2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Kinematic Multibody Model Generation of Deformable Linear Objects from Point Clouds

Markus Wnuk*, Christoph Hinze, Armin Lechler and Alexander Verl

***Abstract*— Control and localization of deformable linear objects (DLOs) require models to handle their deformation. This paper proposes an approach to automatically generate a model from available visual sensor information. Based on point cloud data obtained from a 3D stereo camera, the kinematics of a multibody model formulation are derived. The approach aims to balance the tradeoff between computational complexity and model accuracy. This is achieved with a geometric error criterion that reduces the introduced degrees of freedom (DOF) of the model to a necessary minimum, representing the continuous shape with as few bodies as possible. The approach is evaluated analytically and validated with an experimental scenario of DLO manipulation.**

I. Introduction

Whether in industrial manufacturing processes, medical applications or in everyday life, deformable linear objects (DLOs) such as cables, wires, or hoses, are a very important class of objects, frequently encountered in our environment. Potential applications for robotic handling of DLOs include wire harness assembly for electronic devices, cable routing in switch cabinets, mounting of wire harnesses in the automotive industry, or even surgical assistance. However, implementing robotic manipulation for such tasks is difficult to achieve [1], [2].

This can be attributed to the deformation behaviour of DLOs. When a robot manipulates a DLO it deforms as exemplary shown in Fig. 1, which makes the automation of manipulation tasks highly challenging. Interactions and contacts with the object result in deformations that change the shape, vary the physical properties and lead to occlusions complicating the design of stable control algorithms or robust state estimators. Therefore, many approaches rely on models to capture the deformation behaviour in order to support the robotic system with additional information about the physical properties of the manipulated object during handling.

A commonly used approach is discretizing the DLO into several rigid segments interconnected by joints to reduce the infinite-dimensional configuration space of the continuous DLO to a distinct set of degrees of freedom (DOF) [3], [4]. Such a multibody representation of the DLO is a simplified approximation of the real world, but it allows to consider the kinematics, dynamics and penetration constraints the object is subjected to during manipulation [5].

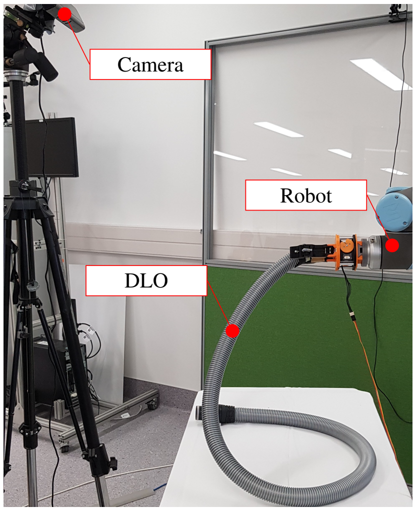


Fig. 1: Example of a robot manipulating a DLO. The scene is observed by a 3D stereo camera.

While many approaches have considered to improve the model's accuracy by estimation of dynamics parameters, not much attention has been given to the identification of the underlying kinematics. However, the kinematics play an important role for model based approaches on control and tracking of DLO with regards to accuracy and efficiency. Depending on the introduced number of DOF the accuracy rises, but so does the computational complexity. Therefore, the number of joints and segment lengths of the discretization need to be considered. Within this paper, we introduce a novel approach to derive a kinematics formulation from sensor data acquired during manipulation.

II. State of The Art

One of the earliest approaches addressing robotic manipulation of DLO is presented in [6]. This work was later extended for grasping point determination [7] and position control [8]. These early approaches introduce the idea to combine of a model of the DLO with visual information obtained from a camera. Despite this idea proved effective for certain applications, the approaches remain limited to a two-dimensional (2D) configuration space where the DLO is constrained to move in a plane.

Since then, many model-based control approaches for robotic manipulation of DLO emerged. They yield the advantage to provide additional information about the geometrical

The research leading to this publication has received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the International Research Training Group "Soft Tissue Robotics" (GRK 2198/1) and under Germany's Excellence Strategy EXC 2075 390740016.

The authors are with the Institute for Control Engineering of Machine Tools and Manufacturing Units, University of Stuttgart, Seidenstr. 36, 70174 Stuttgart, Germany

* Corresponding author. markus.wnuk@isw.uni-stuttgart.de

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

and physical properties of the manipulated object, which can be used for planning and control of the manipulation [9]. In practical applications these models are combined with available measurement data to track the deformation [10]. However, this requires well parametrized models to allow for robust state estimation.

Previous research addressed this problem by using parameter estimation schemes, aiming to adapt the model to the task by sensor data. Offline estimation of model parameters by measuring only the interaction between the robot and the manipulated object was proposed by [11] and [12]. In [13] a controller is presented that relies on an adaptive deformation model where the parameters are estimated online during manipulation. The approach avoids the identification of the underlying geometrical or physical relations by only using visual detectable feature points of the object. Yet, this an approach neglects information about the physical properties of the manipulated DLO and its environment, making preliminary planning of manipulation steps infeasible, limiting the approach to find only locally optimal trajectories during manipulation. [14] have shown that multibody models are a suitable choice for planning and control of DLO manipulation because they are capable to capture geometrical as well as physical properties. They successfully control a dynamic movement of a DLO by planning a trajectory based on the model information. However, their approach remains restricted to a 2D scenario where the DLO is freely hanging in front of a camera. Practical problems such as collisions or contact with the environment are neglected.

Early works, such as [15] and [16], show the significance of considering contact when manipulating DLOs and provide an understanding of how geometrical information of the DLO can be used to infer on contact situations with visual information. But, when a three-dimensional (3D) configuration space of the DLO is allowed, occlusion becomes a major problem because obtaining an unambiguous representation of the DLO's shape from visual data tends to be difficult. [17] reconstruct the geometry of a DLO from point cloud data under self-occlusion. Using Iterative Closest Point (ICP) they combine several point clouds from a stereo camera taken from different perspectives and then estimate the central axis of the DLO with geometrical point-chain model. However, their model does not consider physical properties. Approaches that also include physical information have been presented in [5], [18], [19]. These approaches rely on multibody simulations and match the observed point cloud data with the simulation with a probability based registration method. This allows to track the geometry of the DLO with an especially high robustness against disrupted sensor data. Yet, their focus is on the registration method, while the required multibody model is to be defined manually without closer consideration of the underlying kinematics. But considering the kinematics is crucial as each introduced DOF has to be registered to the measured data and, hence, increases the computational complexity.

III. Model Generation

The objective of the proposed model generation concept is to allow robotic systems to identify a suitable kinematic multibody description of a DLO autonomously during manipulation. This means to choose the number and location of the DOF such that the deformation behaviour of the manipulated DLO is captured accurately enough while maintaining as much computational efficiency as possible. The concept to achieve this trade-off is shown in Fig. 2. It consists of two main components: 1) a skeletonization algorithm to obtain a continuous representation $\boldsymbol{y}$ of the DLO from a acquired point cloud $\mathcal{P}_{ac}$ and 2) a discretization step to obtain a multibody representation while limiting the introduced geometrical error with an error criterion.

A. Representation of the Deformable Linear Object

The DLO can be represented by a line given by the geometric center for each infinitesimal thin cross-section of the DLO. We call this the skeleton line and model it as a continuous spatial curve in Euclidean space $\boldsymbol{f}(s) : \mathbb{R} \mapsto \mathbb{R}^3$ with local coordinate $s \in [0, 1]$, running along the length L of the DLO. The curve can be described with a local FrenetSerret frame as

$$\boldsymbol{R}(s) = \begin{bmatrix} \boldsymbol{f}'(s) & \boldsymbol{f}''(s) & \boldsymbol{f}'(s) \times \boldsymbol{f}''(s) \end{bmatrix},$$

where $\boldsymbol{f}'(s)$ is the tangent to $\boldsymbol{f}(s)$, and $\boldsymbol{f}''(s)$ is an orthogonal vector to $\boldsymbol{f}'(s)$ pointing to the center of the osculating circle to $\boldsymbol{f}(s)$.

Our goal is to obtain a geometric approximation of the continuous skeleton line by representing it as multibody model. The model's kinematics are therefore described as a kinematic chain consisting of a number of n rigid segments of length l_i interconnected by $n + 1$ flexible joints (start and end point are regarded as joints as well). Following the notation of [20], the kinematics for this model are then given by $i = 1, 2, ..., n$ homogeneous transformations

$$^{W}\boldsymbol{T}_i = \begin{bmatrix} ^{W}\boldsymbol{R}_i & ^{W}\boldsymbol{r}_i \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{1}$$

where$^{W}\boldsymbol{R}_i$ is the rotation between the two coordinate systems and $^{W}\boldsymbol{r}_i$ is the respective translation from the origin of the world coordinate system to the body-fixed coordinate system at the center of each segment.

The origin of each segments local coordinate system can directly be expressed in the joint coordinates, as

$$^{W}\boldsymbol{r}_i = \boldsymbol{x}_i + \frac{1}{2}(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i). \tag{2}$$

Since this representation is fully described by the xyz-positions of the joints $\boldsymbol{x}_i$, we are interested in obtaining these joint coordinates from the continuous skeleton line $\boldsymbol{f}(s)$.

However, the skeleton line is only accessible indirectly by available sensor data which are given as a point cloud of the surface of the observed DLO. This set of points is thus described as

$$\mathcal{P}_{ac} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_M\}, \in \mathbb{R}^{Mx3}. \tag{3}$$

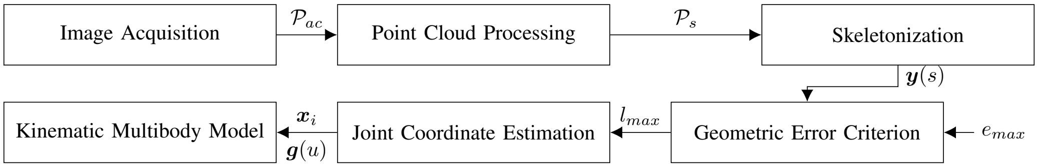


Fig. 2: Overview of the concept of the model generation.

Please note, that for model generation an absolute perspective on the DLO is necessary. This assumes the point cloud is a sufficiently dense representation of the DLO containing the whole object.

B. Point Cloud Preprocessing

The obtained point cloud $\mathcal{P}_{ac}$ represents the surface of the DLO, while our desired representation is given by center of the object's cross-section. Therefore, the acquired point cloud is projected onto the center of the DLO.

For the projection the points of $\mathcal{P}_{ac}$ are projected by the radius r of the DLO along their surface normals $\boldsymbol{n_i}$ to obtain the projected point cloud

$$\mathcal{P}_p = \{\boldsymbol{p}_p : \boldsymbol{p}_{p,i} = \boldsymbol{p}_{m,i} - r\boldsymbol{n_i}\}. \tag{4}$$

The set of surface normals $\mathcal{N} \in \mathbb{R}^{Mx3}$ can be obtained by least-squares fitting of a local plane using the k nearest neighbors of each point [21], or by using all neighbours within a certain radius [22]. For DLO with sufficiently small radius the projection step can be neglected, as the observed point cloud is already close to the center of the cross-section.

The projected point cloud data $\mathcal{P}_p$ can not directly be used to estimate the continuous skeleton line as the point cloud is usually not ordered, so there is no mapping between the obtained points and the local coordinate s of the skeleton line. Hence, the point cloud needs to be sorted to get the values of the skeleton line coordinate s corresponding with the projected points $\boldsymbol{p}_p \in \mathcal{P}_p$.

To approximately sort the points, a subset within the Euclidean distance r_n around an arbitrary chosen initial point $\boldsymbol{p}_0 \in \mathcal{P}_p$ is chosen as

$$\mathcal{P}_\epsilon(\boldsymbol{p}_0) = \{\boldsymbol{p} \in \mathcal{P}_p : \|\boldsymbol{p}_0 - \boldsymbol{p}\| \leq r_n\}. \tag{5}$$

A fast search of the neighbours around $\boldsymbol{p}_0$ is achieved by organizing the points in an efficient structure as e.g. k-d tree.

For a sufficiently large search radius (larger than the diameter D_{DLO} of the DLO: $r_n > D_{DLO}$) the selected subset $\mathcal{P}_\epsilon(\boldsymbol{p}_0)$ has a local direction vector $\boldsymbol{d}$ that approximates the tangent of the continuous formulation $\boldsymbol{f}'(s)$. To find the local direction in a least-squares sense we formulate it as the minimizing argument of

$$\begin{aligned} \boldsymbol{d} &= \arg\min_{\boldsymbol{d}} \sum_i (\langle \boldsymbol{n}_i, \boldsymbol{d} \rangle)^2 = \arg\min_{\boldsymbol{d}} \langle \boldsymbol{N}\boldsymbol{d}, \boldsymbol{N}\boldsymbol{d} \rangle \\ &= \arg\min_{\boldsymbol{d}} \|\boldsymbol{N}\boldsymbol{d}\|, \qquad (6) \\ &\text{s.t.} \quad \langle \boldsymbol{d}, \boldsymbol{d} \rangle = 1, \end{aligned}$$

where $\boldsymbol{n}_i \in \mathcal{N}_\epsilon(\boldsymbol{p}_0)$ and $\boldsymbol{N}$ is the matrix of stacked local normal vectors whose i-th row is given by $\{\boldsymbol{N}\}_i = \boldsymbol{n}_i^{\mathrm{T}}$.

The solution is obtained with a singular value decomposition (SVD) of the matrix of local normals $\boldsymbol{N}$ as

$$\boldsymbol{N} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}},$$

where the column of V corresponding to the smallest singular value solves Equation 6 according to the min-max principle. The topological order of the point clouds is then obtained by estimating a set $\mathcal{S}_s$ of local material coordinates s_s, sorting the points along the length of the DLO. The s_s coordinates are estimated by projecting any point $\boldsymbol{p}_i \in \mathcal{P}_\epsilon(\boldsymbol{p}_0)$ onto its locally neighbouring line $\boldsymbol{l}_s = \boldsymbol{p}_0 + d$ as shown in Fig. 3. The sorting procedure iterates in both directions of the DLO by choosing the point within a search radius r_n with the most positive and most negative coordinate value $s_{s,E}$ as new starting point $\boldsymbol{p}_0$. The result is a set of points $\mathcal{P}_s$, which are sorted along the length of the skeleton line, yielding a approximation s_s of the corresponding s-coordinate values of the points.

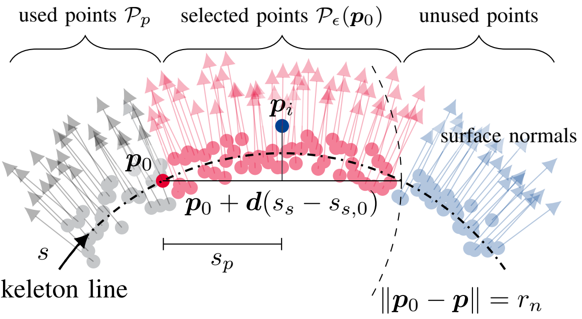

used points $\mathcal{P}_p$
selected points $\mathcal{P}_\epsilon(\boldsymbol{p}_0)$
unused points
$\boldsymbol{p}_i$
surface normals
$\boldsymbol{p}_0$
$\boldsymbol{p}_0 + \boldsymbol{d}(s_s - s_{s,0})$
s
skeleton line
s_p
$\|\boldsymbol{p}_0 - \boldsymbol{p}\| = r_n$

Fig. 3: Approximation of the order of points along a coordinate s_s. Starting from point $\boldsymbol{p}_0$, all points within search radius r_n are projected on a line whose direction $\boldsymbol{d}$ is determined perpendicular to the estimated surface normals.

C. Skeletonization

The set of sorted point cloud data $\mathcal{P}_s$ is used to identify the skeleton line $\boldsymbol{f}(s)$. The skeleton line is build from weighted sum of N basis functions as

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{W}, s) = \sum_{j=1}^{N} \{\boldsymbol{W}\}_j^{\mathrm{T}} \phi_j(s), \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{N \times 3}$ row-wise contains the weights in each spatial direction for each basis function $\phi_j(s)$.

A basis $\boldsymbol{\Phi} = \{\boldsymbol{\Phi} : \{\boldsymbol{\Phi}\}_j = \phi_j(s), j \in 1,2,...,N\}$ can generally be obtained with any orthogonal function space, e.g. with sinusoidal basis functions as in [7]. We use radial basis function (RBF) due to the proven property that a weighted sum of RBFs is able to approximate any $\mathcal{C}^1$ continuous function on a compact subset with arbitrary precision [23], [24]. Specifically, inverse quadratic functions defined as

$$\phi_j(s, \mu_j, \tau) = \frac{1}{1 + \tau^2 \|s - \mu_j\|^2} : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^3, \quad (8)$$

are used, where the width τ of each basis is fixed and the centers μ_j are evenly spaced along the material coordinate $s \in [0,1]$.

This allows to find the skeleton line, defined by the M measured points $\boldsymbol{p}_{p,i} \in \mathcal{P}_s$, by choosing the weights $\boldsymbol{W}$ so that they minimize the squared error

$$\begin{aligned}\boldsymbol{W} &= \arg\min_{\boldsymbol{W}, s_i} \frac{1}{2} \sum_{i=1}^{M} \langle \boldsymbol{e}_i, \boldsymbol{e}_i \rangle \\ &= \arg\min_{\boldsymbol{W}, \boldsymbol{s}} \frac{1}{2} \sum_{i=1}^{M} \|\boldsymbol{p}_{p,i} - \boldsymbol{f}(\boldsymbol{W}, s_i)\|_2^2. \quad (9)\end{aligned}$$

However, the coordinates s_i, which correspond to measured points $\boldsymbol{p}_{p,i}$, are generally unknown. Hence, the squared error depends not only on the weights $\boldsymbol{W}$ but also on the corresponding local coordinates $\boldsymbol{s}_i$ of every measurement point.

Under the assumption that the sorting yields good estimates for the corresponding location in material coordinates s, we are free to use the estimated s_s values and set $s_i = s_s$. This relation holds a mapping of inputs and measured outputs $(s_i, \boldsymbol{p}_{p,i}), i = 1,2,...,M$, such that the weights $\boldsymbol{W}$ remain the only unknowns. Now the skeletonization can be solved by a linear regression.

With known s_i Equation 7 becomes linear in its weights. We can formulate a feature matrix by evaluating the basis $\boldsymbol{\Phi}$ at all estimated s_i

$$\begin{aligned}\boldsymbol{\Phi}_s &= \begin{bmatrix}\boldsymbol{\Phi}(s_1) & \dots & \boldsymbol{\Phi}(s_M)\end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix}\phi_1(s_1) & \dots & \phi_n(s_1) \\ \vdots & \ddots & \vdots \\ \phi_1(s_M) & \dots & \phi_n(s_M)\end{bmatrix} \quad \in \mathbb{R}^{M \times N}\end{aligned}$$

and a measurement vector $\{\boldsymbol{Y}\}_i = \boldsymbol{p}_{p,i}^{\mathrm{T}}$ with columns $\boldsymbol{Y}_{\{x,y,z\}}$ for each Cartesian direction.

Then the regression reads

$$\boldsymbol{W}_c = \arg\min_{\boldsymbol{W}_c} \frac{1}{2} \|\boldsymbol{\Phi}_s \boldsymbol{W}_c - \boldsymbol{Y}_c\|_2^2 + \beta \|\boldsymbol{W}_c\|_2^2, \quad (10)$$

for $c \in \{x,y,z\}$ with β as parameter for L_2-regularization of the weights to avoid overfitting [25]. We choose β to minimize the variance of the residuals. The solution of Equation 10 can then be given analytically in its closed form as

$$\boldsymbol{W}_c = (\boldsymbol{\Phi}_s^{\mathrm{T}} \boldsymbol{\Phi}_s + \beta \boldsymbol{I}_N)^{-1} \boldsymbol{\Phi}_s^{\mathrm{T}} \boldsymbol{Y}_c, \; c \in \{x,y,z\}. \quad (11)$$

Substituting the calculated weights in Equation 7 yields a continuous representation of the DLO obtained from the measured point cloud data.

D. Geometric Error Criterion

To substitute the continuous representation of a DLO with serially connected rigid segments, the skeleton line has to be approximated with a piecewise-linear approximation. We require all segments to be of equal length to preserve the homogeneous deformation behaviour along the length of the DLO in the derived kinematics. Allowing segments of different length would imply a varying kinematic rigidity and, hence, contradict the assumption that the properties, such as stiffness or second moment of inertia, do not vary along the DLO's one-dimensional entity. The only exception is the length of the last element to compensate for the deviation of lengths between the continuous and piecewise-linear representation.

The piecewise-linear representation $\boldsymbol{g}(u)$ with local coordinate u consists of the union of n piecewise-linear functions $\boldsymbol{g}_i$. It is expressed as

$$\boldsymbol{g}(u) = \bigcup_{i=1}^{n} \boldsymbol{g}_i(u), u \in [0,1], i = 1,2,...,n \;, \quad (12)$$

with

$$\boldsymbol{g}_i(u) = \boldsymbol{x}_i + (u - u_i) \frac{L_{\sum}}{\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|} (\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) \quad (13)$$

and

$$u_i = \sum_{j=1}^{i} \frac{\|\boldsymbol{x}_{j+1} - \boldsymbol{x}_j\|}{L_{\sum}}. \quad (14)$$

Here, i denotes the number of the segment, u_i are the joint coordinates expressed in the local coordinate and $L_{\sum} = \sum_{i=1}^{n} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|$ is the length of the piecewise-linear approximation. This representation is solely dependent on the xyz-positions of the joint coordinates $\boldsymbol{x}_i$ in Euclidean space. Hence, to find $\boldsymbol{g}(u)$ to approximate $\boldsymbol{f}(s)$, the $\boldsymbol{x}_i$ have to be determined.

The geometrical error introduced by approximating the skeleton line with a piecwise-linear representation can be expressed as

$$e_{\mathrm{g}} = \|\boldsymbol{f}(s) - \boldsymbol{g}(u)\|. \quad (15)$$

Note that corresponding s and u are not trivial to define.

Therefore, we demand the joint coordinates $\boldsymbol{x}_i$ to lie on $\boldsymbol{f}(s)$ and assume that the skeleton line has constant curvature between to consecutive joint coordinates. This yields a worst case estimation of the occurring geometrical error which can then be measured as the maximum Euclidean distance perpendicularly from the center of the linear segment to the skeleton line, as depicted in Fig. 4. The error then depends on the local curvature of the skeleton line

$$\kappa = \frac{1}{r_\kappa} = \frac{\|\boldsymbol{f}'(s) \times \boldsymbol{f}''(s)\|}{\|\boldsymbol{f}'(s)\|^3}, \quad (16)$$

which is the reciprocal of the radius r_κ of the local osculation circle to $\boldsymbol{f}(s)$ and the length of the linear segment l_i. This geometric relation can be reformulated as a criterion for the maximum allowed segment length given for a maximum geometrical error.

$$l_{max}(e_{\mathrm{g,max}}, r_\kappa) = 2\sqrt{r_\kappa^2 - (e_{\mathrm{g,max}} - r_\kappa)^2}. \tag{17}$$

If the maximum bending curvature of a DLO is known a priori, the relation can be used to calculate the maximum allowed segment length in advance. Note, that a maximum segment length is desired, because it yields an approximation with fewer segments reducing the DOF of the multibody model representation and yielding lower computation times.

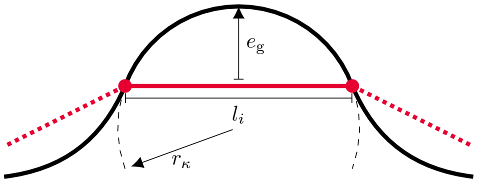


Fig. 4: Error criterion for the segmentation. A 2D representation of a segment of the skeleton line and the corresponding osculating circle.

E. Joint Coordinate Estimation

Given the maximum segment length, the joint positions x_i can be determined. Recall that the segments are required to have the same length, but the last segment is excluded from this requirement as described above. This allows to ensure similar start and end points for both representations.

The joint coordinates are calculated with an iterative procedure. Since, we require the joint coordinates $\boldsymbol{x}_i$ to lie on $\boldsymbol{f}(s)$ resulting in n geometric constraints

$$\boldsymbol{x}_i = \boldsymbol{f}(s_i), \tag{18}$$

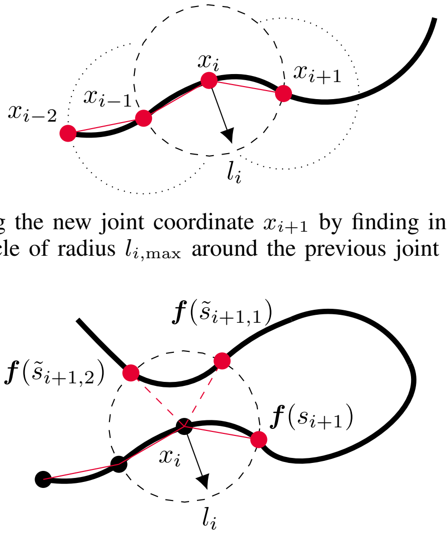
starting with $\boldsymbol{x}_0 = \boldsymbol{f}(s_0)$. From the maximum segment length l_{max} it is required that $\boldsymbol{x}_{i+1}$ and $\boldsymbol{x}_i$ have the Euclidean distance l_{max}, leading to

$$\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\| = \|\boldsymbol{f}(s_{i+1}) - \boldsymbol{f}(s_i)\| = l_{\mathrm{max}}. \tag{19}$$

Finding the points that meet this requirement means to find the s_{i+1} which satisfies Equation 19. This can be thought of as detecting the intersection of a sphere with radius l_{max} around the joint $\boldsymbol{x}_i$ with $\boldsymbol{f}(s)$ as shown in Fig. 5a. Locally, there should be only two solutions that satisfy Equation 19 yielding $\boldsymbol{f}(s_{i+1})$ and $\boldsymbol{f}(s_{i-1})$ for all $i \in [1, n-1]$. But depending on the global configuration of the DLO, it can intersect the sphere several times, resulting in even more solutions $\tilde{s}_{i+1}$, as demonstrated in Fig. 5b. To address this problem we choose s_{i+1} as the minimum of the set of all found solutions to Equation 19 while also excluding all previous solutions $s \leq s_i$. Hence, we propagate along s by

$$s_{i+1} = \arg\min_{s} \left\{ s \in (s_i, 1] : \|\boldsymbol{f}(s) - \boldsymbol{f}(s_i)\| = l_{\mathrm{max}} \right\}, \tag{20}$$

ensuring that the s_{i+1}, corresponding to the appropriate joint coordinate $\boldsymbol{x}_{i+1}$, is found. The search for a new s_{i+1} is repeated as long as we find new solutions for $s_{i+1} \leq 1$. In case there are no other solutions found except s_{i-1}, the end of the DLO's skeleton line has been reached, and the last joint can be set to $\boldsymbol{x}_{n+1} = \boldsymbol{f}(1)$. The desired joint coordinates are then given by substituting all found solutions s_i into Equation 18. The number of joint coordinates directly specifies the required number of segments to build the kinematics model. Substituting the found joint coordinates $\boldsymbol{x}_i$ into Equation 1 and Equation 2 results in the sought kinematics description.


x_{i-2}
x_{i-1}
x_i
x_{i+1}
l_i
$\boldsymbol{f}(\tilde{s}_{i+1,1})$
$\boldsymbol{f}(\tilde{s}_{i+1,2})$
$\boldsymbol{f}(s_{i+1})$
x_i
l_i

(a) Finding the new joint coordinate x_{i+1} by finding intersections with a circle of radius $l_{i,\mathrm{max}}$ around the previous joint coordinate x_i.

(b) Finding multiple solutions for the (i+1)-th joint

Fig. 5: 2D representation of the iterative procedure to find new joint coordinates.

IV. Evaluation

The evaluation of the concept is twofold. An analytical evaluation of the skeletonization and a validation of the model generation step. Both need an error metric that allows to quantify the occurring geometrical errors.

A. Error Metric

Several metrics can be considered for measuring geometric errors between different representations of DLOs, such as partial curve mapping, area between curves, discrete Fréchet distance, dynamic time warping (DTW), or curve length objective functions. Please refer to [26] for a detailed explanation. We base our evaluation on the metric of DTW, which we extend for the purpose of evaluation in 3D Euclidean space and therefore call modified dynamic time warping (MDTW).

DTW is a technique popular in pattern and speech recognition to determine the similarity of recorded signals. For a detailed overview refer to [27]. The definition of MDTW is given as follows: Assume two curves $\boldsymbol{v}$ and $\boldsymbol{q}$ of different lengths. We discretize the curves in H and G sample points $\boldsymbol{v}(s_i)$ and $\boldsymbol{q}(s_j)$, $i = 1, 2, ..., H$ and $j = 1, 2..., G$. Then, we find a mapping $w_l = (s_{i,l}, s_{j,l})$ with $l = 1, 2, ..., G$ between $\boldsymbol{v}(s_i)$ and $\boldsymbol{q}(s_j)$ such that the cumulative Euclidean distance d_{cml} is minimized according to

$$d_{cml}(w_l) = \min \sum_{l=1}^{K} \|\boldsymbol{q}(s_{j,l}) - \boldsymbol{v}(s_{i,l})\|, \tag{21}$$

where K is the number of tuples, which is called the length of the warping path.

The minimization is done in a two-step process by calculating the cumulative Euclidean distance for all tuples (s_i, s_j) in a cost matrix at first and then backtracking through the cost matrix to obtain the mapping w_l. The mapping w_l then determines the correspondence of all points $\boldsymbol{v}(s_{i,l})$ and $\boldsymbol{q}(s_{j,l})$. The modified error metric between two corresponding points is then defined as

$$e_{\text{MDTW},l} = \|\boldsymbol{v}(s_{i,l}) - \boldsymbol{q}(s_{j,l})\|. \tag{22}$$

Fig. 6 shows a graphical example of the introduced MDTW error metric.

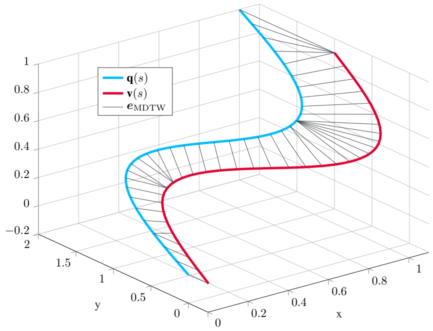
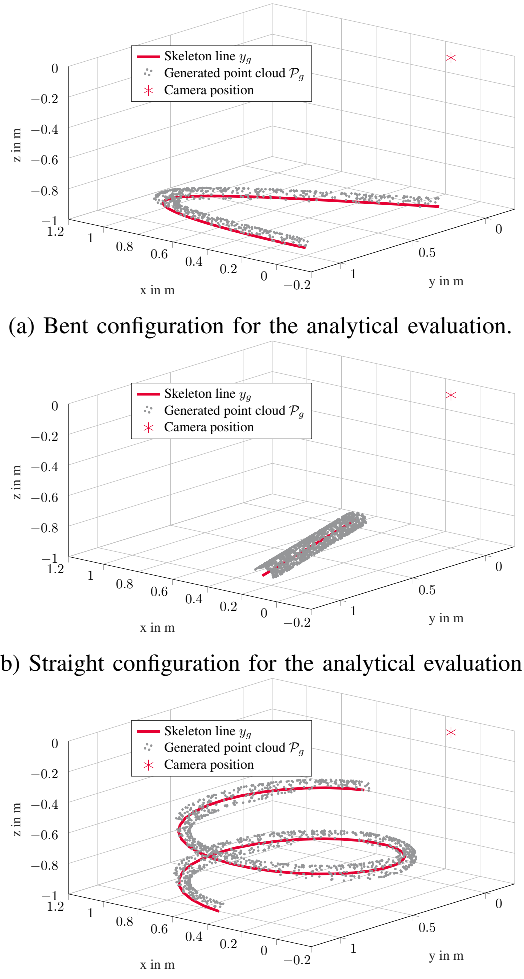


Fig. 6: Error metric defined by modified dynamic time warping (MDTW). The error is measured between points minimizing the cumulative distance between two curves.

To obtain an absolute measure quantifying the error over the whole length of the DLO, we use the statistical mean $\mu = \frac{1}{K}\sum_{i=1}^{K} e_{MDTW}$ and standard deviation $\sigma = \frac{1}{\sqrt{K-1}}\sum_{i=1}^{K}\sqrt{(e_{MDTW} - \mu)^2}$.

B. Analytical evaluation of skeletonization

Within the analytical evaluation the performance of the skeletonization is evaluated based on virtual generated point cloud data, which are obtained from predefined analytical skeleton lines. Three different configurations of skeleton lines $\boldsymbol{y}_g$ are created. We refer to these configurations as bent, straight and helical. The analytical functions corresponding to these configurations are given in Table I. From these skeleton lines, point clouds $\mathcal{P}_g$ with $M = 5000$ data points are generated under the assumption of a DLO diameter $d_{DLO} = 40$ mm. A graphical representation of the configurations is provided in Fig. 7. To each generated point cloud a skeleton line is fitted using the skeletonization with $N = 21$ basis functions, $k = N/4$ and $\beta = 10^{-2}$ as L_2 regularization parameter for the linear regression.



(a) Bent configuration for the analytical evaluation.

(b) Straight configuration for the analytical evaluation.

(c) Helical configuration for the analytical evaluation.

Fig. 7: Different generated configurations used for the analytical evaluation.

The accuracy of the skeletonization is measured by comparing the previously defined function $\boldsymbol{y}_g$ with the obtained skeleton line using the MDTW error metric. The results are summarized in Table II. We interpret the results with respect

TABLE I: Functions used for the analytical evaluation.

Config	Function
bent	$\boldsymbol{y}_g(s) = (\sin(\pi s), 2\cos(\pi s) - 1, 0)^T$
straight	$\boldsymbol{y}_g(s) = (-0.3s + 0.55, s - 1, 0)^T$
helical	$\boldsymbol{y}_g(s) = (\frac{1}{2}\sin(3\pi s) + \frac{1}{2}, \frac{1}{2}\sin(3\pi s) + \frac{1}{2}, \frac{1}{2}s - 1)^T$

TABLE II: Error between the analytical skeleton line and the obtained skeleton line by linear regression by discretizing 1000 sample points on each skeleton line.

Configuration	bent	straight	helical
μ of MDTW in mm	2.42	1.66	1.99
σ of MDTW in mm	2.22	3.63	1.89

to *SI*-units to remain referable to a practical application and to maintain comparability with the following experimental validation. We read an error of $10^{-3} = 1\,\text{mm}$.

Over all configurations, the maximum mean is $2.42\,\text{mm}$ and the maximum standard deviation is $3.63\,\text{mm}$. Note that the MDTW error hereby includes errors in length as well as errors in lateral displacement. Most of the error in our analytical evaluation can be attributed to an underestimation of the skeleton line's length by the linear regression. We noted that the straight configuration could not be approximated as well by as the two other configurations by the chosen RBF. This is displayed by the comparatively high standard deviation for this configuration. The occurring mean error can be attributed by a significant amount to incorrect point cloud projection due to errors in surface normal estimation. A refined preprocessing by additional removal of outliers in the projected point cloud could benefit the accuracy of the estimation.

C. Experimental validation

The setup for the experimental validation consists of a Microsoft Xbox 360 Kinect V1 for 3D point cloud acquisition, which is mounted about $80\,\text{cm}$ above the DLO. A vacuum hose with 40 mm diameter and 1.70 m length is used as test object. The model generation is performed in MATLAB on an Intel Core i7-6600U CPU with 2.60GHz and 12GB RAM. Refer to Fig. 1 for a depiction of the setup.

To experimentally evaluate the concept, we perform all individual steps of the model generation according to Fig. 2 in a practical manipulation scenario. The reference DLO is positioned by the robotic manipulator in front of the 3D camera such that the sensor has an absolute perspective on the DLO. Starting from the image acquisition the Kinect takes an image of the scene and provides a point cloud. From the point cloud of the scene the DLO is extracted via background subtraction and the point cloud is downsampled to $M = 5000$ points. The downsampled point cloud is given as input $\mathcal{P}_{ac}$ to the implemented multibody model generation framework. The generation is run with a maximum geometric error of $e_{\max} = 10\,\text{mm}$. Table III summarizes the input parameters and results of the experiment. For the linear regression the same parameters as in the previous analytical evaluation are used.

From the provided input information, the static configuration of the reference DLO is reconstructed. Fig. 8 shows the result of the model generation process. While Fig. 8a displays the obtained skeleton line, which is extended to a volumetric representation by projecting cross-sections of the radius r of the DLO along its length, Fig. 8b contains the equivalent, automatically generated multibody approximation consisting of $n = 16$ segments with $l_i = 10.59\,\text{cm}$ length. The multibody model representation is superimposed with the input point cloud data for reference.

To analyze the geometric error introduced by the segmentation the continuous representation $\boldsymbol{f}(s)$ is compared to the piecewise-linear approximation $\boldsymbol{g}(u)$ by MDTW. The results are depicted in Fig. 9, showing e_{MDTW} over the material coordinate s. As expected, the absolute error remains below the specified error maximum $e_{\max}$ over the whole length of the skeleton line. e_{MDTW} approaches the allowed maximum only at the area around $s_{\kappa,max} = 0.33$, where the DLO has the highest curvature.

TABLE III: Parameters and results of the experimental automated model generation.

Parameter	Symbol	Value	Unit
Input parameters			
DLO length	L_{DLO}	1.7	m
DLO diameter	D_{DLO}	40	mm
maximum geom. error	$e_{\max}$	10	mm
size of point cloud	M	$5 \cdot 10^3$	-
sorting search radius	r_n	160	mm
Results			
number of segments	n	16	-
segment length	l_i	10.59	cm
length of multibody model	$L_{\sum}$	1.65	m

V. Conclusion

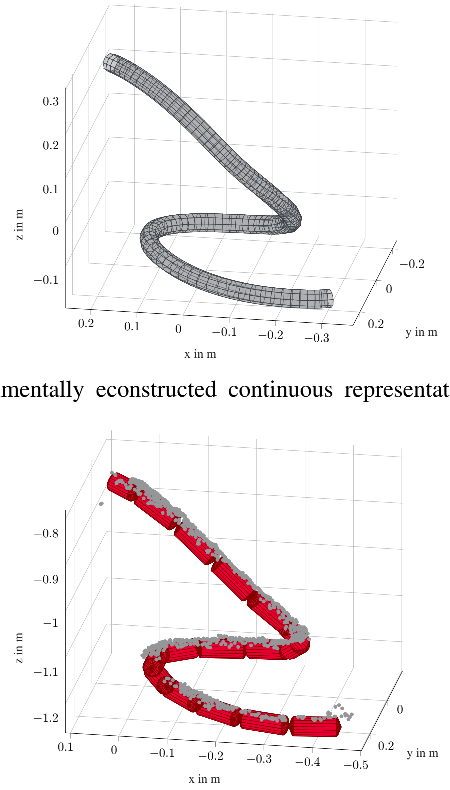
The practical evaluation shows that the proposed method is able to generate a multibody kinematics description from point cloud data without prior knowledge of the object. By approximately sorting the point cloud data along the DLO coordinate, a linear regression problem results and allows to efficiently extract the skeleton line. The proposed model generation takes physical properties into account by using the curvature of the DLO for the calculation of appropriate segment lengths to determine the resulting DOF of the model. With a practical experiment it could be shown, that the obtained kinematics model is able to represent the deformed DLO's geometrical shape within the specified error bounds. A current drawback of the proposed concept is that it requires a high quality data set of the DLO. Disrupted or highly noisy sensor data can lead to errors in the skeleton line estimation and subsequently to a incomplete model representation. Therefore, future works aims to strengthen the robustness of the model generation process. Furthermore, the generated model can be used to predict the physical deformation behaviour of DLOs for manipulation planning and control by adding dynamics parameters such as masses and inertia.

References

[1] A. Papacharalampopoulos, S. Makris, A. Bitzios, and G. Chryssolouris, "Prediction of cabling shape during robotic manipulation," *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 1-4, pp. 123–132, 2016.

[2] J. Zhuand, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2019.

[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.



(a) Experimentally econstructed continuous representation of the DLO.

(b) Experimentally reconstructed multibody model of the DLO. The acquired point cloud data are superimposed with the model.

Fig. 8: Representation of the continuous and reconstructed DLO.

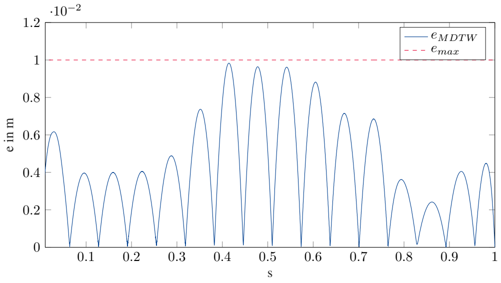


Fig. 9: MDTW error over the material coordinate between the estimated skeleton line and the linear-piecewise approximation.

[4] M. Wnuk, A. Pott, W. Xu, A. Lechler, and A. W. Verl, “Concept for a simulation-based approach towards automated handling of deformable objects a bin picking scenario,” in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2017, pp. 1–6.

[5] T. Tang and M. Tomizuka, “Track deformable objects from point clouds with structure preserved registration,” *The International Journal of Robotics Research*, 2019.

[6] H. Wakamatsu, S. Hirai, and K. Iwata, “Modeling of linear objects considering bend, twist, and extensional deformations,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, 1995, pp. 433–438 vol.1.

[7] H. Wakamatsu and S. Hirai, “Static Modeling of Linear Object Deformation Based on Differential Geometry,” *The International Journal of Robotics Research*, vol. 23, no. 3, pp. 293–311, 2004.

[8] H. Wakamatsu, K. Takahashi, and S. Hirai, “Dynamic modeling of linear object deformation based on differential geometry coordinates,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1028–1033.

[9] Olivier Roussel and Michel Taïx, “Deformable linear object manipulation planning with contacts,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[10] A. Petit, S. Cotin, V. Lippiello, and B. Siciliano, “Capturing deformations of interacting non-rigid objects using rgb-d data,” in *IROS 2018 - IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2018.

[11] T. Caldwell, D. Coleman, and N. Correll, “Optimal parameter identification for discrete mechanical systems with application to flexible object manipulation,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 898–905.

[12] T. Caldwell, D. Coleman, and N. Correll, “Robotic manipulation for identification of flexible objects,” in *Springer Tracts in Advanced Robotics*. Khatib O., Kumar V., Eds., 2016, vol. 109, pp. 133–147.

[13] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, “Automatic 3-D Manipulation of Soft Objects by Robotic Arms With an Adaptive Deformation Model,” *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[14] M. Hashimoto, T. Ichikawa, S. Inui, and Y. Horiba, “Dynamic manipulation of a string using a manipulator-parameter identification and control based on a rigid body link model,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 5, 2004, pp. 4815–4820.

[15] D. Henrich, T. Ogasawara, and H. Worn, “Manipulating deformable linear objects - contact states and point contacts,” in *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning*, 1999, pp. 198–204.

[16] J. Acker and D. Henrich, “Manipulating deformable linear objects: characteristic features for vision-based detection of contact state transitions,” in *Proceedings of the IEEE International Symposium on Assembly and Task Planning*. IEEE, 2003, pp. 204–209.

[17] K. Mukai, T. Matsuno, A. Yanou, and M. Minami, “Shape modeling of a string and recognition using distance sensor,” in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2015, pp. 363–368.

[18] J. Schulman, A. Lee, J. Ho, and P. Abbeel, “Tracking deformable objects with point clouds,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1130–1137.

[19] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, “State estimation for deformable objects by point registration and dynamic simulation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2427–2433.

[20] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Science+Business Media, 2008.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’92*. ACM Press, 1992, pp. 71–78.

[22] N. J. Mitra and A. Nguyen, “Estimating surface normals in noisy point cloud data,” in *Proceedings of the nineteenth annual symposium on Computational geometry*. ACM, 2003, pp. 322–328.

[23] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, “Layered Neural Networks with Gaussian Hidden Units as Universal Approximations,” *Neural Computation*, vol. 2, no. 2, pp. 210–215, 1990.

[24] F. Girosi and T. Poggio, “Networks and the best approximation property,” *Biological Cybernetics*, vol. 63, no. 3, pp. 169–176, 1990.

[25] A. Y. Ng, “Feature selection, L1 vs. L2 regularization, and rotational invariance,” in *Twenty-first international conference on Machine learning - ICML 04*. ACM, Jul. 2004, p. 78.

[26] C. F. Jekel, G. Venter, M. P. Venter, N. Stander, and R. T. Haftka, “Similarity measures for identifying material parameters from hysteresis loops using inverse analysis,” *International Journal of Material Forming*, vol. 11, no. 3, pp. 355–378, 2018.

[27] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1994, pp. 359–370.