l_q). \tag{9}$$

The unary potentials V_p evaluate the energy for each node p independently, while the pairwise potentials V_{pq} evaluate the energy for pairs of inter-dependent nodes (p, q). In a spline of degree d, a point of the curve is affected by $d+1$ control points. For exact computation of the energy E, one should therefore consider sets of nodes (cliques) of size $d+1$. This can be formulated using higher order MRF. The computational cost of optimization methods for higher order MRF is however prohibitive for our application. We therefore approximate the exact energy E of eq. 4 by considering only unary and pairwise potentials. Our experiments will show that such approximations yield good results in practice.

2) Energy approximations: We consider two different approximations of E, both considering interdependencies between pairs of successive control points, using an MRF chain as shown in fig. 3(a). The first approximation $E^{(1)}_{mrf}$ models the data term and the length constraint with unary potentials. A third pairwise term is used for regularization in addition to the intrinsic spline smoothness, as done in [13]. We found out in our experiments that this term improves the results when the inter-dependencies are neglected in the computation of the rest of the energy. The first approximation is expressed as

$$E^{(1)}_{mrf} = \sum_{p \in G} (V^{(1a)}_p(l_p) + \lambda_1 V^{(1b)}_p(l_p)) + \lambda_2 \sum_{(p,q) \in E} V^{(1c)}_{pq}(l_p, l_q), \tag{10}$$

with

$$V^{(1a)}_p(l_p) = \frac{1}{2} \sum_{i=1}^{2} \int_0^1 \alpha_p(u) (\tilde{I}_i(\mathcal{C}^{2D}_i(\{l_p\}, u))) \mathrm{d}u \tag{11}$$

$$V^{(1b)}_p(l_p) = \int_0^1 \alpha_p(u) \left(1 - \frac{\|\mathcal{C}^{3D'}(\{l_p\}, u)\|}{\|{\mathcal{C}_{ref}}^{3D'}(u)\|}\right)^2 \mathrm{d}u \tag{12}$$

$$V^{(1c)}_{pq}(l_p, l_q) = \|l_p - l_q\|. \tag{13}$$

The notation $\{l_p\}$ indicates that the p-th control point of the 3D curve is modified by the 3D displacement l_p. $\mathcal{C}^{2D}_i(\{l_p\}, u)$ is a point of the corresponding projected curve. The variables $\alpha_p(u)$ weight the influence of control point p over the curve point at position u and are obtained naturally from the basis functions:

$$\alpha_p(u) = R_p(u). \tag{14}$$

The second approximation $E^{(2)}_{mrf}$ models the data term and the length constraint with pairwise potentials:

$$E^{(2)}_{mrf} = \sum_{(p,q) \in E} (V^{(2a)}_{pq}(l_p, l_q) + \lambda_1 V^{(2b)}_{pq}(l_p, l_q)), \tag{15}$$

with

$$V^{(2a)}_{pq}(l_p, l_q) = \frac{1}{2} \sum_{i=1}^{2} \int_0^1 \alpha_{pq}(u) \tilde{I}_i(\mathcal{C}^{2D}_i(\{l_p, l_q\}, u)) \mathrm{d}u \tag{16}$$

$$V^{(2b)}_{pq}(l_p, l_q) = \int_0^1 \alpha_{pq}(u) \left(1 - \frac{\|\mathcal{C}^{3D'}(\{l_p, l_q\}, u)\|}{\|{\mathcal{C}_{ref}}^{3D'}(u)\|}\right)^2 \mathrm{d}u. \tag{17}$$

As above, the notation $\{l_p, l_q\}$ indicates that the p-th control point is modified by the displacement l_p and the q-th control point is modified by displacement l_q. $\mathcal{C}^{2D}_i(\{l_p, l_q\}, u)$ is a point of the corresponding projected curve and the variables $\alpha_{pq}(u)$ weight the influence of the two control points over the curve point at position u. They are defined by the *product model* from [8]:

$$\alpha_{pq}(u) = \frac{R_p(u) R_q(u)}{\sum_{k=1}^{K-1} R_k(u) R_{k+1}(u)}. \tag{18}$$

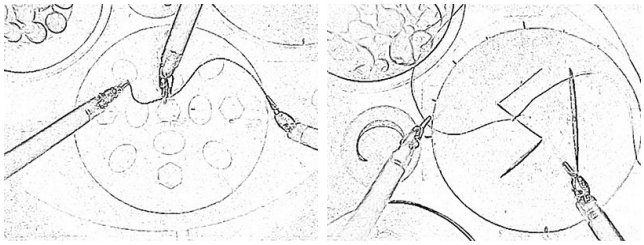


Fig. 4. Outputs of the curvilinear detector used to compute the cost images.

3) Optimization: To solve the aforementioned discrete MRF formulation, we use the FastPD algorithm [14]. This is a computationally efficient approach based on linear programming which has shown good real-time performance for 2D tracking [8]. In practice, note that $E^{(1)}_{mrf}$ is faster to optimize than the more precise energy $E^{(2)}_{mrf}$, since the pairwise evaluations are less time-consuming. The efficiency of the optimization is also driven by the sizes of G and L. Due to the large 3D search space, we use a sparse set of labels $L(r,s)$ sampling seven 3D directions (top-down, left-right, front-back and the four main cube diagonals). $L(r,s)$ depends on two parameters: r, the number of labels in each oriented direction and s, the 3D distance between two consecutive labels in the top-down direction. The possible 3D displacements for $r=2$ are indicated in fig. 3(b). $L(r,s)$ contain $(14r+1)$ labels that sparsely sample a cube of side length $(2rs)$ mm.

E. Cost images

The cost images $\{\tilde{I}_i\}_{i\in\{1,2\}}$ indicate the thread location with low intensities. We apply on the input images $\{I_i\}$ a curvilinear structure detector initially developed for vessel detection in medical images [15]. These curvilinear structures are detected by analyzing the Hessian matrix at each image locations. Outputs of this detector can be seen in fig. 4. Additionally, this detection is thresholded and processed with a Euclidean distance transform in order to create ridges with smooth borders along the detected curves.

F. Tracking of curve extremities

When the curve undergoes large movements, inaccuracies occur especially at the extremities. For instance, the tracked curve can slide along the real thread if its length was not properly initialized. This effect can also occur if the cost images are noisy at the extremities, e.g. due to the presence of the instruments holding the thread. Noisy areas in the cost images can also cause small local 3D deformations. The length constancy then implies that the tracked extremities do not reach the extremities of the real thread. To address these ambiguities and limit their effect, when the thread is held by the instruments or by the tissues, we require the extremities to remain close to their measured locations. Since these measures are approximate, we use the following soft constraint as unary energy term:

$$E_{tips} = \|\mathcal{C}^{3D}(0) - T_0\|_\epsilon + \|\mathcal{C}^{3D}(1) - T_1\|_\epsilon \ , \qquad (19)$$

where T_0 and T_1 are the measured extremity locations. $\|x\|_\epsilon$ is 0 if $\|x\| \leq \epsilon$ and the usual norm $\|x\|$ otherwise.

III. Experiments

In this section, we test the approach on synthetic and real data and compare the two energy approximations $E^{(1)}_{mrf}$ and $E^{(2)}_{mrf}$ defined in section II-D.2. The thread is telemanipulated using a da Vinci surgical robot and observed by a stereo endoscopic camera (Ikegami HD), as shown in fig. 1. In the following experiments, the curves are cubic splines with a fixed number of eight control points. In the energies, we only use a large weight λ for the length constancy constraint. We assume the splines to be initialized at the beginning of the tracking, for instance by providing the extremity locations and running a few iterations of the optimization on a spline initially defined as a straight line. In all experiments, we perform two optimization steps with two label sets $L(r,s)$ and $L(r,\frac{s}{2})$ at each image frame, in order to refine the tracking.

A. Synthetic experiments

We apply the algorithm to synthetic images in which the ground truth position of the thread is known. To generate a meaningful thread deformation pattern, we first track a thread undergoing various transformations from a real sequence. The synthetic data is then obtained by projecting at each time step this thread pattern on two stereo images containing background clutter (similar to fig. 5(a)). The original sequence contains 1000 frames. In order to evaluate the algorithm when larger deformations between two consecutive time steps are present, we also generate 4 other sequences with respective lengths 500, 333, 250 and 200. These sequences replay the original one with acceleration factors 2-5 by skipping frames.

We compare the two energy approximations from section II-D.2 and compute both the 3D and 2D accuracies. The accuracy is the average curve distance between the computed curve and the ground truth (gt) template curve:

$$\text{3D accuracy} = \Delta(\mathcal{C}^{3D}, \mathcal{C}^{3D}_{\mathbf{gt}}) \qquad (20)$$

$$\text{2D accuracy} = \frac{1}{2}\left(\Delta(\mathcal{C}^{2D}_1, \mathcal{C}^{2D}_{1,\mathbf{gt}}) + \Delta(\mathcal{C}^{2D}_2, \mathcal{C}^{2D}_{2,\mathbf{gt}})\right) \qquad (21)$$

with

$$\Delta(\mathcal{C}, \mathcal{C}_{\mathbf{gt}}) = \int_0^1 \mathrm{d}_{min}(\mathcal{C}(u), \mathcal{C}_{\mathbf{gt}})du \ , \qquad (22)$$

and d_{min} is the usual Euclidean distance between a point and a curve. The thread extremities are assumed to be known from the ground truth, as we are interested in evaluating robustness to deformations. For this reason, it is not necessary to use a symmetric curve distance in place of Δ. Results are shown in table I, which also indicates the 3D and 2D deformations undergone by the thread template. These deformations are indicated by the average and maximum displacements of knot points between two consecutive time steps.

Approximation $E^{(2)}_{mrf}$ yields better results as it defines a better approximation of the global energy. The 2D accuracies

Accel.	Deformations				Accuracies			
					Approx. $E^{(1)}_{mrf}$		Approx. $E^{(2)}_{mrf}$	
	3D (mean)	3D (max)	2D (mean)	2D (max)	3D	2D	3D	2D
1	0.03	0.40	0.76	8.10	0.47 ($\pm$0.21)	2.43 ($\pm$1.90)	0.42 ($\pm$0.19)	1.21 ($\pm$1.11)
2	0.05	0.37	1.17	8.39	0.47 ($\pm$0.19)	2.55 ($\pm$2.23)	0.47 ($\pm$0.22)	1.12 ($\pm$1.05)
3	0.06	0.40	1.53	10.37	0.45 ($\pm$0.20)	2.54 ($\pm$2.42)	0.52 ($\pm$0.21)	1.16 ($\pm$0.98)
4	0.08	0.50	1.87	13.21	0.83 ($\pm$0.42)	6.77 ($\pm$5.90)	0.52 ($\pm$0.19)	1.17 ($\pm$0.91)
5	0.09	0.51	2.20	13.80	0.84 ($\pm$0.37)	4.50 ($\pm$3.40)	0.69 ($\pm$0.25)	2.14 ($\pm$2.69)

TABLE I

MEAN 3D AND 2D TRACKING ACCURACIES ON SYNTHETIC DATA WITH STANDARD DEVIATIONS. THE 3D VALUES ARE GIVEN IN MM, THE 2D VALUES IN PIXELS. ACCEL. IS THE FACTOR BY WHICH THE SEQUENCE IS ACCELERATED BY DROPPING FRAMES. THE DEFORMATIONS INDICATE THE MEAN AND MAXIMUM KNOT DISPLACEMENTS BETWEEN TWO CONSECUTIVE FRAMES IN THE GENERATED SYNTHETIC SEQUENCES.

are especially better. This is mainly due to the fact that small 3D errors can yield large 2D errors, because of the small baseline (5mm) of the stereo camera and sometimes noisy images. One also verifies that the errors increase with higher speed and larger knot displacements. Here, we use a large label set ($s = 16$) allowing a maximum displacement of a projected 2D knot of approximately 14 pixels. One also notes that the 3D accuracies are similar for both approaches. This is an interesting result since approximation $E^{(1)}_{mrf}$ is computationally more efficient, as mentioned in sec. II-D.3. It can then be used in online experiments.

B. Experiments on real data

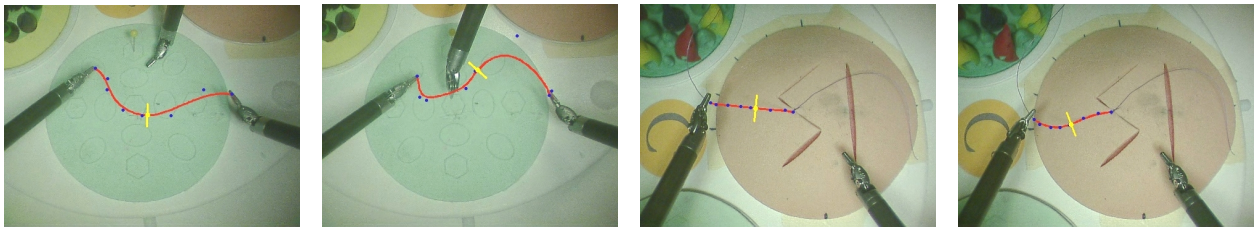
In this experiment, we recorded two sequences of 1544 and 561 frames. In the first sequence, two instruments hold a thread which is deformed by the first instrument and by a third instrument which grasps it, also causing occlusions. In the second sequence, a knot has been tied and one instrument manipulates one side of the thread. These two sequences are illustrated in figure 5(a). For the first sequence, we use the constraint of section II-F. In the second sequence, we use this constraint for one extremity only and keep a fixed position for the other extremity at the knot location. We experimented with approximations $E^{(1)}_{mrf}$ and $E^{(2)}_{mrf}$ and different label sets. Like on the synthetic data, $E^{(2)}_{mrf}$ provides smoother and better 2D results. In both approaches, the tracked thread extremities happen to slide of a few pixels along the thread during large displacements. This effect is caused by unpenalized small 3D deformations due to noisy cost images. It is limited by the extremity constraint, but also reinforced by the limitations of our acquisition system for this experiment: in addition to the short baseline, images of reduced sizes are used. The images are down-sampled by a factor of 3 from an original high definition (HD) resolution (1920×1080). A better tracking of the thread extremities would reduce this tracking ambiguity. Finally, we also notice that the tracking is robust to the 3rd instrument occlusions in the first sequence. Excerpts of the tracking are displayed in fig. 5 and the complete tracking sequences can be seen in the illustrative video.

C. Automatic scissors

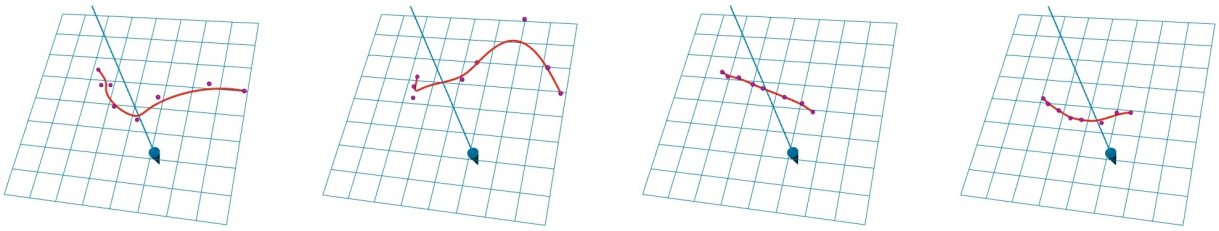
Supposing that the thread is held between two instruments, we implement an *automatic scissors* command for tele-surgery. This command invokes the 3rd instrument to come and cut the thread at its detected center. It is implemented in an assistance system based on a da Vinci robot, which is used exclusively for research purposes. The exact deformations of the thread need to be tracked to make sure that the cut will occur at the right place, since the instrument may move and change the thread shape before the cut takes place. In this experiment, we use a label set with $s = 4$ and the faster approximation $E^{(1)}_{mrf}$. With this parameter, our current single-threaded implementation runs at 5fps (including image processing) on an Intel Xeon 2.4GHz core and at 2fps (including HD frame grabbing and processing of half-size images) on the demonstration computer connected to the robot, which runs with a core at 2.0GHz. For this particular application where the thread undergoes slow deformations, the thread can be successfully tracked and cut, since the low framerate is compensated by the larger capture range provided by the label set. Excerpts of the cutting are shown in fig. 6 and full sequences are shown in the demonstration video.

IV. DISCUSSION AND CONCLUSION

To develop robotic assistance systems that require automatic thread manipulation, the exact position of the thread needs to be known. A thread is however a highly deformable curvilinear structure, which can undergo large deformations and displacements. In this paper, we propose an approach based on discrete optimization for thread tracking from stereo. We model the 3D thread as well as its 2D projections with non-uniform rational B-splines. We then embed the parameter optimization in a discrete Markov random field optimization framework that provides a large tracking capture range. Two approximations of the global energy are compared in synthetic experiments. We also show that accurate results can be obtained on real data. In particular, we demonstrate our approach for an *automatic scissors* command during tele-surgery. This experiment shows very promising results, since the thread can be tracked and cut under a low tracking framerate.



(a) Reprojected spline $\mathcal{C}$ with control points and indication of the center.



(b) Different 3D view of spline $\mathcal{C}$.

Fig. 5. Illustration of the tracking on two sequences. A complete performance can be seen in the illustrative video.

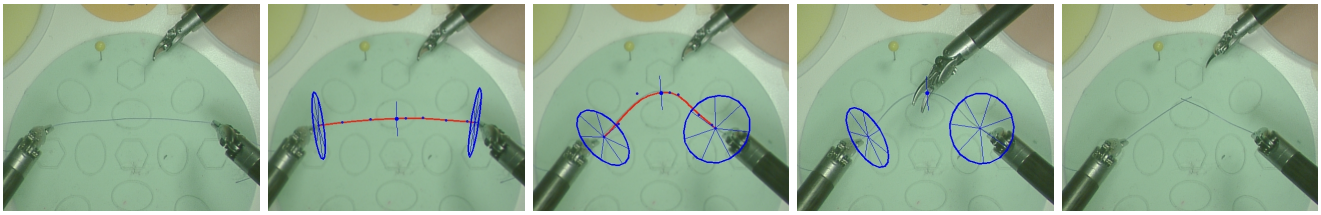


Fig. 6. Example of automatic thread cutting by the 3rd instrument with the *automatic scissors* command.

In the experiments, we use thin threads of 6-8cm. The use of longer threads will generate larger deformations, possibly including loops. In future work, we plan to address these situations and to perform further experiments on endoscopic images. We also plan to investigate the use of non-uniform discrete sets of labels to fasten the optimization and better discretize the search space. Finally, fine manipulation tasks are usually constrained by the task objective and the physical environment. Prior on the deformations could therefore be learned from several recordings of the same task and be used to further improve the tracking accuracy and fasten the optimization.

V. Acknowledgments

The authors would like to thank Simon DiMaio from Intuitive Surgical™ and Anton Deguet from the Johns Hopkins University for their support with the Intuitive Research API. They also would like to thank Ben Glocker for providing his MRF optimization framework. This work is funded by NSF grants CPS-0931805. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Kass, A. P. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV*, vol. 1, no. 4, pp. 321–331, 1988.

[2] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *PAMI*, vol. 17, no. 3, pp. 294–302, 1995.

[3] A. A. Amini, R. W. Curwen, and J. C. Gore, "Snakes and splines for tracking non-rigid heart motion," in *ECCV*, 1996, pp. 251–261.

[4] T.-J. Cham and R. Cipolla, "Stereo coupled active contours," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, p. 1094, 1997.

[5] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.

[6] G. Shechter, F. Devernay, È. Coste Manière, and E. Mcveigh, "Temporal tracking of 3D coronary arteries in projection angiograms," in *Medical Imaging*, vol. 4684. SPIE, 2002.

[7] P. Wang, T. Chen, Y. Zhu, W. Zhang, S. K. Zhou, and D. Comaniciu, "Robust guidewire tracking in fluoroscopy," in *CVPR*, 2009, pp. 691–698.

[8] T. H. Heibel, B. Glocker, M. Groher, N. Paragios, N. Komodakis, and N. Navab, "Discrete tracking of parametrized curves," in *CVPR*, 2009.

[9] H. G. Mayer, F. J. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," in *IROS*, 2006, pp. 543–548.

[10] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *ICRA*, 2010, pp. 2074–2081.

[11] L. Piegl and W. Tiller, *The Nurbs Book*, 2nd ed. Springer, 1997.

[12] S. Z. Li, *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., 2001.

[13] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, "Dense image registration through mrfs and efficient linear programming," *Medical Image Analysis*, vol. 12, no. 6, pp. 731–741, 2008.

[14] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, approximately optimal solutions for single and dynamic mrfs," in *CVPR*, 2007.

[15] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Muliscale vessel enhancement filtering," in *MICCAI*, 1998, pp. 130–137.