

TSL: Tracking Deformable Linear Objects for Bimanual Shoe Lacing

Haining Luo , Graduate Student Member, IEEE, Yiannis Demiris , Senior Member, IEEE

Abstract—Robot manipulation of Deformable Linear Objects (DLOs) in tasks such as Shoe Lacing (SL) is challenging due to the objects' high dimensionality. One of the key components to successful manipulation is accurate state estimation. Existing methods focus on the DLO itself without considering the interaction with the scene objects. The estimated DLO relation with the scene objects is an important factor in task planning. In SL, the shoelace and shoe relation decides whether the shoelace should be inserted, and the target eyelet for insertion. In this work, we propose an algorithm, Tracking for Shoe Lacing (TSL), which combines probabilistic registration with physical simulation, to preserve the topological relations between the shoelace and the shoe. Leveraging the physical simulation, it also identifies system anomalies during the tracking process. We propose two evaluation metrics: *Sequential H-Signature*, which qualitatively describes the shoelace-shoe relation to assess algorithm performance, and Dynamic Time Warping (DTW) distance, which quantitatively evaluates tracking errors against the groundtruth. The result shows that TSL achieves tracking error on a par with the state-of-the-art algorithm, and it is the only method that correctly estimates the shoelace-shoe relations across all experimental conditions.

Index Terms—Perception for Grasping and Manipulation, RGB-D Perception, Visual Tracking.

I. INTRODUCTION

MANIPULATION of Deformable Linear Object (DLO) is common in everyday life. In DLO manipulation tasks, achieving an accurate 3D understanding of the DLO state is essential [1], [2]. Tasks such as cable harnessing and surgical suturing require precise control of the cable and thread states. Compared with rigid objects, DLOs exhibit more variations in their topology, which is an important factor to consider during tracking [3]. Previous works have tackled the preservation of the topology of the object itself from both the perspective of the local segment scale and the global structure [4]–[7]. However, in real-world applications, DLOs commonly interact with the surrounding environment and it is also important to consider its relation with the scene objects. For example, in surgical suturing, the success of an insertion is determined by whether the thread passes through appropriate areas of the wound. Similarly, in cable harnessing, the stability and durability of the structure depend on whether the cable properly loops around the nails or merely passes by them. The relation with the scene objects provides spatial understanding for task planning to produce realistic and executable plans.

Both authors are with the Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom (e-mail: haining.luo18, y.demiris@imperial.ac.uk). This research was supported by a Royal Academy of Engineering Chair in Emerging Technologies.

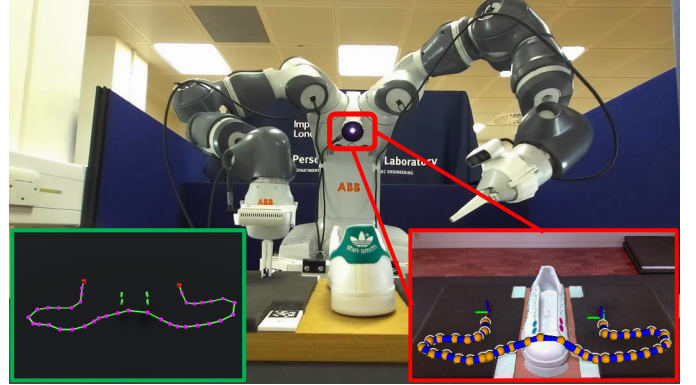


Fig. 1: TSL for shoelace tracking in robot SL. The DLO shoelace is represented by a set of ordered key nodes and the edges connecting adjacent points. The bottom left figure shows the physical simulation constructed for TSL, including the tracked shoelace and the eyelets. The estimated shoelace states are projected to the image space in the bottom right. The spheres indicate the tracked key nodes.

In this work, we focus on bimanual Shoe Lacing (SL), a representative DLO manipulation task that requires precise shoelace state control, as well as the understanding of the overall shoelace topology that dynamically evolves during the interaction with the shoe. Correctly estimating the shoelace-to-shoe relation is crucial for task planning, e.g., the estimated shoelace path should pass through eyelets after successful insertions; otherwise, appropriate recovery actions should be taken. Meanwhile, shoelace state tracking is challenging due to the constant occlusion from the shoe and the manipulator. Previous work on SL inaccurately estimates shoelace length with the distance between eyelets, which limits the success rate and has become one of the bottlenecks of the task [8].

To address the challenges, we propose Tracking for Shoe Lacing (TSL), an algorithm that tracks the shoelace states while preserving its topology relative to the scene objects. It combines probabilistic registration with physical simulation to enforce physical and topological constraints. The states of the shoelace, the shoe and End-Effector (EE), and are maintained and updates separately for efficiency, while the physical simulation also monitors anomalies such as excessive shoelace tension or insertion failures.

We compare the performance of TSL with existing methods [3], [6], [7] using human-demonstrated data, real robot data (Figure 1), and simulated data where we get the groundtruth for evaluation. Since the previous works do not consider the DLO-scene objects relations, there have been no standardised evaluation metrics. To qualitatively showcase the correctness of the estimated shoelace-eyelet relation, we

propose a representation, *Sequential H-Signature* on top of the existing *H-signature*. To numerically compare the tracking accuracy, we propose using DTW distance as the error metric. The results show that TSL achieves tracking distance error on a par with the state-of-the-art algorithm, and correctly estimates the shoelace-eyelet relation in all tested cases while the other algorithms do not produce the correct *Sequential H-Signatures*.

To summarise, our contributions are: 1) We tackle the novel task of shoelace tracking in SL with our proposed algorithm TSL, which combines probabilistic registration with physical simulation to correctly estimate the shoelace topology during tracking. 2) We propose two new performance metrics, a representation *Sequential H-Signature* of the DLO topological states for qualitative evaluation, and DTW distance for quantitative measures. 3) We evaluate the performance of TSL against baseline algorithms using the proposed signature with real robot, simulated and human demonstration data.

II. RELATED WORK

A. State Estimation of DLOs

Common state estimation approaches discretise DLOs as a set of ordered key nodes [3], where the DLO is represented by these nodes and the edges connecting them. DLO tracking is typically formulated as a point set registration problem, with one set representing the maintained key nodes (also referred to as state points) and the other consisting of point clouds from new observations.

During tracking, point clouds are aligned with the maintained state using point registration. Key nodes are treated as Gaussian centroids, with Gaussian Mixture Model (GMM) used to compute correspondences through probability distributions [9]. Node positions are then updated using Expectation-Maximization (EM), while Coherent Point Drift (CPD) is designed to preserve the overall structure of the DLO, referred to as the global topology [10]. To further regularise the local topology, Global-Local Topology Preservation (GLTP) was introduced using Locally Linear Embedding [11], [12]. Tang et al applied GLTP to DLO tracking, demonstrating the robustness of the estimator to occlusions [4].

However, probabilistic registration alone is not sufficient to produce physically plausible estimations [3]. Two research directions are derived here. The first incorporates dynamic simulation to enforce physical constraints [4], [13]. The second applies selected posterior constraints, either after the EM step or directly to the cost terms in the M-step, to restrict the DLO states, as demonstrated in Constrained Deformable CPD (CDCPD) [5] Constrained Deformable CPD 2 (CDCPD2) [6], and TrackDLO [7]. In contact-rich tasks such as SL, writing the constraints for the DLO-scene objects relations becomes challenging. In this work, we leverage existing physics solvers to detect collisions and preserve the topological relations between the shoelace and the shoe.

B. Topology in DLO tracking

The aforementioned methods tackle the preservation of the local and global structure but lack the consideration of interactions with the environment. This limitation arises primarily

from the tasks addressed. To this end, we introduce the concept of homotopy. Homotopy refers to a property of the DLO that remains invariant under continuous deformation [14]. Two paths are *homotopic* or belong to the same *homotopy class* if one can be continuously deformed into the other. In 3D space, homotopy classes can only be introduced by obstacles with one or more holes, or by obstacles stretching to infinity in two directions [14]. For tasks such as rope shaping, there is only one homotopy class. For SL, the eyelets create new homotopy classes. To constrain the movement of the tracked shoelace, we add the eyelets as obstacles in the physical simulation.

The concept of homotopy has long been studied in motion planning of mobile robots [15]. In the path planning of tethered robots, the homotopy classes of robot paths are identified to prevent entanglement [16], [17]. To avoid obstacles, *H-signature* is proposed to characterise the topological class of paths in a given environment [14], [18]. Mitrano et al further enhance this representation by incorporating the robot state into *H-signature* [19]. A similar representation is also used in guiding DLOs for insertion tasks [20]. While being extensively studied in motion planning, it cannot be directly applied to evaluate the tracking performance. The continuous signature values it provides for each of the obstacles do not reflect the sequence of the obstacles being passed through. In tasks like SL, the sequence of the eyelets being passed through is crucial for the success of the task. In this work, we leverage the *H-signature* to propose a new metric for the qualitative evaluation of the tracking performance.

III. THE TSL ALGORITHM

A. Structure of TSL

There are two types of topology that we consider: the topological property with respect to the DLO itself; as well as the properties with respect to the scene objects. In the context of SL, the DLO is represented by the shoelace, and the scene object corresponds to the shoe. TSL tackles this with 2 components: 1) The registration module which estimates the Gaussian centroids and variance given a new frame (Section III-B) while preserving the topology of the shoelace. 2) The physical simulation that maintains a SL environment and constantly updates its internal shoelace, EE and eyelet states based on the registration results (Section III-C). This module preserves the shoelace-eyelet relations and detects system anomalies (Section III-D).

The states of three components are maintained in TSL: the EE, the shoelace, and the eyelets. The shoelace is modelled as a series of key nodes, among which the manipulated ones follow the movement of the EE. As discussed in Section II-B, we represent the shoes with their eyelets in TSL. The eyelets are modelled as closed-loop points in 3D space with the pose derived from eyelet detection.

The state updates of the three components are performed in parallel to improve efficiency (Algorithm 1). 1) Prediction: the update of the EE states q . 2) Adjustment: the update of the shoelace state represented by the key nodes $Y = [y_1, \dots, y_M] \in \mathbb{R}^{M \times 3}$ based on new observed point cloud $X = [x_1, \dots, x_N] \in \mathbb{R}^{N \times 3}$, where M is the number of key

nodes and N is the size of the point cloud. 3) Correction: the update of the eyelet states Z .

Algorithm 1: The Tracking for Shoe Lacing algorithm

Input: New Observation
Output: Shoelace States

```

1 do prediction in parallel
2   Update EE target  $q$  from the observation
3   Calculate forces to update simulated EEs
4   Apply forces to EEs in simulation
5   Estimate shoelace tension
6 do correction in parallel
7   Detect eyelet poses  $Z$ 
8   Update  $Z$  in simulation
9   Check shoelace-eyelet relations
10 do adjustment in parallel
11   if Selected as key frame then
12     Get simulated states  $\hat{Y}$ 
13     Register states  $Y$  to new observation  $X$ 
14     Calculate particle regularisation forces
15     Apply forces to shoelace particles in simulation
16 return states

```

The registration function receives new observations and decides if this frame should trigger adjustment. It then applies for registration and estimates the new shoelace states, which are consecutively sent to the physical simulation for adjustment. Meanwhile, the EE states are updated in the simulation based on the robot commands; and the scene object states are updated based on the detected poses. The simulation environment constantly monitors the collision between the shoelace and eyelets and determines whether an insertion is successful.

B. Statistical Registration

We consider the problem of estimating shoelace states with new observation as non-rigid registration and use an EM algorithm to estimate the new states. We consider the key nodes $y_m (m \in 1, \dots, M)$ as the Gaussian centroids and the observed point cloud as the sampled points from the GMM. The probability distribution of $x_n (n \in 1, \dots, N)$ is:

$$p(x_n) = p(m)p(x_n|m), \quad (1)$$

where $p(m)$ is the membership probability of each Gaussian and $p(x_n|m)$ represents the distribution of the m th Gaussian. Consider that the registration is in a D dimensional space. Based on Bayes' theorem, the posterior probability distribution with the old (x_n, σ^2) is updated by the E-step:

$$p(m|x_n) = \frac{\exp\left(-\frac{\|x_n - y_m\|^2}{2\sigma^2}\right)}{\sum_{m=1}^M \exp\left(-\frac{\|x_n - y_m\|^2}{2\sigma^2}\right) + \frac{(2\pi\sigma^2)^{\frac{D}{2}}}{(1-\mu)^N}}. \quad (2)$$

We would like to maximise the log-likelihood of the observation which gives the cost function in the M-step:

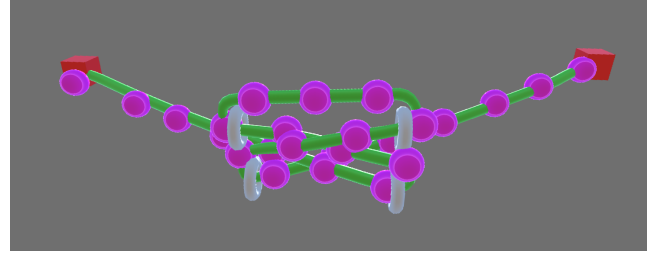


Fig. 2: Physical simulation environment of the proposed TSL. The simulated shoelace correctly passes through the eyelets. We simplified the shoelace and the shoe structure to speed up the simulation. The eyelets are represented as rings of particles. The spheres show the target key node positions and regulation forces are applied to adjust the shoelace particles towards them. The boxes indicate the EEs.

$$Q_{GMM} = \sum_{m=1}^M \sum_{n=1}^N p(m|x_n) \frac{\|x_n - y_m\|^2}{2\sigma^2} + \frac{N_p D}{2} \log(\sigma^2), \quad (3)$$

where $N_p = \sum_{m=1}^M \sum_{n=1}^N p(m|x_n)$.

To regularise the shoelace structure, we use a cost term for cross-frame consistency with Coherent Point Drift (CPD), and a term for centroid movement consistency with Locally Linear Embedding. We refer readers to [21] for the details. The final cost function to minimise is

$$Q = Q_{GMM} + Q_{CPD} + Q_{LLE}. \quad (4)$$

C. Physical Simulation

To maintain the shoelace-eyelet relation, we construct a physical simulation environment. The environment maintains the states of the EE, the shoelace, and the eyelets (Figure 2). The environment detects collisions between the shoelace and the eyelets and prevents the shoelace from penetrating the eyelets. The physical model of the shoelace also includes physics constraints such as stretching, friction and damping. Meanwhile, the simulation environment keeps the complete states of the shoelace and monitors system anomalies.

The states estimated by the registration module are statistically reasonable, but not restricted by physical laws. We use a physics engine to regulate the states. Equivalent to the posterior constraints in CDCPD2, our simulation environment includes physical constraints. Known correspondence points are restricted and treated as rigid kinematic bodies. Stretching constraints for the shoelace are added by minimising the displacement between particles along the inter-particle direction. Collision constraints are added by adjusting predicted contact points between a shoelace edge and other edges or objects. Apart from the constraints used in CDCPD2, bending constraints, friction and damping are also added. Using physics simulation expands the physics regulations that can be applied to the shoelace and allows physically interpretable parameters.

Due to the nature of physical simulation, the states of simulated objects cannot be changed instantly. Abrupt robot movements or shoelace state changes cause significant interruptions of the simulated physics. Therefore, we convert updating signals to regulation forces and pull the components into desired states. To avoid sudden drastic motion, we use a

PD force controller to regulate the movement. This reduces the *tunnelling effect* where the simulated shoelace penetrates through other objects. The simulated shoelace states are also regulated by PD force controllers.

During robot SL, the shoelace often remains stationary. The robot might place and re-grasp the shoelace to adjust its arm configuration. Tasks such as eyelet perception and motion planning contribute to the stationary periods. Meanwhile, if the registered shoelace configuration did not change noticeably due to slow robot motion or occlusion, it is not necessary to adjust the simulated states for every frame. To this end, we introduce the notion of keyframes. A keyframe is when the simulation updates the shoelace configuration. As we represent the shoelace with a GMM, the cost function Q_{GMM} reflects the quality of the estimated states to the current frame. We select the keyframes using its matrix form

$$\bar{Q}_{GMM} = \frac{1}{\sigma^2 N} (\text{Tr}(X^T d(P^T 1) X) - 2 \text{Tr}((PX)^T Y) + \text{Tr}(Y^T d(P1) Y)), \quad (5)$$

where $\text{Tr}(A)$ is the trace of a matrix A , and $d(a)$ is a diagonal matrix constructed from a vector a . A frame is selected when the cost is over an empirical threshold, depending on the fidelity of the simulated model. This reduces unnecessary updates and computation.

D. Anomaly Detection

The simulation environment provides complete states of simulated objects. In the case of SL, the physics engine can continuously monitor the relative positions of the shoelace and the eyelets; as well as the tension of the shoelace.

We define the eyelet coordinate systems so that the eyelet skeleton lies in XZ planes and Y is the vertical axis. By representing the shoelace particles in the eyelet spaces, shifts in the sign of particle Y coordinates indicate crossing over the XZ plane. An eyelet is passed through if this crossing lies within the skeleton on the XZ plane. The shoelace configuration Y that passes through an eyelet Z is represented as

$$Y_{passed} = \{\exists y_m \in Y \mid y_{m,y} \cdot y_{m-1,y} < 0, y_m \in \text{int}(Z)\}, \quad (6)$$

where $\text{int}(Z)$ is the interior of the eyelet Z .

Meanwhile, the physical simulation monitors the tension of the shoelace, represented by the strain along the tensile axis. This is calculated by

$$\text{Strain}(y) = \frac{\|y_{m+1} - y_m\|}{\|y_{m+1}^0 - y_m^0\|}, \quad (7)$$

$$\text{Strain} = \sum_{m=1}^M \text{Strain}(y_m). \quad (8)$$

The particle information within the physics engine can be easily retrieved and used for monitoring. Figure 3 shows the strain of the shoelace during an experiment. The red region indicates higher strain.

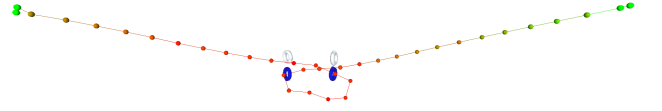
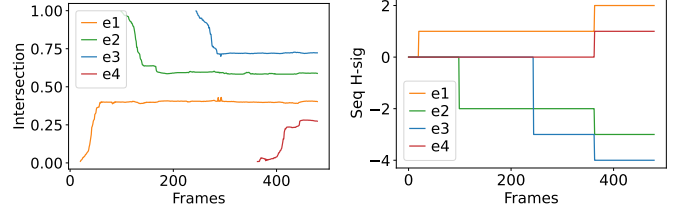


Fig. 3: Anomaly monitoring during an experiment. The blue obstacles have been passed through. The red region indicates higher strain.



(a) Intersections with eyelets

(b) Sequential H -Signature

Fig. 4: Example of intersections on the shoelace with the eyelets and Sequential H -Signature of the shoelace. The intersections range from 0-1 with 0 and 1 indicating the two ends of the shoelace. At frame 500 of this example, the shoelace in sequence passes through $e4, e1, e2, e3$. The signature gives the index of an eyelet within this sequence and its sign indicates the direction of the passing. At frame 500, the Sequential H -Signature is $[2, -3, -4, 1]$.

IV. EVALUATION METRICS

A. Sequential H -Signature

Although H -signature is designed to capture the topological relation to the scene objects, it does not provide the sequential information of the relations. For tasks such as SL, the order of obstacles that the DLO passes through is crucial for the success of the task. To evaluate the correctness of the estimated shoelace-eyelet relations, we propose a form of representation namely Sequential H -Signature (Figure 4).

Suppose a scene object skeleton S is a current conductive wire with unit current and $B(y)$ is its Virtual Magnetic Field over a point y on the DLO. Then we have

$$B(y) = \int_S \frac{(\vec{s} - \vec{y}) \times d\vec{s}}{\|\vec{s} - \vec{y}\|^3}. \quad (9)$$

H -signature $\mathcal{H}$ is defined as the line integral of the Virtual Magnetic Field along the DLO:

$$\mathcal{H}(S) = \int_Y B(y) \cdot dy. \quad (10)$$

If the absolute H -signature value is over 0.5, the DLO is considered to pass through the scene object. The key node y that produces the maximum Virtual Magnetic Field is the closest point y_{m^*} to the skeleton:

$$y_{m^*} = \begin{cases} \text{argmax}_y B(y), & |\mathcal{H}(S)| > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Suppose $M_{sorted}^* \in \mathbb{N}^R$ is the list obtained by sorting $M^* \in \mathbb{N}^R$ in descending order. We define $\mathcal{H}_{seq} \in [1, R] \cap \mathbb{N}$ as the rank of each m^* in M_{sorted}^* ,

$$\mathcal{H}_{seq} = \{\text{rank}(m^*) \mid m^* \in M^*\}. \quad (12)$$

Sequential H -Signature $\mathcal{H}_{seq}$ provides information on both the state of the DLO with respect to the scene object and the order in which the scene objects are encountered. An example is provided in Figure 4.

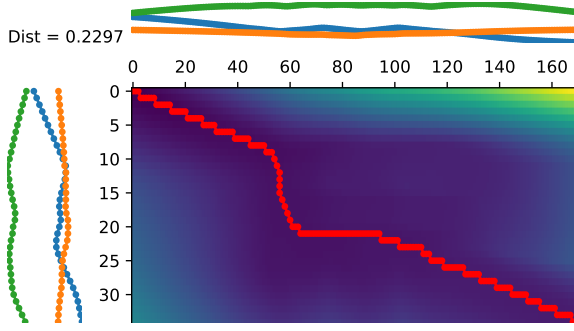


Fig. 5: DTW between estimated states and the groundtruth at a frame. The red diagonal line shows the correspondence between the individual nodes of the sequences. The DTW distance is the minimum cost of aligning the two state sequences.

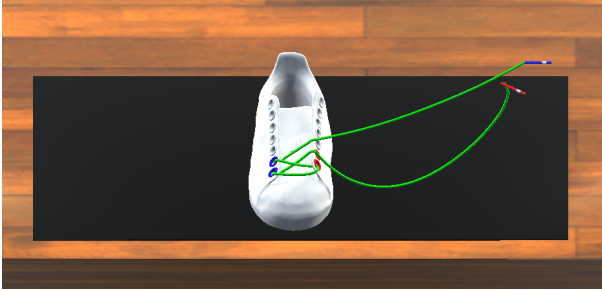


Fig. 6: Simulation environment used to collect simulated data [22].

B. Distance with Dynamic Time Warping

To evaluate the similarity between the estimated and ground truth states, we propose to use the DTW distance. DTW is a measure of similarity between two sequences that may vary in time or length (Figure 5). In the context of DLO tracking, the ground truth sequence Y and the estimated sequence $\hat{Y}$ often differ in the number of nodes and may not have a one-to-one correspondence. The DTW distance is defined as the minimum cumulative cost required to align the two sequences, where the cost of aligning two points is given by their Euclidean distance. The DTW distance is defined as:

$$\text{DTW}(Y, \hat{Y}) = \min_{\pi} \sum_{(i,j) \in \pi} \|y_i - \hat{y}_j\|, \quad (13)$$

where π is the warping path that minimises the cost.

V. EXPERIMENTS

In this section, we evaluate the performance of TSL, as well as the baseline algorithms CDCPD2, TrackDLO, and GLTP. The testing data is acquired from a bimanual robot, a simulated SL scene (Figure 6), and human demonstrations. The simulated data is used to verify the tracking accuracy using DTW. An analysis sweeping through ranges of physical parameters is conducted to examine their effects on the accuracy. In addition, a separate simulation dataset is used to study how tracking accuracy changes with an increasing number of eyelets.

A. Implementation

Accurately detecting and managing collisions between the shoelace and eyelets, as well as self-collisions of the shoelace,

is challenging due to the *tunnelling effect* [23]. At sufficiently high speeds, shoelace particles can penetrate through eyelets between consecutive frames. We construct the simulation environment with a multi-purpose game engine Unity¹ and a particle-based physics solver Obi² specialised for soft objects. Obi surface-based collision effectively reduces the *tunnelling effect* and provide reliable collision detection. To efficiently communicate between the simulation and the registration module with low latency, we use the cross-language communication Google Remote Procedure Call (gRPC) framework. The server is implemented in C++ with the registration module and the client in C# with the simulation environment.

We assume that the shoelace is distinctively coloured and free of tangles or occlusions at the beginning. This simplification could be relaxed in future work using data-driven methods [24]. Point clouds corresponding to the shoelace areas are segmented using colour thresholding. The initial states are estimated by extracting the skeletons of the mask and bridging the segments [7], [25]. All tested algorithms use the same initial states. In TSL, the estimated initial states are used to define the control particles and construct the Obi Rope. The EEs are simulated with points rigidly connected to the first and last particles of the shoelace. The eyelets are represented by a ring of interconnected particles with their normals pointing towards the inside of the shoe.

To collect simulated data, we use an environment from our previous work (Figure 6) [22]. We record the eyelet, aglet and shoelace particle positions as groundtruth. The shoe model is a 3D scan of the real shoe used in human demonstration. The other shoe from the same pair is used in robot data collection. The robot data is collected with the system proposed in our previous work [8], where a dual-arm ABB YuMi IRB 14000 is used to complete the task. The system combines PDDL-based high level reasoning for overall strategy planning and multimodal low-level sensorimotor primitives to execute the low-level actions. We record the eyelet poses detected by the system and EE poses computed from the joint states. For the human demonstration data, the eyelet poses and EE positions are manually labelled. Both the physical robot data and the human demonstration are recorded with Intel Realsense L515s. We lace 4 eyelets and record RGBD images in all cases. Data is recorded as *ROS Bags* and the algorithms are tested on a platform with an Intel 8700K and an Nvidia GTX 1080.

B. Qualitative Analysis with Sequential H-Signature

Figure 7 shows the results after lacing the second and fourth eyelets with all three data sources. Each section includes: (1) the frame image overlaid by estimated states; (2) a close-up view of the eyelet region (top right); (3) a 3D plot of the states around the eyelets (top left), where coloured eyelets indicate that the shoelace passes through, and white eyelets indicate otherwise; and (4) the *Sequential H-Signatures* that follow the same colour scheme as their corresponding eyelets (bottom).

The left two columns of Figure 7 shows the results with simulated data. Overall, TSL, TrackDLO and CDCPD2 achieved

¹Unity Engine: <https://unity.com/>

²Obi: <https://obi.virtualmethodstudio.com/>



Fig. 7: Tracking results after lacing the second and fourth eyelets. Each section provides the frame image overlaid by the estimated states. Close-up views around eyelets are displayed at the top right of a section. The top left of each section presents a 3D plot of the estimated states around the eyelets, where coloured eyelets indicate that the shoelace passes through, and white indicate otherwise. The *Sequential H-Signatures* follow the same colour scheme as their corresponding eyelets and are shown at the bottom of each section. TSL consistently produces the correct signatures across all cases, whereas the baselines estimate the shoelace to be outside the eyelets (zero in the signatures).

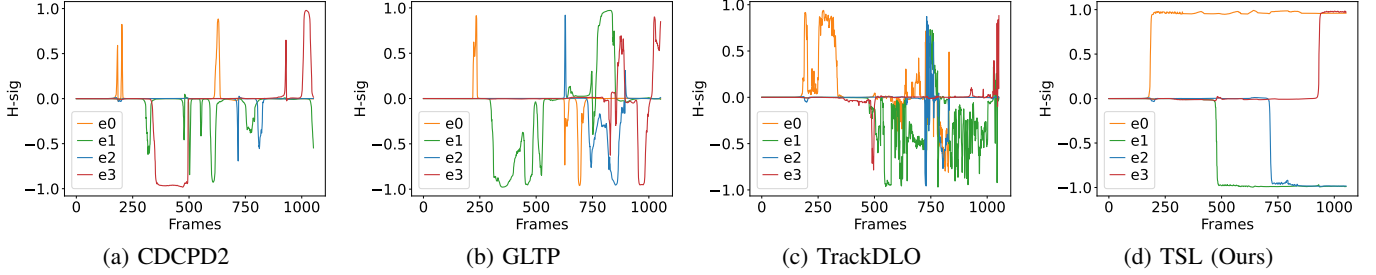


Fig. 8: *H-signature* of the shoelace with simulated data. Positive values indicate the shoelace is passing through the eyelet same as the eyelet's normal direction. The shoelace is considered to be passing through an eyelet if the absolute value is over 0.5. TSL maintains a stable shoelace-eyelet relation for all eyelets, whereas the baselines fail to preserve the shoelace paths through the eyelets.

similar levels of shoelace coverage. The close-up view for CDCPD2 shows the shoelace path does not cover all the eyelets. As shown in the top right 3D plots, the shoelace estimated by CDCPD2 does not pass through these eyelets. This is further verified by the *Sequential H-Signature* where CDCPD2 produces zeros for these eyelets. TrackDLO maintains better alignment between the estimated states and the actual shoelace compared to CDCPD2, but it also generates incorrect *H-Signatures*. The results from GLTP only partially align with the shoelace structure, and its *Sequential H-Signature* indicates that the shoelace does not pass through any eyelets.

To further examine the evolution of the shoelace-eyelet relationship, we plot the *H-signature* in Figure 8. The signature is 1 or -1 when the shoelace passes through the corresponding eyelet in the same or opposite direction of the eyelet normal. Although the *Sequential H-Signatures* produced by CDCPD2, TrackDLO, and GLTP are incorrect, their tracked shoelace paths do momentarily pass through the eyelets immediately after insertion. However, due to the absence of constraints to preserve the topology, the tracked shoelaces frequently penetrate the eyelets throughout the task. As a

result, the *H-Signatures* for CDCPD2, TrackDLO, and GLTP are unstable, whereas TSL consistently preserves the correct shoelace-eyelet relationship across all eyelets.

With respect to the human demonstration data (the middle two columns of Figure 7), the results are similar to the simulated condition. TSL correctly estimates the shoelace-eyelet relation and produces the correct *Sequential H-Signatures*, while the results of other algorithms fail to preserve the shoelace paths through the eyelets. CDCPD2 have key nodes overall aligned with the eyelet, but nodes near the eyelets are squeezed outside and edges near the shoelace tips are bent. This shows how CPD tries to maximise the coverage of the tracked nodes against the segmented shoelace masks. GLTP pushes the key nodes close to the shoelace and keeps a good coverage, however, the structure of the shoelace is lost. TrackDLO lost track of the tips affected by occlusion, where the motion of the visible part of the shoelace does not reflect the behaviour of the hidden region. In Figure 9a, we highlight critical frames in which the tip, along with a significant portion of the shoelace, is occluded, making it difficult to estimate the tip motion based on the visible shoelace components.

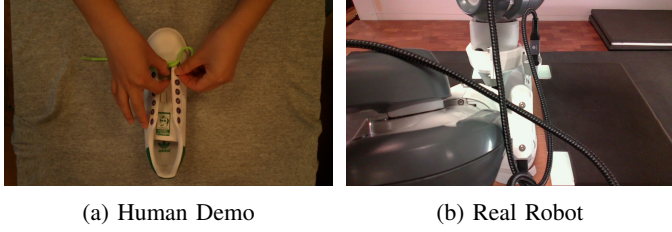


Fig. 9: Occlusions during the tracking process. In SL, the manipulators inevitably occlude the shoelace from different perspectives. TSL keeps the tracking process to continue with its simulated model and the EE movement.

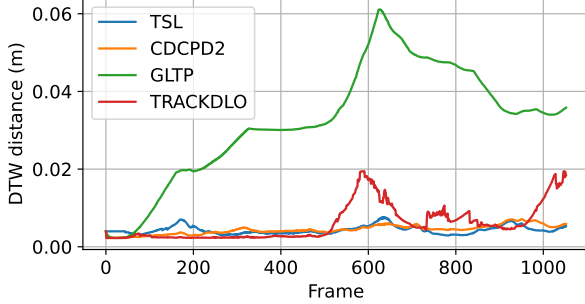


Fig. 10: DTW distance for all tested methods. TSL and CDCPD2 achieve similar levels of tracking error. Although CDCPD2 is not able to estimate the shoelace-eyelet relation correctly, the difference makes a small impact on the overall tracking error.

The bottom section of Figure 7 shows the results from the real robot data. Compared with the previous two experiments, all tested algorithms show worse performance in overlapping the tracked nodes with the shoelace. Although TSL kept the shoelace through the eyelets, the estimated structure does not completely align with the shoelace. The baseline algorithms also exhibited varying degrees of mismatch with the shoelace, which we attribute to severe occlusions and the robot’s fast motion. Due to the complexity of the SL task, the robot introduces significant occlusions regardless of camera placement. For instance, in Figure 9b, the shoelace is almost completely obscured from view. Meanwhile, YuMi EEs move at a speed of 300 mm/s which is significantly faster than the EEs in the other two experiments. This also caused severe motion blur in the RGB images, which decreased the quality of the segmented shoelace mask. The quality of depth information is also affected by the fast robot’s movement. The tracking accuracy can be potentially improved with slower motion, however, this will hinder manipulation performance.

C. Quantitative Analysis with the DTW Distance

To understand whether correctly estimating the shoelace-eyelet relation impacts the overall tracking error, we plot the DTW distance of the tested algorithms in Figure 10. While TSL achieves correct *Sequential H-Signatures*, overall, its tracking error is similar to that of CDCPD2, with mean values at 4.32 and 4.36 mm/node respectively. Without the EE information, TrackDLO achieves a 5.98 mm/node mean error. GLTP instead reached a mean error at 32.48 mm/node. Since the regions near the eyelets form a small portion of

TABLE I: Tracking Error (mm) with different simulation parameters: Particle Mass (kg), Particle Radius (m) and Max Bending (m). The values used to collect the simulation data are bolded.

Mass	0.001	0.002	0.003	0.004	0.005
Error	4.32 ± 0.10	4.48 ± 0.11	4.80 ± 0.12	5.05 ± 0.13	5.38 ± 0.15
Radius	0.001	0.00125	0.0015	0.00175	0.002
Error	4.48 ± 0.11	4.32 ± 0.10	4.32 ± 0.10	4.34 ± 0.10	4.36 ± 0.11
Bend	0.001	0.005	0.01	0.015	0.02
Error	4.51 ± 0.10	4.36 ± 0.10	4.32 ± 0.10	4.34 ± 0.10	4.33 ± 0.10

TABLE II: Tracking Error with different number of eyelets

# Eyelets	2	4	6	8	10
Error	4.75 ± 1.26	4.99 ± 1.31	4.98 ± 1.28	4.84 ± 1.35	4.69 ± 1.35

the shoelace, the performance difference in the shoelace-eyelet relation only makes a small difference to the tracking error.

The choice of physical parameters is an important factor in DLO simulation. In the previous experiments, we use the same parameter values as those used for generating the simulated data, and applied empirically tuned values for other datasets to maximise visual similarity with the real shoelace. To investigate whether inaccuracies in physical parameters affect tracking error, we conduct a sensitivity analysis by sweeping through ranges of three key physical parameters: particle mass, particle radius and max bending. Particle mass influences how particles respond to constraints such as collisions and bending. Particle radius affects the number of particles generated. Max bending defines the threshold at which the bending constraint becomes active, effectively controlling the shoelace’s stiffness.

Table I presents the results of this experiment, with the values used during data collection highlighted in bold. For the mass parameter, tracking error increases with higher mass values, as the regulating forces become less effective when particles are heavier. The mass could not be reduced further, as the shoelace becomes too light and oscillates under the regulating forces. The radius sweep was limited to a relatively narrow range due to constraints imposed by the eyelet size. Nonetheless, it is clear that tracking error increases as the radius deviates from the original value. For the bending parameter, which serves as the threshold for activating bending constraints, tracking becomes less accurate as the value decreases. Meanwhile, once the value exceeds that used in data collection, the error differences become negligible. This suggests that making the simulated shoelace softer than in the dataset does not lead to improved tracking accuracy.

In previous experiments, TSL was evaluated using 4 eyelets. To examine the impact of increasing homotopy classes on tracking accuracy, we tested TSL on a simulated dataset with 10 eyelets. As shown in Table II, the tracking error initially increases and then decreases as the number of eyelets grows. This trend is attributed to the latency between the tracked and actual states during fast movements. Initially, the tracked states lag behind the actual states, resulting in larger errors. However, as the SL progresses, more sections of the shoelace become constrained, or effectively ‘frozen’, between the eyelets. The influence of latency is reduced as the remaining free shoelace branches are shorter. As a result, fewer key nodes fall behind, leading to a reduction in overall tracking error.

From the time aspect, CDCPD2 runs at on average 83.76 ms/frame, GLTP at 71.20 ms/frame, TrackDLO at 52.83

ms/frame, and TSL at 114.18 ms/frame. Benefitting from the lightweight physics solver and the simplified simulation setup (Figure 2), an adjustment cycle in simulation uses only 25.00 ms. However, comparing with the baselines, the time cost for TSL remains one of its limitations. In future work, the time performance of the current implementation could potentially be improved with more software engineering optimising communication between the registration module and the simulation environment, e.g., more efficient serialisation to avoid conversion between Unity and Eigen data formats.

Overall, TSL correctly estimates the shoelace-eyelet relation in all experimental conditions while the other three algorithms are not able to produce the correct *Sequential H-Signatures*. In terms of the tracking error, TSL achieves a similar level of accuracy as the CDCPD2. With the support of TSL, future SL systems may improve their performances from two perspectives: 1) Task planning: Leveraging the estimation of shoelaces states and the relation with eyelets, systems can detect tangles or failures and act accordingly; 2) Motion planning: The estimated states and shoelace dynamic models can inform the planning of new actions. We hope that this algorithm will foster the future advancement of SL systems.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we present TSL for tracking shoelaces and correctly estimating their topology in the task of bimanual SL. The algorithm leverages statistical registration to estimate the states of the shoelaces and physical simulation to maintain the topology. The performance of proposed TSL is evaluated with simulated data, human demonstration data and real robot data. The experiments demonstrate that TSL successfully preserves the shoelace topology in all experimental conditions, with tracking error on par with the state-of-the-art CDCPD2. One of the potential improvements of this work is to learn the physical parameters. This can be done by training a neural network to estimate the parameter similarity through contrastive learning [26]. Other directions include improving the performance under heavy occlusion and image motion blur [27], [28], as well as exploring learning-based tracking methods [29].

REFERENCES

- [1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello *et al.*, “Challenges and outlook in robotic manipulation of deformable objects,” *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.
- [2] S. Zhaole, H. Zhou, L. Nanbo, L. Chen, J. Zhu, and R. B. Fisher, “A robust deformable linear object perception pipeline in 3d: From segmentation to reconstruction,” *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 843–850, 2023.
- [3] T. Tang and M. Tomizuka, “Track deformable objects from point clouds with structure preserved registration,” *The International Journal of Robotics Research*, vol. 41, no. 6, pp. 599–614, 2022.
- [4] 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 (IROS)*. IEEE, 2017, pp. 2427–2433.
- [5] C. Chi and D. Berenson, “Occlusion-robust deformable object tracking without physics simulation,” in *2019 IEEE/RSJ int. conf. on intelligent robots and systems (iros)*, 2019.
- [6] Y. Wang, D. McConachie, and D. Berenson, “Tracking partially-occluded deformable objects while enforcing geometric constraints,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 199–14 205.
- [7] J. Xiang, H. Dinkel, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl, “Trackdlo: Tracking deformable linear objects under occlusion with motion coherence,” *IEEE Robotics and Automation Letters*, 2023.
- [8] H. Luo and Y. Demiris, “Bi-manual robot shoe lacing,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8772–8775.
- [9] H. Chui and A. Rangarajan, “A feature registration framework using mixture models,” in *Proceedings IEEE workshop on mathematical methods in biomedical image analysis. MMBIA-2000 (Cat. No. PR00737)*. IEEE, 2000, pp. 190–197.
- [10] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [11] S. Ge, G. Fan, and M. Ding, “Non-rigid point set registration with global-local topology preservation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 245–251.
- [12] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [13] T. Wang and Y. Yamakawa, “Real-time occlusion-robust deformable linear object tracking with model-based gaussian mixture model,” *Frontiers in Neurobotics*, vol. 16, p. 886068, 2022.
- [14] S. Bhattacharya, M. Likhachev, and V. Kumar, “Identification and representation of homotopy classes of trajectories for search-based path planning in 3d,” in *Robotics: Science and Systems*, 2011, pp. 27–30.
- [15] E. Hernandez, M. Carreras, and P. Ridao, “A comparison of homotopic path planning algorithms for robotic applications,” *Robotics and Autonomous Systems*, vol. 64, pp. 44–58, 2015.
- [16] M. Cao, K. Cao, S. Yuan, T.-M. Nguyen, and L. Xie, “Neptune: nonentangling trajectory planning for multiple tethered unmanned vehicles,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2786–2804, 2023.
- [17] T. Yang, J. Liu, Y. Wang, and R. Xiong, “Self-entanglement-free tethered path planning for non-particle differential-driven robot,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7816–7822.
- [18] E. Hernández, M. Carreras, J. Antich, P. Ridao, and A. Ortiz, “A topologically guided path planner for an auv using homotopy classes,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2337–2343.
- [19] P. Mitrano and D. Berenson, “The grasp loop signature: A topological representation for manipulation planning with ropes and cables,” *arXiv preprint arXiv:2403.01611*, 2024.
- [20] W. Wang, D. Berenson, and D. Balkcom, “An online method for tight-tolerance insertion tasks for string and rope,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2488–2495.
- [21] G. Fan and M. Ding, “Non-rigid point set registration with global-local topology preservation.”
- [22] H. Luo and Y. Demiris, “Benchmarking and simulating bimanual robot shoe lacing,” *IEEE Robotics and Automation Letters*, 2024. [Online]. Available: <https://doi.org/10.1109/LRA.2024.3440101>
- [23] S. Redon, A. Kheddar, and S. Coquillart, “Fast continuous collision detection between rigid bodies,” in *Computer graphics forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 279–287.
- [24] K. Lv, M. Yu, Y. Pu, X. Jiang, G. Huang, and X. Li, “Learning to estimate 3-d states of deformable linear objects from single-frame occluded point clouds,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7119–7125.
- [25] A. Keipour, M. Bandari, and S. Schaal, “Deformable one-dimensional object detection for routing and manipulation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4329–4336, 2022.
- [26] F. Zhang and Y. Demiris, “Learning garment manipulation policies toward robot-assisted dressing,” *Science robotics*, vol. 7, no. 65, p. eabm6010, 2022.
- [27] Y. Yang, J. A. Stork, and T. Stoyanov, “Learning differentiable dynamics models for shape control of deformable linear objects,” *Robotics and Autonomous Systems*, vol. 158, p. 104258, 2022.
- [28] A. Caporali, P. Kicki, K. Galassi, R. Zanella, K. Walas, and G. Palli, “Deformable linear objects manipulation with online model parameters estimation,” *IEEE Robotics and Automation Letters*, 2024.
- [29] Y. Yang, J. A. Stork, and T. Stoyanov, “Particle filters in latent space for robust deformable linear object tracking,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 577–12 584, 2022.