Unsupervised 3D Link Segmentation of Articulated Objects With a Mixture of Coherent Point Drift

Jaegoo Choy, Geonho Cha, and Songhwai Oh, *Member, IEEE*

***Abstract*—In this letter, we address the 3D link segmentation problem of articulated objects using multiple point sets with different configurations. We are motivated by the fact that a point set of an object can be aligned to point sets with different configurations by applying rigid transformations to links. Since existing 3D part segmentation datasets do not provide motion-based annotations, we propose a novel dataset of articulated objects, which are annotated based on its kinematic models. We define the point set alignment process as a probability density estimation problem and find the optimal decomposition of the point set and deformations using the EM algorithm. In addition, to improve the segmentation performance, we propose a regularization loss designed with a physical prior of decomposition. We evaluate the proposed method on our dataset, demonstrating that the proposed method achieves the state-of-the-art performance compared to baseline methods. Finally, we also propose an effective target manipulating point proposer, which can be applied to collect multiple point sets from an unknown object with different configurations to better solve the 3D link segmentation problem.**

***Index Terms*—3D link segmentation, unsupervised method, 3D link segmentation dataset.**

I. Introduction

MANIPULATING articulated objects is one of the core problems in robotics, which is hard to handle because of its inherent high degrees of freedom. However, if we know the kinematics model of an articulated object, it will be much easier to manipulate the object. Hence, kinematics-model-based manipulations have been extensively studied in robotics [1]–[3]. Here, decomposing an object into several links in kinematics, is one of the most important steps in understanding the kinematics of an object.

Due to its importance, extensive work on 3D part segmentation has been conducted in many ways. A main approach is utilizing massive data with 3D part annotation to learn a deep neural network in a supervised manner [4]–[6]. These datasets consist of extensive samples in daily categories (e.g., scissors, chairs, and cups) with category-level or instance-level annotations. However, supervised-learning-based methods could have a generalization issue, i.e., their performances might be degraded in case of unseen object categories. In addition, they need massive data with expensive annotations. To resolve this issue, unsupervised-learning-based approaches have been proposed [7]–[9].

It should be noted that existing 3D part segmentation datasets may not be appropriate to study the kinematics model of objects since their annotations are not based on motions of objects. For example, in the ShapeNet dataset, a chair is divided into a backrest, a seat, and legs, but the chair is considered a rigid body in the aspect of its kinematics model. Hence, we need a dataset which is suitable for kinematics studies.

In this letter, to resolve these issues, we propose a novel unsupervised link segmentation algorithm and a novel KinArt3D dataset.[1] Our model is designed with the following intuition: given different point sets of the same object, the target point sets can be aligned by dividing the source point set into several links and applying a rigid transformation to each link, as illustrated in Fig. 1. The goal of our model is to find an optimal decomposition

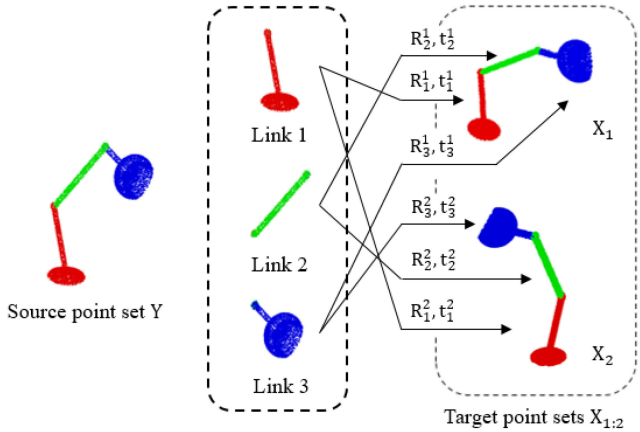

Fig. 1. A visualization of the point sets and the corresponding transformation parameters. In this example, the source point set Y maps to two target point sets $X_{1:2}$. From the shared link decomposition information and transformation parameters $\{R_{1:3}^1, t_{1:3}^1\}$ and $\{R_{1:3}^2, t_{1:3}^2\}$, Y is aligned to X_1 and X_2.

Manuscript received February 24, 2022; accepted May 16, 2022. Date of publication June 8, 2022; date of current version June 15, 2022. This letter was recommended for publication by Associate Editor R. Carloni and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korea Government (MSIT) under Grants 2019-0-01371, Development of Brain-Inspired AI with Human-Like Intelligence, and 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning. *(Corresponding author: Songhwai Oh.)*

Jaegoo Choy and Songhwai Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea (e-mail: jaegu.choy@cpslab.snu.ac.kr; songhwai@snu.ac.kr).

Geonho Cha is with the Clova AI, NAVER Corp, Seongnam 13561, Korea (e-mail: geonho.cha@cpslab.snu.ac.kr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3180444, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3180444

[1]This dataset is available at https://rllab-snu.github.io/projects/mcpd/doc.html

of the point set and transformations as a probability density estimation problem. We adopt the probability-based approach as it is robust to noise, outliers, and missing points, which can frequently occur in practical situations. The probability density is a Gaussian mixture model weighted by a latent variable inferring the decomposition of the point set. After the optimization procedure, we can figure out the links of the object. Here, it is the optimal decomposition which allows us to align different configurations of point sets by applying rigid transformations to links. However, due to the inherent high degrees of freedom of the problem, the results might not be satisfactory. To mitigate this issue, we propose a regularization loss designed with a physical prior of decomposition, which can be used in the optimization step. We demonstrate the effectiveness of the proposed method in the experiments section by comparing the performance of the proposed method with other state-of-the-art methods.

In addition, for practical robotic uses, we also propose a target manipulating point proposer. In most environments, such as a robotic manipulating problem, since an object is given rather than point sets of the object, it is necessary to gather point sets of various configurations of an object. In that situation, the target manipulating point proposer enables the robotic manipulator to obtain target points so that the proposed method can achieve high performance. In MuJoCo simulator [10], we demonstrate the effectiveness of the target manipulating point proposer using the objects in the KinArt3D dataset.

In summary, the contributions of this letter can be summarized as:

- We propose the KinArt3D dataset which can be used as a benchmark dataset in the 3D link segmentation of articulated objects.
- We propose a novel method of decomposing an articulated object into several links in an unsupervised way.
- Our method shows the state-of-the-art performance on the KinArt3D dataset. Specifically, the mean IoU of our method on the KinArt3D dataset is improved by 41% compared to the baseline methods.
- We also propose a target manipulating point proposer.

II. Related Work

In this section, we summarize related studies in four parts: 3D part dataset, deep-neural-network-based 3D part segmentation algorithms, point set registration methods, and online kinematics estimation methods.

3D part datasets: ShapeNet [11] is a large-scale 3D CAD dataset consists of 220,000 models. In addition, its subset, ShapeNetCore [11], covers 55 object categories with about 51,300 unique 3D models and provides detailed semantic annotations. PartNet [6] is a large-scale dataset providing fine-grained and hierarchical instance-level part semantic annotations. It consists of 26,671 3D models covering 24 object categories, and 14 categories are selected from ShapeNet. However, they only provide rigid 3D models, making utilization of them in robot manipulation studies difficult. Thus, we propose the KinArt3D dataset that provides 3D models with articulated motions to resolve this issue.

3D part segmentation algorithms: Thanks to the increased computation capacity and large-scale datasets, supervised learning methods using neural networks show favorable performance in 3D part segmentation [7]–[9]. The multi-scale U-Net proposed by Lyu *et al.* [7] maps point clouds into a 2D image space using 2D convolutional networks. 3D Graph Convolution Networks (3D-GCN) of Lin *et al.* [8] is a graph drawing algorithm that projects point clouds into 2D image representations while preserving the local information among the points in each cloud. Thomas *et al.* [9] presented a deformable convolution which is robust to varying point cloud densities. In spite of the high performance of the supervised methods, its application to general objects is challenging, as they require expensive semantic annotations for the training.

Point set registration: Point set registration has been studied since 1990 s, and there are many related studies, but we describe only the most general methods. Iterative Closet Point (ICP) [12], [13] alternatively finds the correspondences by searching the nearest points and the rigid transformations which minimize the distance between the point sets until convergence. However, it is quite sensitive to the initial condition. To alleviate this issue, variations of ICP have been studied to solve the shortcomings of ICP. Specifically, Gold *et al.* [14] proposed a robust ICP algorithm which is less sensitive to an initial condition. The most popular method among probability-based point set registration methods is coherent point drift (CPD) [15] which formulates maximum likelihood estimation problem using a Gaussian mixture model. However, the existing point set registration methods do not handle articulated objects, which inhibits their practical utilizations.

Online kinematics: Tzionas *et al.* [16] proposed an algorithm that generates a 3D mesh from a depth image and segments the mesh by motion tracking in animation. Nunes *et al.* [17] performed unsupervised online learning of the 3D kinematic structures of random articulated objects by introducing a similarity matrix of a point cloud. However, these methods have the limitation of requiring correspondence between point sets.

III. KinArt3D Dataset

The KinArt3D dataset provides instance-level link segmentation annotation for 91,075 shapes of 92 unique models from eight articulated object categories. Fig. 2 shows some examples of our dataset. We first manually created a kinematic model for each object and prepared a mesh model of each link. Fig. 3 illustrates five links of an example arm object and its tree structure. The edge of the tree means that there is a joint between the two connected links and stores three pieces of information: joint type, axis, and range. In the KinArt3D dataset, we consider 1D revolute joints and 2D universal joints. By uniformly sampling joint angles from its ranges, calculating the global 3D transformation of each link, and taking the calculated 3D transformation to each link, we can create a large number of different shapes from a single object. The remaining task is to check collisions between different links, and if there is no collision, merge all links and sample points from the merged mesh. In conclusion, by elaborately decomposing the object into a few links and generating a tree

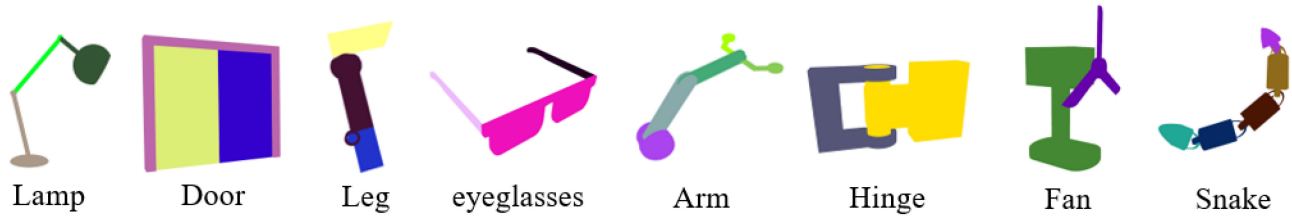

Fig. 2. An illustration of eight object categories in the KinArt3D dataset.

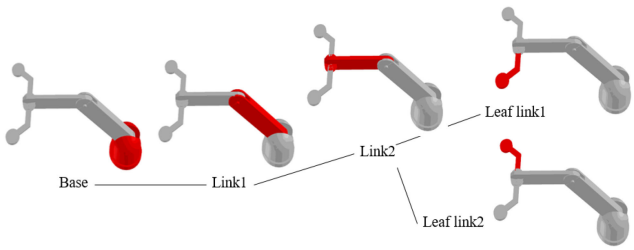

Fig. 3. A visualization of the kinematic chain of an arm object.

structure based on its kinematic model, we can get a large set of points with precisely labeled links without human segmentation.

By using the farthest sampling method [23], we sample 2,048 points from the merged meshes and normalize them so that their coordinates are between -1 and 1. In addition, we also voxelize the merged meshes with 64^3 resolutions for the unsupervised baseline method.

IV. Problem Formulation

Our goal is to align the source point set Y to the S target point sets $\mathrm{X}_{1:S} = \{\mathrm{X}_1, \ldots, \mathrm{X}_S\}$ by decomposing Y into K links and rigidly transforming the decomposed links, as illustrated in Fig. 1. Each target point set consists of N points. To this end, we find the parameters $\theta = \{\Pi, \mathrm{R}_{1:S,1:K}, \mathrm{t}_{1:S,1:K}, \sigma^2\}$. Π is the conditional probability matrix with M rows and K columns of which the (i,j)-th element π_{ij} represents the probability that the i-th source point belongs to the j-th link, where M is the number of points in the source point set. Here, π_{ij} satisfies the following two conditions: $0 \leq \pi_{ij} \leq 1$ for $\forall i \in \{1, \ldots, M\}, \forall j \in \{1, \ldots, K\}$ and $\sum_{j=1}^{K} \pi_{ij} = 1$ for $\forall i \in \{1, \ldots, M\}$. We also introduce a discrete latent variable Z that determines the link from which the source point originates. In other words, $p(\mathrm{Z}_i = j | y_i)$ is the probability that the i-th source point belongs to the j-th link, and it is equal to π_{ij}. R_{sk} and t_{sk} are the rigid transformation parameters, rotation and translation, respectively, of the k-th link to the s-th target point set. And σ is the standard deviation for generating a probability model of the source point set Y.

A. Rigid Link Registration

Similar to CPD [15], the points in Y become centroids to define the GMM probability density function. The conditional probability of the s-th point of the n-th target point set x_{sn}, given the m-th point of the source point set y_m and the state $\mathrm{Z}_m = k$, is as follows:

$$p(x_{sn}|y_m, \mathrm{Z}_m = k, \theta) = \frac{\exp^{-\frac{1}{2\sigma^2}\|x_{sn} - \mathrm{R}_{sk} y_m - \mathrm{t}_{sk}\|^2}}{(2\pi\sigma^2)^{D/2}} := g(x_{sn}, \mathrm{R}_{sk} y_m + \mathrm{t}_{sk}, \sigma^2), \quad (1)$$

where $g(x_{sn}, \mathrm{R}_{sk} y_m + \mathrm{t}_{sk}, \sigma^2)$ is a normal distribution with mean $\mathrm{R}_{sk} y_m + \mathrm{t}_{sk}$ and the standard deviation σ, and D is the dimension of the point sets.

We optimize the parameters θ by using Expectation-Maximization [19]. To this end, we define the expected value of the log-likelihood probability function with respect to a conditional probability distribution of the hidden variable Z given observations and the current estimates of the parameters $\hat{\theta}$ as follows:

$$Q(\theta|\hat{\theta}) = \mathbb{E}_{\mathrm{Z}|\mathrm{X}_{1:S}, \mathrm{Y}, \hat{\theta}} \left[\log p(\mathrm{X}_{1:S}, \mathrm{Y}, \mathrm{Z}|\theta)\right]. \quad (2)$$

The E-step computes the value $Q(\theta|\hat{\theta})$ and the M-step finds parameters θ which maximizes Q given the current estimates of the parameters $\hat{\theta}$. From now, to simplify equations, the sum or product over multiple variables is replaced with a single summation or product notation (e.g., $\sum_{sm} f(s,m) \equiv \sum_{s=1}^{S} \sum_{m=1}^{M} f(s,m)$).

1) E-Step: Since the conditional probability function $\hat{\mathrm{T}}_{snmk} \equiv p(\mathrm{Z}_m = k | x_{sn}, y_m; \hat{\theta})$ can be calculated as

$$\frac{\hat{\pi}_{mk} g(x_{sn}, \hat{\mathrm{R}}_{sk} y_m + \hat{\mathrm{t}}_{sk}, \hat{\sigma}^2)}{\sum_{j=1}^{K} \hat{\pi}_{mj} g(x_{sn}, \hat{\mathrm{R}}_{sj} y_m + \hat{\mathrm{t}}_{sj}, \hat{\sigma}^2)}, \quad (3)$$

we can compute $Q(\theta|\hat{\theta})$ as follows:

$$Q(\theta|\hat{\theta}) = \sum_{snmk} \hat{\mathrm{T}}_{snmk} \left[\log \frac{1}{M} + \log \pi_{mk} - \frac{D}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \|x_{sn} - \mathrm{R}_{sk} y_m - \mathrm{t}_{sk}\|^2 \right]. \quad (4)$$

2) M-Step: In the M-step, we fix the current estimates of the parameters $\hat{\theta}$ and obtain the parameters θ which maximizes $Q(\theta|\hat{\theta})$. After taking a partial derivative of the objective with respect to t, π and σ^2, we obtain the parameters which make the corresponding derivatives equal to zero.

Taking partial derivative of Q with respect to t_{sk} and equate it to zero, we obtain $\mathrm{t}_{sk} = \mu_{skx} - \mathrm{R}_{sk} \mu_{sky}$, where the mean vectors μ_{skx} and μ_{sky} for the s-th target point set and the k-th

link are defined as:

$$\mu_{skx} = \frac{\sum_{nm} \hat{\mathrm{T}}_{snmk} x_{sn}}{\sum_{nm} \hat{\mathrm{T}}_{snmk}}, \ \mu_{sky} = \frac{\sum_{nm} \hat{\mathrm{T}}_{snmk} y_m}{\sum_{nm} \hat{\mathrm{T}}_{snmk}}. \quad (5)$$

In the same way, the σ^2 can be updated as:

$$\sigma^2 = \frac{\sum_{snmk} \hat{\mathrm{T}}_{snmk} \left\| x_{sn} - \mathrm{R}_{sk} y_m - \mathrm{t}_{sk} \right\|^2}{\sum_{snmk} \hat{\mathrm{T}}_{snmk} D}. \quad (6)$$

Substituting t into the objective function and rewriting it in a matrix form, we obtain

$$\mathrm{R}_{sk} = \arg\max_{\mathrm{R}_{sk}} \mathrm{tr}\left(\mathrm{A}_{sk}^T \mathrm{R}_{sk}\right), \ \mathrm{A}_{sk} = \mathrm{X}_{sk}^T \hat{\mathrm{T}}_{sk}^T \mathrm{Y}_{sk}, \quad (7)$$

where $\mathrm{X}_{sk} = \mathrm{X}_s - 1\mu_{skx}$, $\mathrm{Y}_{sk} = \mathrm{Y} - 1\mu_{sky}$, and $\hat{\mathrm{T}}_{sk}$ is a $N \times M$ matrix of which the (i, j)-th element is $\hat{\mathrm{T}}_{sijk}$. From [20], the optimal R_{sk} can be calculated as UCV^T, where USV^T is SVD of A_{sk} and $\mathrm{C} = \mathrm{diag}(1, \ldots, 1, \det(\mathrm{UV}^T))$.

Finally, the optimization rule for π with a constraint can be represented as

$$\pi = \arg\max_{\pi} \sum_{snmk} \hat{\mathrm{T}}_{snmk} \log \pi_{mk} \quad \text{s.t.} \sum_k \pi_{mk} = 1 \text{ for } \forall m. \quad (8)$$

It can be solved by using the Lagrange multiplier theorem [21] and we obtain $\pi_{mk} = \frac{\sum_{sn} \hat{\mathrm{T}}_{snmk}}{\sum_{snk} \hat{\mathrm{T}}_{snmk}}$.

B. Regularized Rigid Link Registration

In Section IV-A, we propose a novel method to divide the source point set into its kinematic links without correspondences between point sets. We further improve the method by introducing a novel regularization term which can be used in conjunction with the objective function. We employ the KL divergence to make the conditional probability π of neighboring points similar during the optimization step. The regularization term is defined and approximated as

$$R(\Pi) = -\sum_{i,j=1}^{M} w_{ij} D_{\mathrm{KL}}(\pi_i || \pi_j) \approx -2 \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik} \log \frac{\pi_{ik}}{\pi_{mk}}, \quad (9)$$

where the binary coefficient w_{ij} denotes whether the distance between y_i and y_j is less than ϵ or not.

Since the regularization term is only affected by Π, the optimization rules for the other parameters are the same as in Section IV-A. As described in Section IV-A, we utilize the Lagrange multiplier theorem and the optimal π_{mk} with the regularization loss is

$$\pi_{mk} = \frac{\sum_{sn} \hat{\mathrm{T}}_{snmk} + 2 \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik}}{\sum_{snk} \hat{\mathrm{T}}_{snmk} + 2 \sum_k \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik}}. \quad (10)$$

When two conditions $\sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik} \ll \sum_{sn} \hat{\mathrm{T}}_{snmk}$ and $\sum_k \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik} \ll \sum_{snk} \hat{\mathrm{T}}_{snmk}$ are satisfied, the optimal π_{mk} can be approximated as $\frac{\sum_{sn} \hat{\mathrm{T}}_{snmk}}{\sum_{snk} \hat{\mathrm{T}}_{snmk}}$ which is identical to the results for unregularized optimization in Section IV-A. We summarize the entire optimization steps of the regularized rigid link registration in Algorithm 1.

Algorithm 1: Regularized Rigid Link Registration.

Initialize $\mathrm{R}_k^s = \mathrm{I}, \mathrm{t}_k^s = 0$, for $\forall k \in \{1, \ldots, K\}$ and $\forall s \in \{1, \ldots, S\}$
Initialize $\sigma^2 = \frac{1}{SMND} \sum_{snm} \left\| x_{sn} - y_m \right\|^2$
Initialize $\Pi_{M \times K}$ using GMM clustering method
Set a binary matrix W ($w_{ij} = 1$ if $\|y_i - y_j\| < \epsilon$ else 0)
repeat
  $\mathrm{T}_{nmk}^s = \frac{\pi_{mk} g(x_{sn}, \mathrm{R}_{sk} y_m + \mathrm{t}_{sk}, \sigma^2)}{\sum_{j=1}^{K} \pi_{mj} g(x_{sn}, \mathrm{R}_{sj} y_m + \mathrm{t}_{sj}, \sigma^2)}$
  for s $\leftarrow$ 1 to S **do**
    for k $\leftarrow$ 1 to K **do**
      $\mu_{skx} = \frac{\sum_{nm} \mathrm{T}_{snmk} x_{sn}}{\sum_{nm} \mathrm{T}_{snmk}}, \ \mu_{sky} = \frac{\sum_{nm} \mathrm{T}_{snmk} y_m}{\sum_{nm} \mathrm{T}_{snmk}}$
      $\mathrm{X}_{sk} = \mathrm{X}_s - 1\mu_{skx}, \mathrm{Y}_{sk} = \mathrm{Y} - 1\mu_{sky}$
      $\mathrm{A}_{sk} = \mathrm{X}_{sk}^T \mathrm{T}_{sk}^T \mathrm{Y}_{sk}$, and compute SVD of A_{sk}
      $\mathrm{R}_{sk} = \mathrm{UCV}^T$, where $\mathrm{C} = \mathrm{diag}(1, \ldots, 1, \det(\mathrm{UV}^T))$
      $\mathrm{t}_{sk} = \mu_{skx} - \mathrm{R}_{sk} \mu_{sky}$
    end for
  end for
  $\sigma^2 = \frac{\sum_{snmk} \mathrm{T}_{snmk} \|x_{sn} - \mathrm{R}_{sk} y_m - \mathrm{t}_{sk}\|^2}{\sum_{snmk} \mathrm{T}_{snmk} D}$
  $\pi_{mk} = \frac{\sum_{sn} \mathrm{T}_{snmk} + 2 \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik}}{\sum_{snk} \mathrm{T}_{snmk} + 2 \sum_k \sum_{\substack{i=1 \\ i \neq m}}^{M} w_{im} \pi_{ik}}$
until converge

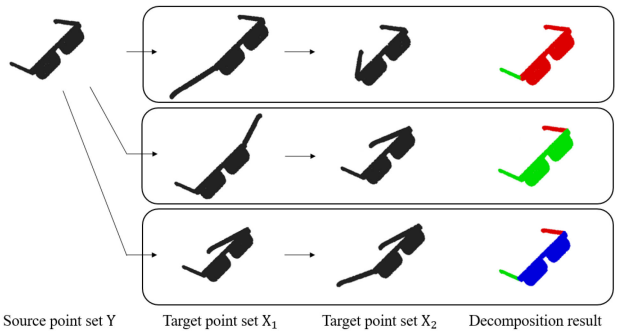

Fig. 4. Some results of the regularized link registration process. A source point set is used for the process with different target point sets. In the first and second cases, the source point set is decomposed into only two links.

V. Target Manipulating Point Proposer

When running the proposed algorithm on the KinArt3D dataset, all point sets are randomly sampled from the pool of point sets of the same object in the dataset. However, this method cannot be applied when facing an unknown articulated object on the table because only one source point set is observed. It is necessary to create new target point sets with different configurations by manipulating the object to run our algorithm. However, gathering target point sets by randomly manipulating the object may not help to understand the object's structure. When collecting the target point sets through completely random manipulation, sometimes not all links are detected, as shown in Fig. 4. Only in the third case, the target point set enables the

proposed method to find all object links. Meanwhile, in the other two cases, the target point set X_2 is not informative. As such, the need for a method to gather informative target point sets leads us to suggest a target manipulating point proposer.

The ultimate goal of the target manipulation point proposer is to select some manipulation point $x \in \mathrm{X}_h$ given the target point sets $\mathrm{X}_{1:h}$ and the source point set Y. Instead of directly sampling the target manipulating point from the last target point set X_h, we sample the target manipulating point from the source point set Y and manipulate the corresponding point from the last target point set X_h. The criteria for selecting the target manipulation point from the source point set Y and how to find the target point $x \in \mathrm{X}_h$ corresponding to an arbitrary source point $y_m \in \mathrm{Y}$ will be explained.

After the optimization of parameters θ with the source point Y and the target point sets $\mathrm{X}_{1:h}$, for $m \in \{1, \ldots, M\}$, we can infer the point $x \in \mathrm{X}_h$ which is the most closely related to the point $y_m \in \mathrm{Y}$ by using the parameters θ as follows:

$$x^*, k = \arg \max_{x \in \mathrm{X}_h, j} \pi_{mj} g(x, \mathrm{R}_{hj} y_m + \mathrm{t}_{hj}, \sigma^2). \quad (11)$$

For convenience, the point in the target point set X_h corresponding to y_m is denoted as x_{hm*}.

The proposer introduces M-dimensional probabilities Φ that satisfy $\sum_{m=1}^{M} \phi_m = 1$ and choices $y_i \in \mathrm{Y}$ over the probabilities Φ. At the initial state where no target point set is given, all values of Φ are initialized to $\frac{1}{M}$ to uniformly sample point y_i. After the new target point set X_h is collected, the parameters θ are optimized through the Algorithm 1, and the probabilities Φ are updated based on the parameters θ. We employ the Gaussian distribution with mean x_{hi^*} and the standard deviation σ^2 to make the target point proposal not close to the previous manipulating point. Furthermore, we employ the entropy of Π to focus the proposal on the unstable source points. For example, suppose that it is learned that the point y_i belongs to the j-th link. Then π_{ij} will be close to one while π_{ik} will be close to zero for $k \neq j$. Hence, the entropy of $\pi_i = [\pi_{i1} \cdots \pi_{iK}]$ will be small. On the other hand, the "unstable" source points such as points near joints of the object have relatively high entropy. The optimization rule for Φ is as follows:

$$\begin{aligned} \phi_m \leftarrow \phi_m - \alpha \cdot \mathrm{MINMAX}\left(g(x_{hm*}, x_{hi*}, \sigma^2)\right) - \\ \beta \cdot \mathrm{MINMAX}\left(\sum_{k=1}^{K} \pi_{mk} \log \pi_{mk}\right), \\ \phi_m \leftarrow \mathrm{MINMAX}(\phi_m), \end{aligned} \quad (12)$$

where MINMAX is min-max normalization over variable m, i.e., $\mathrm{MINMAX}(f(m)) = \frac{f(m) - \min_m(f(m))}{\max_m(f(m)) - \min_m(f(m))}$. When the α is high, the proposer mainly samples the target point far from the previous manipulating points. Conversely, when the β is high, the proposer mainly samples the target point where segmentation is unstable, especially near joints.

Algorithm 2: Target Manipulating Point Proposer.

1: Receive the source point set Y
2: Initialize the point proposal probabilities $\Phi \leftarrow \frac{1}{M}$
3: Initialize target point set buffer $\bar{\mathrm{X}} \leftarrow \{\}$
4: **for** $h \leftarrow 1$ to max _ step **do**
5: Sample one point $y_i \in \mathrm{Y}$ over the probabilities Φ
6: Manipulate the matched point $x_{hi^*} \in \mathrm{X}_h$ ($\mathrm{X}_1 = \mathrm{Y}$)
7: Store the new target point set X_{h+1} in $\bar{\mathrm{X}}$
8: Optimize the parameters θ as illustrated in section 4.B
9: $\phi_m \leftarrow \phi_m - \alpha \cdot \mathrm{MINMAX}(g(x_{hm*}, x_{hi*}, \sigma^2))$ - $\beta \cdot \mathrm{MINMAX}(\sum_{k=1}^{K} \pi_{mk} \log \pi_{mk})$
10: Normalize Φ
11: **end for**

VI. Evaluation and Results

Test metrics: Evaluation metrics are mean IoU per category following the practices in [23]. We follow the evaluation protocol of [24], where the Hungarian method is used to find the best allocation of segments to the ground truth. Empty segments are introduced if the number of the estimated segments are less than that of the ground truth.

A. Link Decomposition on KinArt3D Dataset

1) Baseline Description: We compare the performance of the proposed method to various state-of-the-art methods [7]–[9], [22] to demonstrate the effectiveness of the proposed method. For comparison with the proposed method, we use three supervised 3D part segmentation methods, U-Net [7], 3Dgcn [8], and KPConv [9], and one unsupervised method BAE-Net [22]. We split our dataset into train and test sets based on the ratio of 8:2. The supervised methods are trained with the sampled 2048 points and its ground-truth segmentation label. On the other hand, the unsupervised method is trained using 64^3 voxels.

2) Quantitative Results: In Table I, we report the mean IoU scores per category and overall mean IoU scores. The proposed method with regularization achieves the best performance compared to the other methods. We use the following setting for the evaluation: $M = 2048$, $N = 2048$, $K = 7$, $S = 4$, and $\epsilon = 0.05$. Here, we observe that the performance of the supervised methods on our data set is not as high as that on Shapenet-Part dataset [11]. (Note that U-Net, 3Dgcn, and KPConv achieve 0.846, 0.821, and 0.851 in class mean IoU and in 0.888, 0.851, and 0.864 in instance mean IoU on Shapenet-Part dataset.) Those algorithms achieve similar performance only on glasses models, where all the objects have the same kinematics, of our dataset. According to the observation, we argue that the supervised methods have bad generalization performance. Since the mean IoU of the regularized method is higher than that of the unregularized method in all the categories except for the door as shown in Table I, we can conclude that adding the regularization term can improve the segmentation performance.

TABLE I
OVERALL SEGMENTATION RESULTS ON KINART3D DATASET

	supervision	class mIoU	instance mIoU	glasses	lamp	leg	arm	hinge	door	fan	snake
U-NET [7]	✓	0.555	0.540	0.813	0.537	0.177	0.424	0.490	0.615	0.761	0.619
3Dgcn [8]	✓	0.515	0.498	0.794	0.660	0.208	0.392	0.254	0.605	0.644	0.565
KPConv [9]	✓	0.525	0.510	**0.944**	0.552	0.080	0.304	0.230	0.491	0.630	0.757
BAE-NET [22]		0.528	0.552	0.478	0.679	0.583	0.416	0.684	0.585	0.327	0.472
Ours (w/o reg)		0.736	0.735	0.834	0.801	0.799	0.679	0.692	**0.674**	0.708	0.698
Ours		**0.784**	**0.792**	0.902	**0.852**	**0.813**	**0.715**	**0.835**	0.621	**0.808**	**0.729**

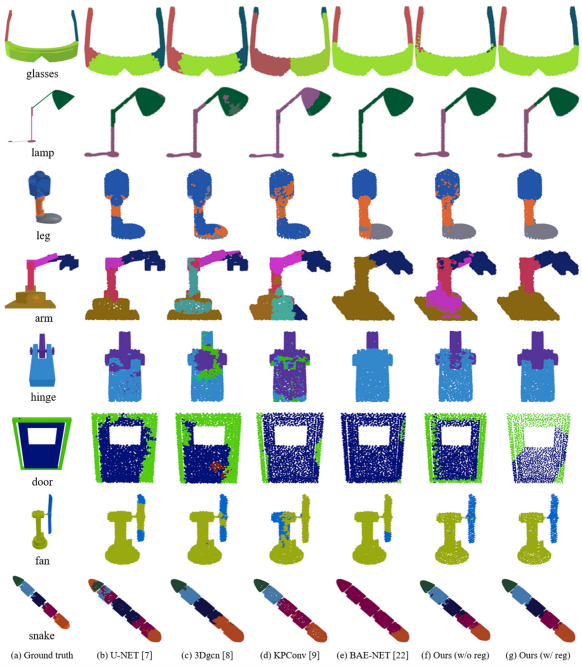

Fig. 5. An illustration of segmentation results on examples in KinArt3D dataset.

3) Qualitative Results: We also report the qualitative results in Fig. 5. We can demonstrate that the proposed method shows the best performance for all examples. Furthermore, the regularized method generally shows better results than the unregularized method. On the other hand, supervised methods show poor performance for examples of categories other than glasses. From this result, we can qualitatively observe that the supervised methods have bad generalization performance. Note that our method consistently shows satisfactory performance even if the number of links of the model in the same category varies. While supervised methods are vulnerable to unseen models in test data, the proposed method can accurately infer models in test data as our method estimates the decomposition without any prior knowledge of the model. This result is more conspicuous in the KinArt3D dataset, because the models in the KinArt3D dataset contain very diverse shapes and segmentation information even though they belong to the same category.

4) Results on Perturbed Data: Since there is a discrepancy between the point sets collected in the real environment and the simulated data, we test the proposed method on the KinArt3D dataset under varying levels of missing points, noise, and outliers. We generate missing points by randomly removing some points from the point set. The noise is sampled from the normal

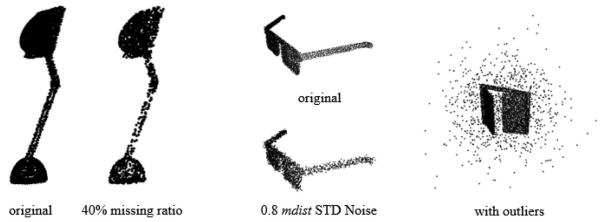

Fig. 6. Examples of missing points, noise, and outliers.

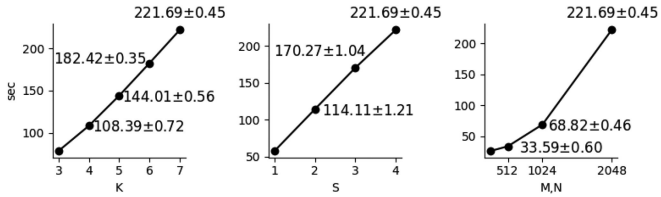

Fig. 7. The computational time measurement results of the proposed method in various implementation variables. (left) The number of links, K. (middle) The number of target point sets, S. (right) The number of points (M = N).

distribution with zero mean and standard derivation considering *mdist* which is the mean distance between a pair of nearest points and added to the point set position. The outliers sampled from the standard normal distribution are added to the point set. We have found out that a simple outlier removal method, namely the radius outlier removal method [25], is highly effective. Some examples of missing points, noise, and outliers are illustrated in Fig. 6 and the results of the proposed method are shown in Table II. It shows that the proposed method is robust against missing points, noises, and outliers.

5) Computation Time: To show the practicality of the proposed method, we measure the computation time of the proposed method in various implementation variables. Algorithm 1 is implemented in Python using the CUDA accelerated library and all results are measured on GeForce GTX 1080 Ti. The required calculation time is 221.69 ± 0.45 seconds under the following settings: $M = 2048$, $N = 2048$, $K = 7$, and $S = 4$. Fig. 7 shows the computation times while changing one variable and fixing the other variables. It shows that there is a linear relationship between the computation time and the implementation variables. A small deviation of the computation time indicates that the computation time of the proposed method does not depend on the complexity of objects.

6) Further Analysis for Experimental Results: Reconstruction results of the proposed method: We illustrate the process of aligning the target point set from the source point set in Fig. 8. At the final step, the transformed point set is closely aligned to the target point set, and the segmentation result is

TABLE II
RESULTS OF THE PROPOSED METHOD ON PERTURBED DATASET. THE MISSING RATIO IS THE RATIO OF THE NUMBER OF MISSING POINTS TO THE TOTAL NUMBER OF POINTS. THE NOISE STD IS THE STANDARD DEVIATION OF THE NOISE PROPORTIONAL TO *MDIST*. THE OUTLIERS IS THE NUMBER OF OUTLIERS ADDED TO THE POINT SET

missing ratio	lamp	leg	fan
0.0	0.791	0.779	0.803
0.1	0.792	0.784	0.769
0.2	0.779	0.779	0.767
0.4	0.754	0.754	0.725

Noise STD	glasses	leg	snake
0.0 *mdist*	0.905	0.790	0.744
0.2 *mdist*	0.897	0.779	0.708
0.4 *mdist*	0.891	0.727	0.640
0.8 *mdist*	0.872	0.682	0.599

outliers	arm	hinge	door
0	0.720	0.851	0.610
256	0.704	0.848	0.605
512	0.700	0.811	0.606
1024	0.677	0.813	0.592

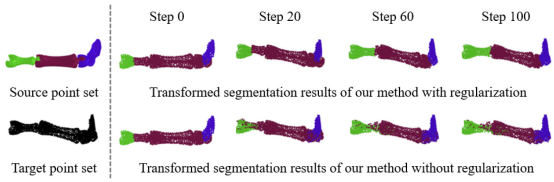

Fig. 8. An illustration of segmentation and reconstruction results per step.

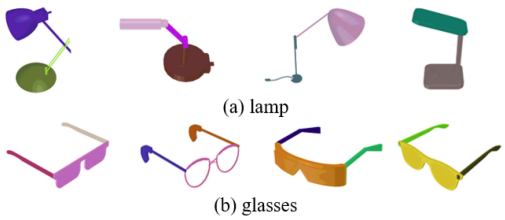

Fig. 9. An illustration of sample objects in lamp and glasses category.

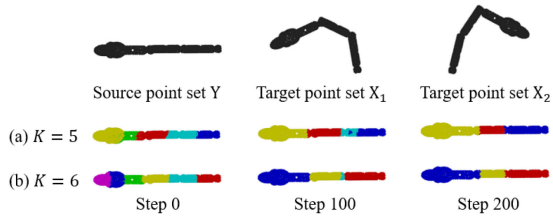

Fig. 10. An illustration of segmentation results depending on the number of links. Optimization results when K is (a) 5 and (b) 6.

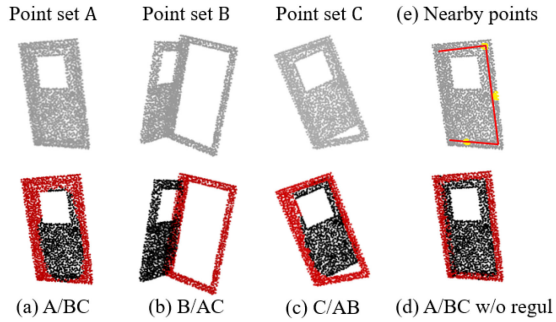

Fig. 11. An illustration of segmentation results depending on its source point set. Segmentation results of the proposed method when the source point set is (a) A, (b) B, and (c) C and the target point sets are the others. (d) Segmentation results of the unregularized method when the source point set is A. (e) Visualization of the range of nearby points applied to regularization.

also satisfactory. The successful aligning of two point sets, even though no correspondence between the source point set and the target point set is given, is a remarkable result.

The reason why the proposed method outperforms supervised methods except for the glasses category: The proposed method shows higher performance than supervised methods in all categories except the glasses category, as shown in Table I. It is contradictory that supervised methods perform better than unsupervised methods for most problems. The reason for this unusual result is that the KinArt3D dataset is annotated based on the kinematics of the object. For example, as shown in Fig. 9, a lamp object can have a different number of links depending on its kinematics, which negatively affects supervised methods. However, in the glasses category, since all objects have the same number of links and similar shapes, the supervised method shows its strength. In fact, KPConv shows the best performance in the glasses category.

The reason why the proposed method is robust to a varied number of links: When using K higher than the actual number of links, the optimization results of the proposed method are shown in Fig. 10. Even if K is larger than three (the actual number of links of the object), any excessive links are compressed during optimization, and only three links remain at the end. Because the proposed method derives results that match the actual number of links of the object regardless of K, the proposed method shows robust results in categories with various numbers of links.

Negative effects of the regularization term on door category: The unregularized method shows better results quantitatively and qualitatively than the regularized method for doors, as shown in Table I and Fig. 5. In Fig. 11(e), when the source point set is A, the ground-truth segmentation of the door dramatically changes bordering on the red line, whereas the regularization term makes the points in the yellow circle belong to the same link. These opposing goals make the regularized method's performance lower than that of the unregularized method in special cases such as a closed door.

B. Target Manipulating Point Proposer on MuJoCo

We use UR5 and Robotiq 2R-85 gripper of MuJoCo for the data collecting task of articulated objects. For convenience, objects in some categories, such as a lamp, are fixed to the table. All procedures are the same as described in Algorithm 2, and the point set is restored using its joint angles obtained from the simulator. After the point to be manipulated is determined, the robot gripper grabs the point, moves it randomly, and releases it. A process of collecting data with the proposed proposer is illustrated in Fig. 12.

To demonstrate the effectiveness of the proposed method, we report the mean IoU scores per category of the proposed method and random method. The random method is a method of randomly sampling manipulating points from the source point set by fixing the probabilities Φ to $\frac{1}{M}$ and not updating it.

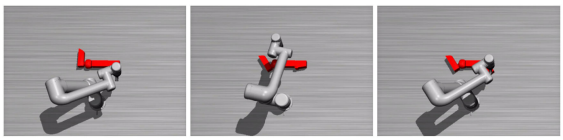

Fig. 12. An illustration of collecting data using the target manipulation point proposer on a leg object in simulation.

TABLE III
RESULTS OF TARGET MANIPULATING POINT PROPOSER AND COMPARISON WITH THE RANDOMIZED METHOD

iterations	glasses	lamp	leg	arm	hinge	door	fan	snake
1	0.394	0.536	0.531	0.270	0.579	0.271	0.525	0.287
2	0.636	0.684	0.582	0.463	0.653	0.549	0.598	0.491
3	**0.876**	0.800	0.703	0.416	**0.746**	0.645	0.426	0.499
4	0.852	**0.811**	**0.778**	**0.552**	0.725	**0.737**	**0.750**	**0.616**
4 (all random)	0.716	0.781	0.690	0.493	0.614	0.455	0.637	0.470

TABLE IV
SEGMENTATION RESULT OF THE TARGET MANIPULATING POINT PROPOSER WITH RESPECT TO VARIOUS SETTINGS OF α AND β

α / β	glasses	lamp	leg	arm	hinge	door	fan	snake
1.0 / 0.0	0.849	0.797	0.715	0.533	0.643	0.623	0.706	0.604
0.0 / 1.0	0.410	0.622	0.527	0.348	0.492	0.447	0.534	0.431
0.5 / 0.5	**0.852**	**0.811**	**0.778**	**0.552**	**0.725**	**0.637**	**0.750**	**0.616**

Table III, which shows the mean IoU scores per category of the proposed and randomized methods, proves the effectiveness of the proposed method. The results in Table III are measured at $\alpha = 0.5$ and $\beta = 0.5$ and the iterations is the number of target point sets, h, in Algorithm 2. The target manipulating point proposer consistently outperforms the randomized method.

Table IV shows the performance measured by changing the α and β in (10). The performance of $\alpha = 1$ is higher than the performance of $\beta = 1$, indicating that sampling unvisited points is more beneficial than sampling unstable points in understanding the structure of an object. However, the highest performance at $\alpha = 0.5$ and $\beta = 0.5$ means it is better to mix the two sampling strategies.

VII. CONCLUSION

We have introduced a novel 3D link segmentation algorithm for articulated objects given a group of point sets. While we align one point set to other point sets, we decompose the point set into several links and find rigid transformations of each link. We have formulated the alignment problem as a probability density estimation where the point set is represented as a Gaussian mixture model. Since the existing 3D part datasets do not match the link annotation from the kinematics point of view, we have also proposed a novel dataset of articulated objects based on its kinematics model. We have solved the optimization problem using the EM algorithm. In addition, we have proposed a regularization term to prevent adjacent points belonging to different links. We have evaluated the proposed method on our dataset, where the proposed method achieves the state-of-the-art performance compared to various baseline methods. Finally, we have proposed a method for efficiently collecting target point sets of an unknown object using a robotic manipulator and demonstrated it experimentally.

REFERENCES

[1] D. Katz, Y. Pyuro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded relational representation," in *Robot.: Sci. Syst.*, 2008.

[2] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 5003–5010.

[3] S. Zimmermann, R. Poranne, and S. Coros, "Dynamic manipulation of deformable objects with implicit integration," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 4209–4216, Apr. 2021.

[4] Z. Wu *et al.*, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[5] L. Yi *et al.*, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016.

[6] K. Mo *et al.*, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 909–918.

[7] Y. Lyu, X. Huang, and Z. Zhang, "Learning to segment 3D point clouds in 2D image space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12255–12264.

[8] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3D graph convolution networks for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1800–1809.

[9] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.

[10] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Conf. Int. Robots Syst.*, 2012, pp. 5026–5033.

[11] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[12] P. J. Besl and N. D. McKay, "Method for registration of 3D shapes," *Sensor Fusion IV: Control Paradigms Data Structures*, vol. 1611, pp. 586–606, 1992.

[13] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994.

[14] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[15] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[16] D. Tzionas and J. Gall, "Reconstructing articulated rigged models from RGB-D videos," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 620–633.

[17] U. M. Nunes and Y. Demiris, "Online unsupervised learning of the 3D kinematic structure of arbitrary rigid bodies," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3809–3817.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Neural Inf. Process Syst.*, 2017, pp. 5105–5114.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. Roy. Stat. Society: Ser. B. (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[20] A. Myronenko and X. Song, "On the closed-form solution of the rotation matrix arising in computer vision problems," 2009, *arXiv:0904.1613*.

[21] I. D. Brian Beavis, *Optimization and Stability Theory for Economic Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[22] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang, "BAE-NET: Branched autoencoder for shape co-segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8490–8499.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[24] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

[25] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.