SOFT ROBOTICS
Volume 9, Number 2, 2022
Mary Ann Liebert, Inc.
DOI: 10.1089/soro.2020.0110

ORIGINAL ARTICLE

Skeletonizing the Dynamics of Soft Continuum Body from Video

Katsuma Inoue, Yasuo Kuniyoshi, Katsushi Kagaya, and Kohei Nakajima

Abstract

Soft continuum bodies have demonstrated their effectiveness in generating flexible and adaptive functionalities by capitalizing on the rich deformability of soft material. Compared with a rigid-body robot, it is in general difficult to model and emulate the morphology dynamics of a soft continuum body. In addition, a soft continuum body potentially has an infinite degree of freedom, requiring considerable labor to manually annotate its dynamics from external sensory data such as video. In this study, we propose a novel noninvasive framework for automatically extracting the skeletal dynamics from video of a soft continuum body and show the applications and effectiveness of our framework. First, we demonstrate that our framework can extract skeletal dynamics from animal videos, which can be effectively utilized for the analysis of soft continuum body including animal motion. Next, we focus on a soft continuum arm, a commonly used platform in soft robotics, and evaluate the potential information-processing capability. Normally, to control such a high-dimensional system, it is necessary to introduce many sensors to completely capture the motion dynamics, causing the deterioration of the material's softness. We illustrate that the evaluation of the memory capacity and sensory reconstruction error enables us to verify the minimum number of sensors sufficient for fully grasping the state dynamics, which is highly useful in designing a sensor arrangement for a soft robot. Also, we release the software developed in this study as open source for biology and soft robotics communities, which contributes to automating the annotation process required for the motion analysis of soft continuum bodies.

Keywords: continuum robot, morphological computation, computer vision

Introduction

Living organisms incorporate elastic body tissues to realize smooth and adaptive behavior in uncertain environments. Motivated by the ubiquity of soft structures in creatures, soft robots have been developed that incorporate the deformability of soft material.[1,2] In addition, the diverse spatiotemporal pattern of soft continuum bodies has recently been highlighted as a novel tool for implementing adaptive behavioral controllers,[3–6] sensors,[7–12] and information-processing devices.[13–15] To sum, the dynamic property of soft material will be exploited to realize the versatile functionality in developing next-generation robots.

It is, however, challenging to quantitatively capture skeletal dynamics of a soft continuum body in biology and soft robotics. Unlike a conventional rigid-body robot, soft continuum bodies are often continuous, and modeling their dynamics potentially requires an infinite state space. Owing to the intrinsic nonlinearity and hysteresis of soft materials, soft continuum bodies generate a rich variety of dynamic deformation patterns when actuated, making it difficult to construct a precise model describing the deformation dynamics.[16,17] Moreover, the morphology displacement may be able to be measured by embedded sensors. However, the implanted sensors often impair a material's softness, limiting the number to be used. Therefore, to completely grasp the deformation dynamics of a soft continuum body, it is desirable to extract the skeletal dynamics by noninvasive external sensors such as video cameras or laser rangefinders.

In the field of computer vision and imaging science, skeletonization has been an important topic for finding compact representations of objects from the image for many years.[18] Blum's pioneering work[19] first formulated the concept of object skeletons and established the foundation of

Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan.

skeletonization. Blum's skeleton is obtained by the *grassfire transform process* and analytically defined as a set of collision points of two independent curves propagating from the object boundary at a constant velocity.[20] Based on the grassfire transform process, many approaches have been developed, including geometric approaches approximating the skeleton using the Voronoi diagram[21–23] and continuous curve propagation approaches emulating grassfire propagation with partial differential equations.[24–27] The skeletonization technique has been widely employed in various image processing and computer vision applications. In particular, medical imaging widely uses skeletonization to extract the centerline of blood vessels and arteries from computed tomography imaging.[28,29]

Many frameworks have also been proposed to extract skeletal dynamics from the video recording of the motion of a soft continuum body. To analyze the complicated behavior of an octopus, a framework for extracting a three-dimensional (3D) arm trajectory was developed using multiple video cameras.[30,31] The skeletonization algorithm is easily accomplished by simulating Blum's grassfire process on the digital grids. By parameterizing the contour with elliptic Fourier descriptors, it is possible to describe the morphology dynamics of soft continuum bodies.[37] Also, deep neural network (DNN) models that track characteristic points on video have recently been proposed, which would be powerful options for skeletonizing soft continuum bodies.[33,38] Although these approaches based on computer vision are useful in skeletonizing soft continuum body dynamics, they have several drawbacks. For example, the endpoint coordinates of the skeleton should be manually specified for all video frames in the octopus arm tracking system used in Refs.[30,31] The model-free method based on the elliptic Fourier descriptors[37] is not suitable for extracting skeleton dynamics because it does not provide direct information of the skeletal coordinates. It is necessary to prepare annotation data and fine-tune the model in the methods based on DNN (Table 1).[33] Also, the markers were directly attached or written on the soft continuum body as a reference point,[8,14,39,40] which involves an invasive process and cannot be used, especially with animals.

In this study, we propose a novel framework called SSS (Soft Skeleton Solver) for skeletonizing soft continuum body dynamics based on a background subtraction algorithm and a skeletonization algorithm[36] using a *fast marching method* (FMM).[35] By employing the minimum distance field and the traveling time field calculated during the skeletonization algorithm, our framework can effectively and automatically extract the endpoint coordinates and skeleton curve of the soft continuum body on all frames except the first one. Furthermore, by specifying the resolution and tracking parameters, it is possible to extract the skeleton curve with arbitrary accuracy. Below, we list the contributions of this article:

- Our proposed method automates the annotation process of specifying the skeleton's tip points, which significantly enhance the extraction efficiency and reduce the manual operation costs.
- Unlike skeletonization methods based on DNN, our proposed method does not require pretraining, which alleviates the annotation and training costs.
- We demonstrate that our methods can fully capture the deformation dynamics of soft bodies in a noninvasive manner, which could be effectively employed for designing the optimal sensor placement.

In this article, we first demonstrate that our framework can extract skeletal dynamics from dead fish ''swimming'' and brittle star movement videos. We also show that both the microscopic and macroscopic features of the animal motion are effectively reflected in the analysis. In addition, we verify the minimum number of sensors sufficient for fully grasping the state dynamics of a soft silicone rubber arm, a typical platform in soft robotics, from the video. Normally, to completely capture the deformation dynamics, a sufficient number of sensors should be embedded in the body. However, implanting sensors in the soft components often reduces its deformability and motion variety. We exhibit that our framework effectively offers a noninvasive indicator to design the sensor arrangement on a soft robot through the two demonstrations measuring the information-processing capacity and the reconstruction error of the actual sensor dynamics. Finally, we discuss the usefulness and future extension of our framework. Our software used in this study is open source and released on a website, which should be especially helpful for biologists and soft roboticists who wish to analyze the dynamic movement of soft continuum bodies.

Proposed Method

In this study, we propose an iterative skeletonizing framework for soft continuum bodies composed of the following three steps (Fig. 1A–C). The skeletonization process is automatically completed for all frames except the first one by extending the centerline estimation algorithm[36] based on the FMM algorithm. We explain the detailed algorithm of each step through a demonstration with a five-armed brittle star video (Fig. 1). Five skeletal curves should be extracted in this demonstration.

Basal point estimation

First, one of the two endpoints of the skeletal curve is estimated (referred to as the *basal point*). Initially, the

Table 1. Comparison with Other Markerless Methods That Can Skeletonize Soft Continuum Bodies ($N =$ the Number of Frames)

Method	*Algorithm*	*Pretraining*	*Resolution*	*Manual specification of tip points*
Octopus arm tracking system[30,31]	Thinning algorithm[32]	Not required	Adjustable	$O(N)$ (required for every frames)
DeepLabCut[33]	ResNet[34]	Required	Fixed	not required after the pretraining
Soft Skeleton Solver (ours)	Fast marching method[27,35,36]	Not required	Adjustable	$O(1)$ (only first frame)

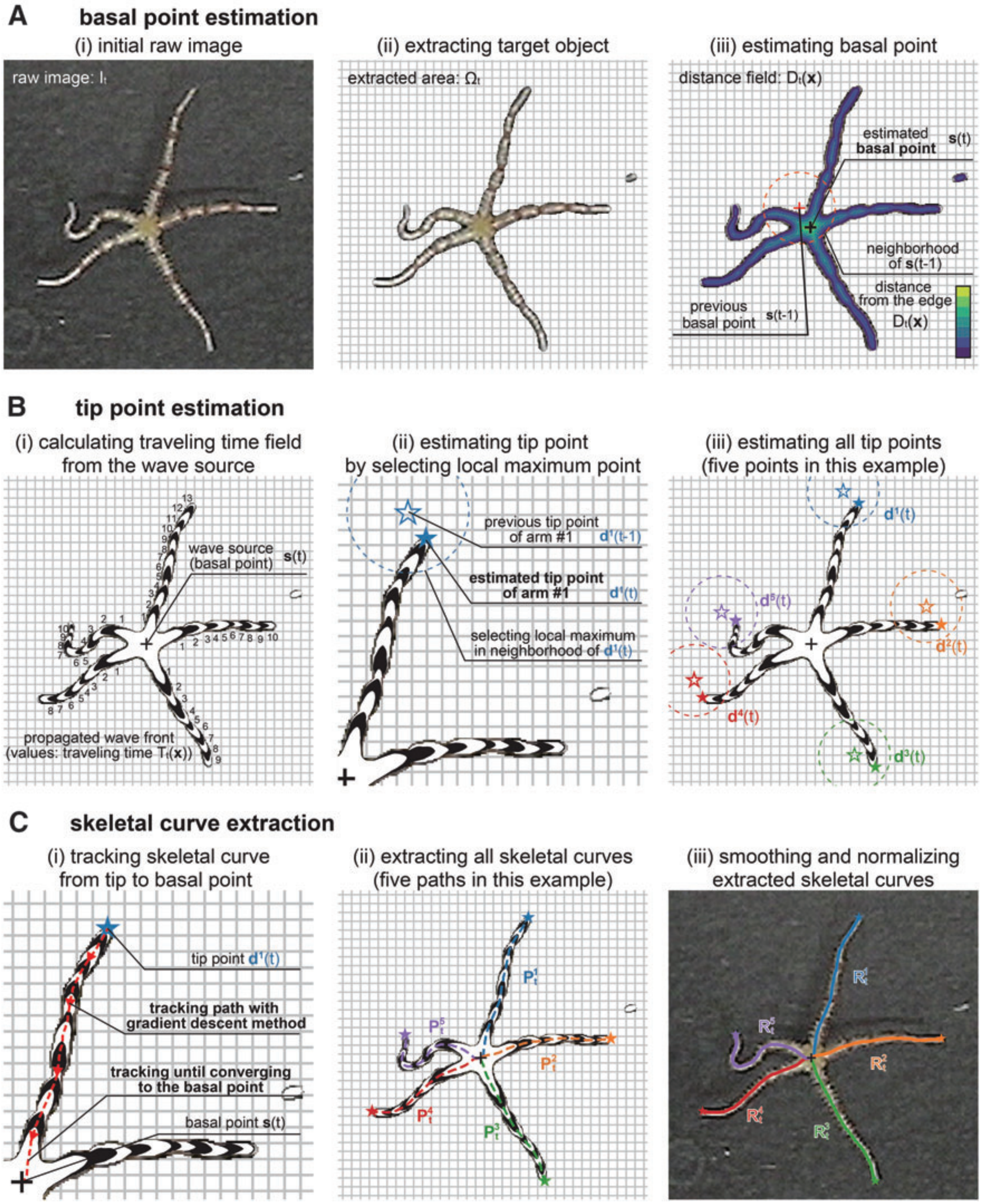

A basal point estimation
(i) initial raw image
raw image: I_t
(ii) extracting target object
extracted area: Ω_t
(iii) estimating basal point
distance field: $D_t(\mathbf{x})$
estimated basal point $\mathbf{s}(t)$
neighborhood of $\mathbf{s}(t-1)$
distance from the edge $D_t(\mathbf{x})$
previous basal point $\mathbf{s}(t-1)$
B tip point estimation
(i) calculating traveling time field from the wave source
wave source (basal point) $\mathbf{s}(t)$
propagated wave front (values: traveling time $T_t(\mathbf{x})$)
1 2 3 4 5 6 7 8 9 10 11 12 13
(ii) estimating tip point by selecting local maximum point
previous tip point of arm #1 $\mathbf{d}^1(t-1)$
estimated tip point of arm #1 $\mathbf{d}^1(t)$
selecting local maximum in neighborhood of $\mathbf{d}^1(t)$
(iii) estimating all tip points (five points in this example)
$\mathbf{d}^1(t)$
$\mathbf{d}^5(t)$
$\mathbf{d}^2(t)$
$\mathbf{d}^4(t)$
$\mathbf{d}^3(t)$
C skeletal curve extraction
(i) tracking skeletal curve from tip to basal point
tip point $\mathbf{d}^1(t)$
tracking path with gradient descent method
tracking until converging to the basal point
basal point $\mathbf{s}(t)$
(ii) extracting all skeletal curves (five paths in this example)
P_t^1
P_t^5
P_t^2
P_t^4
P_t^3
(iii) smoothing and normalizing extracted skeletal curves
R_t^1
R_t^5
R_t^2
R_t^4
R_t^3

FIG. 1. Detailed description of the algorithm. The algorithm has three steps. Each step processes data from left to right figure (i–iii). **(A)** Basal point estimation. In this demonstration using the brittle star video, the farthest point from the edge was selected as the basal point. **(B)** Tip point estimation. Five tip points were estimated corresponding to the five arms in this demonstration. **(C)** Skeletal curve estimation. The skeletal curve was estimated to connect the basal point and the tip points. The solution was obtained by backtracking along the gradient of the traveling time field. Color images are available online.

region of target object Ω_t is extracted from the raw image $I_t (t = 1, 2, 3, \cdots)$ (Fig. 1A-i, ii). Here, we used a simple background subtraction algorithm to binarize the image I_t based on the pixel values with an appropriate threshold. Note that the extracted region Ω_t is assumed to be simply connected, that is, enclosed by a single closed curve and having no holes in the region. Next, the minimum Euclid distance field $D_t(\boldsymbol{x})$ is calculated from the contour of Ω_t [$\boldsymbol{x} \in \Omega_t$ is a grid coordinate; $D_t(\boldsymbol{x})$ represents the distance between $\boldsymbol{x}$ and the nearest boundary of Ω_t]. By applying the FMM algorithm, the computational complexity for the $D_t(\boldsymbol{x})$ calculation can be suppressed to $O(HW \log(HW))$ for grid number HW (H and W are the height and width of the video, respectively). Based on the distance field $D_t(\boldsymbol{x})$, the coordinate of basal point $\boldsymbol{s}(t)$ is estimated. Since the points on the skeletal curve are distributed on the *ridge line* of the distance field, the basal point $\boldsymbol{s}(t)$

can be estimated from the local maximum point of $D_t(\boldsymbol{x})$. Especially in this brittle star demonstration, five ridge lines corresponding to the five arms intersect at the maximum point of distance field $D_t(\boldsymbol{x})$. Therefore, we selected the local maximum point of $D_t(\boldsymbol{x})$ on the ε-neighborhood of the previous basal point $\boldsymbol{s}(t-1)$ as the next basal point $\boldsymbol{s}(t)$ (ε is set to an appropriate value according to the video size). Note that this estimation algorithm can be flexibly modified or replaced with another one depending on the target object morphology. For example, a similar algorithm was used by fixing the Y-coordinate of the basal point for the soft octopus arm video presented in the Designing Sensory Configuration for Soft Robotic Arm section (see the Appendix for detail). Also, the previous basal point $\boldsymbol{s}(0)$ is manually set for the initial basal point $\boldsymbol{s}(1)$ on the first frame I_1. We developed a user interface to set the endpoint coordinates for the first frame (Supplementary Videos S1 and S2).

Tip point estimation

Next, another endpoint of the skeletal curve is estimated (referred to as the *tip point*). First, the following speed vector field $F_t(\boldsymbol{x})$ is calculated based on the distance field $D_t(\boldsymbol{x})$ with the following equation (Fig. 1B-i):

$$F_t(\boldsymbol{x}) := \exp(\alpha D_t(\boldsymbol{x})), \tag{1}$$

where α is a constant value adjusting the convexity of $F_t(\boldsymbol{x})$ (we used $\alpha = 0.5$ for calculation stability). Next, consider a closed curve Γ_t propagating normal to itself with the speed $F_t(\boldsymbol{x})$ from the *wave source* $\boldsymbol{s}(t)$. Then, the traveling time field $T_t(\boldsymbol{x})$ is calculated. $T_t(\boldsymbol{x})$ denotes a time when Γ_t passes over $\boldsymbol{x}$. Especially in a special case where the wave front moves in one direction with the velocity F_t, the relationship between $T_t(\boldsymbol{x})$ and $F_t(\boldsymbol{x})$ can be formulated with the following equation:

$$|\nabla T_t(\boldsymbol{x})| F_t(\boldsymbol{x}) = 1. \tag{2}$$

This is called the eikonal equation, whose solution can be efficiently acquired by the FMM algorithm as with the calculation of the distance field.[36]

After the calculation of $T_t(\boldsymbol{x})$, the tip point $\boldsymbol{d}^n(t)$ is estimated (Fig. 1B-ii). Here, n is an arm index ($n = 1 \cdots 5$ in this demonstration). The travel time field $T_t(\boldsymbol{x})$ takes a local maximum value at the farthest points from the wave source $\boldsymbol{s}(t)$. Therefore, we selected the local maximum point of $T_t(\boldsymbol{x})$ as the new tip point $\boldsymbol{d}^n(t)$ on the ε-neighborhood of the previous tip point $\boldsymbol{d}^n(t-1)$. Also, the previous tip points $\boldsymbol{d}^n(0)$ of $\boldsymbol{d}^n(1)$ on the first frame I_1 are manually set, and those on the rest of the frames are automatically gained.

Skeletal curve extraction

Finally, the N_{dim} point sequences $R_t^n = \{\boldsymbol{r}_i(t)\}_i (i = 1 \cdots N_{\mathrm{dim}})$ distributed at regular intervals on the skeletal curve is extracted by connecting the basal point $\boldsymbol{s}(t)$ and each tip point $\boldsymbol{d}^n(t)$ (below, the arm index n is omitted for simplicity). We consider the following skeletal curve C_t minimizing the accumulate value of the cost function among the curves connecting $\boldsymbol{s}(t)$ and $\boldsymbol{d}(t)$:

$$C_t := \underset{C}{\operatorname{argmin}} \int_C U_t(C(s)) ds, \tag{3}$$

$$U_t(\boldsymbol{x}) := \exp(-\alpha D_t(\boldsymbol{x})) \left(= (F_t(\boldsymbol{x}))^{-1} \right). \tag{4}$$

The minimum cost path between $\boldsymbol{s}(t)$ and $\boldsymbol{d}(t)$ is found by backtracking along a gradient ∇T_t from $\boldsymbol{d}(t)$ until reaching $\boldsymbol{s}(t)$ (Fig. 1C-i, ii).[36] The second-order Runge–Kutta method (RK2) was used for approximating the gradient with a constant width δ in Algorithm 1. This algorithm yields a point sequence $P_t = \{\boldsymbol{p}_i(t)\}_i (i = 1 \cdots N_{\mathrm{raw}})$ on the skeletal curve C_t. Note that N_{raw} does not necessarily match N_{dim}.

Algorithm 1 Backtracking in Skeletal Curve Extraction

```
$P_t = \{\phi\}$          ▷ point sequence on the skeletal curve
$\boldsymbol{p} = \boldsymbol{d}(t)$
while $\|\boldsymbol{p} - \boldsymbol{s}(t)\| > \varepsilon$ do
    $P_t \leftarrow P_t \cup \{\boldsymbol{p}\}$
    $\boldsymbol{p} \leftarrow \boldsymbol{p} + \delta \nabla_x T_t(\boldsymbol{x})$     ▷ approximating $\nabla_x T_t$ with RK2
$P_t \leftarrow P_t \cup \{\boldsymbol{s}(t)\}$
```

After the backtracking process, the smoothed point sequence $Q_t = \{\boldsymbol{q}_i(t)\}_i (i = 1 \cdots N_{\mathrm{raw}})$ is obtained from point sequence P_t with the following smoothing algorithm:

$$\boldsymbol{q}_i(t) = \sum_{j=-K}^{K} \boldsymbol{p}_{\max(\min(i+j, N_{\mathrm{raw}}), 1)}(t), \tag{5}$$

where K is a constant value adjusting the smoothing strength. Then, Q_t is interpolated to construct a smoothed curve $C_t^Q : [0, \infty) \to \Omega_t$ (a cubic interpolation was used). Finally, the point sequence R_i on C_t^Q is reconstructed to satisfy $|R_t| = N_{\mathrm{dim}}$ and $\|\boldsymbol{r}_{i-1}(t) - \boldsymbol{r}_i(t)\| = \mathrm{const.}$ for all $i = 2 \cdots N_{\mathrm{dim}}$. The skeletal resolution can be arbitrarily set by adjusting the width δ in the gradient-descent process and the number of points N_{dim}. Through the iterative algorithm repeating the above three steps, our framework can automatically extract the skeletal dynamics of soft continuum bodies.

Case Studies

In this section, we demonstrate the applications of our framework to biological and soft robotic data.

Analysis of biological data

First, we demonstrate the effectiveness of our framework by skeletonizing animal movement. We prepared a *dead trout swimming* video (published in Refs.,[41,43] 178 frames) as a simple task and extracted the spine dynamics. This video displays the ability of a dead fish body to swim upstream by employing the Karman vortices generated by a D-shaped obstacle. In this demonstration, we used the manually annotated time series data of the head position as the basal point dynamics $\boldsymbol{s}(t)$, and the skeletal dynamics and tail position $\boldsymbol{d}(t)$ were automatically extracted with our framework. Figure 2A shows the extracted spine dynamics ($N_{\mathrm{dim}} = 1000$). As can be seen from the figure, our framework effectively extracted the continuum spine dynamics (Supplementary Video S3).

Next, we exhibit that our framework can be applied to skeletonizing a multi-armed object. We prepared a five-armed brittle star (*Ophiactis brachyaspis*) video (published in Ref.,[42] 843 frames). It was reported that brittle star randomly selects the *leading arm* opposite to one that is stimulated and has a tendency to move forward in the direction of the leading arm while synchronizing the bilateral arms adjacent to the leading arm.[42] Especially, there exist two candidate arms to be selected in response to the external stimulus in a five-armed brittle star. In the video, the tester provided a stimulus to the tip of arm #5 (purple). Then, the brittle star selected arm #2 (orange) as the leading arm until around the 500th frame, and arm #3 (green) after that. Figure 2B shows the relative coordinates dynamics ($N_{\mathrm{dim}} = 1000$) of each arm and its movement analysis. Note that the skeletal dynamics for all the frames were automatically extracted except for the first one. Positive correlation values were globally obtained on the two correlation matrices; one between arm #1 and arm #3 and another between arm #2 and arm #4 (surrounded by a dotted line in Fig. 2B), which is consistent with the observed leading arm selection and synchronized arm movement (Supplementary Video S4). In this way, our framework efficiently extracts the skeletal dynamics of the soft continuum bodies and provides useful information for understanding the comprehensive movement of animals.

Also, our proposed method can be used to skeletonize fuzzy soft body objects. We prepared a video recording the behavior of *Hydra vulgaris*,[44] whose body is semitransparent and thus generally hard to skeletonize. Our proposed method is, however, applicable to skeletonizing such a blurry body when the background is homogeneous since a simple binarization process can extract the semitransparent object (Supplementary Video S5). In this way, our proposed method can extract the object structure more robustly in a case where the background environment can be easily controlled (e.g., in a laboratory environment).

Designing sensory configuration for soft robotic arm

Next, we demonstrate that our framework offers a noninvasive indicator for designing the sensory configuration of a soft robot. Here, we prepared a movie recording of *soft octopus arm* movement (published in Refs.,[15,46] 36,321 frames). The soft octopus arm is a typical soft continuum robotic body in which a servomotor and 10 bend sensors are attached to a silicone rubber arm (Fig. 3A). A soft continuum arm is a commonly used platform in the field of soft robotics.[13,14,47] Also, the soft octopus arm is a primary mechanical device for *physical reservoir computing*,[5,6,48–53] where the complicated time series responses generated on the soft material are exploited for machine-learning tasks. In particular, the soft octopus arm converts binary motor commands $u(t)$ with the switching time interval τ_{state} into continuous sensory dynamics $\boldsymbol{x}_{\mathrm{sensor}}(t)$ (Fig. 3B, we fixed the interval to $\tau_{\mathrm{state}} = 11$). Originally, 10 bend sensors were embedded to extract complex spatiotemporal deformation patterns [i.e., $\boldsymbol{x}_{\mathrm{sensor}}(t) \in \mathbb{R}^{10}$], which was insufficient for fully grasping the deformation dynamics. However, owing to reduced flexibility, the number of attachable sensors was limited. We estimated the sufficient number of sensors to capture the deformation dynamics by evaluating the potential information-processing capacity of the soft octopus arm.

We prepared for 10,010 points of extracted skeletal dynamics R_t ($N_{\mathrm{dim}} = 10,010$). Then, to correspond to the actual sensor, the tangent vectors $\boldsymbol{x}(t) \in \mathbb{R}^{10,000\times 2}$ were calculated from R_t by the following formula:

$$\boldsymbol{x}_i(t) = (\boldsymbol{r}_{i+10}(t) - \boldsymbol{r}_i(t))^T.$$

Figure 3C and D displays the skeletal dynamics $\boldsymbol{x}(t)$ and sensory dynamics $\boldsymbol{x}_{\mathrm{sensor}}(t)$ in response to the binary sequence $\boldsymbol{u}(t)$ (Supplementary Video S6). Here, we assumed that these tangent vectors corresponded to three-axis accelerometers measuring the direction of gravity. A three-axis accelerometer is often embedded into soft robotic components to measure the displacement of the material deformation.[54–56] In other words, we estimated the number of required three-axis accelerometers to fully exploit the potential computational resource.

To evaluate the information-processing capability, we prepared a *short-term memory task* that measured the memory property for a random input signal. The short-term memory task requires system to reconstruct past input before n segments, $u(t' - n)$, from current state $\boldsymbol{x}(t')$ with a linear logistic regression model.[57] Below, we introduced $t := \tau_{\mathrm{state}} t'$; that is, only the dynamics on every $\tau_{\mathrm{state}} = 11$ steps were considered. The linear weight $\boldsymbol{w}$ on the model was trained to approximate $u(t' - n)$ as follows:

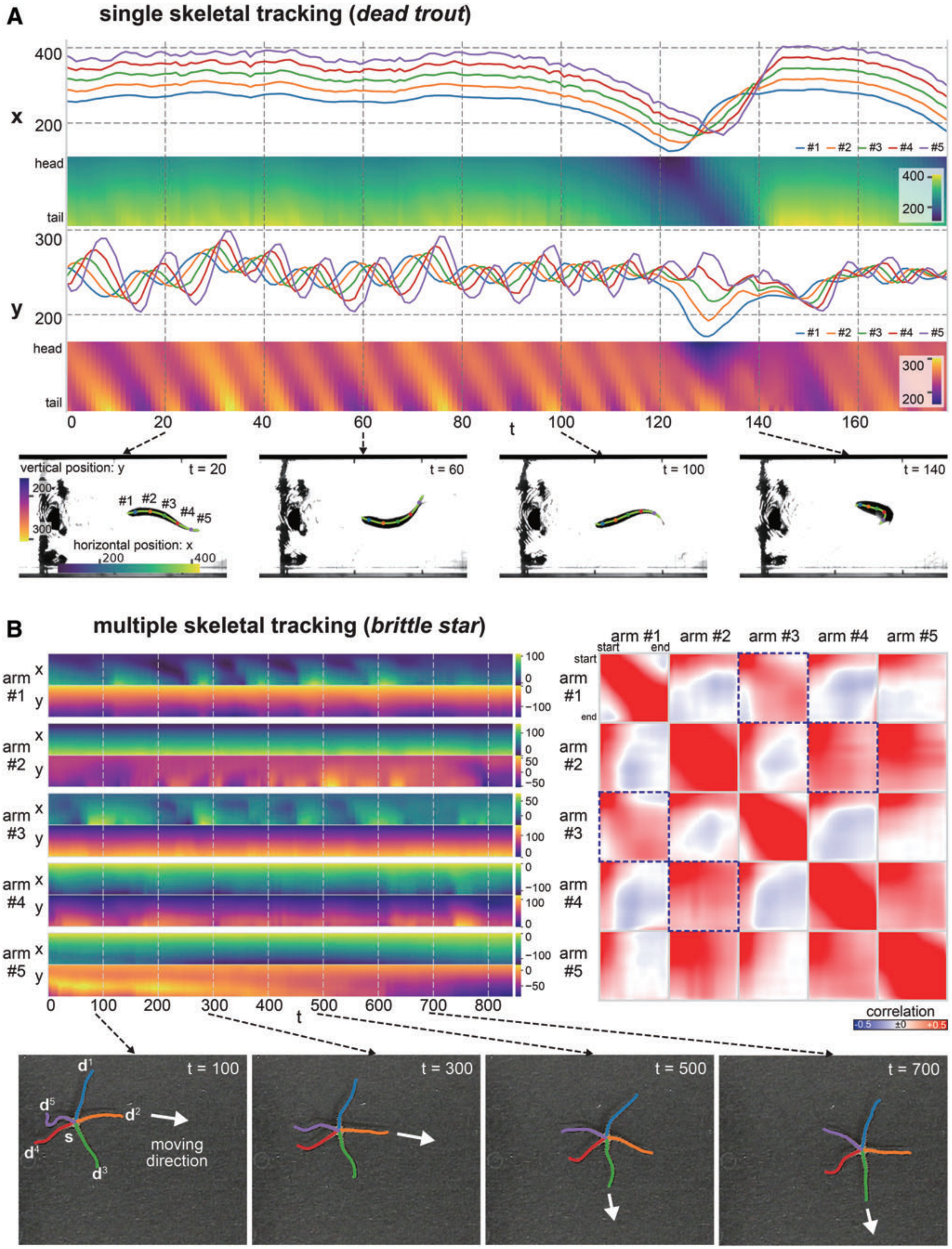
$$u(t' - n) \approx y(t'),$$

$$y(t') := \begin{cases} 1 & (p(t') > 1/2) \\ 0 & (\text{otherwise}) \end{cases},$$

$$p(t') := 1/(1 + \exp(\boldsymbol{w}\boldsymbol{x}(t'))).$$

Since the logistic regression model has a minimal nonlinearity and no memory property, the task performance significantly reflects the degree of the computational capability of the system. Here, we introduced mutual information MI_n between the output $y(t')$ and target $u(t' - n)$ as an evaluation measure. MI_n takes a value within $[0, 1]$ and approaches one as the performance increases (see the Appendix for the MI algorithm). Also, we calculated the *performance capacity* $C_{\mathrm{memory}} := \sum_n MI_n$ to assess the overall computational capacity. We prepared 1250 time steps of training data

FIG. 2. Demonstration of skeletal dynamics extraction. **(A)** Extracted spine dynamics of the dead trout. We used a video published in the study of Beal *et al.*[41] and the manually annotated basal points. The *XY*-coordinate dynamics of each point on the spine are plotted (the colormap shows the 1000-dimensional dynamics of *XY*-coordinates from the head to the tail). The skeletal curve and tip point were automatically extracted by our framework (Supplementary Video S3). **(B)** Skeletal dynamics of five-armed brittle star and its movement analysis. We used a brittle star video published in the study of Wakita *et al.*[42] The skeletal curve and endpoints were automatically extracted by our framework (Supplementary Video S4). Each arm is indexed in clockwise order from the upper one. The *left* colormap plots the dynamics of the relative coordinates $\boldsymbol{r}_{\mathrm{i}}^{\mathrm{n}}(\mathrm{t}) - \boldsymbol{s}(\mathrm{t})$. The *right* colormap shows the correlation matrix of velocity $\|\Delta\boldsymbol{r}_{\mathrm{i}}^{\mathrm{n}}(\mathrm{t})\|$. Color images are available online.


A
single skeletal tracking (dead trout)
x
400
200
—#1 —#2 —#3 —#4 —#5
head
tail
400
200
y
300
200
—#1 —#2 —#3 —#4 —#5
head
tail
300
200
0
20
40
60
80
t
100
120
140
160
vertical position: y
200
300
t = 20
#1 #2 #3 #4 #5
horizontal position: x
200
400
t = 60
t = 100
t = 140
B
multiple skeletal tracking (brittle star)
arm #1
x
y
100
0
0
−100
arm #2
x
y
100
0
50
0
−50
arm #3
x
y
50
0
100
0
arm #4
x
y
0
−100
100
0
arm #5
x
y
0
−100
0
−50
0
100
200
300
400
t
500
600
700
800
arm #1
start
end
arm #2
arm #3
arm #4
arm #5
start
arm #1
end
arm #2
arm #3
arm #4
arm #5
correlation
-0.5
±0
+0.5
t = 100
d1
d5
d2
d4
s
d3
moving direction
t = 300
t = 500
t = 700

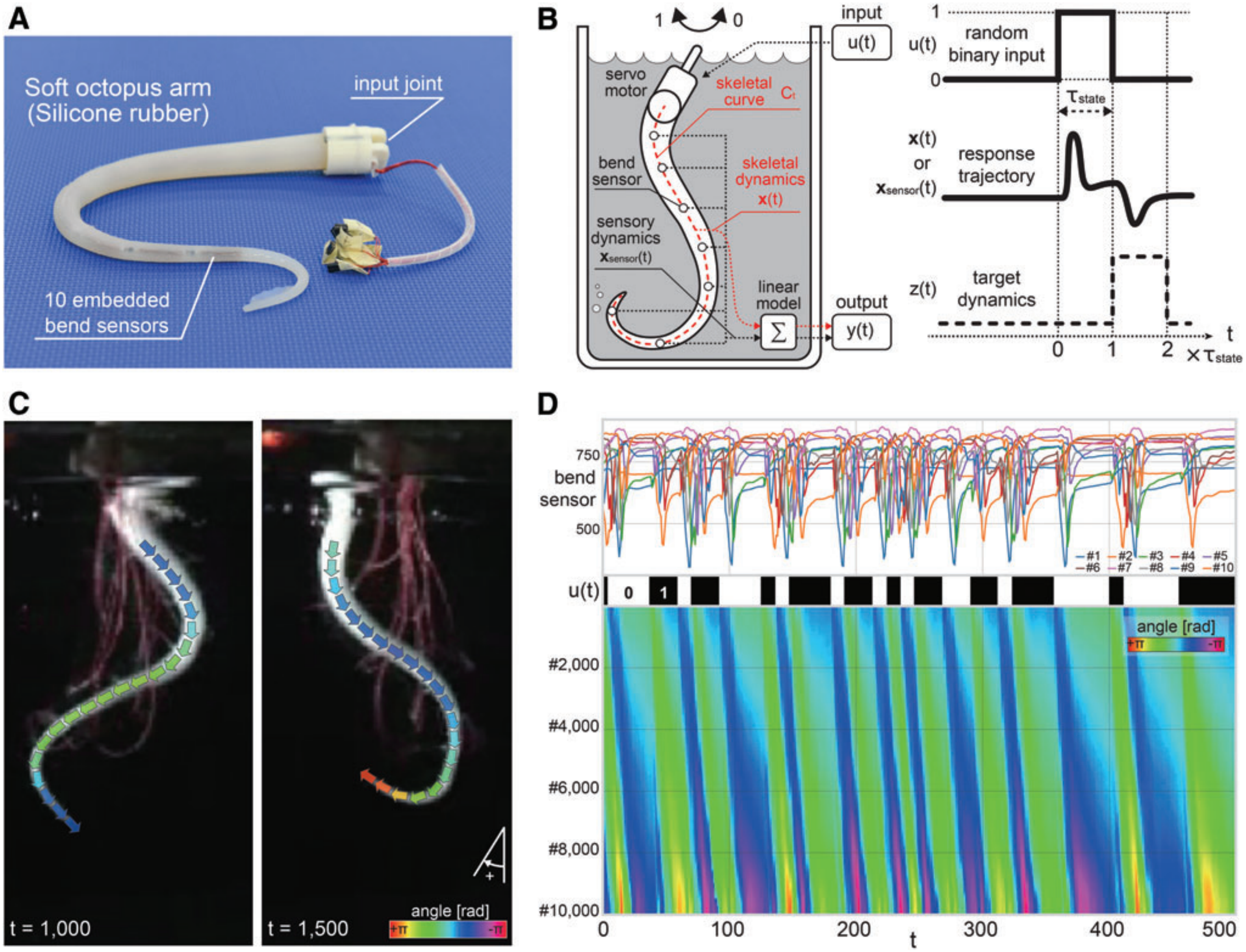


FIG. 3. Soft octopus arm setup. **(A)** Overview of soft octopus arm.[45] **(B)** Experimental setup for evaluation of the information-processing capability. **(C)** Dynamics of the extracted skeletal dynamics and corresponding contact vector. The *color* represents the angle of the tangent vector. **(D)** Response curve comparison. *Top*: 10 bend sensor dynamics $\boldsymbol{x}_{\text{sensor}}$. Each *label* represents the index of the sensor. *Middle*: Input time series $u(t)$. *Bottom*: Dynamics of the angle of the tangent vector $\boldsymbol{x}(t)$. The y-axis in the colormap corresponds to the position on the soft octopus arm (i.e., #1: the *top* basal point, #10,000: the *bottom* tip point). Refer also the Supplementary Video S6. Color images are available online.

and 1250 time steps of evaluation data. In addition, to investigate the dependence of the number of sensors on the information-processing capacity, N_{select} elements among 20,000 nodes in $\boldsymbol{x}(t)$ were randomly chosen (we tested $N_{\text{select}} = 1, 2, 5, 10, 20, 50, 100, 200, 500,$ and 1000). We also measured the performance with the bend-sensor dynamics $\boldsymbol{x}_{\text{sensor}}(t')$ as the baseline.

Figure 4A depicts the performance function MI_n and memory capacity C_{memory}, suggesting that the performance monotonically improved as N_{select} increased and was comparable with sensory dynamics at $N_{\text{select}} = 10$. Moreover, the memory capacity saturated at around $N_{\text{select}} = 10^2$. These results revealed the following two points: (i) our system brought out the information-processing capability of the soft octopus arm more than the actual 10 bend sensors, and (ii) the computational capacity of the soft octopus arm can be sufficiently extracted by embedding 10^2 accelerometers.

Furthermore, we demonstrate that our framework can estimate the number of required sensors to extract deformation dynamics even without input information $u(t)$. We evaluated the reconstruction error of the bend-sensor dynamics $\boldsymbol{x}_{\text{sensor}}(t)$ using the extracted skeletal dynamics $\boldsymbol{x}(t)$. A linear ridge regression model was used, and the normalized mean square error (NMSE) between the output $y(t)$ and target dynamics $z(t)$ was measured with the following equation:

$$\mathrm{NMSE}(y, z) := \frac{\mathbb{E}\left[(y-z)^2\right]}{\mathrm{Var}(z)}$$

$$= \frac{\sum_t (y(t) - z(t))^2}{\sum_t (z(t) - \bar{z}(t))^2}, \qquad (7)$$

where $\bar{z}(t)$ is the average of $z(t)$. Also, the sum of NMSEs over 10 bend-sensor dynamics was calculated to measure the overall reconstruction accuracy.

Figure 4B displays the reconstruction error over N_{select}. The bend-sensor dynamics were effectively reconstructed

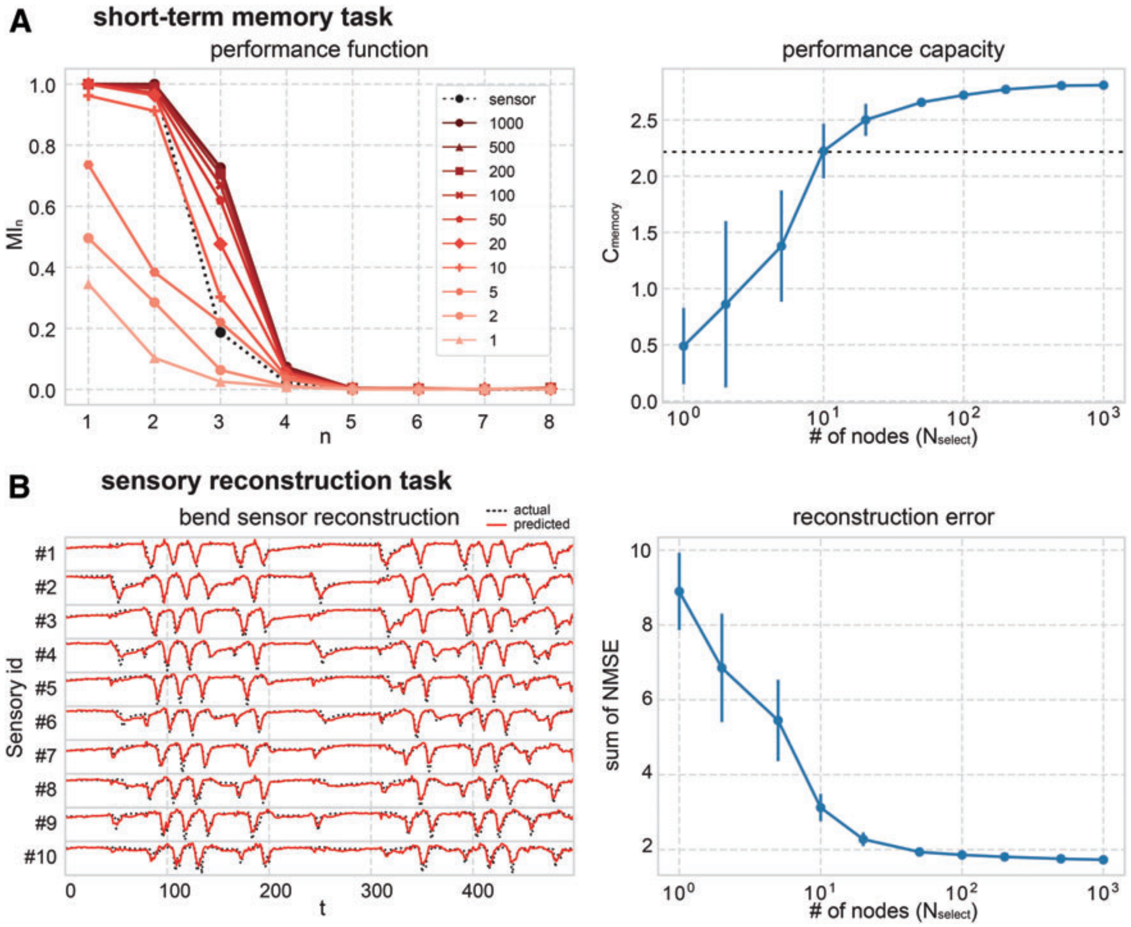


FIG. 4. **(A)** Results of the short-term memory task. *Left*: Performance function. The averaged mutual information values MI_n over 10 trials are plotted. MI_n with 10 bend sensors are plotted (*black dotted line*). The *number in the labels* denotes the dimension of selected dynamics. *Right*: Performance capacity ($\sum_n MI_n$). The *dotted line* shows the capacity with the actual sensory data. The *error bar* shows the standard deviation over 10 trials. **(B)** Reconstruction of sensor time series. *Left*: Reconstruction of the 10 bend sensor dynamics. The figure displays both the actual sensor values (*dotted*) and predicted values by the linear model (*red*). *Right*: Reconstruction error. The sum of NMSEs for the 10 sensor dynamics $\sum_{i=1}^{10} NMSE(y_i, z_i)$ is plotted. The *error bar* represents the standard deviation over 10 trials. NMSE, normalized mean square error. Color images are available online.

using the extracted skeletal dynamics. Also, the NMSE evaluation showed that the reconstruction accuracy was monotonically improved and saturates at around $N_{\text{select}} = 10^2$, which was consistent with the results of the short-term memory task. In this way, the number of required accelerometers to extract deformation dynamics can be estimated even with sensory information of different modalities.

As with the case studies in the Analysis of Biological Data section skeletonizing biological data, our framework, of course, can be used to extract the skeletal dynamics of the other soft robotic systems, which offer useful information for control (see the Supplementary Video S7 skeletonizing a soft manipulator from a video published by Truby *et al.*[58]). Also, 3D skeletal coordinates can be easily implemented with multiple videos from different viewpoints. We prepared a simple setup recording behavior of the soft octopus arm from two cameras arranged in the vertical direction and reconstructed the 3D arm dynamics (Supplementary Video S8 and Appendix). This 3D skeletonization would help estimate the posture of soft robotic systems.

Discussion

In this article, we proposed a framework for automatically extracting skeletal dynamics from video information of a soft continuum body. Since most of the annotation process are automated compared with the conventional methods, our framework can efficiently skeletonize soft continuum bodies. Also, the skeletal curve can be extracted with arbitrary accuracy by adjusting the tracking width and normalization parameter.

In the Analysis of Biological Data section, we exhibited that our framework efficiently extracted the skeletal dynamics from the video recording of animal movement. We showed that our framework can simultaneously extract multiple skeletal dynamics through a demonstration with a brittle star video. Also, we illustrated that our framework is applicable to analyze animal behavior. In the brittle star demonstration, for example, we demonstrated that the macroscopic arm movements of the five-armed brittle star were effectively reflected in the analysis with correlation matrices.

We show that *O. brachyaspis* continuously switches the leading arm, which was neither quantitatively evaluated nor visualized in a previous study.[42] The extracted arm dynamics of *H. vulgaris* can be used to study the mechanism of the behavioral generation when it is analyzed with the imaged cell activity data[44] In this way, our framework is promising for studying animal behaviors.

In the Designing Sensory Configuration for Soft Robotic Arm section, we verified the minimum number of sensors sufficient for fully grasping the state dynamics by evaluating the information-processing capacity through the short-term memory task. We also demonstrated that the number of required sensors can be estimated without input information through the reconstruction of actual sensory dynamics. From the viewpoint of morphological computation, it is desirable to know the potential computational capacity of a body before deciding the sensor configuration.[59–62] However, an implanted sensor often worsens a soft material's deformability, which limits the number of attachable sensors. Our framework can be employed to estimate the optimal sensor positions of soft robots in a noninvasive manner, which is helpful in the design of soft robot. For example, by optimizing the sensory placements to maximize a measurement called *effective dimension*,[63] we can reduce the redundant sensors and efficiently capture the internal state of the soft body with a limited number of sensors. Another possible scenario is that our framework can be used to estimate the minimum dimension to model the deformation dynamics of the soft body. Since the redundant sensors are wiped out by the optimization, the number of obtained optimal sensors would be related to the required dimension for modeling the soft body. To sum, our framework offers a useful indicator in designing soft robot setups.

Finally, we discuss the possible direction for extending our framework. The accuracy of our framework depends highly on the performance of the background subtraction algorithm. In this study, we prepared movies where the background and the object could be easily binarized by a single threshold value. It is, however, necessary to introduce advanced background subtraction algorithms such as Refs.[64,65] especially when the background has a complicated pattern such as in a natural environment. Moreover, our framework cannot extract the skeleton of a soft continuum body overlapping on the image, limiting the scalability of our framework in the control of soft robots. To solve the problem, a 3D volume video should be used instead of a two-dimensional video. In particular, the FMM algorithm, a core algorithm in our framework, can be easily extended to a 3D volume image, which should be developed in future work.

Acknowledgments

We thank K. Tanaka and Y. Minami for providing the mechanical setups for our demonstration.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

This work was based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). Katsuma Inoue was supported by JSPS KAKENHI grant number JP20J12815. Katsushi Kagaya was supported by JSPS KAKENHI grant numbers JP19H05330 and JP18K19336. Kohei Nakajima was supported by JSPS KAKENHI grant number JP18H05472.

Supplementary Material

Supplementary Video S1
Supplementary Video S2
Supplementary Video S3
Supplementary Video S4
Supplementary Video S5
Supplementary Video S6
Supplementary Video S7
Supplementary Video S8

References

1. Rus D, Tolley MT. Design, fabrication and control of soft robots. Nature 2015;521:467–475.
2. Laschi C, Mazzolai B, Cianchetti M. Soft robotics: technologies and systems pushing the boundaries of robot abilities. Sci Robot 2016;1:eaah3690.
3. Brown E, Rodenberg N, Amend J, *et al.* Universal robotic gripper based on the jamming of granular material. Proc Natl Acade Sci U S A 2010;107:18809–18814.
4. Shepherd RF, Ilievski F, Choi W, *et al.* Whitesides, Multigait soft robot. Proc Natl Acade Sci U S A 2011;108: 20400–20403.
5. Caluwaerts K, D'Haene M, Verstraeten D, *et al.* Locomotion without a brain: physical reservoir computing in tensegrity structures. Artif Life 2013;19:35–66.
6. Caluwaerts K, Despraz J, Işçen A, *et al.* Design and control of compliant tensegrity robots through simulation and hardware validation. J R Soc Interface 2014;11:20140520.
7. Chin K, Hellebrekers T, Majidi C. Machine learning for soft robotic sensing and control. Adv Intell Syst 2020. [Epub ahead of print]; DOI: 10.1002/aisy.201900171.
8. Judd E, Soter G, Rossiter J, *et al.* Sensing through the body-- non-contact object localisation using morphological computation (2019 2nd IEEE International Conference on Soft Robotics, RoboSoft). Seoul, Korea: IEEE, 2019, pp. 558–563.
9. Honda S, Zhu Q, Satoh S, *et al.* Textile-based flexible tactile force sensor sheet. Adv Funct Mater 2019;29:1807957.
10. Gao W, Ota H, Kiriya D, *et al.* Flexible electronics toward wearable sensing. Acc Chem Res 2019;52:523–533.
11. Wakamatsu K, Inoue K, Hagiwara D, *et al.* Mixing state estimation of peristaltic continuous mixing conveyor with distributed sensing system based on soft intestine motion (2020 3rd IEEE International Conference on Soft Robotics, RoboSoft). Seoul, Korea: IEEE, 2020, pp. 208–214.
12. Sakurai R, Nishida M, Sakurai H, *et al.* Emulating a sensor using soft material dynamics: a reservoir computing approach to pneumatic artificial muscle (2020 3rd IEEE International Conference on Soft Robotics, RoboSoft). New York, NY: IEEE, 2020, pp. 710–717.
13. Nakajima K, Hauser H, Li T, *et al.* Information processing via physical soft body. Sci Rep 2015;5:10487.
14. Nakajima K, Schmidt N, Pfeifer R. Measuring information transfer in a soft robotic arm. Bioinspir Biomim 2015;10: 035007.
15. Nakajima K, Hauser H, Li T, *et al.* Exploiting the dynamics of soft materials for machine learning. Soft Robot 2018;5: 339–347.

16. Trivedi D, Rahn CD, Kier WM, *et al.* Soft robotics: biological inspiration, state of the art, and future research. Appl Bionics Biomech 2008;5:99–117.
17. George Thuruthel T, Ansari Y, Falotico E, *et al.* Control strategies for soft robotic manipulators: a survey. Soft Robot 2018;5:149–163.
18. Saha PK, Borgefors G, di Baja GS. A survey on skeletonization algorithms and their applications. Pattern Recognit Lett 2016;76:3–12.
19. Blum H. A Transformation for Extracting New Descriptors of Shape, vol. 4. Cambridge: MIT Press, 1967.
20. Blum H, Nagel RN. Shape description using weighted symmetric axis features. Pattern Recognit 1978;10:167–180.
21. Brandt JW, Algazi VR. Continuous skeleton computation by voronoi diagram. CVGIP Image Understanding 1992; 55:329–338.
22. Ogniewicz RL, Ilg M. Voronoi skeletons: theory and applications. CVPR 1992;92:63–69.
23. Ogniewicz RL, Kübler O. Hierarchic voronoi skeletons. Pattern Recognit 1995;28:343–359.
24. Kimia BB, Tannenbaum AR, Zucker SW. Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space. Int J Comput Vis 1995;15:189–224.
25. Kimmel R, Shaked D, Kiryati N, *et al.* Skeletonization via distance maps and level sets. Comput Vis Image Underst 1995;62:382–391.
26. Leymarie F, Levine MD. Simulating the grassfire transform using an active contour model. IEEE Trans Pattern Anal Mach Intell 1992;1:56–75.
27. Siddiqi K, Bouix S, Tannenbaum A, *et al.* Hamilton-jacobi skeletons. Int J Comput Vis 2002;48:215–231.
28. Bouix S, Siddiqi K, Tannenbaum A. Flux driven automatic centerline extraction. Med Image Anal 2005;9:209–221.
29. Yang G, Kitslaar P, Frenay M, *et al.* Automatic centerline extraction of coronary arteries in coronary computed tomographic angiography. Int J Cardiovasc imaging 2012;28:921–933.
30. Yekutieli Y, Mitelman R, Hochner B, *et al.* Analyzing octopus movements using three-dimensional reconstruction. J Neurophysiol 2007;98:1775–1790.
31. Kazakidi A, Zabulis X, Tsakiris DP. Vision-based 3D motion reconstruction of octopus arm swimming and comparison with an 8-arm underwater robot (2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA). New York, NY, USA: IEEE, 2015, pp. 1178–1183.
32. Lam L, Lee S-W, Suen CY, *et al.* Thinning methodologies-a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 1992;14:869–885.
33. Mathis A, Mamidanna P, Cury KM, *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat Neurosci 2018;21:1281.
34. He K, Zhang X, Ren S, *et al.* Deep residual learning for image recognition (2016 IEEE conference on computer vision and pattern recognition (CVPR)), Las Vegas, NV, USA, 2016, pp. 770–778.
35. Sethian JA. A fast marching level set method for monotonically advancing fronts. Proc Natl Acad Sci U S A 1996; 93:1591–1595.
36. Hassouna MS, Farag AA. Robust centerline extraction framework using level sets (2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05, vol. 1), San Diego, CA, USA, USA). New York, NY, USA: IEEE, 2005, pp. 458–465.
37. Digumarti KM, Trimmer B, Conn AT, *et al.* Quantifying dynamic shapes in soft morphologies. Soft Robot 2019;6:733–744.
38. Cao Z, Hidalgo G, Simon T, *et al.* Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv Preprint 2018;arXiv:1812.08008.
39. Nakajima K, Li T, Sumioka H, *et al.* Information theoretic analysis on a soft robotic arm inspired by the octopus (2011 IEEE International Conference on Robotics and Biomimetics Karon Beach, Thailand, Thailand). New York, NY, USA: IEEE, 2011, pp. 110–117.
40. Nakajima K, Li T, Kang R, *et al.* Local information transfer in soft robotic arm (2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China). New York, NY, USA: IEEE, 2012, pp. 1273–1280.
41. Beal D, Hover F, Triantafyllou M, *et al.* Passive propulsion in vortex wakes. J Fluid Mech 2006;549:385–402.
42. Wakita D, Kagaya K, Aonuma H. A general model of locomotion of brittle stars with a variable number of arms. J R Soc Interface 2020;17:20190374.
43. Liao JC. Neuromuscular control of trout swimming in a vortex street: implications for energy economy during the karman gait. J Exp Biol 2004;207:3495–3506.
44. Yamamoto W, Yuste R. Whole-body imaging of neural and muscle activity during behavior in *Hydra vulgaris*: effect of osmolarity on contraction bursts. ENEURO, 2020;7: ENEURO.0539-19.2020.
45. Nakajima K, Li T, Akashi N. Soft timer: dynamic clock embedded in soft body (Robotic Systems and Autonomous Platforms: Advances in Materials and Manufacturing). Sawston, Cambridge: Woodhead Publishing in Materials, 2018, pp. 181–196.
46. Nakajima K, Li T, Hauser H, *et al.* Exploiting short-term memory in soft body dynamics as a computational resource. J R Soc Interface 2014;11:20140437.
47. Nakajima K. Muscular-hydrostat computers: Physical reservoir computing for octopus-inspired soft robots. In Shigeno S, Murakami Y, Nomura T, eds. Brain Evolution by Design. Japan: Springer, 2017, pp. 403–414.
48. Nakajima K, Hauser H, Kang R, *et al.* A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. Front Comput Neurosci 2013;7:91.
49. Nakajima K, Hauser H, Kang R, *et al.* Computing with a muscular-hydrostat system (Proceedings of 2013 IEEE International Conference on Robotics and Automation, ICRA). New York, NY: IEEE, 2013, pp. 1496–1503.
50. Zhao Q, Nakajima K, Sumioka H, *et al.* Spine dynamics as a computational resource in spine-driven quadruped locomotion (Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS). New York, NY: IEEE, 2013, pp. 1445–1451.
51. Inoue K, Nakajima K, Kuniyoshi Y. Soft bodies as input reservoir: role of softness from the viewpoint of reservoir computing (2019 International Symposium on Micro-NanoMechatronics and Human Science, MHS). Nagoya, Japan: IEEE, 2019, pp. 1–7.
52. Nakajima K. Physical reservoir computing–an introductory perspective. Japn J Appl Phys 2020;59:060501.
53. Tanaka K, Yang SH, Tokudome Y, *et al.* Flapping-wing dynamics as a natural detector of wind direction. Adv Intell Syst 2020. [Epub ahead of print]; DOI: 10.1002/aisy.202000174.
54. Li S, Katzschmann R, Rus D. A soft cube capable of controllable continuous jumping (2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS). Hamburg, Germany: IEEE, 2015, pp. 1712–1717.

55. Tomo TP, Wong WK, Schmitz A, *et al.* A modular, distributed, soft, 3-axis sensor system for robot hands (2016 IEEE-RAS 16th International Conference on Humanoid Robots, Humanoids). Cancun, Mexico: IEEE, 2016, pp. 454–460.
56. Zimmer J, Hellebrekers T, Asfour T, *et al.* Predicting grasp success with a soft sensing skin and shape-memory actuated gripper (2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS). Macau, China: IEEE, 2019, pp. 7120–7127.
57. Jaeger H. Short Term Memory in Echo State Networks, vol. 5. Germany: GMD-Forschungszentrum Informationstechnik, 2001.
58. Truby RL, Wehner M, Grosskopf AK, *et al.* Soft somatosensitive actuators via embedded 3D printing. Adv Mater 2018;30:1706383.
59. Pfeifer R, Bongard J. How the Body Shapes the Way We Think: A New View of Intelligence. Massachusetts: MIT Press, 2006.
60. Pfeifer R, Lungarella M, Iida F. Self-organization, embodiment, and biologically inspired robotics. Science 2007; 318:1088–1093.
61. Hauser H, Ijspeert AJ, Füchslin RM, *et al.* Towards a theoretical foundation for morphological computation with compliant bodies. Biol Cybern 2011;105:355–370.
62. Hauser H, Ijspeert AJ, Füchslin RM, *et al.* The role of feedback in morphological computation with compliant bodies. Biol Cybern 2012;106:595–613.
63. Abbott LF, Rajan K, Sompolinsky H. Interactions between intrinsic and stimulus-evoked activity in recurrent neural networks. In Ding M, Glanzman D, eds. The dynamic brain: an exploration of neuronal variability and its functional significance. Oxford, UK: Oxford University Press, 2011, pp. 1–16.
64. Braham M, Van Droogenbroeck M. Deep background subtraction with scene-specific convolutional neural networks (2016 international conference on systems, signals and image processing, IWSSIP). Bratislava, Slovakia: IEEE, 2016, pp. 1–4.
65. Babaee M, Dinh DT, Rigoll G. A deep convolutional neural network for video sequence background subtraction. Pattern Recognit 2018;76:635–649.

Address correspondence to:
Katsuma Inoue
Graduate School of Information Science
and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo 113-8656
Japan

E-mail: k-inoue@isi.imi.i.u-tokyo.ac.jp

Appendix

Basal Point Estimation

Different basal point estimation algorithms were used in the three demonstrations: dead trout, brittle star, and soft octopus arm. In the demonstration of the dead trout video, the manually annotated head dynamics were used as basal point $\mathbf{s}(t)$. In the soft octopus arm case, a point maximizing the value of distance field D_t was selected as $\boldsymbol{s}(t)$ in the ε-neighborhood of the previous basal point $\boldsymbol{s}(t-1)$:

$$\boldsymbol{s}(t) = \underset{x \in A_t}{\arg\max}\, D_t(\boldsymbol{x}), \tag{A1}$$

$$A_t := \{\boldsymbol{x} \in \Omega_t \mid \parallel \boldsymbol{x} - \boldsymbol{s}(t-1) \parallel < \varepsilon\}. \tag{A2}$$

In the soft octopus arm extraction, the Y-coordinate of $\mathbf{s}(t)$ was fixed at y_0 and the X-coordinate was set to maximize the value of distance field D_t:

$$\boldsymbol{s}(t) = \underset{x \in B_t}{\arg\max}\, D_t(\boldsymbol{x}), \tag{A3}$$

$$B_t := \{[x, y]^T \in \Omega_t \mid \parallel [x, y]^T - \boldsymbol{s}(t-1) \parallel < \varepsilon \wedge y = y_0\}. \tag{A4}$$

Mutual Information

The mutual information $MI(X; Y)$ between two random variables X, Y is calculated by the following formula:

$$MI(X; Y) = \sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} P_{XY}(x, y) \log\left(\frac{P_{XY}(x, y)}{P_X(x) P_Y(y)}\right), \tag{A5}$$

where P_{XY} is the joint probability of X, Y.

Estimation of Three-Dimensional Skeletal Coordinates

The three-dimensional skeletal coordinates shown in Supplementary Video S8 were estimated by properly mapping the front Z-coordinate (vertical positions) to the side one. This mapping was obtained by minimizing the following cost function $\mathcal{L}\left(\left\{t_i^{\mathrm{s}}\right\}_i\right)$ for given values of $\left\{t_i^{\mathrm{f}}\right\}_i$:

$$\mathcal{L}\left(\left\{t_i^{\mathrm{s}}\right\}_i\right) := \sum_i \left| z_{\mathrm{f}}\left(t_i^{\mathrm{f}}\right) - z_{\mathrm{s}}\left(t_i^{\mathrm{s}}\right) \right|, \tag{A6}$$

where $z_{\mathrm{f}} : [0, 1] \to \mathbb{R}$ and $z_{\mathrm{s}} : [0, 1] \to \mathbb{R}$ are parametric representations of the front and side Z-coordinates, respectively (we used quadratic interpolation), $\left\{t_i^{\mathrm{f}}\right\}_i$ and $\left\{t_i^{\mathrm{s}}\right\}_i$ are monotonically increasing sequences s.t. $0 = t_1^k \leq t_1^k \leq \cdots \leq t_N^k = 1 (k = \mathrm{f}, \mathrm{s})$. Each skeleton was scaled to have the same value range of the Z-coordinate.