

Deformable Tracking of Textured Curvilinear Objects

Nicolas Padoy
padoy@jhu.edu

Gregory Hager
hager@jhu.edu

Johns Hopkins University
Baltimore, Maryland, USA

Abstract

The evaluation and automation of tasks involving the manipulation of deformable curvilinear objects, such as threads and cables, requires the real-time estimation of the 3D shapes of these objects from images. This estimation is however extremely challenging due to the small amount of available visual information, the inherent geometric ambiguities, and the large object deformations.

We propose an approach for tracking deformable curvilinear objects using solely visual information from one or more calibrated cameras. The key idea is to formulate the shape estimation as a deformable 1D template tracking problem. The object is first textured with a pattern of different alternating colors. The tracking problem is then expressed as an energy minimization over a set of control points parameterizing a 3D NURBS modeling the object. Assuming the object's in-extensibility, we propose a novel energy based on a texture-sensitive distance map.

We demonstrate the benefits of this energy in synthetic and real experiments, using data illustrating the deformation and manipulation of a thread with a da Vinci robot. In particular, we show that the approach allows for deformable tracking in the absence of normal motion along the curve, a challenging practical situation that occurs when the thread is dragged by one extremity.

1 Introduction

Threads and wires are deformable 3-dimensional (3D) and curvilinear objects which are commonly manipulated by humans in various medical and manufacturing tasks. Several applications, including computer-assisted evaluation, augmented reality guidance, and autonomous robotic manipulation [12, 15, 20] would benefit from the real-time estimation of the 3D shapes of these deformable objects from images. This estimation is however challenging due to multiple factors: 1) little information is available within an image to visually detect and distinguish a curvilinear object due to its thin and usually uniform appearance; 2) different 3D shapes may lead to the same visual perception, even in a stereo setting in case portions of the objects lie in an epipolar plane; and 3) the motions and deformations can be large, depending on the stiffness of the object. Additionally, one desired property of the tracking in the aforementioned applications is the possibility to track consistently specific points along the object, such as the extremities or the mid-point.

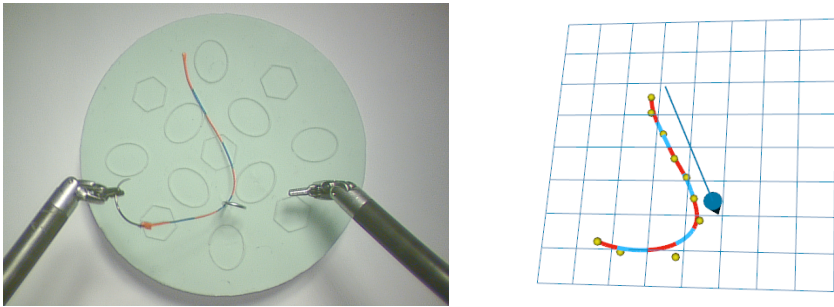


Figure 1: Two da Vinci instruments manipulating a thread textured with a color pattern (left) and corresponding 3D tracking model (right).

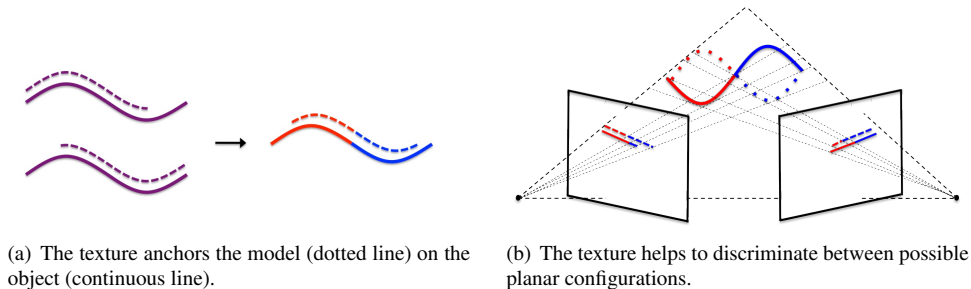
Common curvilinear objects usually have a uniform appearance. Even though different constraints, like in-extensibility and motion continuity, can be used to estimate the shape more tractably, the tracking of a curvilinear object with uniform appearance suffers from visual ambiguities, such as projective ambiguities or drift along the curve. In particular, the curvilinear property of the object is not a sufficient cue, as the environment in which the object is manipulated often contains multiple edges. The solution proposed in this paper is to texture the object with a multi-color pattern, namely a one-dimensional template. Using more than a single color makes the tracking more robust to visual ambiguities, as illustrated in fig. 2. Moreover, such a texture is coarse enough to be reliably detected on a curvilinear object, contrary to feature points.

In the proposed approach, we model the deformable object directly in 3D with a rationale non-uniform B-spline (NURBS) and formulate the tracking problem as an energy minimization over the spline control points. We propose an energy containing three terms to enforce the curvilinear and texture appearances of the object, while maintaining an approximate arc-length parameterization. The advantage of using a single 3D model, as opposed to 2D models coupled by the epipolar geometry, as used in [5] for contour tracking, is the immediate incorporation of natural 3D constraints, such as length constancy. Additionally, using the projective property of NURBS, the 2D splines can be directly computed from the 3D control points using the known projection matrices. The approach also generalizes to an arbitrary number of images.

To enforce the texture appearance, minimizing the sum of square differences between the texture template and the image (*direct tracking*), as for instance done for tracking deformable surfaces [18], seems a natural choice. Such a term is however not robust to larger motions due to the thin structure of the object. The convergence radius is small and using image smoothing to increase the radius causes the object's texture to become lost in the process. For this reason, we propose to define the texture appearance term from a texture sensitive distance map, indicating for each texture element the distance to the closest matching image pixel.

Our contribution is twofold: 1) we tackle a challenging problem that is needed by the robotics community but usually circumvented due to its difficulty; 2) we propose and study a novel approach based on 1D textures.

The main application targeted in this research is suture tracking during tele-surgery, for surgical skills evaluation and robotic automation. Without loss of generality, we therefore focus in this paper on tracking a thread which is remotely manipulated by the instruments of a da



(a) The texture anchors the model (dotted line) on the object (continuous line).

(b) The texture helps to discriminate between possible planar configurations.

Figure 2: Example of ambiguities resolved by the use of a 1D texture on the object.

Vinci robot as shown in fig. 1.

The remainder of this paper is as follows: section 2 describes the related work, and section 3 describes the thread modeling and tracking approach. Section 4 presents comparisons and experiments on synthetic and real data. Section 5 discusses and concludes this work.

2 Related work

Methods for robotic automation of tasks that involve the manipulation of a thread, such as knot tying, have been presented in [14, 21]. These works, however, do not perceive the thread. Instead, variability in the thread location and the need for perception are suppressed either by the use of pre-determined initial conditions [21] or by the use of a third arm [14].

In [10], the physical model proposed in [9] for graphical simulation of flexible rods is used to estimate the static shape of a thread in an engineered environment. The authors first estimate the physical parameters, such as twisting and bending coefficients. The 3D shape of the thread is then iteratively reconstructed edge by edge by starting from one given extremity and using a precise image segmentation. In spite of the great interest of this modeling for automatic manipulation, it is currently difficult to use such an approach for tracking during real thread manipulation. An immediate challenge is to detect and model the physical contacts between the thread and the environment, that affect the physical modeling. Additionally, the physical properties of the thread can change during its manipulation due to the damaging effects of the graspers. In previous work [16], a 3D thread tracking approach has been developed for automatically cutting a thread using a robotic tool. The approach is based on discrete optimization and only makes use of ridge information from the images, but requires constraints on the locations of the thread extremities that are provided by the tool positions.

Different approaches have been proposed in the computer vision community to track contours using snakes or B-splines within 2D images or 3D volumetric data [1, 7, 9, 10]. These approaches usually use edge information or orthogonal profiles to attract the curve to the nearest contour. Shape priors can also be included [9] to constrain the deformations and make the 2D tracking robust to clutter. Contours differ however from pure curvilinear objects, because they have a stable visual side that provides more visual information. Additionally, they do not deform as much as a thread does. It should also be noted that these approaches do not evaluate whether specific points can be tracked consistently along the contours, but only focus on the shape. The tracking of 2D deformable surfaces has been studied in [2, 18]. But compared to a deformable thread, such objects provide more visual information and often less deformations.

The closest application to ours comes from the medical community, where the tracking of purely curvilinear objects such as vessels or catheters is necessary. [14] track vessels and [8, 21] track catheters in 2D angiographic images. In addition to the differences in setups and image modalities, since the physical displacements are constrained by human anatomy, the deformations are also different from those generated when manipulating a thread.

3 Methods

In the following section, we assume that the thread is observed by two calibrated cameras with known projection matrices P_1 and P_2 . The two corresponding images are denoted by I_1 and I_2 . Note that the approach described below does not depend on the second camera, as long as a 3D initialization of the tracking is provided.

3.1 Thread modeling

We model the thread using a low dimensional parameterization based on a NURBS spline representation [21]. NURBS have been used successfully for static curve reconstruction [13, 22] and are also useful for 3D curve tracking due to their projective invariance [16]. The 3D curve modeling the thread is denoted by $\mathcal{C}^{3D}(\mathcal{Q}, u)$, where \mathcal{Q} is a set of n 3D control points and u is the curve parameter:

$$\mathcal{C}^{3D}(\mathcal{Q}, u) = \sum_{i=1}^n R_{i,d}(u) Q_n, \quad u \in [0, 1]. \quad (1)$$

$R_{i,d}$ are the rationale spline basis functions. NURBS curves allow weights for the control points. They are set to 1 in the 3D curve to reduce the dimensionality of the estimation. The 2D spline projections are then denoted by \mathcal{C}_1^{2D} and \mathcal{C}_2^{2D} and directly defined from the projected 3D control points by using suitable weights [16, 17, 22].

The color pattern texturing the thread is composed of alternating colors and represented by a general function associating the curve parameter u to its color $c(u)$:

$$c(u) : u \in [0, 1] \longrightarrow S,$$

where S is a color space. For generality, we do not require the two cameras to possess the same color-calibrations, but maintain instead two representations of the texture by using two functions: c_i with $i \in \{1, 2\}$. The pattern representation is supposed to be known in advance or to be learned from the first image. Figures 2(a) and 2(b) illustrate how texturing the object with a color pattern provides additional constraints that address the drift and projective ambiguities mentioned in the introduction. Combined with length constancy and smoothness constraints, the texture addresses most of the ambiguities in practical situations, as long as the pattern is chosen such that the maximum length of each color segment is small enough so as not to allow for multiple bendings of the thread within the segment.

It is important to note that, for consistency, c and \mathcal{C}^{3D} need to be defined using the same curve parameterization. Under the in-extensibility assumption, we choose to use an arc-length parameterization. Because rationale curves cannot be approximated exactly by arc-length, except for lines [2], we use an approximate arc-length parameterization, as explained below.

3.2 Energy

The tracking problem is formulated as an energy minimization over the spline parameters \mathcal{Q} . The energy E is defined as the sum of two terms: $E = E_{ext} + E_{int}$, where the first term is a data term enforcing the curvilinear and texture appearances. The second term enforces the in-extensibility constraint. We define the data term as

$$E_{ext} = \frac{1}{2(K+1)} \sum_{i=1}^2 \sum_{k=0}^K \left(\alpha \mathcal{D}_i^{tub} (\mathcal{C}_i^{2D}(u_k))^2 + \beta \mathcal{D}_i^{tex} (\mathcal{C}_i^{2D}(u_k), u_k)^2 \right), \quad (2)$$

where $\mathcal{D}_i^{tub}(x)$ is a distance map indicating the distance to the closest ridge from position x in image I_i and $\mathcal{D}_i^{tex}(x, u)$ is a texture sensitive distance map indicating the distance to the closest pixel with color $c_i(u)$. (u_1, \dots, u_K) defines a uniform sampling of the parameters and α, β are weights balancing the two terms.

An additional regularization term enforces the in-extensibility constraint. We noticed in our experiments that a global length constancy constraint forcing $(1 - \int_0^1 \|\mathcal{C}^{3D'}(u)\| du / L_{ref})^2$ to zero, where L_{ref} is the initial length, and used in combination with the aforementioned appearance constraints, fails to maintain proper parameterization during deformations. This is due to the coarse texture information. We therefore introduce a term that penalizes both local changes of length and local changes of arc-length parameterization:

$$E_{int} = \frac{\gamma}{K} \sum_{k=0}^{K-1} \left(1 - \frac{\int_{u_k}^{u_{k+1}} \|\mathcal{C}^{3D'}(u)\| du}{L_{loc}} \right)^2, \quad (3)$$

with $L_{loc} = L_{ref}/K$ and γ being a weighting coefficient. This new term also allows for faster tracking: it is not required to recompute at each iteration the arclength parameterization nor the values of the basis functions at the sampling points, which are also defined from the arclength parameterization.

Minimizing the energy E is a non-linear least-squares problem with respect to the 3D control points \mathcal{Q} . In our experiments, we use the Levenberg-Marquardt approach for minimization. When using a pair of stereo images, the Jacobian obtained by stacking the partial derivatives is of size $(5K+4) \times 3n$. The discrete functions \mathcal{D}_i^{tub} and \mathcal{D}_i^{tex} are stored as images, as explained below, and their derivatives are computed using Sobel filters and image interpolation.

3.3 Distance maps

To compute $\mathcal{D}_i^{tub}(x)$, a ridge detector suitable for a tubular object is first applied on the image I_i . We use a Frangi filter [Frangi], which analyses the images' Hessian structure and enables precise localization of the ridge center. The ridges are then segmented using a user-defined threshold and $\mathcal{D}_i^{tub}(x)$ is defined by the distance map over the segmented ridges.

$\mathcal{D}_i^{tex}(x, u)$ is computed from automatic analysis of the function $c(u)$ to generate an appropriate color model of the pattern. The one-dimensional texture is first clustered into consistent color clusters $(b_r)_{1..R}$ using Gaussian mixture models. During tracking, the different texture colors are segmented in the current images using the computed Gaussians. $\mathcal{D}_i^{tex}(x, u)$ is obtained from distance maps computed over the R segmentations of I_i that correspond to the R different color clusters of the pattern. To ensure robustness to small occlusion and illumination changes, morphological closing is applied on the segmentations before the computation of the distance maps.

3.4 Geometric optimization

Due to the large deformations encountered by the thread, it may be necessary of adapt the number of knots/control points of the spline during the tracking. The employed strategy evaluates an approximate spline curvature using the control points and adds or removes control points using an approach similar to the one presented in [10] for the physical simulation of a thread.

4 Experiments

4.1 Synthetic experiments

We present in this section several experiments that demonstrate the usefulness of the texture. The first experiments analyze the convergence of the approach and the influence of the size of the color stripes. In particular, they show that the texture reduces the tracking error in the presence of larger displacements. The last experiment shows that the texture allows for accurate tracking in situations where the thread is dragged by one end and does not have any motion normal to the curve.

4.1.1 Definition of the tracking error

Most works in curve tracking evaluate the error by using a distance between the computed curve and a ground truth (gt) curve [8, 11]. Such errors, that we call *curve errors*, can be defined symmetrically as

$$e_{\text{crv}}^{3D} = \Delta(\mathcal{C}^{3D}, \mathcal{C}_{\text{gt}}^{3D}) \quad \text{and} \quad e_{\text{crv}}^{2D} = \frac{1}{2} \sum_{i=1}^2 \Delta(\mathcal{C}_i^{2D}, \mathcal{C}_{i,\text{gt}}^{2D}) \quad (4)$$

with

$$\Delta(\mathcal{C}, \mathcal{C}_{\text{gt}}) = \frac{1}{2} \left(\int_0^1 d_{\min}(\mathcal{C}(u), \mathcal{C}_{\text{gt}}) du + \int_0^1 d_{\min}(\mathcal{C}_{\text{gt}}(u), \mathcal{C}) du \right), \quad (5)$$

where d_{\min} is the usual Euclidean distance between a point and a curve. These curve errors are not fully suitable for thread tracking and manipulation, because they do not properly evaluate the tracking of specific points. Supposing the thread is in-extensible, in the case of tracking a suture composed of a needle and of a thread, one is interested in tracking points at specific distances behind the needle, so that they can for instance be used to grasp the thread. We therefore define two *arclength errors* e_{acl}^{2D} and e_{acl}^{3D} based on r specific points from the thread uniquely defined by their arclength parameters $\{v_k | 1 \leq k \leq r\}$:

$$e_{\text{acl}}^{3D} = \frac{1}{r} \sum_{k=1}^r \|\mathcal{C}^{3D}(v_k) - \mathcal{C}_{\text{gt}}^{3D}(v_k)\|, \quad e_{\text{acl}}^{2D} = \frac{1}{2r} \sum_{i=1}^2 \sum_{k=1}^r \|\mathcal{C}_i^{2D}(v_k) - \mathcal{C}_{i,\text{gt}}^{2D}(v_k)\| \quad (6)$$

In the following experiments, we use three points defined by $v_1 = 0.25, v_2 = 0.5, v_3 = 0.75$. Since the thread extremities are only constrained by one side, we exclude them from this evaluation to clearly show the advantage of the pattern for points inside the thread.

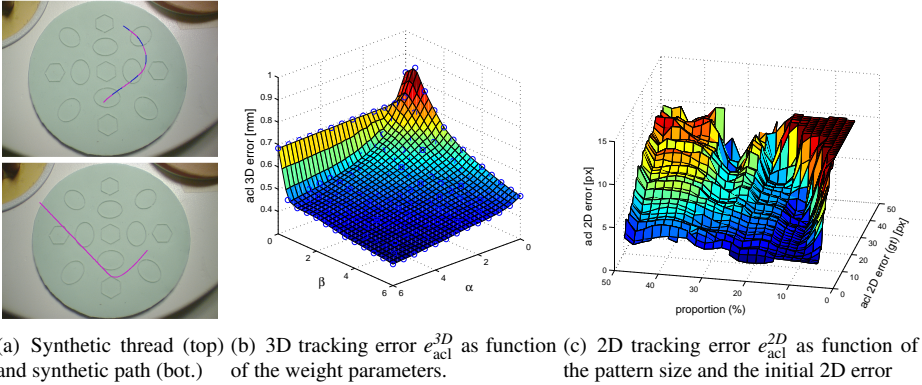


Figure 3: Synthetic data reprojected on a typical background for better visualization (a) and evaluation of the influence of the weight parameters used in the cost function (b,c).

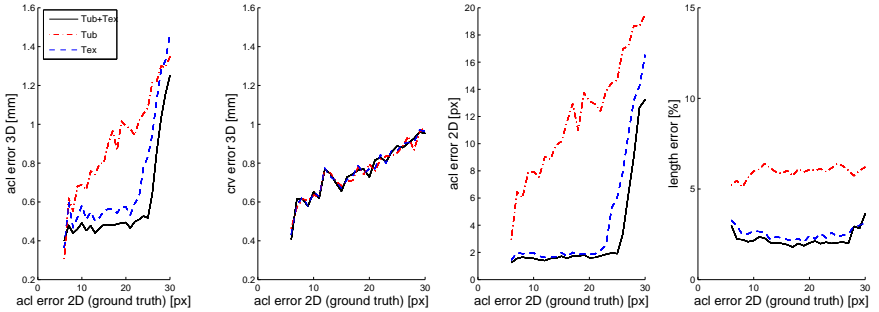


Figure 4: Convergence results for three different cost functions, using solely the tubular term (*Tub*), solely the texture term (*Tex*) or both (*Tub+Tex*). Results are presented as a function of the initial 2D error. a) 3D arclength error e_{ac1}^{3D} , b) 3D curve error e_{cv}^{3D} ; c) 2D arclength error e_{ac1}^{2D} , d) length error

4.1.2 Convergence experiments

In these experiments, a 3D virtual thread as shown in fig. 3(a) is projected, together with its pattern, on a white background. The 3D tracking model is then initialized at the position of the virtual thread and randomly perturbed in 3D by rigid translations. The perspective projection creates various initial 2D configurations. Then, several tracking approaches based on different cost functions are compared on the synthetic images to recover the original thread shape. In this process, since synthetic data is used, the ridge and texture distance maps do not contain any outliers. We present results averaged over 1000 random initializations.

The original stereo camera from the da Vinci robot that we use for the real experiments has a small baseline (5mm) for a large focal length (1770px). In order not to bias the error computations by this very specific setup, where small 3D displacements may be hardly visible in the images for a thread located at about 9 cm away from the camera, we use a baseline of 2cm in the synthetic experiments.

First, fig. 3(b) shows the error e_{ac1}^{3D} while varying the two weight parameters α, β in the cost function. In particular, we notice that the absence of a pattern term ($\beta = 0$) causes a

	ACL 3D (mm)	CRV 3D (mm)	ACL 2D (px)	CRV 2D (px)	LM ITER
Tub	0.52 (\pm 0.47)	0.42 (\pm 0.53)	9.21 (\pm 8.47)	3.41(\pm 5.04)	59.1
Tex	0.20 (\pm 0.12)	0.14 (\pm 0.08)	0.75 (\pm 0.30)	0.41 (\pm 0.14)	43.5
Tub + Tex	0.09 (\pm 0.05)	0.06 (\pm 0.03)	0.53(\pm 0.21)	0.27 (\pm 0.06)	63.8

Table 1: Tracking motion along the curve. Errors using three different cost functions.

larger error. Detailed convergence results are displayed in fig. 4 as a function of the initial 2D error $e_{\text{acl,gt}}^{2D}$ between the initial randomly perturbed 3D tracking model and the ground truth. Fig. 4(a) and 4(c) show the errors e_{acl}^{3D} and e_{acl}^{2D} for three cost functions: the curve *Tub+Tex* has parameters $\alpha = \beta = 4$, the curve *Tub* has parameters $\alpha = 4, \beta = 0$ and the curve *Tex* has parameters $\alpha = 0, \beta = 4$. One observes clearly that the presence of the texture reduces the errors, especially when the initial displacement increases. Fig. 4(b) shows that the usual curve error e_{crv}^{3D} , contrary to the arclength error e_{acl}^{3D} , fails to capture the proper tracking error. Finally, fig. 4(d) shows the errors in maintaining the initial length after convergence. The error is defined as $\frac{|L_{\text{final}} - L_{\text{initial}}|}{L_{\text{initial}}}$ and presented as a percentage. We notice that the use of the texture also helps to maintain the thread length in the multi-objective function that is optimized.

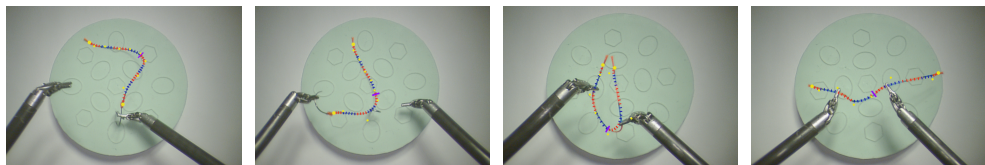
The convergence ability is both related to the size of the color pattern and to the initial displacement. This is illustrated in fig. 3(c) which shows the error e_{acl}^{2D} as a function of both the pattern size, defined as the ratio of the length of one color segment to the total thread length, and of the initial displacement error $e_{\text{acl,gt}}^{2D}$. For a fixed pattern size, the errors are computed over 1000 random initializations. The surface shows two trends: the initial 2D image displacement, starting from which the tracking fails, increases with the pattern size. This occurs up to a pattern size of about 30%. After this value, the pattern fails to improve the tracking, because there are too few color changes to constrain the convergence.

4.1.3 Motion along the curve

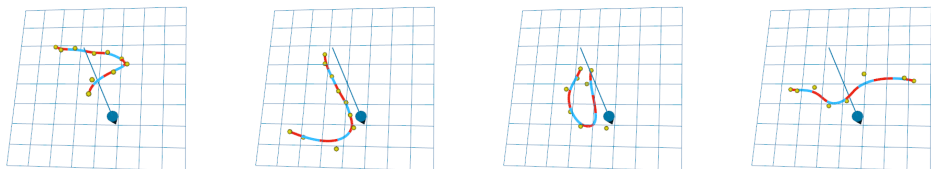
In this experiment, we evaluate the tracking approach against a pure tangential motion. A predefined 3D path is defined as shown in fig. 3(a). A virtual 3D thread follows this path in 30 displacements. The 3D tracking model is correctly initialized at the initial virtual thread position. The synthetic images, containing the projections of the virtual thread on a white background, are then used for the tracking. Tracking errors are displayed averaged along the whole path in table 1 for the three cost functions described previously. We see that the texture permits to accurately track tangential motion along the curve. When using only tubular information, which is equivalent here to using a mono-color thread, the tracking model tends to bend in the depth direction to satisfy both the length and visual constraints for this small baseline. Table 1 also indicates the average number of iterations required up to convergence for each cost function.

4.2 Real experiments

We have recorded several videos of an operator manipulating a thread over a background containing edges, using a da Vinci robot, to further evaluate the tracking qualitatively. Manipulations involve deformations in multiple directions in 3D space, laterally, and in depth, with the thread being grasped at different positions. "Motion along the curve" is demon-



(a) Original image overlaid with the reprojected tracking spline, its control points, and indication of the thread center.



(b) A different 3D view of the tracking spline, where the arrow indicates the camera direction.

Figure 5: Tracking illustration on two sequences showing large thread deformations.

strated by passing the thread, attached to a needle or not, through a ring or by dragging it along obstacles.

We use half-size full high-definition (HD) stereo images to perform the tracking. These images are obtained with a thread located at roughly 9cm in front of a stereo camera with a baseline of 5mm and a focal length of 1770px. The thread is of length 8cm and contains nine color stripes. Results of the tracking with overlay of the 3D reconstructed thread are displayed in fig. 5 and available in the demonstration video. Due to the small baseline, a few projective ambiguities are not always penalized by the data term, leading to two limitations in the reconstruction: 1) the extremities tend to curve in the depth direction as they are constrained only on one side; 2) small bendings in the depth direction sometimes occur along the thread. These two effects are especially visible in thread portions lying in the epipolar plane, but limited due to the constraints imposed by the pattern and the in-extensibility. As typical in visual tracking, one also notices that incorrect distance maps can throw the tracking off, especially if complete colors stripes are not detected. We therefore perform a slight over-segmentation of the colors and rely on the ridge detector to enforce proper localization on the thread centerline. Our current implementation runs on two cores (2.40 GHz) at about 10 fps. Using regions of interest and GPU parallelization to perform the image processing would significantly increase the frame rate and allow for tracking on the full stereo HD images.

5 Discussion and conclusion

We have presented an approach for tracking in 3D deformable curvilinear objects, such as a thread. This is a challenging problem due to the thin structure of the object and the lack of visual information available to reliably perform the tracking. In particular, as shown in our experiments, in the case of curvilinear objects with uniform appearance, the tracking is lost upon motion along the curve. Since it is not realistic to mark thin objects with distinguishable feature points, we instead propose a solution consisting of texturing the curvilinear object with a 1D pattern. We then formulate the tracking as a 3D NURBS estimation problem and propose a cost function that enforces both the tubular and texture appearances of the object

as well as the arclength parameterization of the 3D spline model. We show in synthetic experiments that this approach allows for accurate tracking by using a new error based on curve points uniquely defined by their arclength. We also demonstrate results on challenging videos in which a thread is manipulated by the tools of a da Vinci robot.

In future work, we plan to improve the modeling for tackling scenarios in which the thread can undergo major deformations and occlusions, like during knot tying manipulations. In particular, we plan to integrate prior information about the current temporal step of the manipulation task, which can be learnt from demonstration.

Acknowledgments: This work is funded by NSF grant 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] Amir A. Amini, Rupert W. Curwen, and John C. Gore. Snakes and splines for tracking non-rigid heart motion. In *Proc. of ECCV*, pages 251–261, 1996.
- [2] Adrien Bartoli, Mathieu Perriollat, and Sylvie Chambon. Generalized thin-plate spline warps. In *Proc. of CVPR*, 2007.
- [3] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete Elastic Rods. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):63:1–63:12, aug 2008.
- [4] Giulio Casciola and Serena Morigi. Reparametrization of nurbs curves. *International Journal of Shape Modeling*, 2:103–116, 1996.
- [5] Tat-Jen Cham and Roberto Cipolla. Stereo coupled active contours. In *Proc. of CVPR*, pages 1094–1099, 1997.
- [6] Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. Multi-scale vessel enhancement filtering. In *Proc. of MICCAI*, pages 130–137, 1998.
- [7] Davi Geiger, Alok Gupta, Luiz A. Costa, and John Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *PAMI*, 17(3):294–302, 1995.
- [8] Tim Hauke Heibel, Ben Glocker, Martin Groher, Nikos Paragios, Nikos Komodakis, and Nassir Navab. Discrete tracking of parametrized curves. In *Proc. of CVPR*, pages 1754–1761, 2009.
- [9] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
- [10] Shervin Javdani, Sameep Tandon, Jie Tang, James F. O’Brien, and Pieter Abbeel. Modeling and perception of deformable one-dimensional objects. In *Proc. of ICRA*, pages 1607–1614, 2011.
- [11] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

- [12] Julien Lenoir, Laurent Grisoni, Christophe Chaillou, and Philippe Meseure. Adaptive resolution of 1d mechanical b-spline. In *Proc. of GRAPHITE*, pages 395–403, 2005.
- [13] Hanna Martinsson, François Gaspard, Adrien Bartoli, and Jean-Marc Lavest. Energy-based reconstruction of 3d curves for quality control. In *Proc. of EMMCVPR*, pages 414–428, 2007.
- [14] Hermann Georg Mayer, Faustino J. Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jürgen Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. In *Proc. of IROS*, pages 543–548, 2006.
- [15] Nicolas Padoy and Gregory D. Hager. Human-machine collaborative surgery using learned models. In *Proc. of ICRA*, pages 5285–5292, 2011.
- [16] Nicolas Padoy and Gregory D. Hager. 3d thread tracking for robotic assistance in tele-surgery. In *Proc. of IROS*, pages 2102–2107, 2011.
- [17] Les Piegl and Wayne Tiller. *The Nurbs Book*. Springer, 2nd edition edition, 1997.
- [18] Rogério Richa, Philippe Poignet, and Chao Liu. Deformable motion tracking of the heart surface. In *Proc. of IROS*, pages 3997–4003, 2008.
- [19] Guy Shechter, Frédéric Devernay, Ève Coste Manière, and Elliot Mcveigh. Temporal tracking of 3D coronary arteries in projection angiograms. In *Proc. of Medical Imaging. SPIE*, 2002.
- [20] Jur van den Berg, Stephen Miller, Daniel Duckworth, Humphrey Hu, Andrew Wan, Xiao-Yu Fu, Ken Goldberg, and Pieter Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *Proc. of ICRA*, pages 2074–2081, 2010.
- [21] Peng Wang, Terrence Chen, Ying Zhu, Wei Zhang, Shaohua Kevin Zhou, and Dorin Comaniciu. Robust guidewire tracking in fluoroscopy. In *Proc. of CVPR*, pages 691–698, 2009.
- [22] Yi Jun Xiao and Y. F. Li. Optimized stereo reconstruction of free-form space curves based on a nonuniform rational b-spline model. *J. Opt. Soc. Am. A*, 22(9):1746–1762, 2005.