Noname manuscript No.
(will be inserted by the editor)

Shape Estimation for Elongated Deformable Object using B-spline Chained Multiple Random Matrices Model

Gang Yao · Ryan Saltus · Ashwin Dani

Received: date / Accepted: date

Abstract In this paper, a B-spline chained multiple random matrix model (RMM) representation is proposed to model geometric characteristics of an elongated deformable object. The hyper degrees of freedom structure of the elongated deformable object make its shape estimation challenging. Based on the likelihood function of the proposed B-spline chained multiple RMM, an expectation-maximization (EM) method is derived to estimate the shape of the elongated deformable object. A split and merge method based on the Euclidean minimum spanning tree (EMST) is proposed to provide initialization for the EM algorithm. The proposed algorithm is evaluated for the shape estimation of the elongated deformable objects in scenarios, such as the static rope with various configurations (including configurations with intersection), the continuous manipulation of a rope and a plastic tube, and the assembly of two plastic tubes. The execution time is computed and the accuracy of the shape estimation results is evaluated based on the comparisons between the estimated width values and its ground-truth, and the intersection over union (IoU) metric.

This work was supported by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA's Space Technology Research Grants Program.

Gang Yao
E-mail: gang.yao@uconn.edu

Ryan Saltus
E-mail: ryan.saltus@uconn.edu

Ashwin Dani
E-mail: ashwin.dani@uconn.edu

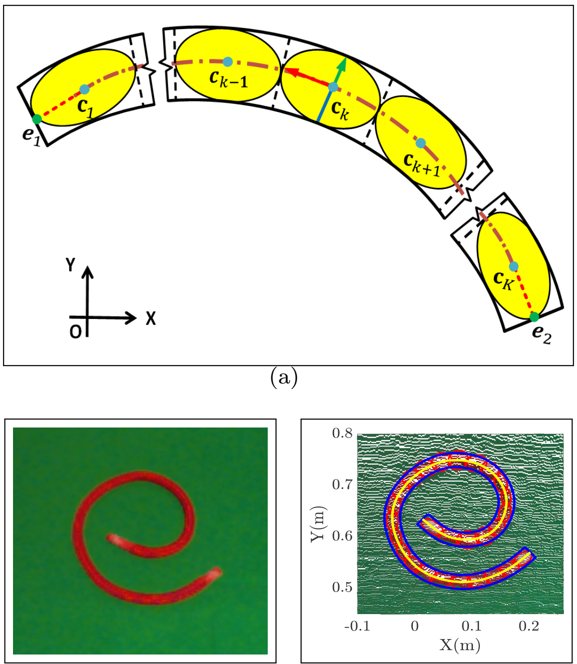
Department of Electrical and Computer Engineering, University of Connecticut, Storrs, Connecticut, USA.

Keywords State Estimation · Elongated Deformable Object · Random Matrices

1 Introduction

Elongated deformable objects are deformable objects which are characterized by a length that is much longer than their width [1, 2]. Objects of this type are commonly encountered in daily life, such as ropes, tubes, and trains. Automatic manipulation tasks such as grasping, completing surgical sutures or assembling cable harnesses are challenging due to the hyper degrees of freedom structure of the elongated deformable objects [1, 3, 4]. To improve the manipulation performance, it is necessary to provide an accurate perception model of the elongated deformable object as feedback to the robotic manipulators. The shape estimation of the elongated deformable object using data collected from the perception sensors, such as RGB-D cameras, is a challenging problem. In this paper, a shape estimation methodology for elongated deformable objects using a chained multiple random matrices representation is developed.

Some works use the physics-based simulation models such as linked capsules [5, 6], mass-spring models or finite element method model [7] to represent the elongated deformable object by considering the physical constraints of the object [1, 5–8]. A variety of registration algorithms are used to find the correspondences between the measurements and the predefined nodes on the simulated physics model. For example, the Gaussian mixture model (GMM) incorporating coherent point drift regularization is applied to register the rope nodes of a dynamic simulation model to the noisy point cloud, using a



(a)

(b)

(c)

Fig. 1 (a) Representation of an elongated deformable object as a B-spline curve chained multiple random matrices model; (b) The RGB image of the rope; (c) The point cloud and the shape estimation result of the rope, where the red ellipses are chained by the yellow B-spline curve and the blue curves are the offset curves from the B-spline.

stereo camera in [5]. The iterative closest point (ICP) method is used to estimate the rigid transformation from a point cloud to a 3D volumetric mesh, generated by the finite element method in [7]. A modified expectation-maximization (EM) algorithm is designed to directly register a point cloud from an RGB-D camera to a predefined mechanical model by a physical simulator in [6]. However, the physical simulation models only work for linked rigid objects or elastic objects. The accuracy of the algorithms depends on the physics-based prior and the physical simulation. A physically accurate model is used in [8] to model the elongated deformable object, and the physics-based priors are estimated by minimizing a generic energy function based on the images from a calibrated 3-camera rig.

Elongated deformable objects are also modeled by graphs or splines in [9,10]. A linear graph (line segments and not the shape) is used to represent a rope, and particle filters based on a predefined score function are used to infer the rope configuration in [10]. A region adjacency graph based on the super-pixels from the image of wires is developed to model the elongated deformable object in [9]. A method based on topological model and knot theory is developed to recognize the rope conditions in [11]. Based on the point cloud from an RGB-D camera, Bézier curve chained rectangles are used to approximate an elongated deformable object, the corresponding likelihood function is proposed, and the progressive Gaussian filter is used for the state estimation in [2]. However, the paper doesn't consider the situation where the elongated deformable object intersects with itself. A nonuniform rational B-spline (NURBS) curve is used to model a thin, deformable surgical suture thread by minimizing the image matching energy between the projected stereo NURBS image and the segmented thread image [4].

In this paper, an elongated deformable object is modeled as chained ellipses as shown in Fig. 1 (a), where the centers of the ellipses are located on a B-spline curve (brown dash-dotted line). Each ellipse (in yellow) is represented by a random matrix model (RMM), of which the center represents the location and the covariance matrix represents the shape of the ellipse. The RMM approximation of an elliptical object is widely used for extended object tracking. Extended object tracking methods track the position of the centroid and estimate the shape of the object, given sparse point measurements on the object at each time frame (cf. [12–14]). The proposed method is based on the point cloud of the elongated deformable object obtained by an RGB-D camera. The point cloud generated from the elongated deformable object is sparse but provides the position information which can be used as the measurements.

The B-spline chained RMMs can also be used when both the shape and the length of the object are changing, e.g., during the manipulations, or when assembling two elongated deformable objects. In addition, the physical simulation model of the elongated deformable object is not required to be built beforehand. Technical contributions of the paper are briefly summarized as follows:

- A set of chained multiple RMMs is proposed to approximate the elongated deformable object, of which the centers are enforced to be located on a B-spline curve. The corresponding likelihood function is derived. The proposed model both localizes the elongated deformable object and estimates its shape.
- A modified EM algorithm is proposed for the shape estimation of the elongated deformable object, based on the proposed log-likelihood function. The control points of the B-spline curve, the number of measurement points associated with each RMM and the covariance matrices of the RMMs are the parameters to be estimated by the EM algorithm.

– Since the log-likelihood function is nonconvex, it is necessary to have a good initialization for the EM algorithm. Knot sequence of the B-spline curve also needs to be generated from the unordered measurement points. A split and merge algorithm is proposed for the initialization which uses the Euclidean minimum spanning tree (EMST) and the breadth first search (BFS) method.

The rest of the paper is organized as follows. In Section 2, the B-spline curve chained RMMs, its likelihood function and a corresponding EM method are presented to estimate the position and shape of the elongated deformable object. In Section 3, a split and merge method based on the EMST and BFS is proposed to initialize the chained RMMs. The shape estimation results of the rope in different configurations, including non-intersection and intersection, based on measurements from the RGB-D camera, are shown in Section 4. The estimation results of the rope as well as the plastic tube in scenarios such as continuous manipulations and the assembly of two plastic tubes are also shown in this section. Conclusions and future work are given in Section 5.

2 B-spline Chained Random Matrices Model

2.1 B-spline Curve Representation

A point $\mathbf{p}(t) \in \mathbb{R}^{2\times 1}$ on the B-spline curve of degree d can be interpolated with parameter t from a polynomial, which is defined as a linear combination of $n+1$ control points (de Boor points) $\mathbf{b}_i \in \mathbb{R}^{2\times 1}$ and basis functions $N_{i,d}$ given by [15]

$$\begin{aligned}\mathbf{p}(t) &= \sum_{i=0}^{n} \mathbf{b}_i N_{i,d}(t)\\ 0 \le t &\le n-d+1\end{aligned} \tag{1}$$

The basis function $N_{i,d}(t)$ is defined based on a non-decreasing knot sequence $\{t_i : i = 0 \cdots n+d+1\}$ as [15]

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t) \tag{2}$$

which is a recursion function and

$$N_{i,0}(t) = \begin{cases} 1 & t \in [t_i, t_{i+1}) \\ 0 & otherwise \end{cases} \tag{3}$$

and the knot values t_i of the knot sequence are generated by

$$t_i = \begin{cases} 0 & 0 \le i < d+1 \\ i-d & d+1 \le i \le n \\ n-d+1 & n < i \le n+d+1 \end{cases} \tag{4}$$

2.2 Chained Multiple Random Matrices Model

The B-spline chained multiple random matrices model is used to model the elongated deformable object in this section as shown in Fig. 1. The multiple random matrices model is the sum of equally weighted K RMMs, which is defined as

$$p(\mathbf{Z} \mid \theta) = \sum_{k=1}^{K} w_k \phi(\mathbf{Z}_k | L_k, \mu_k, \Sigma_k) \tag{5}$$

where $\theta = \{(L_k, \mu_k, \Sigma_k)\}_{k=1}^{K}$, L_k is the number of measurements assigned to the kth RMM, μ_k and Σ_k are the mean and covariance of the kth cluster, $\phi(\mathbf{Z}_k|L_k, \mu_k, \Sigma_k)$ is the probability distribution of the measurement points $\mathbf{Z}_k = \{\mathbf{z}_{k,l}\}_{l=1}^{L_k}$ from the kth cluster, $\mathbf{Z} = \{\mathbf{z}_r\}_{r=1}^{N_r}$ is the set of total number of measurement points, and the weights of the clusters are assumed to be equal as $w_k = \frac{1}{K}$.

The kth cluster is approximated as an ellipse, and the measurement points inside the ellipse are assumed to be distributed as a Gaussian distribution with mean μ_k and covariance Σ_k. The probability of the measurement points $\mathbf{Z}_k = \{\mathbf{z}_{k,l}\}_{l=1}^{L_k}$ from the kth ellipse represented by RMM is [14]

$$\begin{aligned}\phi(\mathbf{Z}_k|L_k, \mu_k, \Sigma_k) &= \prod_{l=1}^{L_k} \mathcal{N}(\mathbf{z}_{k,l}; \mu_k, \Sigma_k)\\ &\propto \mathcal{N}(\overline{\mathbf{z}}_k; \mu_k, \frac{\Sigma_k}{L_k}) \times W(\overline{\mathbf{C}}_k; L_k - 1, \Sigma_k)\end{aligned} \tag{6}$$

where the center $\overline{\mathbf{z}}_k$ is

$$\overline{\mathbf{z}}_k = \frac{1}{L_k} \sum_{l=1}^{L_k} \mathbf{z}_{k,l} \tag{7}$$

and the scattering matrix $\overline{\mathbf{C}}_k$ is

$$\overline{\mathbf{C}}_k = \sum_{l=1}^{L_k} (\mathbf{z}_{k,l} - \overline{\mathbf{z}}_{k,l})(\mathbf{z}_{k,l} - \overline{\mathbf{z}}_{k,l})^{\mathrm{T}} \tag{8}$$

and $W(\overline{\mathbf{C}}_k; L_k - 1, \Sigma_k)$ is a Wishart density in $\overline{\mathbf{C}}_k$ with $L_k - 1$ degrees of freedom. The statistical sensor errors

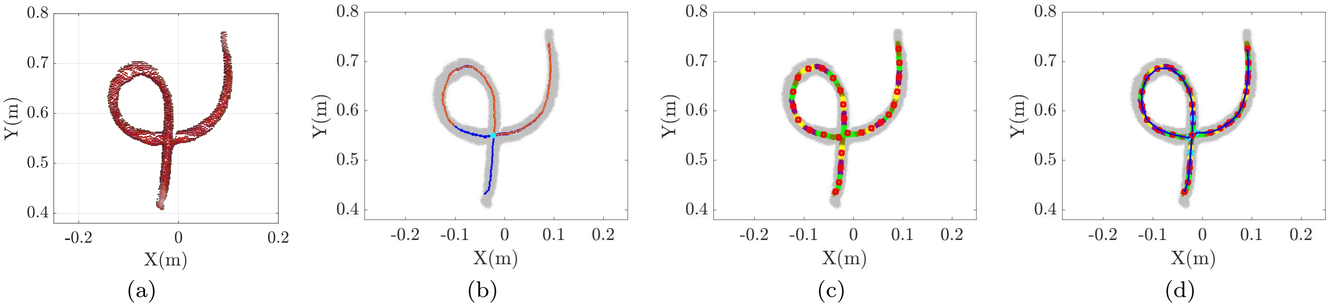


Fig. 2 Illustration of the initialization procedure: (a) Point cloud from the rope after background subtraction; (b) EMST of the points, the longest path (orange line segments) and the common vertex (in cyan); (c) Small segments represented by different colors and the corresponding centers (red points); (d) Graph of centers (blue line segments) and the common vertex and its 4 closest centers (in cyan).

are assumed to be neglected, and the physical extension of the target dominates the spread of measurements in (6) [14]. If the sensor errors are within the same order of the magnitude of the target extension, they cannot be neglected anymore. In this case, the covariance is $\Sigma_k = \Sigma_p + \mathbf{R}$, where Σ_p is a symmetric positive definite random matrix representing the physical extension, and $\mathbf{R}$ is the covariance matrix of sensor errors [14].

The brown dash-dotted curve in Fig. 1 (a) is the B-spline curve, which crosses through the centers of the clusters. Multiple RMMs with centers located on the B-spline curve constitute the elongated deformable object. The probability of the measurement points $\mathbf{Z}_k = \{\mathbf{z}_{k,l}\}_{l=1}^{L_k}$ from the kth elliptical cluster is defined as

$$\phi(\mathbf{Z}_k|L_k, \mathbf{c}_k, \Sigma_k) = \int\limits_{\mu_k} \phi(\mathbf{Z}_k|L_k, \mu_k, \Sigma_k) \cdot p(\mu_k|\mathbf{c}_k) d\mu_k \tag{9}$$

where $\phi(\mathbf{Z}_k|L_k, \mathbf{c}_k, \Sigma_k)$ is the probability distribution of the measurement points from the kth cluster and $p(\mu_k|\mathbf{c}_k) = \delta(\mu_k - \mathbf{c}_k)$ is the Dirac delta function to enforce the center μ_k to be located on the B-spline curve $\mathbf{c}_k$. K center points are sampled from the B-spline curve as

$$\mathbf{c}_k = \mathbf{p}(t_k), k = 1 \cdots K \tag{10}$$

where $\mathbf{c}_k$ is the kth center point, and $\mathbf{p}(t_k)$ is the corresponding point on the B-spline curve, which is only determined by the control points of the B-spline curve $\mathbf{b}_i$ in (1), and t_k is the corresponding parameter value of the B-spline curve determined by the centripetal method [16].

Assuming the kth cluster center $\mathbf{c}_k$ is located on the B-spline curve, by putting (6) and (10) into (9) the probability distribution of the measurement points from the kth cluster is

$$\phi(\mathbf{Z}_k|L_k, \mathbf{c}_k, \Sigma_k) \propto \mathcal{N}(\bar{\mathbf{z}}_{k,l}; \mathbf{c}_k, \frac{\Sigma_k}{L_k}) \times W(\overline{\mathbf{C}}_k; L_k - 1, \Sigma_k) \tag{11}$$

The B-spline curve chained multiple random matrices representation is the sum of equally weighted K RMMs, which is redefined as

$$p(\mathbf{Z} \mid \theta) = \sum_{k=1}^{K} w_k \phi(\mathbf{Z}_k|L_k, \mathbf{c}_k, \Sigma_k) \tag{12}$$

2.3 Expectation-Maximization Method

In this subsection, the EM algorithm is used to estimate the parameters $\theta = \{(L_k, \mathbf{c}_k, \Sigma_k)\}_{k=1}^{K}$ of the B-spline chained RMMs. The EM algorithm finds the parameters $\theta^* = \{(L_k^*, \mathbf{c}_k^*, \Sigma_k^*)\}_{k=1}^{K}$ corresponding to the maximum likelihood by iterating between the expectation step and the maximization step.

The **expectation step** assigns each of the N_r measurement points $\mathbf{Z} = \{\mathbf{z}_r\}_{r=1}^{N_r}$ to each of clusters by

$$k^* = \arg\max_{k=1\cdots K} p_k(\mathbf{z}_r) \tag{13}$$

where $p_k(\mathbf{z}_r) = \mathcal{N}(\mathbf{z}_r; \mathbf{c}_k, \Sigma_k)$. The parameter L_k is then determined by counting the number of points in each cluster. After the assignment of the measurements, the mean $\bar{\mathbf{z}}_k$ and scattering matrix $\overline{\mathbf{C}}_k$ are calculated using (7) and (8). The corresponding parameter t_k in (10) is also recalculated based on the centripetal method [16].

The **maximization step** estimates the parameters θ by maximizing the log-likelihood function of the chained RMMs. The log-likelihood function of the chained RMMs in (12) is

$$\mathcal{L}(\theta) = \sum_{k=1}^{K} \Big\{ -\frac{L_k}{2}(\mathbf{c}_k - \overline{\mathbf{z}}_k)^{\mathrm{T}} \varSigma_k^{-1}(\mathbf{c}_k - \overline{\mathbf{z}}_k) \\ -\frac{L_k+1}{2}\log|\varSigma_k| - \frac{1}{2}\mathrm{tr}(-\frac{1}{2}\overline{\mathbf{C}}_k \varSigma_k^{-1}) \Big\} + \mathrm{Const} \tag{14}$$

which is maximized by iterative re-weighted least squares method [17]. Rewrite (1) in the matrix-vector form as $\mathbf{c}_k = \mathbf{B}_k \mathbf{b}$, where $\mathbf{B}_k$ is the block diagonal matrix as $\mathbf{B}_k = \mathbf{blkdiag}(\mathbf{n}_k^{\mathrm{T}}, \mathbf{n}_k^{\mathrm{T}}) \in \mathbb{R}^{2\times 2(n+1)}$, $\mathbf{n}_k = \left[N_{0,d}(t_k), \cdots N_{n,d}(t_k)\right]^{\mathrm{T}} \in \mathbb{R}^{(n+1)\times 1}$, and $\mathbf{b} = \left[\mathbf{b}_x^{\mathrm{T}}, \mathbf{b}_y^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{2(n+1)\times 1}$, where $\mathbf{b}_x^{\mathrm{T}}$ and $\mathbf{b}_y^{\mathrm{T}}$ are the control points in x and y coordinates. Taking the derivative of the log-likelihood function $\mathcal{L}(\theta)$ with respect to the control points $\mathbf{b}$ and positive symmetric matrix $\varSigma_k$ separately and setting them equal to 0 yields

$$\mathbf{b} = \mathbf{Q}^{+}\mathbf{M} \tag{15}$$

where $\mathbf{Q}^{+}$ is the Moore–Penrose inverse of $\mathbf{Q} = \sum_{k=1}^{K} L_k \mathbf{B}_k^{\mathrm{T}} \varSigma_k^{-1} \mathbf{B}_k$ and $\mathbf{M} = \sum_{k=1}^{K} L_k \mathbf{B}_k^{\mathrm{T}} \varSigma_k^{-1} \overline{\mathbf{z}}_k$, and

$$\varSigma_k = \frac{1}{L_k+1}\sum_{l=1}^{L_k}(\mathbf{z}_{k,l} - \mathbf{c}_k)(\mathbf{z}_{k,l} - \mathbf{c}_k)^{\mathrm{T}} \tag{16}$$

The iteration between (15) and (16) is carried out until the value of the log-likelihood function in (14) stops increasing or the optimization reaches the predefined maximum iteration number.

The orientation of the ellipse and its semi-major (red arrow) and semi-minor (green arrow) axes are determined by the eigenvalues and eigenvectors of $4\varSigma_k$ as shown in Fig. 1 (a), assuming that the measurements are uniformly distributed inside the ellipse and approximated as a Gaussian distribution in the proposed model by the moment matching method [14].

In every ellipse, each line perpendicular to the B-spline curve that passes through the center is found. The length of the line segment (blue line segment as shown in Fig. 1 (a)) between the center and its intersection with the ellipse is calculated. Then, half of the width of the rope is determined by the average length of the calculated line segments. The offset curves are drawn by off-shifting the B-spline curve by half of the calculated width of the rope.

The two ends ($\mathbf{c_1}$ and $\mathbf{c}_K$ in Fig. 1 (a)) of the B-spline curve are the centers of the ellipses representing the two terminal parts of the rope. The two ends ($\mathbf{e_1}$ and $\mathbf{e}_2$ in Fig. 1 (a)) of the rope are determined by the intersections between these ellipses and the lines (red dashed lines in Fig. 1 (a)) tangent to the B-spline curve at the centers ($\mathbf{c_1}$ and $\mathbf{c}_K$ in Fig. 1 (a)) of these corresponding ellipses. The length of the rope is determined by the length of the B-spline curve and the lengths of the line segments (red dashed lines in Fig. 1 (a)) between the two ends of the rope and the centers of the corresponding ellipses.

3 Initialization for B-spline Chained RMMs

The elongated deformable object may have parts which are very close to one another or have intersections with itself. The initialization step is to generate the general configuration of the elongated deformable object from the point cloud and to trace a B-spline curve embedded in the unordered measurement points. A split and merge method is proposed to initialize the algorithm in this section. The rope is first split into small segments and then the configuration of the rope is obtained by building the graph of the centers of the small segments. The initialization procedure is shown in Fig. 2.

Before the initialization procedure, the pre-processing stage is done to find the medial skeleton of the point cloud. The point cloud of the rope after background subtraction is shown in Fig. 2 (a). Then, the point cloud is converted into a binary image which is then dilated and thinned. The pixels (in red) after dilation and thinning are obtained as shown in Fig. 3. The intersection part of the rope (in green) is linearized by the Bresenham algorithm (see [11]). Then the pixels are converted back to the point cloud with position information as shown in Fig. 2 (b). Other methods to find the medial skeleton of the point cloud can also be used in this stage [18].

3.1 Split Step

The first step of the split and merge method is to divide the obtained point cloud into smaller segments, as shown in Algorithm 1. First, an EMST is constructed based on the points. The EMST is an acyclic edge-weighted graph $T = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertices set and $\mathcal{E}$ is the set of the edges connecting every two vertices $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}$ and $i \neq j$. The weight of the edge is defined as the Euclidean distance of the two vertices $\epsilon = \|\mathbf{v}_i - \mathbf{v}_j\|$ [19, 20]. The EMST is constructed by selecting the set of connected edges to ensure the summation of the weights of the edges is

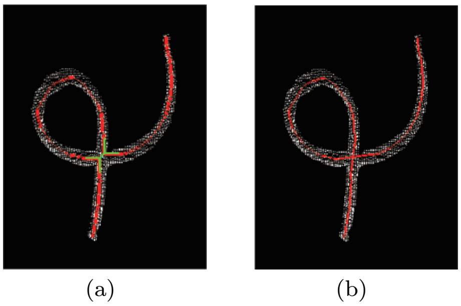


Fig. 3 The binary image of the point cloud (in white) from the rope and the linearization of the intersection points: (a) The pixels (in red) obtained by the dilation and thinning of the binary image, and the intersection points (in green); (b) The pixels linearized by the Bresenham algorithm.

minimum. The Prim's algorithm is used to generate the EMST [20].

After the construction of the EMST, the longest path of the EMST is found by two breadth first searches (BFS) [20]. The first BFS is used to traverse the EMST with a random chosen vertex from the EMST, and the path with the largest weight and one end point of the longest path are found. Then, another BFS traverses the EMST, starting with the end point that was found in the prior iteration. The longest path $P = (\mathcal{V}_p, \mathcal{E}_p)$ is found which is also the longest path of the EMST $T = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}_p \subset \mathcal{V}$ and $\mathcal{E}_p \subset \mathcal{E}$. The orange line segments shown in Fig. 2 (b) are the longest path of the EMST $T = (\mathcal{V}, \mathcal{E})$.

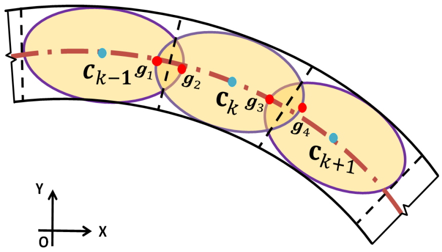


Fig. 4 Adjustment of the ellipses with overlaps.

Algorithm 1: The segmentation of the elongated deformable object

```
Set the threshold $\gamma$;
Obtain the point cloud $\mathbf{Z} = \{\mathbf{z}_r\}_{r=1}^{N_r}$;
Build the EMST $T = (\mathcal{V}, \mathcal{E})$;
Find the longest path of the EMST $P = (\mathcal{V}_p, \mathcal{E}_p)$ by
  BFSs;
Delete $\mathcal{E}_p$, then there are many small clusters (trees)
  $\mathcal{C} = \{\mathcal{C}_h\}_{h=1}^{J_h}$;
Initialize empty sets $\mathcal{A}$ and $\mathcal{B}$;
Initialize $h = 1$;
while $h \leq J_h$ do
    Calculate $D_h$ in (17);
    if $D_h > \gamma$ then
        Append $\mathcal{C}_h$ to $\mathcal{A}$;
    end
    else
        $\mathcal{B} = \mathcal{B} \cup \mathcal{C}_h$;
    end
    $h = h + 1$;
end
Append $\mathcal{B}$ to $\mathcal{A}$;
```

After the construction of the EMST and the longest path $P = (\mathcal{V}_p, \mathcal{E}_p)$ is found, the EMST is segmented into smaller disconnected clusters (or trees) by deleting the $\mathcal{E}_p$ from the EMST T. Then, there are some small trees $\mathcal{C} = \{\mathcal{C}_h\}_{h=1}^{J_h}$ left. The distance D_h between $\mathcal{C}_h$ and $\mathcal{V}_p$ is defined as

$$D_h = \sup\{\inf\{\|\mathbf{u}_i - \mathbf{v}_j\| \, |\forall \mathbf{v}_j \in \mathcal{V}_p\} | \forall \mathbf{u}_i \in \mathcal{C}_h\} \tag{17}$$

where $\mathbf{v}_j$ are vertices from $\mathcal{V}_p$ and $\mathbf{u}_i$ are vertices from $\mathcal{C}_h$. If D_h is larger than a predefined threshold γ, the cluster $\mathcal{C}_h$ and the $\mathcal{V}_p$ are from different regions of the rope, instead of the small branches of the P. After the O_n different parts (or smaller clusters) of the rope $\mathcal{A} = \{\mathcal{A}_n\}_{n=1}^{O_n}$ are found, the common vertices $\mathcal{V}_C = \{\mathbf{v}_c | \mathbf{v}_c \in \mathcal{A} \backslash \mathcal{B}, \mathbf{v}_c \in \mathcal{V}_P, c = 1, \cdots, O_n - 1\}$ of the O_n clusters are also found, where the points of $\mathcal{B}$ are from the same region of the rope as $\mathcal{V}_P$. The common vertices have more than 2 edges connected with them. If the distance between some common vertices are smaller than the predefined threshold γ, their mean values are used to represent them. The EMST is segmented into $O_n = 2$ clusters (orange and blue line segments) shown in Fig. 2 (b).

Each cluster is an EMST and the longest path with two end points are found by two BFSs. Starting with one end point, the cluster is divided into M_n smaller segments $\mathcal{S}_n = \{\mathcal{S}_{u,n}\}_{u=1}^{M_n} \subset \mathcal{A}_n$, with each segment $\mathcal{S}_{u,n}$ satisfying

$$\sup\{\|\mathbf{z}_i - \mathbf{z}_j\| \, |\forall \mathbf{z}_i, \mathbf{z}_j \in \mathcal{S}_{u,n}, i \neq j\} \leq H \tag{18}$$

where H is a predefined parameter and $\mathbf{z}$ is the point inside the smaller segments $\mathcal{S}_{u,n}$. The center of each segment $\mathbf{c}_{u,n}$ is calculated as

$$\mathbf{c}_{u,n} = \frac{1}{L_{u,n}} \sum_{i=1}^{L_{u,n}} \mathbf{z}_i, \mathbf{z}_i \in \mathcal{S}_{u,n} \tag{19}$$

where $L_{u,n}$ is the number of points in segment $\mathcal{S}_{u,n}$. The segments are shown with different colors and the red points are the centers of the segments in Fig. 2 (c).

Algorithm 2: The ordering of the centers and B-spline interpolation

```
Set the distance $H$;
Given the $O_n$ clusters $\mathcal{A} = \{\mathcal{A}_n\}_{n=1}^{O_n}$;
Given the common vertices
  $\mathcal{V}_C = \{\mathbf{v}_c | \mathbf{v}_c \in \mathcal{A} \backslash \mathcal{B}, \mathbf{v}_c \in \mathcal{V}_p, c = 1, \cdots, O_n - 1\}$;
Initialize $n, c = 1$;
while $n \leq O_n$ do
    Divide $\mathcal{A}_n$ into smaller segments
      $\mathcal{S}_n = \{\mathcal{S}_{u,n}\}_{u=1}^{M_n}$ satisfies (18);
    Calculate $\mathbf{c}_{u,n}$ in (19);
    Create the graph $T_m = (\mathcal{V}_m, \mathcal{E}_m)$ with
      $\mathbf{c}_{u,n} \in \mathcal{V}_m$ and $\{\mathbf{c}_{u,n}, \mathbf{c}_{u+1,n}\} \in \mathcal{E}_m$ and
      $u = 1, \cdots, M_n - 1$;
    $n = n + 1$;
end
while $c \leq O_n - 1$ do
    Find the centers $\mathcal{Q} = \{\mathbf{c}_q\}_{q=1}^{4} \subset \mathcal{V}_m$ closest to
      $\mathbf{v}_c$, and delete edges between them;
    Calculate vectors $\mathbf{d}_q$;
    Find the two pairs $(i^*, j^*)$, based on (20);
    Create the edges $e$;
    $\mathcal{E}_m = \mathcal{E}_m \cup e$;
    $\mathcal{V}_m = \mathcal{V}_m \cup \mathbf{v}_c$;
    $c = c + 1$;
end
Complete the graph $T_m = (\mathcal{V}_m, \mathcal{E}_m)$;
Fit B-spline by the ordered centers and readjust the
  centers;
```

3.2 Merge Step

Previously the elongated target is subdivided into small segments and the centers of the segments are found. In this step, the order information of the centers is generated and a B-spline curve is traced, which represents the global shape and configuration information of the target as shown in Algorithm 2. At first, a graph $T_m = (\mathcal{V}_m, \mathcal{E}_m)$ is created, such that the center of each segment constitutes the vertex $\mathbf{c}_{u,n} \in \mathcal{V}_m$ and the edges are built by connecting the centers of nearby segments in each cluster $\mathcal{A}_n$ as $\{\mathbf{c}_{u,n}, \mathbf{c}_{u+1,n}\} \in \mathcal{E}_m$ and $u = 1, \cdots, M_n - 1$.

For the common vertex $\mathbf{v}_c \in \mathcal{V}_C$, the closest four centers are found $\mathcal{Q} = \{\mathbf{c}_q\}_{q=1}^{4} \subset \mathcal{V}_m$. The edges between these four centers are deleted and the new edges between them are found in the following way. The vectors $\mathbf{d}_q = \frac{\mathbf{v}_c - \mathbf{c}_q}{\|\mathbf{v}_c - \mathbf{c}_q\|}$ are calculated, and the pair (i^*, j^*) is found that meets

$$\underset{i,j \in q, i \neq j}{\arg\min} \|\mathbf{d}_i + \mathbf{d}_j\| \tag{20}$$

The vertex $\mathbf{v}_c$ is added to the graph as $\mathcal{V}_m = \mathcal{V}_m \cup \mathbf{v}_c$. Then, the edges are generated as $e_1 = \{\mathbf{c}_{i^*}, \mathbf{v}_c\}$ and $e_2 = \{\mathbf{v}_c, \mathbf{c}_{j^*}\}$, which connect between $\mathbf{v}_c$ and the two centers $\mathbf{c}_{i^*}$ and $\mathbf{c}_{j^*}$, giving the minimum value of (20). The edges e_1 and e_2 are added to the graph T_m as $\mathcal{E}_m = \mathcal{E}_m \cup e_1$ and $\mathcal{E}_m = \mathcal{E}_m \cup e_2$. The 2 remaining centers are connected as another edge e_3 and it is added to the graph T_m as $\mathcal{E}_m = \mathcal{E}_m \cup e_3$. Finally, the graph $T_m = (\mathcal{V}_m, \mathcal{E}_m)$ is completed by adding an edge to the 2 centers of the closest end segments from different clusters to make all the centers ordered.

The initial B-spline curve is interpolated by the ordered centers using (15) and (16). The ellipses may have overlaps as shown in Fig. 4. The centers are adjusted to decrease the overlap between ellipses by using the following formula:

$$\mathbf{c}_k = \frac{(\mathbf{g}_1 + \mathbf{g}_2) + (\mathbf{g}_3 + \mathbf{g}_4)}{4} \tag{21}$$

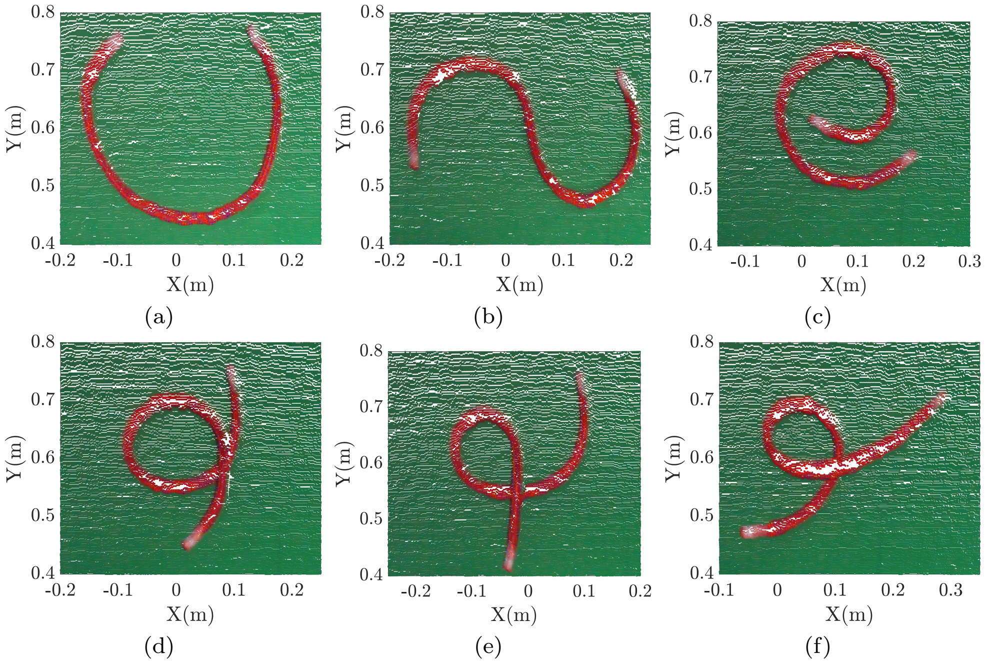
where $\mathbf{c}_k$ is the center of the kth ellipse, $\mathbf{g}_1$ and $\mathbf{g}_4$ are the endpoints of the kth ellipse, $\mathbf{g}_2$ is the endpoint of the $(k-1)$ th ellipse and $\mathbf{g}_3$ is the endpoint of the $(k+1)$ th ellipse. The centers are deleted if their distances to the common vertices are smaller than H.

4 Experimental Results

In order to validate the proposed B-spline chained RMMs, a series of experiments are performed to estimate the shape of the elongated deformable objects. Intersection over union (IoU) is used as the metric to evaluate the accuracy of shape estimation of the proposed algorithm. The IoU is defined as the area of intersection of the estimated shape and the true shape divided by the union of the two shapes [2]

$$\mathbf{IoU} = \frac{area(\theta^*) \cap area(\hat{\theta})}{area(\theta^*) \cup area(\hat{\theta})} \tag{22}$$

where $\hat{\theta}$ is the true shape parameters and θ^* is the estimated shape parameters. IoU is between 0 and 1, where the value 1 corresponds to a perfect match between the estimated area and the ground-truth. Since the ground-truth of the position and the shape of the elongated deformable object is difficult to obtain, the measurements from the RGB-D camera are used as the ground-truth [2]. The measurement noise is neglected because it is small compared to the area of the elongated deformable object. The ground-truth is constructed by creating a 1px × 1px rectangle at each measurement point and taking the union of all the rectangles [2]. The dilation and erosion methods are applied to ensure that the pixels are fully connected while preserving the boundary of the target.



(a) (b) (c) (d) (e) (f)

Fig. 5 Point clouds of the rope in different configurations obtained from an RGB-D camera.

4.1 Shape Estimation Results of The Static Rope

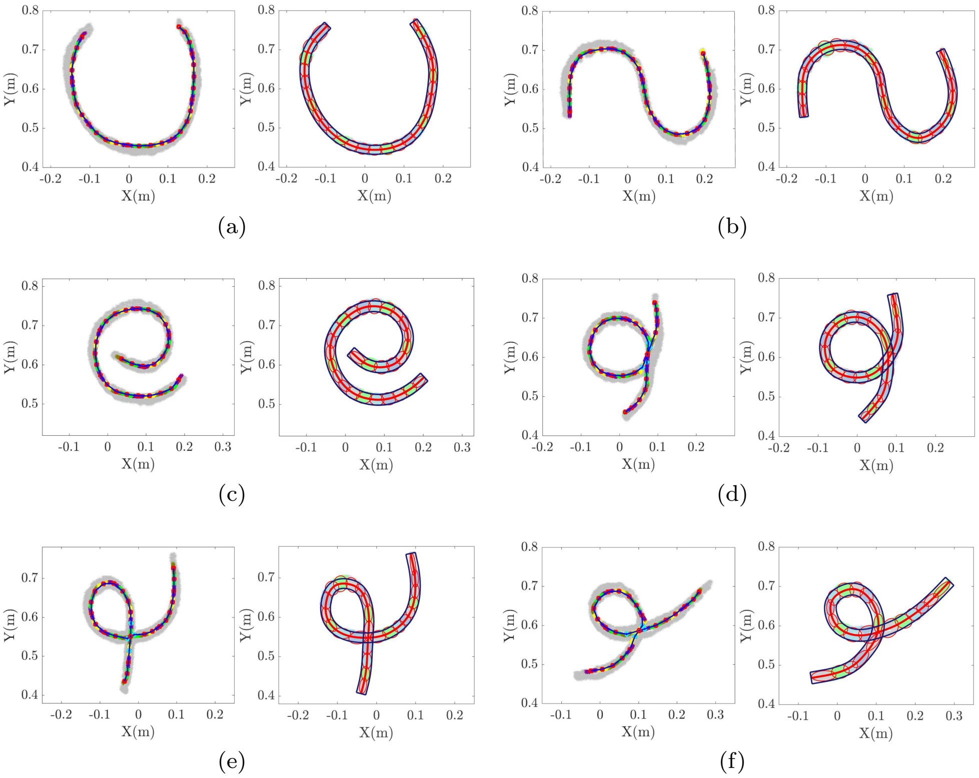
A red nylon dock line with width of 15.8mm and length of 930mm is used in the first part of the experiment. The rope is manipulated to 6 different shapes either with or without intersection. A green cloth is used as the background. A Microsoft® Kinect camera is used to obtain a point cloud of the rope. Plane fitting method and clustering algorithms are applied to subtract the rope point cloud from the background. The point clouds of the rope with 6 different configurations are shown in Fig. 5.

The shape estimation results on 6 different configurations of the rope are shown in Fig. 6. Each sub-figure includes two plots. The left plot is the initialization result. The gray measurement points are the measurements from the rope. The colored segments are the initial segments. The red dots are the centers of each segment and the blue line segments are the initial graph of the centers. The right plot is the estimated shape of the rope. The segmented measurement points are denoted by different colors. The red ellipses represent multiple RMMs and the red curve is the B-spline curve of the centers of the ellipses. The offset curves in blue in Fig 6 are created by shifting the estimated B-spline curve by half of the estimated width of the rope. The H value in (18) in the initialization procedure is set to 30mm. The degree of the B-spline curve is set to 2 and the number of the control points is set to 13 for all 6 configurations. The maximum iteration number of the EM algorithm is set to 3.

Table 1 The estimated width and IoU values of the rope in 6 different configurations.

Configuration	1	2	3	4	5	6
Width (mm)	11.4	11.2	13.7	13.0	12.3	14.9
IoU	0.752	0.738	0.731	0.732	0.736	0.715
IoU [6]	0.683	0.654	0.632	0.603	0.642	0.520

The estimated width of the rope and the IoU values are shown in Table 1. The estimated width is smaller than the true width (15.8mm) and the average IoU value over 6 configurations is 0.734. Because the surface of the rope is not flat, the detection of the edge points of the rope using Kinect sensor is hard. This causes the estimates of the width of the rope to be smaller than the ground-truth. Algorithm in [6] is also applied to estimate the shape of the rope in 6 configurations, of which the code is publicly available (`http://rll.berkeley.edu/tracking/`). The algorithm in [6] models a virtual rope as a set of serial-linked capsules in a simulation. The radius of the capsules is set as 7.9mm and other parameters are set to the default settings. The shape estimations of the proposed algorithm are more accurate compared with the algorithm in [6], based on the IoU values shown in Table 1.



(a) (b) (c) (d) (e) (f)

Fig. 6 Shape estimation results on the rope in 6 different configurations. For each sub-figure (a)-(f), the left plot is the initialization result and the right plot is the estimated shape of the rope.

Table 2 Typical execution time for initialization and EM algorithms with the rope in 6 different configurations.

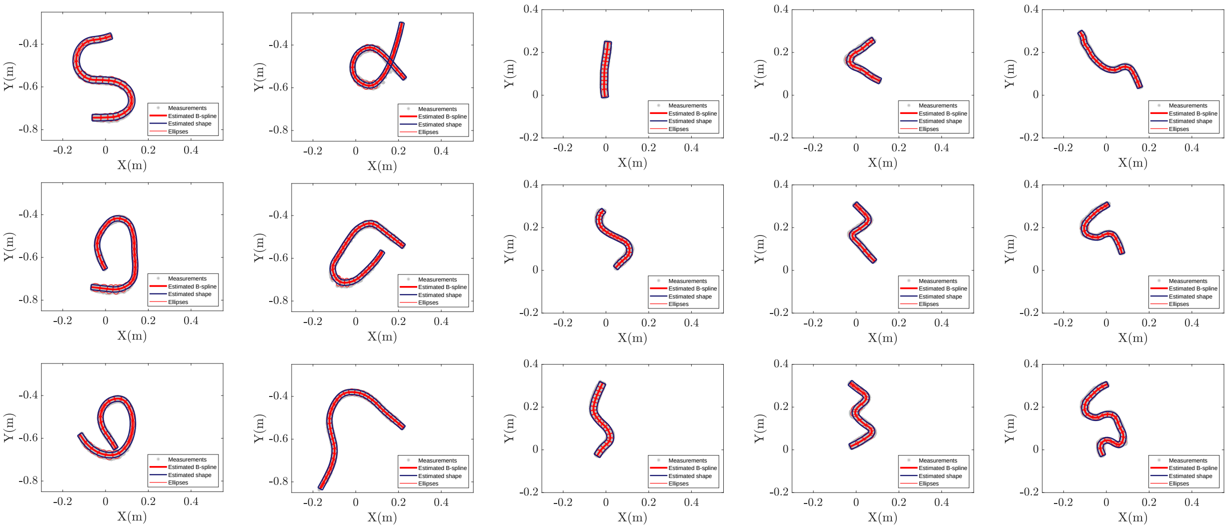
Configuration	1	2	3	4	5	6
Initialization (sec)	0.195	0.196	0.199	0.221	0.212	0.215
EM (sec)	0.296	0.291	0.283	0.253	0.248	0.252

The experiments are performed by measuring the elongated deformable object on a tabletop with an RGB-D sensor. The sensor only detects the top surface of the elongated deformable object closest to the sensor. Thus, the two dimensional shape is estimated by the proposed algorithm and the third dimension is determined by the tabletop. However, the proposed model can be extended to estimate the shape of the rope in three dimensional space. The control points of the B-spline curve in (1) can be extended into three dimensions as $\mathbf{b}_i \in \mathbb{R}^{3\times1}$ [15]. RMM can also be extended into three dimensions to represent an ellipsoid [14].

The codes of the initialization and EM algorithms were run in MATLAB® R2019b on a Windows 10 PC with Intel® $i7-9700k@3.60GHZ$ processor and 32.0GB of RAM. The typical execution time for initialization and EM algorithms are shown in Table 2. The most time consuming parts of the initialization stage are building the EMST and the calculation of the distance D_h in (17). The time complexity of building the EMST is $\mathcal{O}(|\mathcal{E}|\log|\mathcal{V}|)$, where $|\mathcal{E}|$ and $|\mathcal{V}|$ are the number of edges and vertices in graph $T = (\mathcal{V}, \mathcal{E})$ [20]. The calculation of distance D_h in (17) requires $\mathcal{O}(|\mathcal{V}_p|\,|\mathcal{C}_h|)$ operations, where $|\mathcal{V}_p|$ is the number of vertices from $\mathcal{V}_p$ and $|\mathcal{C}_h|$ is the number of vertices from $\mathcal{C}_h$. The time complexity of the EM algorithm is determined by the expectation stage, which assigns the measurements into different ellipses. The time complexity is $\mathcal{O}(KN_r)$, where N_r is the number of the measurements $\mathbf{Z} = \{\mathbf{z}_r\}_{r=1}^{N_r}$ and K is the number of clusters. In order to reduce the execution time for the shape estimation of the elongated deformable object, parallel computing can be used or the number of measurements can be uniformly down sampled.

4.2 Shape Estimation Results During Manipulations and Assembly

The previous experiments show the shape estimation of the rope in 6 different configurations, based on the measurements from one frame. The proposed EM algorithm is also used for shape estimation of the



(a) Rope1 (b) Rope2 (c) Tube1 (d) Tube2 (e) Assembling

Fig. 7 Shape estimation results on the rope and the plastic tube during manipulations. For each sub-figure (a)-(e), the 1^{st} frame, the 25^{th} frame and the 50^{th} frame are shown from top to bottom.

Table 3 The average execution times and average IoU values of the rope and the plastic tube during manipulations.

Video name	Rope1	Rope2	Tube1	Tube2	Assembling
shape change	√	√	√	√	√
length change			√	√	√
execution time (sec/frame)	0.343	0.422	0.450	0.445	0.441
IoU	0.727	0.709	0.799	0.768	0.819

elongated deformable object over multiple frames. The IoU value between the estimated shape of the previous time step and the current measurements is calculated for each frame. If the calculated IoU value is smaller than a threshold, the initialization procedure is rerun in consideration of the abrupt changes of the shape or the length of the elongated object.

Besides the red nylon dock line, a flexible red plastic tube with modifiable shape and length is also used in the experiments. The elongated deformable objects are manipulated in 5 different scenarios. The scenarios and the estimated results are recorded as videos (provided as a single combined video multimedia attachment). Each video has 50 frames of point cloud measurements. The video ‘Rope1’ shows the scenario that the rope is manipulated from the shape ‘s’ to the shape ‘9’. The video ‘Rope2’ demonstrates the scenario when the rope changes from one intersection configuration to non-intersection configuration. The ‘Tube1’ shows that the plastic tube is stretched and squeezed which changes both the shape and the length. The video ‘Tube2’ illustrates that one part of the plastic tube is stretched at a time and the red plastic tube is manipulated from the shape ‘L’ to the shape ‘M’. The last video ‘Assembling’ shows the situation when two plastic tubes are assembled together as one tube.

The typical frames and the estimation results are shown in Fig. 7. The description of the videos and the estimation results, including the average IoU value and the average execution time over 50 frames for each video, are shown in Table. 3. Algorithm in [6] is applied to estimate the shape of the rope in ‘Rope1’ and ‘Rope2’ scenarios. The average IoU values for ‘Rope1’ and ‘Rope2’ scenarios are 0.505 and 0.608 separately. The proposed algorithm achieves better accuracy in terms of IoU. Because the Algorithm in [6] uses linked rigid objects as the simulation model of the rope, it cannot work for the scenarios (e.g. ‘Tube1’, ‘Tube2’ and ‘Assembling’) when the elongated object are changing both length and shape during the manipulations.

5 Conclusions and Future Work

To localize the elongated deformable object and to estimate its shape, a B-spline chained multiple RMMs representation and its corresponding EM algorithm are developed in this paper. Based on the sparse measurements from an RGB-D camera, the proposed algorithm approximates the elongated deformable object as a set of chained ellipses by using a B-spline curve. Each ellipse is represented by an RMM, of which the center represents the location and the covariance matrix determines the shape of the ellipse. All the centers are enforced to be located on a B-spline curve, which represents the shape of the elongated deformable object. The EM algorithm and its initialization method are presented to estimate the control points of the B-spline curve as well as the RMMs. The performance of the proposed shape estimation algorithm is evaluated using real measurements of a red dock line in 6 different configurations. The proposed algorithm is also used to estimate the shapes in scenarios such as the continuous manipulation and the assembly of elongated deformable objects. From the experimental results, it can be concluded that the B-spline curve chained RMM algorithm is capable of estimating the shape of the elongated deformable object configured as intersecting and non-intersecting shapes. The case when the rope has more than one intersection, has knots or is piled up will be studied in the future work.

References

1. A. Shah, L. Blumberg, and J. Shah, "Planning for manipulation of interlinked deformable linear objects with applications to aircraft assembly," *IEEE Transactions on Automation Science and Engineering*, no. 99, pp. 1–16, 2018.
2. A. Zea, F. Faion, and U. D. Hanebeck, "Tracking elongated extended objects using splines," in *19th International Conference on Information Fusion (FUSION)*, 2016, pp. 612–619.
3. J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.
4. R. C. Jackson, R. Yuan, D.-L. Chow, W. S. Newman, and M. C. Çavuşoğlu, "Real-time visual tracking of dynamic surgical suture threads," *IEEE Transactions on Automation science and Engineering*, vol. 15, no. 3, pp. 1078–1090, 2017.
5. T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, 2018.
6. J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1130–1137.
7. A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3d elastic objects with an RGB-D sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3914–3921.
8. S. Javdani, S. Tandon, J. Tang, J. F. O'Brien, and P. Abbeel, "Modeling and perception of deformable one-dimensional objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1607–1614.
9. D. De Gregorio, G. Palli, and L. Di Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Asian Conference on Computer Vision (ACCV)*, 2018, pp. 662–677.
10. W. H. Lui and A. Saxena, "Tangled: Learning to untangle ropes with rgb-d perception," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 837–844.
11. T. Matsuno and T. Fukuda, "Manipulation of flexible rope using topological model based on sensor information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 2638–2643.
12. G. Yao and A. Dani, "Image moment-based random hypersurface model for extended object tracking," in *20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–7.
13. G. Yao, K. Hunte, and A. Dani, "Image moment-based object tracking and shape estimation for complex motions," in *American Control Conference (ACC)*, 2018, pp. 5819–5824.
14. M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, 2011.
15. C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines.* springer-verlag New York, 1978, vol. 27.
16. E. T. Lee, "Choosing nodes in parametric curve interpolation," *Computer-Aided Design*, vol. 21, no. 6, pp. 363–370, 1989.
17. C. M. Bishop, *Pattern recognition and machine learning.* springer, 2006.
18. H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud." *ACM Trans. Graph.*, vol. 32, no. 4, pp. 65–1, 2013.
19. I.-K. Lee, "Curve reconstruction from unorganized points," *Computer aided geometric design*, vol. 17, no. 2, pp. 161–177, 2000.
20. R. Sedgewick and K. Wayne, *Algorithms.* Addison-Wesley Professional, 2011.