3D Motion Segmentation of Articulated Rigid Bodies based on RGB-D Data

Urbano Miguel Nunes
um.nunes@imperial.ac.uk
Yiannis Demiris
y.demiris@imperial.ac.uk

Personal Robotics Laboratory
Imperial College London
London, United Kingdom

Abstract

This paper addresses the problem of motion segmentation of articulated rigid bodies from a single-view RGB-D data sequence. Current methods either perform dense motion segmentation, and consequently are very computational demanding, or rely on sparse 2D feature points, which may not be sufficient to represent the entire scene. In this paper, we advocate the use of 3D semi-dense motion segmentation which also bridges some limitations of standard 2D methods (*e.g.* background removal). We cast the 3D motion segmentation problem into a subspace clustering problem, adding an adaptive spectral clustering that estimates the number of object rigid parts. The resultant method has few parameters to adjust, takes less time than the temporal length of the scene and requires no post-processing.

1 Introduction

The estimation of articulated objects' kinematic structure is an important research topic in computer vision and robotics. This is due to the fact that kinematic structures provide a compact representation regarding skeleton structure and motion about an object, which can be relevant for high level tasks, such as: human activity recognition [1], robotics manipulation [17] and kinematic structure correspondences learning [3]. To accomplish this, several methods have been proposed based on motion segmentation [5, 19] and combining skeleton information [2]. In particular, subspace clustering methods (*e.g.*, [4, 10]) have demonstrated state-of-the-art performance in the context of 2D motion segmentation tasks. In this work, we propose to investigate whether such methods could be extended for 3D motion segmentation of rigid bodies based on RGB-D input. For instance, 3D reconstruction became feasible in real-time due to RGB-D sensors, bridging some limitations of 2D camera sensors (*e.g*. background removal) [7, 12]. In this sense, we cast the problem of 3D motion segmentation of articulated rigid bodies as a subspace clustering problem. In particular, we adopted the sparse subspace clustering method [4], due to its empirical success and theoretical guarantees. Moreover, we propose the addition of self-tuning spectral clustering [21], in order to automatically estimate the number of segments, without relying on prior knowledge or other heuristics. Consequently, the proposed method is a complete pipeline that does not rely on prior object models and the number of constituent rigid body segments, having few parameters to adjust.

The rest of the paper is organized as follows: in Section 2, other motion segmentation methods based on 2D and 3D points are discussed; the proposed method is presented in Section 3, which relies on applying subspace clustering and self-tuning spectral clustering on captured 3D semi-dense point trajectories; in Section 4, quantitative and qualitative results obtained are shown; a final conclusion of our work is discussed in Section 5, as well as current limitations and topics of future research.

2 Related Work

Several motion segmentation and kinematic structure learning approaches have been proposed over the last years. Yan and Pollefeys [19] developed a factorisation-based approach that not only estimates rigid articulated segments, but can also recover non-rigid parts from RGB video sequences. It is capable of estimating the trajectories' rank and also the number of segments, through a recursive two-way spectral clustering. However, it is very dependent on the rank estimation, which is highly prone to noise. Ross *et al.* [16] presented motion segmentation and kinematic structure estimation as a probabilistic fitting model problem, associating a set of 2D feature point trajectories to a segment. Similarly, Sturm *et al.* [17] also proposed a probabilistic framework for kinematic structure learning, using a motion capture system for noise-free input data. Their method required the number of segments to be estimated and it was used in several robotics applications. Fayad *et al.* [5] proposed a simultaneous motion segmentation and 3D reconstruction approach, based on multiple model assignment corresponding to each body part, which was capable of dealing with outliers and complex structures. Ochs and Brox [14] proposed a variational optical-flow approach for point tracking, which was robust against occlusion and long term video analysis. Pairwise affinities between each trajectory were computed in order to perform spectral clustering and a spatial regularity energy minimization was proposed to automatically detecting the number of objects. Although they focus on 2D object segmentation, they provide insightful ideas for dealing with occlusion and long term video analysis. Following these ideas, Keuper *et al.* [8] formulated trajectory segmentation as a variant with minimum cost multicut, instead of using a spectral clustering method. A recent kinematic structure learning approach was proposed by Chang and Demiris [2], where they could estimate highly complex articulated structures and achieve state-of-the-art performance. However, they focused only on 2D image sequences, combining motion and skeleton information.

With the introduction of depth sensors, RGB-D data acquisition became feasible in real-time and some kinematic structure estimation approaches were proposed, taking this provided additional information into account. In [7, 12], an interactive segmentation and kinematic modelling method is proposed based on RGB-D data. The method could estimate the kinematic structure of the object, as well as its constraints (*i.e.* determine revolute or prismatic joints). However, it still relied on 2D features for point tracking, which is not suitable for large video sequences. Also, only objects with two or three body parts were analysed. More recently, motion segmentation from point cloud sequences has garnered attention. Zhang *et al.* [22] proposed a kinematic structure estimation method for complex articulated objects, based on point cloud sequences obtained by depth data. They rely on a two-step non-rigid matching between point clouds based on the Markov Random Field Deformation Model, where each consecutive frame is matched to the first frame, making it very computationally demanding. Although the authors claim this avoids the propagation of tracking errors, there are still significant error fluctuations in the measured segment length and a proper initial frame must be enforced. Yuan *et al.* [20] proposed a space-time co-segmentation method that propagates and merges all segmentation models that were estimated based on each individual frame. The authors propose the acquisition of point cloud sequences from depth data and achieved impressive results. However, they rely on a relatively high time interval between frames for tractability reasons, which makes the algorithm sensitive to large displacements. Lu *et al.* [11] also presented an unsupervised articulated structure estimation method based on point cloud sequences from a single depth camera. They rely on two distinct expectation-maximization optimizers (*i.e.* one for non-rigid point set registration and the other for structured joint estimation) and a constrained motion-based clustering for articulated structure generation. Although they achieve better overall performance compared to other methods, they rely on a selected initial point cloud for registration. Tzionas and Gall [18] presented an approach to reconstruct articulated models from RGB-D data. They focused on tracking points based on deformable mesh motion and then applied spectral clustering to the trajectories obtained. To the best of our knowledge, they have provided the only dataset suitable for evaluation of 3D motion segmentation of rigid bodies, to which we report our results.

3 Methodology

Given a set of complete point trajectories from sequential frames, the goal of this work is to estimate the corresponding 3D articulated kinematic structure. In contrast to previous subspace clustering based motion segmentation methods, where 2D sparse feature points are tracked [4], we consider the case of semi-dense 3D points provided directly by an RGB-D sensor. To this end, a sub-sampled point cloud is generated from raw RGB-D data and the respective point displacements between two consecutive frames are estimated based on the scene flow computed [6]. Then, given complete point trajectories, the task of motion segmentation consists of separating them according to their underlying motions, where it is assumed that a motion exists for each associated body part. Therefore, in this context, the problem of motion segmentation is cast as clustering data into a union of subspaces [4]. Also, the number of subspaces (*i.e.* body parts, segments) is not assumed to be given; thus, it is an unknown parameter to be estimated using self-tuning spectral clustering [21]. An overview of the algorithm is described in Algorithm 1.

Algorithm 1 3D Motion Segmentation of Rigid Bodies

```
Input: Set of 3D point trajectories $\mathbf{X}$.
Output: Number of segments $c$, set of points belonging to each segment $\mathbf{S}$ and respective center positions $\mathbf{M}$.
Solve sparse optimization program (2), with $\mathbf{Y} = \mathbf{X}$ and the affine constraint $\mathbf{1}^\top \mathbf{C} = \mathbf{1}^\top$.
Compute affinity matrix as in Eq. (3).
Compute symmetric normalized Laplacian matrix $\mathbf{L}$ as in Eq. (6).
Find $c$ and respective rotation matrix $\mathbf{R}$ by applying self-tuning spectral clustering to the eigenvectors $\mathbf{V}$ of $\mathbf{L}$.
Find $c$ segments $g$ by clustering rows of $\mathbf{VR}$ using K-means, with prior initialization from non-maximum suppression on the rows of $\mathbf{VR}$.
Compute center position $\mathbf{m}_g$ of each segment, averaging all points belonging to it.
```

3.1 Notations

The i^{th} point of frame f is defined as $\mathbf{x}_i^f \in \mathbb{R}^3$, where $i = 1, \ldots, N$, $f = 1, \ldots, F$. The g^{th} segment estimated $\mathbf{S}_g^f \in \mathbb{R}^{3 \times n_g}$ is composed of a subset of points $\mathbf{s}_g^f \in \mathbb{R}^{3 \times n_g}$ at frame f and the corresponding center position is $\mathbf{m}_g^f \in \mathbb{R}^3$, where $g = 2, \ldots, c$. N is the number of points considered in each frame, F is the number of frames of the sequence of images and n_g is the number of points belonging to segment g.

3.2 Initial formulation

Given a set of point trajectories, corresponding to an image sequence of a rigid articulated object, that are arranged such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^1 & \mathbf{x}_2^1 & \ldots & \mathbf{x}_N^1 \\ \mathbf{x}_1^2 & \mathbf{x}_2^2 & \ldots & \mathbf{x}_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^F & \mathbf{x}_2^F & \ldots & \mathbf{x}_N^F \end{bmatrix} \in \mathbb{R}^{3F \times N} \tag{1}$$

is the data matrix, the objective of motion segmentation is to separate each point trajectory according to their underlying motion. Similar to 2D motion segmentation [19], where the point trajectories of c rigid motions lie in a union of c low-dimensional subspaces of $\mathbb{R}^{2F}$ [4], 3D point trajectories also lie in a union of low-dimensional subspaces of $\mathbb{R}^{3F}$. Thus, the 3D rigid body part motion segmentation may be cast as a subspace clustering problem.

3.3 Subspace clustering based motion segmentation

Sparse subspace clustering [4] relies on the *self-expressiveness property* of the data to find a sparse representation, solving the convex optimization problem

$$\begin{aligned} &\min \|\mathbf{C}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda_z}{2} \|\mathbf{Z}\|_F^2 \\ &\text{s. t.} \quad \mathbf{Y} = \mathbf{Y}\mathbf{C} + \mathbf{E} + \mathbf{Z}, \quad \operatorname{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{2}$$

$\mathbf{C} \in \mathbb{R}^{N \times N}$ is the sparse representation matrix of the data matrix $\mathbf{Y} \in \mathbb{R}^{D \times N}$, $\mathbf{E} \in \mathbb{R}^{D \times N}$ represents sparse outlying entries, $\mathbf{Z} \in \mathbb{R}^{D \times N}$ represents occurring noise and $\operatorname{diag}(\mathbf{C}) \in \mathbb{R}^N$ corresponds to the vector of its diagonal elements. Once the optimization program is solved, a sparse representation of the data $\mathbf{C}$ is obtained. Then, a similarity graph with N nodes representing the data points is created, where the weights on the edges (*i.e.* affinity matrix) are computed as

$$\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^\top. \tag{3}$$

This is done to ensure that if a given data point y_i is written as a linear combination of other data points including y_j (*i.e.* $c_{ij} > 0$), both are connected even if the sparse representation y_j is not written as a linear combination that includes y_i (*i.e.* $c_{ji} = 0$). Ideally, the similarity matrix has c connected components associated to each one of the c subspaces, *i.e.*

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{W}_c \end{bmatrix}, \tag{4}$$

where $\mathbf{W}_l$ corresponds to the similarity matrix of data points belonging to subspace l.

In the context of the present work, $\mathbf{Y} = \mathbf{X} \in \mathbb{R}^{3F \times N}$ and the objective is to find a sparse representation $\mathbf{C}$, solving the optimization program (2) with an additional constraint that captures the affine structure of the motion segmentation problem, *i.e.* $\mathbf{1}^\top \mathbf{C} = \mathbf{1}^\top$. As suggested in [4], it was assumed that there are not sparse outlying entries (*i.e.* without the $\mathbf{E}$ term, since only complete point trajectories are considered). However, noisy point trajectories are expected (*e.g.* due to noisy sensor measurements and consequent innacuracies on scene flow estimation), which are handled by the $\mathbf{Z}$ term and

$$\lambda_z = 800/\mu_z, \quad \mu_z = \min_i \max_{j \neq i} |\mathbf{y}_i^\top \mathbf{y}_j|. \tag{5}$$

3.4 Self-tuning spectral clustering

Applying spectral clustering [13] to the similarity graph created follows, in order to cluster the data into subspaces. Firstly, a symmetric normalized Laplacian matrix of the graph $\mathbf{L} \in \mathbb{R}^{N \times N}$ is computed, *i.e.*

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}, \tag{6}$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with its elements given by

$$d_{ii} = \sum_{j=1}^{N} w_{ij}. \tag{7}$$

Then, c eigenvectors of $\mathbf{L}$ are found corresponding to the associated c largest eigenvalues, forming the matrix $\mathbf{V} \in \mathbb{R}^{N \times c}$ by stacking the eigenvectors column-wise. Lastly, K-means may be employed treating each row of $\mathbf{V}$ as a point, in order to cluster them into c clusters.[1]

However, it is not assumed that the number of clusters c is given as input. A first attempt to estimate c could consist of performing eigenvalue decomposition of $\mathbf{L}$ and counting the number of eigenvalues above a given threshold, which would determine the number of clusters to segment. Although intuitive, this approach introduces an additional parameter to tune, which may be dependent on the initial data matrix. Therefore, determining the number c based on the eigenvalues may be infeasible for general applications.

Another approach was introduced in [21], where a self-tuning spectral clustering algorithm is proposed based on eigenvector analysis, instead of eigenvalues. The main idea is to find a rotation $\mathbf{R} \in \mathbb{R}^{c \times c}$ which best aligns the stacked eigenvector matrix $\mathbf{V}$ with the canonical coordinate system, while computing its cost for every $c = 2, \ldots, N$. The number c^* which minimizes the cost is selected as the number of clusters to segment. The algorithm follows by performing non-maximum suppression on the rows of $\mathbf{V}\mathbf{R}^*$, where $\mathbf{R}^* \in \mathbb{R}^{c^* \times c^*}$ is the rotation matrix that minimized the cost of alignment.

In the context of this work, the self-tuning spectral clustering algorithm [21] was adopted to estimate the number of segments c of the kinematic structure. For computational efficiency, the upper bound on range values of c was heuristically constrained to $c_{\text{upper}} = \lceil \log_2 N \rceil$. Also, the clusters obtained from non-maximum suppression on the rows of $\mathbf{V}\mathbf{R}^*$ were used to initialize K-means algorithm, as suggested in [21] to accommodate for highly noisy data. In the end, a subset of points $\mathbf{S}_g$ belonging to each segment g is obtained, where

[1] Note that the Laplacian could also be computed as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$, as it appears more frequently in spectral graph theory. The only change between these two formulations is concerned with the respective eigenvalues, *i.e.* from λ_i to $1 - \lambda_i$.

$N = n_1 + \cdots + n_c$. The respective center positions $\mathbf{m}_g^f$ are obtained by averaging all points belonging to the associated segment, for each frame.

4 Experiments and Results

The proposed method was evaluated on the dataset provided by [18], which consists of five RGB-D videos of different articulated objects: "donkey", "lamp", "pipe 1/2", "pipe 3/4" and "spray". Since the method requires complete point trajectories, only points that are tracked throughout the whole sequence are included in the set of point trajectories, as in Eq. (1).[2] All experiments were performed using a PC with an Intel Core i7-8700k CPU @ 3.7Ghz (x6) and 32GB of RAM.

4.1 Quantitative Results

Five metrics were evaluated, namely: execution time, number of segments obtained, precision, recall and f-measure. As described in [14], the last three metrics for each pair of segment estimated $\mathbf{S}_i$ and ground-truth segment $\mathbf{S}_j^{\mathrm{GT}}$ are defined as follows, respectively:

$$P_{ij} = \frac{|\mathbf{S}_i \cap \mathbf{S}_j^{\mathrm{GT}}|}{|\mathbf{S}_i|} \quad R_{ij} = \frac{|\mathbf{S}_i \cap \mathbf{S}_j^{\mathrm{GT}}|}{|\mathbf{S}_j^{\mathrm{GT}}|} \quad F_{ij} = \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}}. \tag{8}$$

These metrics try to capture a trade-off between measuring false positives and misses. The Hungarian method [9] is applied, in order to find the best allocation of segments to ground-truth segments. Empty segments are introduced in the case where there are fewer segments estimated than ground-truth segments (*i.e.* recall is zero and precision is defined to be one).

The proposed method is deterministic and only a few parameters require adjustment, namely: parameter λ_z (*i.e.* parameter that balances the noise term in Eq. (2)), the upper bound on range values of number of segments to be estimated c_{upper} and the number of initial sub-sampled points to be tracked N_{init}. Parameter λ_z is studied in [4] in the context of 2D motion segmentation. As shown in [4], the resultant clustering error is not significantly affected for a large range of values; thus, we followed the suggested value, as presented in Eq. (5). The upper bound on range values for estimating the number of segments c_{upper} only affects the size of the search space for the optimal value c^*. In this sense, the proposed search constraint (*i.e.* $c_{\text{upper}} = \lceil \log_2 N \rceil$) is merely suggestive, as the only observable effect was the computational time. This means that, depending on the knowledge of the problem at hand, one could be more conservative by setting the lowest upper bound possible (*e.g.* if a given body is not expected to have more than 10 segments, one could set $c_{\text{upper}} = 10$). Nevertheless, the point to be made is that the method could be applied to other problem, even without changing the upper bound. Therefore, only the effect of the number of initial sub-sampled points N_{init} will be studied. As the proposed method is deterministic, for equal initial conditions, equal results are obtained. Thus, in order to test the robustness and reliability of the method, a randomly selected subset of points is sub-sampled, where its size is determined by N_{init}.

[2] In [4], two approaches are described to deal with missing entries in the data matrix (*i.e.* incomplete point trajectories). However, both only work properly when a small fraction of entries is missing, which may not occur in the context of the present work, mainly due to sensor measurement errors.

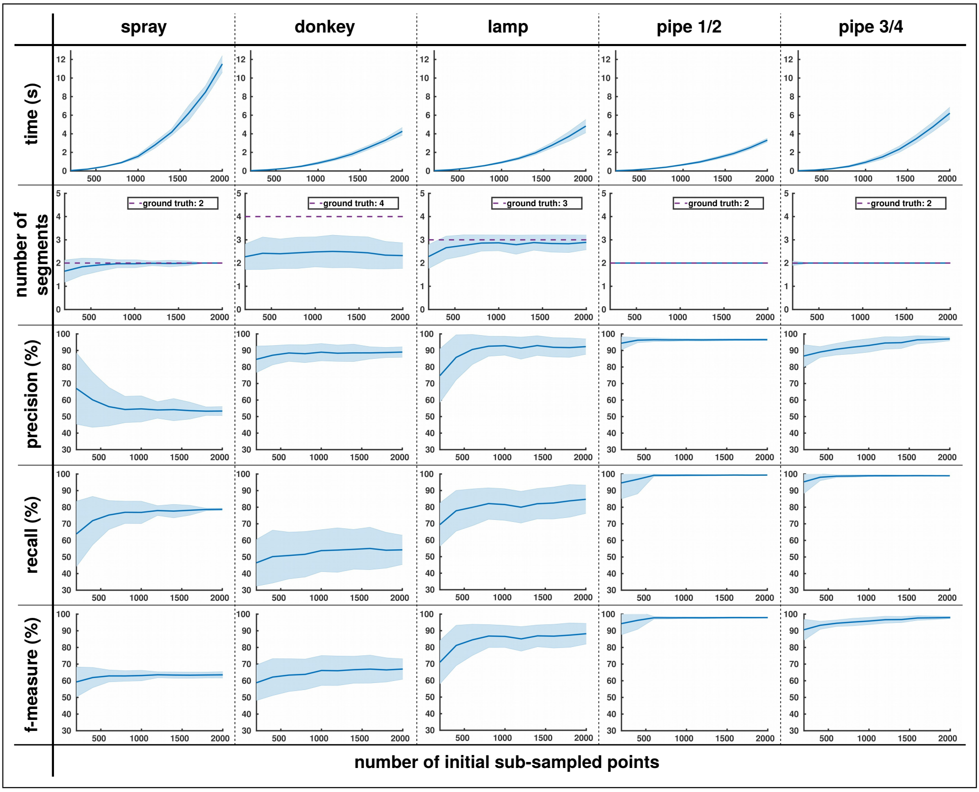

Figure 1: Quantitative results concerning five metrics: time (first row), number of segments estimated (second row), precision, recall and f-measure (third, fourth and fifth rows, respectively). These results were obtained by varying the number of initial randomly sub-sampled points across one hundred trials per number. The mean is represented by the solid line and the standard deviation is represented by the surrounding shaded area.

		Donkey	Lamp	Pipe 1/2	Pipe 3/4	Spray	Average
SSC [4]	Time per point (msec)	1.03	1.14	0.80	1.32	2.39	1.34
	# of segments	2.41/4	2.77/3	2/2	2.00/2	1.93/2	-
	Precision	88.03%	89.68%	96.21%	93.07%	56.04%	84.61%
	Recall	52.53%	80.34%	98.58%	98.46%	75.59%	81.10%
	F-Measure	65.80%	84.75%	97.38%	95.69%	64.37%	82.82%
LRR [10]	# of segments	2.80/4	1.54/3	1.49/2	1.27/2	1.69/2	-
	Precision	60.33%	67.48%	67.13%	70.77%	75.97%	68.34%
	Recall	43.06%	17.32%	53.06%	50.44%	56.16%	44.01%
	F-Measure	50.25%	27.56%	59.27%	58.90%	64.58%	52.11%

Table 1: Summary of average results obtained.

	Donkey	Lamp	Pipe 1/2	Pipe 3/4	Spray	Average
Time per point (msec)	1.14	1.03	1.00	1.32	2.36	1.37

Table 2: Summary of average processing time per point obtained considering $2F$ frames.

Figure 1 shows the quantitative results obtained over one hundred trials for a range of values of N_{init}. The following conclusions can be made:

- *Computational time*: as the number of sub-sampled points increases, so does the execution time of the proposed method, as expected.

- *Number of segments estimated*: besides the case of the "donkey" object, the number of segments estimated converges to the ground-truth for all other objects. Also, the associated standard deviation decreases as the number of initial sub-sampled points increases, suggesting that tracking more points improves the reliability of the method.

- *Precision, recall and f-measure*: besides the "spray" object, increasing of the number of sub-sampled points exhibit slight improvements, both in terms of higher expected values and lower associated variability, for the three metrics.

- There should be a compromise w.r.t. the number of point trajectories considered. The results obtained suggest that 1000 initial points considered might be a good compromise, since no significant improvements are observed for higher values, whilst the computational time still increases. Further discussion about the different results obtained for "donkey" and "spray" objects, concerning number of segments estimated and precision, respectively, will be done considering the qualitative results as well.

In Table 1, a summary of the quantitative results is provided, using sparse subspace clustering (SSC) [4] and low rank representation (LRR) [10] methods to build the affinity matrix $\mathbf{W}$. As shown, higher performance is obtained when the affinity matrix is computed based on SSC, which also corroborates the choice of using it. The computation time for LRR is not provided, since the respective author's code in MATLAB was used, whereas a version of SSC was implemented in C++.[3] It is interesting to note that if 1000 points were tracked the average computation time would be 1.34 seconds. Considering that every video takes more than 3 seconds, this means that the actual computation is faster than the video sequence, suggesting that a real-time online approach relying on this method would be feasible. This topic will be considered in future research.

The influence of the number of frames to be processed on the overall computational time was also evaluated and the results are reported in Table 2, where the length of each sequence

[3] Code available: https://github.com/ImperialCollegeLondon/3DKSL.

was doubled, *i.e*. each sequence was considered forward and then backward. As presented, the average processing time per point is identical, comparing both the original sequence length and the doubled one (*i.e*. comparing each sequence with F and $2F$ frames, respectively). This suggests that the number of frames to be processed does not affect significantly the overall computational cost of the proposed method. Indeed this is true, since in practise we are solving the optimization program (2) via an Alternating Direction Method of Multipliers (ADMM) and we are only considering complete point trajectories for each sequence (*i.e*. without $\mathbf{E}$ term). For a more detailed discussion, please refer to the supplementary material provided.

4.2 Qualitative Results

In Fig. 2, qualitative results are shown (bottom row), as well as the respective objects and ground-truth segmentation. As shown by the quantitative results, for the "lamp", "pipe 1/2" and "pipe 3/4" objects, the method achieves an average f-measure performance above 84%.[4] However the results obtained are lower in the case of the "donkey" and "spray" objects. Concerning the "donkey" object, it is observable that a significant portion of the segmented body is not filled with points. This is due to the fact that severe occlusions occur during the video, in particular between the tip of the arms and the body, and between the head and the body. This means that a significant number of point trajectories are discarded, which could be important for better overall performance. In the case of "spray" object, the associated motion might not exhibit sufficient evidence for a correct motion segmentation (as may be observed in the lower right corner image in Fig. 2). This means that the proposed method is not able to accurately segment subtle motions, which may occur in real-life scenarios, either due to noisy sensor measurements or errors during point tracking. This observation is corroborated by the fact that the precision results obtained for "spray" object (third row of first column, Fig. 1) decrease for higher numbers of initial sub-sampled points, since the ratio of correctly segmented points decreases. These two issues will also motivate further research.

5 Conclusion and Future Work

A complete method for 3D motion segmentation of articulated rigid bodies based on RGB-D data was proposed in this paper. Given a set of complete point trajectories, the method estimates automatically 3D motion segments by solving a sparse optimization program and then applying adaptive spectral clustering to the resultant affinity matrix, in order for the number of body segments to be estimated. The method is deterministic and achieves lower computational time than the duration of the image sequence, depending on the number of points considered. The method was evaluated on a public dataset, achieving consistent results. It has few parameters to adjust, making it suitable for automatic motion segmentation of rigid bodies in a wide range of scenarios. Such scenarios may include visual self-exploratory learning for lifelong autobiographic memories [15]. In this case, it is reasonable to assume that each learning sequence has a relative short duration and severe occlusion does not occur, whereas computational processing and time may be crucial performance indicators. Nevertheless, evaluating the proposed method in more scenarios will be the focus of future research, as

[4] The method could also segment body parts that did not belong to the object, *i.e*. "left hand" in the "pipe 3/4" object sequence. Nevertheless, only the motion segmentation of the actual object was evaluated.

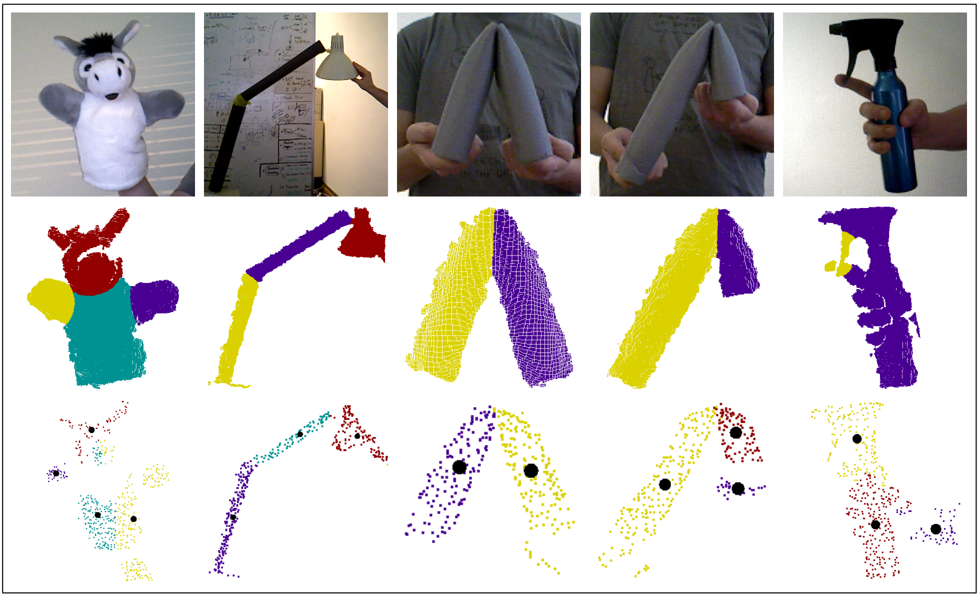

Figure 2: Qualitative results obtained with the same parameters (*i.e.* λ_z as given by Eq. (5), $c_{\text{upper}} = \lceil \log_2 N \rceil$, $N_{\text{init}} = 1000$). The first row of images illustrates the articulated object, from which the motion segments are to be estimated. The second row exemplifies the motion segmentation ground truth, where each colour represents one segment. The third row shows the corresponding obtained motion segments; the black dots correspond to the center coordinates of each segment. Figure best viewed in colour.

well as tackling some of its current limitations, since the proposed method is not capable of handling occlusions or segments entering/exiting the scene.

Acknowledgements

Urbano Miguel Nunes was supported by the Portuguese Foundation for Science and Technology (FCT) under the Doctoral Grant with reference SFRH/BD/130732/2017.

References

[1] Jake K. Aggarwal and Lu Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, October 2014.

[2] Hyung Jin Chang and Yiannis Demiris. Highly articulated kinematic structure estimation combining motion and skeleton information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, in press.

[3] Hyung Jin Chang, Tobias Fischer, Maxime Petit, Martina Zambelli, and Yiannis Demiris. Learning kinematic structure correspondences using multi-order similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, in press.

[4] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and

applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781, November 2013.

[5] João Fayad, Chris Russell, and Lourdes Agapito. Automated articulated structure and 3d shape recovery from point correspondences. In *IEEE International Conference on Computer Vision (ICCV)*, pages 431–438. IEEE, November 2011.

[6] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 98–104. IEEE, May 2015.

[7] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5003–5010. IEEE, May 2013.

[8] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3271–3279. IEEE, 2015.

[9] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

[10] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, January 2013.

[11] Xuequan Lu, Honghua Chen, Sai-Kit Yeung, Zhigang Deng, and Wenzhi Chen. Unsupervised articulated skeleton extraction from point set sequences captured by a single depth camera. In *The Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, 2018, in press.

[12] Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5091–5097. IEEE, May 2016.

[13] Andrew Y. Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[14] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, June 2014.

[15] Maxime Petit, Tobias Fischer, and Yiannis Demiris. Lifelong augmentation of multimodal streaming autobiographical memories. *IEEE Transactions on Cognitive and Developmental Systems*, 8(3):201–213, September 2016.

[16] David A. Ross, Daniel Tarlow, and Richard S. Zemel. Learning articulated structure and motion. *International Journal of Computer Vision*, 88(2):214–237, June 2010.

[17] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, August 2011.

[18] Dimitrios Tzionas and Juergen Gall. Reconstructing articulated rigged models from rgb-d videos. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 620–633. Springer International Publishing, 2016.

[19] Jingyu Yan and Marc Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):865–877, May 2008.

[20] Qing Yuan, Guiqing Li, Kai Xu, Xudong Chen, and Hui Huang. Space-time co-segmentation of articulated point cloud sequences. In *Computer Graphics Forum*, volume 35, pages 419–429. Wiley Online Library, 2016.

[21] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005.

[22] Quanshi Zhang, Xuan Song, Xiaowei Shao, Ryosuke Shibasaki, and Huijing Zhao. Unsupervised skeleton extraction and motion capture from 3d deformable matching. *Neurocomputing*, 100:170–182, January 2013.