UNIVERSITY OF BIRMINGHAM

University of Birmingham
Research at Birmingham

Highly articulated kinematic structure estimation combining motion and skeleton information

Chang, Hyung Jin; Demiris, Yiannis

DOI:
10.1109/TPAMI.2017.2748579

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Chang, HJ & Demiris, Y 2018, 'Highly articulated kinematic structure estimation combining motion and skeleton information', *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 40, no. 9, pp. 2165-2179. https://doi.org/10.1109/TPAMI.2017.2748579

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

H. J. Chang and Y. Demiris, "Highly Articulated Kinematic Structure Estimation Combining Motion and Skeleton Information," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 9, pp. 2165-2179, 1 Sept. 2018, doi: 10.1109/TPAMI.2017.2748579.

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of ‘fair dealing’ under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 24. Apr. 2024

Preprint version; final version available at http://ieeexplore.ieee.org/document/8025409/
IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
DOI: https://doi.org/10.1109/TPAMI.2017.2748579

Highly Articulated Kinematic Structure Estimation combining Motion and Skeleton Information

Hyung Jin Chang, *Member, IEEE,* and Yiannis Demiris, *Senior Member, IEEE*

Abstract—In this paper, we present a novel framework for unsupervised kinematic structure learning of complex articulated objects from a single-view 2D image sequence. In contrast to prior motion-based methods, which estimate relatively simple articulations, our method can generate arbitrarily complex kinematic structures with skeletal topology via a successive iterative merging strategy. The iterative merge process is guided by a density weighted skeleton map which is generated from a novel object boundary generation method from sparse 2D feature points. Our main contributions can be summarised as follows: (i) An unsupervised complex articulated kinematic structure estimation method that combines motion segments with skeleton information. (ii) An iterative fine-to-coarse merging strategy for adaptive motion segmentation and structural topology embedding. (iii) A skeleton estimation method based on a novel silhouette boundary generation from sparse feature points using an adaptive model selection method. (iv) A new highly articulated object dataset with ground truth annotation. We have verified the effectiveness of our proposed method in terms of computational time and estimation accuracy through rigorous experiments with multiple datasets. Our experiments show that the proposed method outperforms state-of-the-art methods both quantitatively and qualitatively.

Index Terms—Highly articulated kinematic structure estimation, adaptive motion segmentation, density weighted silhouette generation from sparse points, adaptive kernel selection.

1 Introduction

Learning the underlying kinematic structure of articulated objects is an active research topic in computer vision and robotics. Kinematic structures contain skeleton information, and also provide motion related information between body parts. This information is beneficial to many higher level tasks such as human action recognition [2], body scheme learning for robotic manipulators [3], articulated objects kinematics recognition and manipulation [4], and finding kinematic correspondences between articulated objects [5]. In this paper, we focus on building the articulated kinematic structure from data, in particular, RGB image sequences with interest points tracked over time. Using images from a monocular RGB camera is advantageous given the proliferation of such cameras in various applications, such as smartphones, webcams, robot eyes, and microscope cameras.

Many algorithms which recover an articulated structure from trajectories of 2D feature points have been used for the automated detection of articulated 3D motion types (*i.e.* folding, rotation, and translation) of relatively simple articulations [6], [7]. They have also been used to build kinematic chains in order to 3D reconstruct articulated objects [6], [8]. These articulated structures derived from motion algorithms are predominantly designed for 3D kinematic structure estimation. In order to solve depth ambiguity problems, neighbouring segments are enforced to be overlapping [8], and joints are located at the intersection of two motion subspaces. Our main target in this work is to find an accurate kinematic structure of arbitrary objects with articulated motion capabilities that range from simple structures to highly complex structures without relying on 3D reconstruction.

Most of the existing kinematic structure estimation methods [6], [8], [9], [10] only exploit motion information. Such methods miss global refinement steps that incorporate topological or kinematic constraints. As shown in [1], considering only motion similarities for structuring body parts results in topologically distant but similarly moving parts being connected, and as such can produce highly implausible structures as output. On the other hand, articulated structure estimation methods which employ shape information [11], [12] have been presented. In this case, the estimated structure is a static skeleton which represents the medial axis of a body and holds topological properties; however such estimation methods cannot represent kinematic properties.

In this paper, we present a novel framework for complex articulated kinematic structure estimation from 2D feature points trajectories extracted from RGB image sequences. We combine motion (a temporal property) and skeleton (a spatial property) information in order to generate a kinematic correlation and topology embedded structure (see Figure 1). We assume that an articulated object is composed of a set of rigid segments, where the structure represents the connections between segments. It is difficult to estimate the number of body parts of the structure in advance, especially when the articulation is complex and the input data is noisy. Thus we introduce a fine-to-coarse strategy which performs iterative merging of over-segmented parts, as guided by the skeletal topology and motion similarity. For the generation of the skeleton distance function from sparse feature points, we present a novel object boundary generation method. As a result, our method does not require any prior knowledge about the object,

- *Both authors are with the Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom, SW7 2AZ.*
E-mail: {*hj.chang, y.demiris*}*@imperial.ac.uk*
- *Research presented in this paper is a continuation of Chang and Demiris [1] and includes results from [1].*

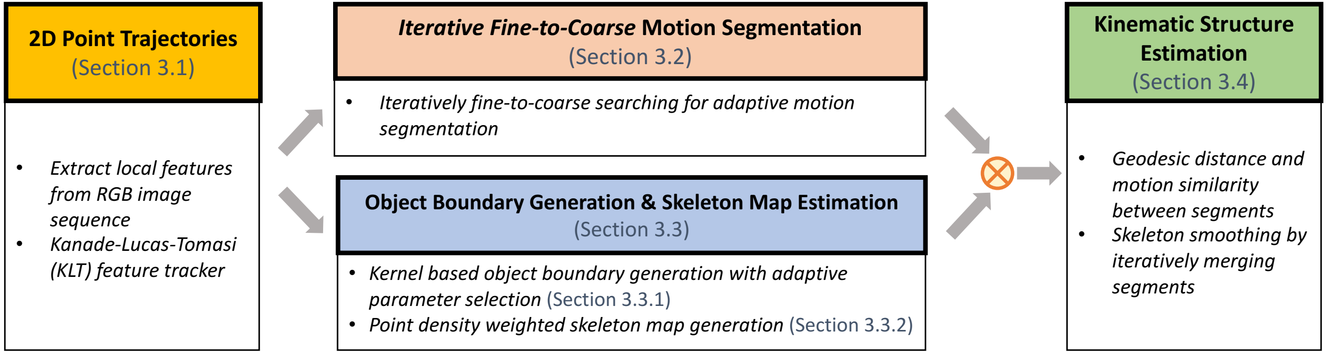

2D Point Trajectories
(Section 3.1)
Extract local features from RGB image sequence
Kanade-Lucas-Tomasi (KLT) feature tracker
Iterative Fine-to-Coarse Motion Segmentation
(Section 3.2)
Iteratively fine-to-coarse searching for adaptive motion segmentation
Object Boundary Generation & Skeleton Map Estimation
(Section 3.3)
Kernel based object boundary generation with adaptive parameter selection (Section 3.3.1)
Point density weighted skeleton map generation (Section 3.3.2)
Kinematic Structure Estimation
(Section 3.4)
Geodesic distance and motion similarity between segments
Skeleton smoothing by iteratively merging segments

Fig. 1. The proposed framework reliably learns the underlying kinematic structure of highly articulated objects without any object model required. From an image sequence, we extract dense 2D feature point trajectories. From the trajectories, we adaptively learn motion segments and the skeleton distance map in parallel. The rigid body parts are segmented out intentionally via an iterative fine-to-coarse motion segmentation because we assume that we have no prior knowledge about the number of segments. The skeleton distance map is generated based on an adaptively estimated object silhouette with a new optimal parameter selection in order to relate the object shape to the skeleton information. We finally estimate an accurate kinematic structure implying both kinematic correlation and skeletal topology followed by a structure refining step which merges the over-segmented parts guided by the skeleton distance map.

such as an object category or the number of constituent rigid parts.Our experiments show that the proposed method outperforms state-of-the-art methods quantitatively and qualitatively.

In Section 2, we discuss other approaches for estimating articulated structures from tracked feature points, as well as for generating object boundaries. In Section 3, we propose a framework for estimating the structure from data. The learning proceeds with estimation of the motion segments and density weighted skeleton distance map performed in parallel. Then, the final structure is found utilising the two intermediate estimation results. In Section 4, we present a new dataset and compare the algorithm with other methods in various aspects. In the final Section 5, we conclude our work by introducing limitations to this approach and areas for future work.

2 Related Work

In line with the three main contributions of this work, we group the literature into a) motion segmentation and kinematic structure building methods, b) object silhouette generation approaches, and c) parameter selection methods for the object shape generation.

Motion segmentation and kinematic structure building: Many motion segmentation approaches [13], [14], [15] have typically used a factorisation method, as proposed by [16]. The factorisation based agglomerative lossy compression (ALC) [17] can even deal with incomplete trajectories, however, the common drawback of factorisation based approaches is their vulnerability to noisy data. Subspace fitting approaches based on a generalised principal component analysis (GPCA) [18] or sparse subspace clustering (SSC) [19] have been widely used for motion segmentation, but they require the exact number of motion segments as input. They also cannot be applied to more than a few subspaces as the number of required samples grows exponentially with the number of subspaces. Jung *et al.* [20] proposed a novel rigid motion segmentation algorithm based on randomised voting (RV). This algorithm showed that state-of-the-art motion segmentation performance could be achieved even under noisy environments. However, it also required an exact number of motion clusters as a prior for good performance, which makes it difficult to be applied to complex articulated sequences.

Tresadern and Reid [21] and Yan and Pollefeys [22] developed the factorisation method [16] based kinematic chain estimation methods for articulated objects. It is effective to segment dependent motions for simple articulations, but cannot deal with high degrees of articulations because the factorization is generally sensitive to non-Gaussian noise, such that only a few tracking errors can spoil the result [23]. Furthermore, Yan and Pollefeys [6] estimated a kinematic chain by modelling the articulated motion as a set of intersecting motion subspaces. The locations of the joints are obtained from the intersections of the motion subspaces of connected parts. This algorithm is highly dependent on the correct detection of the rank of the trajectories and is thus sensitive to noise. There are also many critical tuning parameters in each step, such as the rank estimation parameter, the local sampling size, and the highest dimension size. Overall, this method is very difficult to apply in realistic scenarios. Jacquet *et al.* [7] presented a relative transformation analysis method based on linear subspaces, which focused on detecting the type of articulated 3D motion between two restricted motion parts.

Ross *et al.* [24], [25] proposed probabilistic graphical model approaches to learning the articulated structure from 2D feature tracks. They found the number of joints and their connections adaptively, but their method was sensitive to the prior and had difficulty in recovering from poor initial segmentation. In addition, it had difficulties escaping from local minima. Sturm *et al.* [3], [26] similarly used a probabilistic approach to learning kinematic joints tailored for robot vision applications, body schema learning, and object manipulation. They required fiducial markers on each object part for noise-free input data, and the number of motions had to be given as a prior. A markerless sparse feature tracking-based articulation learning was presented by Pillai *et al.* [27], which did not require prior object models. However, they required RGB-D feature data and could not handle concurrent articulated motions.

An energy based multiple model fitting approach for simultaneous segmentation and 3D reconstruction was proposed by Fayad *et al.* [8]. The benefits of this approach are that neither assumptions about the skeleton structure of the object, nor the number of motion segments are required in advance. They decomposed a set of point tracks into overlapping rigid-bodies and the structure joints are derived from the regions of the overlap. They showed impressive 3D articulated shape reconstruction performances even for complex structures. However, this method focuses more on the full 3D reconstruction of articulated objects requiring overlaps in

Preprint version; final version available at http://ieeexplore.ieee.org/document/8025409/
IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
DOI: https://doi.org/10.1109/TPAMI.2017.2748579

order to resolve the per-rigid-segment depth scale ambiguity in 3D reconstruction.

Our method is focused on building a kinematic structure framework that implies both motion and skeletal information, without 3D reconstruction compared to the earlier mentioned methods [6], [8], [21] which reconstruct 3D shapes as well as kinematic structures.

Object shape boundary generation: There have been various approaches in the prediction of accurate moving object shape boundaries from sequential frames. These approaches can be categorised as motion and colour based, or learning based.

First of all, object boundary detection methods [28] using image processing techniques such as gradient-based contour detection have been applied to each frame to generate the object boundary. However, the resulting boundaries are very noisy and internal textures cause incorrect boundary shapes. Such algorithms also have difficulties in distinguishing between background regions and the object. Furthermore, many methods (*e.g.* [29], [30]) for the boundary estimation of moving objects are based on optical flow [31]. Spoerri [29] has shown that local flow histograms are generally bimodal distributions at motion boundaries, and Black and Fleet [30] improved the accuracy by adopting a probabilistic framework. All optical flow based methods are highly dependent on the optical flow performance, but even using the state-of-the-art optical flow method [32] as input can lead to suboptimal results. There also have been approaches combining motion information with colour information. The object motion boundary estimation is considered as segmenting a video frame into different regions with coherent motion, referred to as layers [33]. Unger *et al.* [34] performed joint estimation of motion layers and optical flow simultaneously, but the joint estimation depends on a complex minimisation of non-convex energy functions, which is unreliable for difficult cases such as fast motion or large displacements. Moreover, the motion layer segmentation often becomes ill-defined when motion boundaries form non-closed regions. Furthermore, Ochs and Brox [35] presented a variational method that fills the gaps between the feature trajectories based on colour and texture, with various works following this approach [36], [37], [38]. In particular, a public benchmark dataset with standardised evaluations was presented [38] and we report our method's performance on this dataset.

Recent methods cast the object boundary detection task into a learning framework relying on a random forest classifier [39]. Dollar and Zitnick [39] formulate the edge detection problem of predicting local edge masks, such as straight lines or T-junctions, in a structured learning framework applied to random decision forests, achieving state-of-the-art performance. Weinzaepfel *et al.* [40] leverage the patch level information, which increases robustness to failures in the optical flow using an estimated flow error. However, they focus on detecting boundaries of moving parts in a short time duration, so most of the time the detected boundaries do not cover the whole objects' shape.

In this work, we propose a new approach for the object boundary generation using sparse feature points. It is formulated as a convex optimisation problem with a kernel trick which finds an arbitrarily shaped boundary enclosing all local feature points. The kernel function makes the boundary flexible, but the kernel parameter significantly affects the resulting shape. Thus, finding a proper parameter becomes an important issue for the adaptive object boundary generation.

Optimal parameter selection methods: To the best of our knowledge, our approach for object boundary estimation from sparse feature points is novel, so the definition of an accurate boundary would be a fundamental issue for selecting an optimal kernel parameter. The boundary can be seen as a separation (*i.e.* classifying) between the inner and outer regions of an object. This interpretation enables us to validate estimated boundaries numerically and to design a new kernel parameter selection. In general, a preferred boundary of a classifier is considered as a discriminating boundary which achieves improved generalisation performance [41]. There have been several approaches to solving the kernel parameter selection problem to achieve high generalisation performance. However, theoretical analysis based approaches [42] give bounds that are too loose, and a heuristic approach based on genetic algorithms [43] requires high computational time. The difficulties in applying artificial outlier generation methods [44], [45] are the generation of well-balanced outliers and the high computational complexity for the validation of the estimated boundary. Kim [46] presented a simple criterion calculating data skewness. However, the method was only verified against very simple toy dataset.

3 Methodology

Our goal is to generate an articulated kinematic structure considering both motion and skeleton information, whilst being accurate and sophisticated enough to handle complicated concurrent motions. To this end, we use only 2D feature points trajectories assuming that one target subject exists in the scene, and we extract features from every part of the object.

The overall concept of the proposed framework is illustrated in Figure 1. In Section 3.1 we define the notations, followed by Section 3.2 which discusses how the adaptive motion segmentation is performed. In Section 3.3 we discuss the generation of the object boundary and the density weighted skeleton distance map from sparse feature points. Finally, in Section 3.4, a kinematic structure generation and refining algorithm using the processed skeleton and motion information is presented.

3.1 Notations

The full 2D feature point trajectories set X is defined as $X = \{X_1, X_2, \ldots, X_N\}$ where X_i represents i^{th} point's trajectory set among N points in homogeneous coordinates. The trajectory set X_i is composed of sequential 2D points x_i^f as $X_i = \{x_i^f | f = 1, ..., F\}$, with f as sequential frame index and F as the total number of frames in the input video. To express motion segments, we use S_g for the disjoint set of points belonging to the g^{th} segment where $g = 1, ..., c$, and c as the total number of segments. y_g^f denotes the centre position of segment S_g obtained by averaging its points at frame f. We express an object region at frame f by Ω^f and its boundary as $\delta\Omega^f$. The terms *object shape boundary* and *silhouette* are used interchangeably.

3.2 Rigid Part Segmentation by Motion Information

The articulated object is composed of a set of rigid body parts, and the rigid parts have been segmented out using local features' motion information [6], [8], [23], [24]. However, it is difficult to estimate the precise number of motion segments while performing motion segmentation, especially when the motions are highly articulated and the input data are noisy. In order to cope with these complicated conditions, we present an 'iterative fine-to-coarse' inference strategy which adaptively estimates an upper-bound number of initial motion segmentation. We use the randomised voting (RV) method [20] as a fundamental motion segmentation method which is relatively fast and robust to noise but requires the number of segments c as input.

3.2.1 Randomised Voting based Motion Segmentation

The RV motion segmentation algorithm [20] performs motion segmentation from 2D feature point trajectories based on the epipolar geometry, which is represented by a fundamental matrix $\mathbf{F}$. The matrix $\mathbf{F}_g$ indicates the motion of body part g, as it encapsulates the intrinsic geometry of the g^{th} body part. The RV motion segmentation is based on two important properties: 1) points from one rigid part share the same matrix $\mathbf{F}$, and 2) a set of points of the same moving part in one frame lies on the corresponding epipolar line in the other frame. Therefore, if the distances between the epipolar line and the points are similar, the points are likely to belong to the same group and vice-versa. Specifically, $\mathbf{F}_g^{(k,l)}$ represents a fundamental matrix estimated by the k^{th} and l^{th} frames of segment g. The Sampson distance ($\mathbf{SD}$) is used to measure the distance d_i between point x_i and the epipolar line as follows:

$$\begin{aligned} d_i^{(k,l)}[g] &= \mathbf{SD}(x_i^k, x_i^l, \mathbf{F}_g^{(k,l)}) \\ &= \frac{{x_i^k}^T \mathbf{F}_g^{(k,l)} x_i^l}{(\mathbf{F}_g^{(k,l)} x_i^k)_1^2 + (\mathbf{F}_g^{(k,l)} x_i^k)_2^2 + (\mathbf{F}_g^{(k,l)} x_i^l)_1^2 + (\mathbf{F}_g^{(k,l)} x_i^l)_2^2}, \end{aligned} \tag{1}$$

where $(\mathbf{F}x)_j^2$ is the square of the j-th entry of vector $\mathbf{F}x$. If the point x_i belongs to segment g and the matrix $\mathbf{F}_g$ is known precisely, the sum of the distances d_i over all frames is approximately zero. Therefore, the motion segmentation label can be found by minimising the function

$$\hat{g}(i) = \arg\min_g \sum_{k=1}^{F} \sum_{l=1}^{F} d_i^{(k,l)}[g], \tag{2}$$

where $\hat{g}(i)$ is an estimated group index of the point i with $\mathbf{F}_g$. Thus, if the fundamental matrices for all the motions are known, the points can be easily segmented. In reality however, both the labels and the fundamental matrices are unknown.

To resolve this chicken and egg problem, an iterative method is used. Firstly, the labels of points are randomly initialised. Based on the initialised labels, the fundamental matrices are estimated and scores for each point are voted (hence the name "randomised voting"). Then, each point is relabelled using the score and the fundamental matrices are updated with the labels. This process is iteratively performed to eventually find the solution. For each iteration, m points, frame k and frame l are randomly selected, and the fundamental matrix is estimated using the selected points by a normalized DLT [47]. Scores for the points are voted for based on the distance computed using Eq.(1). In general, two images are required to estimate a fundamental matrix. Neighbouring frames can be the two views images, but the objects' movements may be too subtle leading to an inaccurate estimate. To avoid this problem, all combinations of views are utilised, and this covers many more movement cases. However, dealing with all cases is difficult as the frame number F increases. Thus, a subset of the combinations of views is utilised by randomly selecting two frames from all frames and updating the estimation iteratively. This method can not work with incomplete feature point trajectories and online learning is not feasible.

For computing the score of point i, the distance $d_i^{(k,l)}[g]$ is calculated. Then the point i is voted by $e^{-\kappa d_i}$, where the parameter λ controls the voting strength by considering $\hat{g}$:

$$h_i[g] = \begin{cases} h_i[g] + e^{-\kappa d_i[g]}, & \hat{g} = g \\ h_i[\hat{g}] - e^{-\kappa d_i[g]} + 1, & \forall \hat{g} \neq g \end{cases}, \tag{3}$$

where $h_i[g]$ is a histogram storing the voting scores with bin $g = 1, ..., c$ for each point i. If the distance value is large, a histogram of the point is accumulated by small values, where the accumulated histogram has c bins. The accumulated value of the bin of the histogram $h_i[g]$ represents the likelihood of point i belonging to the corresponding segment S_g:

$$\hat{g}(i) = \arg\max_g h_i[g] \qquad \text{for} \quad i = 1, ..., N. \tag{4}$$

The randomised voting result converges if the result does not change during T_c iterations (predefined value). Furthermore, if the result keeps changing, the algorithm is terminated after a predefined maximum iteration threshold. All histograms and grouping results over the T trials are aggregated when generating the affinity matrix, and a spectral clustering is used to determine the final grouping.

3.2.2 Iterative Fine-to-coarse Motion Segmentation

As described above, the number of segments c is one of the most important parameters, but it is impractical to set c correctly in advance. We propose an iterative fine-to-coarse search method for greedily estimating the value c based on the intermediate RV motion segmentation results. Despite the greediness of the search method, the overall search process is sufficiently fast, as each run of the RV method is relatively fast.

For estimating the fundamental matrix $\mathbf{F}$, it was shown that a set of eight (or more) corresponding points are required (well-known as the 'eight point algorithm' [48]). Of course, it is possible to estimate the fundamental matrix from seven points, but the estimation from seven points is very sensitive to noise [49]. Since RV utilises the iterative fundamental matrix estimation, at least eight points should be assigned to each segment before the first iteration of the algorithm [47], [48]. Hence, we estimate the initial number of segments as:

$$\hat{c}^{init} = \lfloor N/8 \rfloor. \tag{5}$$

Algorithm 1 Iterative Fine-to-coarse Motion Segmentation

```
Input: $X_i, i = 1, ..., N$                          ▷ Point trajectories
Output: $\hat{c}$ and $S_g, g = 1, ..., \hat{c}$
$t \leftarrow 1$
$\hat{c}^t \leftarrow \lfloor N/8 \rfloor$                 ▷ Initialise the number of segments
repeat
    $S_g^t \leftarrow$ RV motion segmentation$(\{X_i\}_{i=1}^N, \hat{c}^t)$
    $c_{<8} \leftarrow 0$
    for $g = 1, ..., \hat{c}^t$ do
        if $|S_g^t| < 8$ then
            $c_{<8} \leftarrow c_{<8} + 1$
    $\hat{c}^{t+1} \leftarrow \hat{c}^t - c_{<8}$             ▷ Update the estimated number
    $t \leftarrow t + 1$
until $c_{<8} = 0$
```

Even though every segment initially contains more than eight points, there could be some segments having less than eight points through the randomised voting procedure. If there are segments with less than eight points, they are forced to be segmented out (*i.e.* there is an over-segmentation), because the parameter c is too big. Therefore, among the resulting segments, the number of segments with less than eight points is counted ($c_{<8}$), and the new estimated segment number is set to $\hat{c}^{t+1} = \hat{c}^{t} - c_{<8}$. Then, the RV segmentation algorithm is performed iteratively with the decreased segment number $\hat{c}^{t+1}$, until all segments have more than eight points (*i.e.* $c_{<8} = 0$). The iterative fine-to-coarse segmentation procedure is described in Algorithm 1. This procedure differs from general agglomerative clustering which iteratively merges pairs of segments as one moves up the hierarchy. Instead, we find a reasonable number of segments by iteratively changing the estimation value, validating the estimates using RV motion segmentation. At this step, the parts are still over-segmented, but they are corrected by the skeleton information in the following structure refining step (Section 3.4).

For accurate segmentation, the proposed method requires more densely detected feature points (i.e. at least $\hat{c} \times 8$ feature points) than other methods [6], [8], [23], [24], as each segment is obtained by calculating the fundamental matrix. However, as long as each object segment is composed of more than eight points, the point density inside the object does not significantly affect the segmentation result.

3.3 Skeleton Estimation from Sparse Feature Points

Using a skeleton as an abstract representation of an object shape has major benefits; it can contain both essential shape features in a low-dimensional form and structural topology. There have been numerous algorithms for skeleton estimation from a silhouette image of a target object [50], [51], [52], [53]. In this section, we present a novel and adaptive object silhouette generation method from sparse feature points, and a density weighted skeleton distance map generation method using the generated silhouette.

3.3.1 Adaptive Object Boundary Generation

Finding a surrounding boundary of 2D points is a complex problem, especially when the data distribution is non-parametric and sparse. We propose a method to generate an adaptive object boundary ($\delta\Omega^f$) from sparse feature points X^f of each frame by formulating the problem as a frame-by-frame one-class data boundary learning problem. We formulated the problem as an objective function optimisation method based on support vector data description (SVDD) [54]. SVDD tries to find a tight boundary covering all target data while minimising superfluous space. We consider the description boundary as the object boundary $\delta\Omega^f$.

In order to formulate the points' covering boundary with minimal superfluous space, it was found that the description shape is a sphere with minimum volume [54]. As a result, SVDD obtains a spherically shaped closed boundary (an *hypersphere*) enclosing all feature points. It is possible to describe even highly flexible and non-convex data by adopting an implicit mapping into a high-dimensional feature space using a kernel function. The kernel function maps the target data onto a bounded and spherically shaped area in the kernel feature space, and then the hypersphere model would fit the data in the kernel space (this is comparable with using a kernel trick in the Support Vector Machine classifier when the classes are not linearly separable) [54].

The hypersphere at each frame f is characterised by a centre $\mathbf{a}^f$ and radius $\mathbf{R}^f$. The volume of the sphere is minimised by minimising $(\mathbf{R}^f)^2$ [54] , where $(\mathbf{R}^f)^2$ is used to formulate the objective function as a convex problem. The objective function to minimise $(\mathbf{R}^f)^2$ with slack variable $\xi_i^f \geq 0$ and penalty parameter C which controls the trade-off between the volume and the errors is defined as:

$$F(\mathbf{R}^f, \mathbf{a}^f) = (\mathbf{R}^f)^2 + C \sum_i^N \xi_i^f, \tag{6}$$

subject to the following constraints:

$$\|x_i^f - \mathbf{a}^f\|^2 \leq (\mathbf{R}^f)^2 + \xi_i^f, \quad \xi_i^f \geq 0 \quad \forall i. \tag{7}$$

Eq.(6) and Eq.(7) can be combined by introducing Lagrange multipliers $\alpha_i^f \geq 0$ and $\gamma_i^f \geq 0$:

$$L(\mathbf{R}^f, \mathbf{a}^f, \alpha_i^f, \gamma_i^f, \xi_i^f) = (\mathbf{R}^f)^2 + C \sum_i^N \xi_i^f \\ - \sum_i^N \alpha_i^f \Big((\mathbf{R}^f)^2 + \xi_i^f - x_i^f \cdot x_i^f - 2\mathbf{a}^f \cdot x_i^f + \|\mathbf{a}^f\|^2 \Big) - \sum_i^N \gamma_i^f \xi_i^f, \tag{8}$$

L should be minimised with respect to $\mathbf{R}^f, \mathbf{a}^f$ and ξ_i^f; and maximised with respect to α_i^f and γ_i^f. After setting the partial derivatives of L to zero, we obtain the following dual objective function

$$\max_\alpha L(\alpha) = \max_\alpha \Big(\sum_i^N \alpha_i^f x_i^f \cdot x_i^f - \sum_i^N \sum_j^N \alpha_i^f \alpha_j^f x_i^f \cdot x_j^f \Big), \tag{9}$$

which is subject to

$$\sum_i^N \alpha_i^f = 1, \qquad \mathbf{a}^f = \sum_i^N \alpha_i^f x_i^f, \qquad 0 \leq \alpha_i^f \leq C. \tag{10}$$

We solve the objective function and constraint equations by a modified online recursive algorithm [46], [55], which is much faster than a general quadratic programming solver. Analogous to support vector machines (SVM), data can be categorised according to the value of α_i^f; a datum x_i^f with $0 < \alpha_i^f < C$ is called support vector (SV), and x_i^f with $\alpha_i^f = C$ is called an outlier.

By introducing a kernel trick, a non-spherical flexible boundary can be generated by replacing the inner product $(x_i^f \cdot x_j^f)$ with a kernel function $K(x_i^f, x_j^f) = \Phi(x_i^f) \cdot \Phi(x_j^f)$, where Φ is an implicit mapping of the data into high dimensional feature space, and the kernel parameter is indicated as σ^f. We use an exponential radial basis function (eRBF) kernel $K(x_i^f, x_j^f) = exp\Big(- \|x_i^f - x_j^f\| / 2(\sigma^f)^2 \Big)$ which produces a tight piecewise linear solution [56]. Furthermore, as was discussed in [46], the eRBF kernel maps data into a unit-normed feature space as follows:

$$\|\Phi(x)\|^2 = K(x, x) = 1 \longrightarrow \|\Phi(x)\| = 1. \tag{11}$$

This property makes it possible to calculate the *sample margin* [46] of each datum.

Kernel parameter selection using *sample margin*: In this work, we propose a novel optimal kernel parameter selection method using *sample margins* which were first introduced and used for inlier reduction and model selection by Kim *et al.* [46], [57]. The *sample margin* is analogous to the conventional margin definition of SVMs, but it is different in that the margin is defined

Algorithm 2 Object boundary estimation with adaptive kernel parameter selection process

```
Input: $X^f, \Delta\sigma, \sigma_{min}, \sigma_{max}$
Output: $\delta\Omega^f$, $\hat{\sigma}^f$    ▷ Object boundary at frame $f$
$\mathcal{T}_{init} \leftarrow -1/log(p_{init})$,  $\mathcal{T}_{final} \leftarrow -1/log(p_{final})$,  $\mathcal{T}_{current} \leftarrow \mathcal{T}_{init}$    ▷ Initial, final, current temperature
$frac \leftarrow (\mathcal{T}_{final}/\mathcal{T}_{init})^{1/(N_{cycle}-1)}$    ▷ Fractional temperature reduction every cycle
$\sigma_{current} \leftarrow (\sigma_{max} - \sigma_{min}) * random(0,1) + \sigma_{min}$    ▷ Set the current best result as initial random value
$H_{current} \leftarrow H(\gamma(X^f; \sigma_{current}))$    ▷ Train data by Eq.(9) with $\sigma_{current}$ and compute the entropy $H$ according to Eq.(13)
$\Delta E \leftarrow 0$,  $\Delta E_{avg} \leftarrow 0$    ▷ $\Delta E$ is the change in objective function. $\Delta E_{avg}$ is the average change.
for $i = 1$ to $N_{cycle}$ do
    for $j = 1$ to $N_{trial/cycle}$ do
        $\sigma_{new} \leftarrow \sigma_{current} + \Delta\sigma * random(-1,1)$    ▷ Generate a new trial value
        $\sigma_{new} \leftarrow max(min(\sigma_{new}, \sigma_{max}), \sigma_{min})$    ▷ Clip to upper and lower bounds
        $\Delta E \leftarrow |H(\gamma(X^f; \sigma_{new})) - H_{current}|$    ▷ Re-train data with $\sigma_{new}$, and compute $H$ with the $\sigma_{new}$
        if $H(\gamma(X^f; \sigma_{new})) > H_{current}$ then
            $p \leftarrow exp(-\Delta E/(\Delta E_{avg} * \mathcal{T}_{current}))$    ▷ Generate probability of acceptance
            if $random(0,1) < p$ then accept ← TRUE    ▷ Accept the worse solution
            else accept ← FALSE    ▷ Do not accept the worse solution
        else accept ← TRUE    ▷ If objective function is lower, then automatically accept it
        if accept = TRUE then
            $\sigma_{current} \leftarrow \sigma_{new}$,  $H_{current} \leftarrow H(\gamma(X^f; \sigma_{new}))$    ▷ Update currently accepted solutions
            $\Delta E_{avg} \leftarrow (\Delta E_{avg} * N_{accept} + \Delta E)/(N_{accept} + 1)$    ▷ Update $\Delta E_{avg}$
            $N_{accept} \leftarrow N_{accept} + 1$    ▷ Increment number of accepted solutions
    $\mathcal{T}_{current} \leftarrow frac * \mathcal{T}_{current}$    ▷ Lower the temperature for next cycle
$\hat{\sigma}^f \leftarrow \sigma_{current}$
Generate $\delta\Omega^f$ with the selected kernel parameter $\hat{\sigma}^f$ by Eq.(14).
```

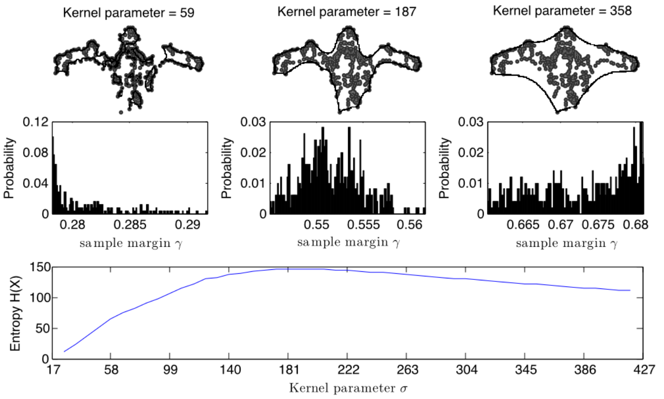


Fig. 2. Object boundary generation results with various kernel parameters. A small parameter value produces over-estimated results with separated boundary regions, and a large value gives an under-estimated boundary result. The distributions of the sample margin γ are shown in the middle with respect to each kernel value. As we can see, the most appropriate boundary is generated by the kernel value which results in the maximum entropy.

not only for the hyperplane but also for all data [46]. The *sample margin* $\gamma(x_i^f)$ is basically a relative distance from a datum to a hyperplane passing through the centre of the hypersphere in a kernel space [46]. It can be calculated as follows for each data point x_i^f:

$$\gamma(x_i^f) = \frac{\mathbf{a}^f \cdot \Phi(x_i^f)}{\|\mathbf{a}^f\|}, \tag{12}$$

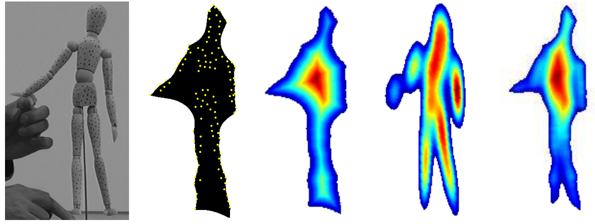
where $\mathbf{a}^f = \sum_i^N \alpha_i^f \Phi(x_i^f)$. Because we are using the eRBF kernel, which produces a unit-normed feature space, the sample margins are within a range of $0 \leq \gamma(x_i^f) \leq 1$ [46]. Each sample margin indicates a normalised relative position between the centre and the boundary of the hypersphere, and different kernel parameters produce different sample margin distributions as well as different description boundaries as shown in Figure 2.

Based on the sample margin property, we propose a new kernel parameter selection criterion to select the optimal σ^f for each frame f by calculating the entropy of the sample margin distribution. We found that if the object boundary is overfitted, the sample margins are distributed toward the boundary of the hypersphere. In contrast, if the boundary is underfitted, the distribution is biased to the centre. We avoid over/underfitting by finding the kernel parameter with the maximum entropy (*i.e.*, evenly spread). This is supported according to the principle of maximum entropy [58]. If no prior knowledge is available about a distribution, then the current state of knowledge is best represented by the probability distribution with the largest entropy. The optimal kernel parameter $\hat{\sigma}^f$ of the current frame feature points X^f can be selected by

$$\begin{aligned} \hat{\sigma}^f &= \arg\max_{\sigma^f} H(\gamma(X^f; \sigma^f)) \\ &= \arg\max_{\sigma^f} \sum_i -p_i log(p_i), \end{aligned} \tag{13}$$

where H is the entropy and p_i is a probability distribution with $p_i = Prob\big(\gamma(x_i^f; \sigma^f)\big)$. In order to find the optimal parameter $\hat{\sigma}^f$ in a search space of kernel values, we utilise the Simulated Annealing (SA) method. The detailed overall selection process using the proposed criterion is given in Algorithm 2. σ_{max} is set to *'image width'* and σ_{min} to *'image width*$\times 0.01$', and the value of $\Delta\sigma$ is empirically set to *'image width*$\times 0.1$'. The parameters of SA are as follows: number of cycles $N_{cycle} = 10$, number of trials per cycle $N_{trial/cycle} = 10$, number of accepted solutions $N_{accept} = 1$, and probability of accepting a bad solution at the start $p_{init} = 0.7$, and at the end $p_{final} = 0.001$.

Object boundary generation: For each frame f, the object boundary $\delta\Omega^f$ can be considered as a set of all pixel points q



(a) (b) $\delta\Omega^f$ (c) Π^f (d) $\mathcal{D}^f$ (e) Ψ^f

Fig. 3. Comparison between skeleton maps. (a) Puppet image (b) Generated object boundary $\delta\Omega^f$ from feature points X^f (yellow dots) (c) The skeleton distance map Π^f. The arms and legs are not well-separated and detailed shape information is lost. (d) The density map $\mathcal{D}^f$. (e) The density weighted skeleton map Ψ^f. By re-weighting the skeleton distance map according to the density map, detailed shape characteristics as well as skeleton distance information can be retained. Figure best viewed in colour.

of image space $\mathcal{I}$ lying at the same distance to the centre $\mathbf{a}^f$ of the hypersphere as the radius $\mathbf{R}^f$. The object boundary can be generated with the selected optimal kernel parameter $\hat{\sigma}^f$ by

$$\delta\Omega^f = \{q|\forall q \in \mathcal{I}, \|q - \mathbf{a}^f\|^2 = 1 - 2\sum_i \alpha_i^f K(q, x_i^f; \hat{\sigma}^f) + \sum_{i,j} \alpha_i^f \alpha_j^f K(x_i^f, x_j^f; \hat{\sigma}^f) = (\mathbf{R}^f)^2\}. \quad (14)$$

3.3.2 Density Weighted Skeleton Distance Function

The shape information provided by the silhouette is represented in a simpler but more informative format by a skeletonisation method. The skeleton can provide useful additional characteristics by preserving the original object's topological and hierarchical properties. A skeleton of an object, $\Upsilon(\Omega^f)$, is defined as the set of all centre points of maximal circles contained in an object Ω^f, which is a medial axis of an object [50]. It can be formulated as the locus of points at equal distance from at least two boundary points as described in [53]:

$$\Upsilon(\Omega^f) = \{p \in \Omega^f | \exists q, r \in \delta\Omega^f, q \neq r : dist(p, q) = dist(p, r)\}. \quad (15)$$

The skeleton contains both shape features and topological structures of the original objects. As a good representation of the skeleton, a distance transform [51] is defined as a function that returns the closest distance to the boundary for each point p. Using the obtained object boundary $\delta\Omega^f$, the distance function $\Pi^f(p)$ of Ω^f is defined as [53]:

$$\Pi^f(p) = \min_{q \in \delta\Omega^f} \big(dist(p, q)\big), \quad (16)$$

for all points $p \in \Omega^f$. The distance metric is usually the Euclidean distance $dist(p, q) = \|p - q\|_2$. Using the distance function is attractive as its computation is relatively simple, and the skeleton can be generated as the ridges of the distance function.

However, as seen in Figure 3, the generated object boundary from feature points is abstracted too much. That is, detailed shape information is lost, and the skeleton distance map Π^f is overweighting the central area of the object. In order to retain the detailed shape information and to give more weight to that area of the skeleton map, we generate a density map of the feature points X^f using a kernel-based adaptive density estimation method proposed by [59]. The density map is represented as $\mathcal{D}^f(p; X^f)$; and we normalise each point of the map $\mathcal{D}^f(p)$ between 0 and 1 by $\mathcal{D}^f(p) = \mathcal{D}^f(p)/max(\mathcal{D}^f)$. As we can see in Figure 3(d), detailed shape information is preserved in the density map. Then, we generate a new density weighted skeleton map Ψ^f by combining the skeleton distance map and the density as follows:

$$\Psi^f(p) = \mathcal{D}^f(p) \times \Pi^f(p). \quad (17)$$

The density weighted skeleton map preserves detailed shapes as well as the overall object's topological information. Figure 3 shows the effects of the weighted map, especially Figure 3(e) shows that the two legs of the puppet are well-represented.

3.4 Kinematic Structure Building

In this section, we present a way to generate the kinematic structure of the articulated object combining the motion segments and the density weighted skeleton map. We assume that the kinematic structure is not cyclic (as in [6]), which covers most articulated objects. We utilise a graphical model $G = (V, E)$ to determine the topological connections between motion segments. All motion segment centres $y_1^f, ..., y_{\hat{c}}^f$ are treated as nodes V in a complete graph. The proximity $E(y_k, y_l)$ between segment k and l is defined as:

$$E(y_k, y_l) = \operatorname*{median}_{f \in F}\{\zeta(y_k^f, y_l^f; \Psi^f) \times \|\dot{y}_k^f - \dot{y}_l^f\|\}, \quad (18)$$

which is a combination of geodesic distance in the density weighted skeleton distance map Ψ and moving velocity difference. $\dot{y}_k^f$ indicates segment k's motion velocity calculated by $\dot{y}_k^f = y_k^f - y_k^{f-1}$. For the final proximity estimation between two segments over all frames F, we take the median value in order to be robust to outliers.

Given the density weighted skeleton distance map Ψ, the geodesic distance between two points $\mathbf{p}$ and $\mathbf{q}$ is defined as:

$$\zeta(\mathbf{p}, \mathbf{q}; \Psi^f) = \min_{\Gamma \in \mathcal{P}_{\mathbf{p},\mathbf{q}}} \sum_{n=1}^{l(\Gamma)} \frac{1}{\Psi^f(p_n)}, \quad (19)$$

where Γ is a path connecting the two points and $\mathcal{P}_{\mathbf{p},\mathbf{q}}$ is the set of all possible paths. Thus Eq. (19) defines the minimum distance between two points in the object region via the skeletal topology path as shown in Figure 4. The proposed proximity measure separates segments that are topologically apart and move with different velocities. Two segments with small edge weight have a large possibility to be connected. We generate the graph's minimum spanning tree as the kinematic structure of the object.

Finally, we run a structure refining step that iteratively merges segments guided by the density weighted skeleton map Ψ^f. This step is necessary for correcting a few over-segmentation errors, as the motion segments before this step are initially over-segmented considering only motion information. Furthermore, skeleton information can be embedded through this refining process. If a segment S_k largely deviates from the medial skeleton axis, then the $\Psi^f(y_k)$ becomes small (*i.e.* $\Psi^f(y_k) < \tau$). The threshold τ

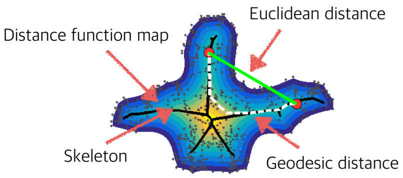


Fig. 4. The white dotted line shows the geodesic distance ζ between two points, and the green solid line shows the Euclidean distance. The black solid line is the skeleton of the object. The geodesic distance represents the minimum distance following the skeleton. Figure best viewed in colour.

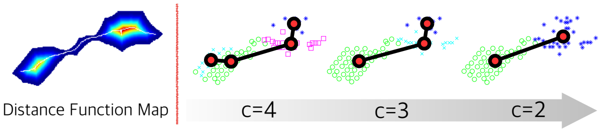


Fig. 5. Refining process for removing over-segmentation by iterative segments merging guided by skeleton information. Best viewed in colour.

TABLE 1
Properties of the dataset. The newly introduced data (boldface) are more challenging because they are composed of concurrent and highly articulated motions for longer frame sequences.

Dataset	# of segm.	# of points	# of frames	motion concur
arm	2	77	30	no
toy truck	3	83	60	no
dancing	6	268	40	yes
head	2	99	60	yes
yellow crane	4	97	50	yes
puppet	7	114	563	no
iCub body	7	573	250	yes
iCub hand	8	154	280	yes
robot arm	8	144	737	yes
Baxter	11	484	454	yes

is adaptively set as the minimum distance value of the skeleton Υ^f; $\tau = \min \Psi(\Upsilon^f)$. Then, the deviated S_k is merged with a connected neighbour segment S_l which has a larger $\Psi^f(y_l)$ value (*i.e.* $S'_k = \{S_k \cup S_l\}$, and $\hat{c} = \hat{c} - 1$). Then, we reconstruct the kinematic structure until all segment centres are located close to the skeleton (see Figure 5).

4 Experiments

Dataset The proposed method was evaluated with widely used sequences such as 'arm' [60], 'toy truck', 'dancing', 'head', 'yellow crane' and 'puppet' (all [6], see Table 1), but these data sequences are relatively simple. Here, we introduce new challenging sequences which are prolonged and composed of complexly articulated motions[1]. We avoided recording severe self-occlusions because the 2D feature point tracker [61] cannot handle occlusions, and overcoming the occlusion issue is not the main purpose of the dataset. However, the new sequences still contain diverse complex motions such as articulations, concurrency, rotations, affine transformations, and scaling. We summarised the dataset properties in Table 1. Similar to the conventional dataset, background regions are masked and the feature points are extracted only from object regions. We have assigned the feature points to ground truth motion segments (S^{GT}), and the ground truth segment centres (y_{GT}) were annotated by computing the mean of the assigned features. The ground truth kinematic structure was generated by manually connecting the centres considering the topology and kinematics. To encourage the use of our novel approach we release our code along with the new dataset[2]. All experiments were performed using a PC with an Intel Core i7-4770 CPU @ 3.40GHz (x8) and 32GB of RAM.

4.1 Experimental Analysis

We have performed various experiments to validate core modules of the proposed algorithm: the iterative motion segmentation and adaptive object boundary generation. In order to evaluate the performance, we used standard motion segmentation evaluation measures as described in [38]: precision (P), recall (R), and F-measure (F). These metrics have proven valuable in capturing a similar trade-off between false positives and misses [38]. S_i indicates a subset of points belonging to estimated segment i, and S_j^{GT} is a subset of points of a ground truth segment j. The precision measures the ratio of correctly segmented features, and the recall measures the fraction of ground truth covered by the estimated segments. The F-measure is the harmonic mean of the precision and recall. The precision ($P_{i,j}$), recall ($R_{i,j}$), and F-measure ($F_{i,j}$) for each pair of segment i and ground truth segment j are defined as follows:

$$P_{i,j} = \frac{|S_i \cap S_j^{GT}|}{|S_i|} \quad R_{i,j} = \frac{|S_i \cap S_j^{GT}|}{|S_j^{GT}|} \quad F_{i,j} = \frac{2P_{i,j}R_{i,j}}{P_{i,j} + R_{i,j}}. \tag{20}$$

As described in [38], the best allocation of segments to ground truth segments is found by the Hungarian method, a one-to-one matching algorithm that maximises the F-measure. In case there are fewer estimated segments than ground truth segments, empty segments are introduced for the calculation, which makes their recall be zero and we define the precision as one.

As our method is based on randomised voting, the results vary slightly across different trials. We have investigated that the evaluation statistics do not change significantly after around one hundred trials. Thus, all the statistics of experimental results are obtained from the one hundred trials. We would like to emphasise that all experiments were performed with the same parameter settings, which means that no parameter tuning was needed for the different sequences.

4.1.1 Motion Segment Convergence

We validated whether the proposed iterative fine-to-coarse merging process with the refining process can estimate the correct number of segments. As shown in Figure 6, the number of segments converged closely to the ground truth value over time. There are two stages for the convergence: the faster first stage is the iterative fine-to-coarse motion segmentation procedure (Section 3.2), and the second stage is the structure refining procedure guided by the skeleton topology (Section 3.4). From Figure 6, it is evident that the fine-to-coarse merging usually leads to over-segmentation of the scene. This over-segmentation is successfully corrected once the skeleton information is integrated via structure refining. This shows a reasonable justification of using skeleton information for motion segmentation at the first place and supports the key idea.

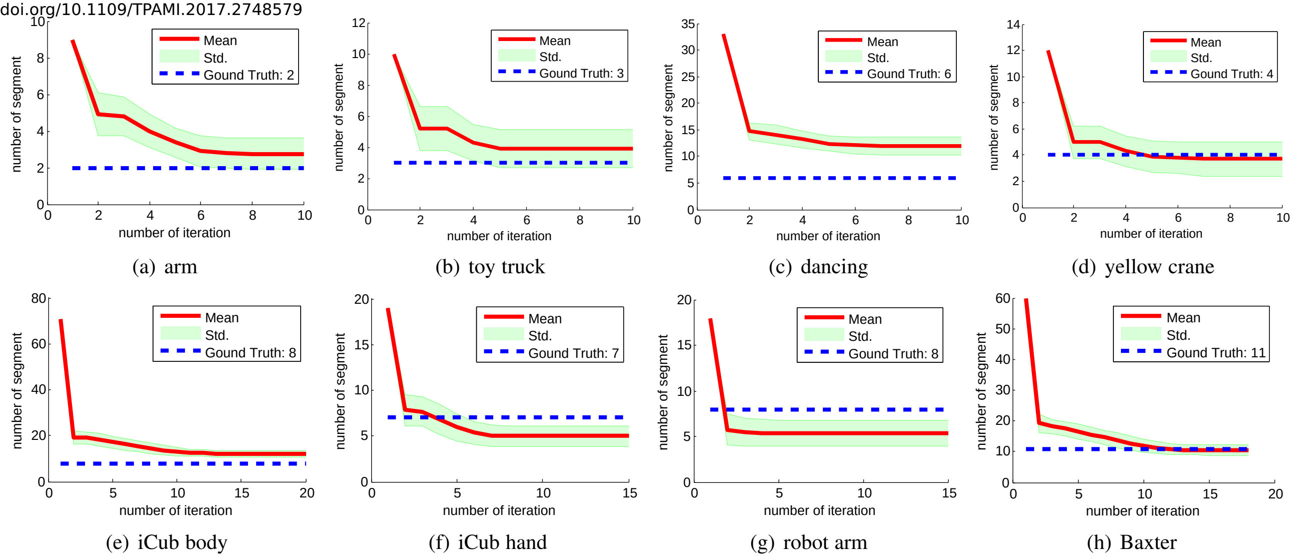
4.1.2 Object Boundary Generation with Adaptive Kernel Parameter Selection

In this section, we validate the goodness of the proposed adaptive object boundary generation. The object boundary shape is highly dependent on the proper kernel parameter selection, so through various experiments, we firstly tested whether the selected kernel parameter is similar to the ground truth. Secondly, we qualitatively compared our sparse motion feature based results with other methods using various image cues. Finally, the proposed method was tested on the standard Freiburg-Berkeley Motion Segmentation dataset (FBMS-59) using the standard evaluation criteria [38].

Validation of the parameter selection method: As discussed in Section 3.3.1, the best kernel parameter is selected according to the maximum entropy. The entropy value distribution displays a generally concave shape. If there is no large scale change

[1] We utilised three robots: OWI-535 Robotic Arm Edge (http://www.owirobot.com), iCub (http://www.icub.org), and Baxter (http://www.rethinkrobotics.com/baxter/)

[2] http://www.imperial.ac.uk/personalrobotics



(a) arm (b) toy truck (c) dancing (d) yellow crane

(e) iCub body (f) iCub hand (g) robot arm (h) Baxter

Fig. 6. The number of motion segments converges closely to the ground truth using the proposed iterative process.

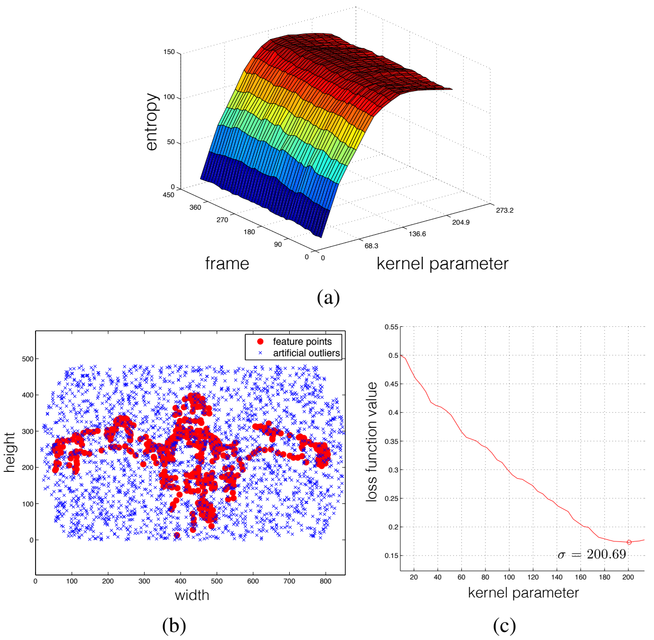


Fig. 7. (a) The entropy value distribution over frames. The optimal kernel values for the 'Baxter' sequence over frames are $182.3(\pm 7.4)$. The selected kernel values are relatively consistent while the object moves. (b) Test data and artificially generated outliers for the 'Baxter' sequence. (c) The loss function result of 'Baxter' sequence. The kernel parameter with the minimum loss function value (200.69) is the best selected values for each sequence by [44]. Note that the false positive rate are calculated based on artificial outliers.

during the motion, we observed that the distribution of the entropy value does not change much and the optimal kernel values are almost similar for all frames as shown in Figure 7(a). This is because the feature points are from rigid body parts, whose relative distances do not change much through motion, and thus the sample margin distribution is not varied either. We have found that these properties remain the same for all data sequences, so we show further analyses using the 'Baxter' sequence as an example.

By comparing with the parameter selection results of Landgrebe *et al.* [44], which has been widely considered as a baseline method for classifiers, we show the effectiveness of the proposed criterion in terms of computational efficiency and accuracy. To select the best parameter for producing a well-balanced boundary, Landgrebe *et al.* [44] use loss function minimisation, where the loss function is the sum of the true positive rate TP_r and the false positive rate FP_r. By measuring the data classification rates they try to find a moderate boundary which is neither overfitted nor underfitted. However, in our problem, outliers are unavailable. In order to consider the false positive rate, artificial outliers are generated as in [62]. The overall loss function L now becomes $L = \frac{1}{2}(1 - TP_r) + \frac{1}{2}FP_r^{a.o.}$ where $FP_r^{a.o.}$ is the false positive rate of the artificial outliers. The artificial outliers were generated using a uniform distribution around the feature points as described in [62], which is shown in Figure 7(b). The overall loss function values calculated by [44] are shown in Figure 7(c), and the kernel parameter which gives the minimum loss function value (200.69) is selected for the sequence by [44]. However, both [44], [62] are computationally expensive, as first artificial outliers need to be generated, then more data need to be tested (target data + artificial outliers), which is finally followed by a greedy search. On the contrary, our proposed method determines the best kernel parameter only using the given feature points. Although Kim's method [46] does not require the outlier generation either, it is based on a greedy search which takes more time than our method. As shown in Table 2, our experiments demonstrate that the proposed method finds the closest parameters to the ground truth values while needing less computational time compared to Landgrebe's [44] and Kim's [46] methods. This is because we do not need to generate artificial outliers, and perform the search in a more strategic manner.

Comparison with various image cue based methods: We have compared our object boundary result with other object boundary generation methods using various cues (see Figure 8). First of all, the grey image based edge detection (Canny edge detection) and the colour cue based detection [39] find internal / background edges, as well as edges belonging to the object boundary with the same importance, which makes the extraction of the object's outer boundary difficult. The state-of-the-art object motion boundary generation method [40] with RGB static cues and motion cues detects moving parts of the object correctly but does not describe the object's overall shape (see 8(d)). The dense trajectory clustering method [23] is focused on motion cues, so homogeneous regions are not segmented correctly. The proposed method generates the overall shape based on the extracted sparse local feature positions. Although the shape is abstracted, it is

Preprint version; final version available at http://ieeexplore.ieee.org/document/8025409/
IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
DOI: https://doi.org/10.1109/TPAMI.2017.2748579

TABLE 2
The selected kernel parameter values and time requirements for various methods. We compare the proposed criterion with the baseline method (Landgrebe *et al.* [44]) and [46]. As a result, the proposed method finds the closest values to manually selected ground truth values (σ_{GT}), while being $2.8 \sim 4.6$ times faster than others. The ϵ_{diff} means an absolute difference between the selected parameter value ($\hat{\sigma}$) and the ground truth value (σ_{GT}) as $\epsilon_{diff} = |\hat{\sigma} - \sigma_{GT}|$.

Dataset	Ground Truth Param. ($param_{GT}$)	Kim [46]		Landgrebe *et al.* [44]		Proposed	
		Time (sec)	ϵ_{diff}	Time (sec)	ϵ_{diff}	Time (sec)	ϵ_{diff}
arm	96	2.8 (±0.2)	41.6 (±0.0)	12.5(±0.1)	57.6 (±0.0)	4.5 (±0.0)	**0.3** (±6.0)
toy truck	90	3.0 (±0.1)	3.6 (±0.0)	15.9 (±0.1)	1.8 (±0.0)	4.5 (±0.1)	**0.0** (±0.1)
dancing	144	35.6 (±0.5)	10.8 (±0.0)	205.9 (±5.5)	3.6 (±7.4)	51.4 (±0.6)	**2.1** (±6.5)
head	72	4.0 (±0.4)	18.0 (±0.0)	22.5 (±2.2)	12.6 (±0.0)	5.9 (±0.1)	**1.5** (±2.4)
yellow crane	108	4.6 (±1.5)	28.8 (±0.0)	17.2 (±0.1)	10.8 (±9.6)	5.9 (±0.1)	**0.4** (±5.0)
puppet	112	5.2 (±1.2)	20.8 (±0.0)	27.5 (±0.4)	1.6 (±0.0)	7.0 (±0.1)	**1.2** (±5.9)
iCub body	118	393.7 (±56.0)	2.8 (±0.0)	1450.4 (±70.8)	19.5 (±2.3)	446.3 (±8.3)	**2.8** (±5.4)
iCub hand	66	13.6 (±1.0)	1.2 (±0.0)	54.4 (±9.0)	9.2 (±3.5)	11.7 (±0.2)	**0.7** (±2.9)
robot arm	155	5.7 (±0.6)	3.4 (±0.0)	41.7 (±6.6)	6.3 (±5.0)	9.7 (±0.1)	**2.1** (±8.1)
Baxter	187	262.5 (±38.4)	26.5 (±0.0)	1045.8 (±60.0)	11.6 (±3.0)	269.2 (±7.6)	**2.9** (±12.0)

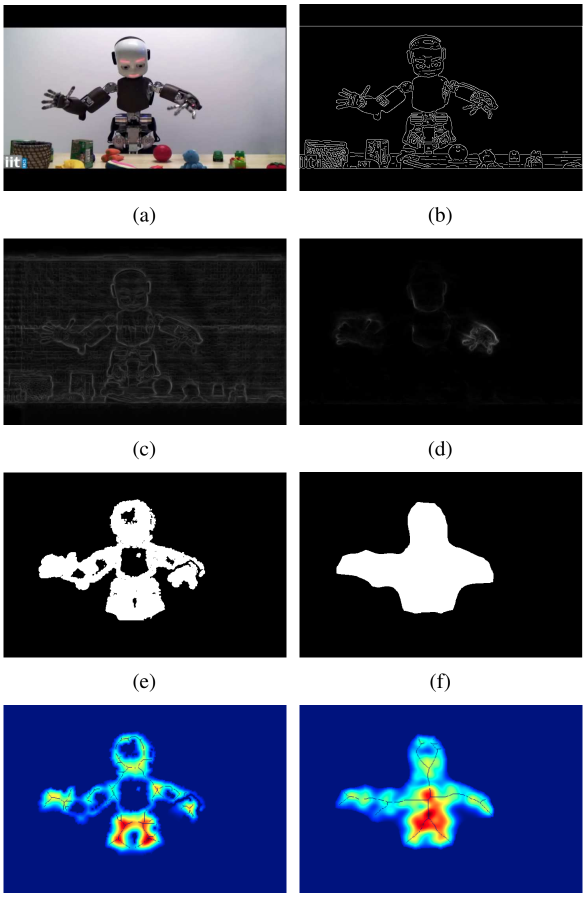

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 8. Comparisons of the moving object boundary generation of various approaches. (a) One frame of the ‘iCub body’ sequence. (b) Canny edge detection results from a grey image. (c) Using only static RGB cues and the learning approach based edge detector [39]. Internal edges and background object edges are detected as well. (d) State-of-the-art motion boundary estimation [40] using RGB + optical flow cues. (e) Clustering dense point trajectories [23]. (f) Proposed method’s boundary generation result. (g) The density weighted skeleton map Ψ^f using the generated boundary((e)) of the dense point trajectories [23]. (h) The density weighted skeleton map Ψ^f using the proposed boundary generation result ((h)).

sufficient to extract the overall skeleton.

Validation on standard object segmentation dataset (FBMS-59) [38]: The FBMS-59 dataset can be used in measuring the object boundary generation accuracy as it provides pixelwise labels, and it was already used for testing dense segmentation performances [38]. We compared with the aforementioned sparse and dense segmentation methods (MoSeg [38]), as well as with SSC [19] and ALC [17]. As shown in Table 3, our proposed method achieved intermediate performances, as our object boundary generation method does not target the most precise object segmentation, but rather generates an overall object silhouette to estimate the skeletal topology. As shown in Figure 8(g) and 8(h), the proposed method typically under-segmented the boundary. The proposed method slightly underperforms [38] on simple articulations but copes well with strong articulations instead.

4.2 Comparisons with Other Kinematic Structure Estimation Methods

We have compared the proposed kinematic structure estimation performance with other methods. We used the code provided by the authors for the sparse motion segmentation method (Moseg Sparse) [38]. All other methods were implemented as described in their respective papers using Matlab libraries, where [6] is typically considered as a baseline, and [8] is the best performing method up-to-now.

We would like to note that methods presented in [6], [8], [63] are predominantly designed from the perspective of 3D kinematic structure estimation and not only the 2D structure. Our proposed method is mainly focused on motion segmentation and 2D kinematic chain generation without 3D reconstruction, so in this paper, we did not evaluate the 3D reconstruction accuracy. In particular the energy based multiple model fitting method [8] is originally designed to enforce segments to overlap in order to resolve the per-rigid-segment depth scale ambiguity, but as we only consider 2D structures in this paper, we report performances both with and without overlaps (the overlap can be controlled by changing the parameter λ of the original paper [8]). In the overlap allowing condition ($\lambda = 0.5$), which is same as the original condition of [8], the joint centres are derived from the

TABLE 3
Acronyms are **P**: Average precision, **R**: Average recall, and **F**: F-measure as described in [38]. Trajectory sampling rates were 8 for both MoSeg methods [38]. All experimental numbers except those of the proposed method are adopted from [38]. We show results for the first 10 frames.

	Training set (29 sequences)			Test set (30 sequences)		
	P	R	F	P	R	F
MoSeg Dense [38]	92.97%	63.18%	75.24%	87.41%	58.73%	70.26%
MoSeg Sparse [38]	92.77%	65.44%	76.75%	87.44%	60.77%	71.71%
SSC [19]	67.62%	73.04%	70.22%	61.64%	60.63%	61.13%
Naive	72.63%	51.63%	60.36%	57.96%	53.41%	55.60%
ALC [17]	54.31%	54.80%	54.56%	53.11%	56.40%	54.70%
Proposed	83.37%	59.07%	65.53%	82.38%	61.60%	67.68%

Preprint version; final version available at http://ieeexplore.ieee.org/document/8025409/
IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
DOI: https://doi.org/10.1109/TPAMI.2017.2748579

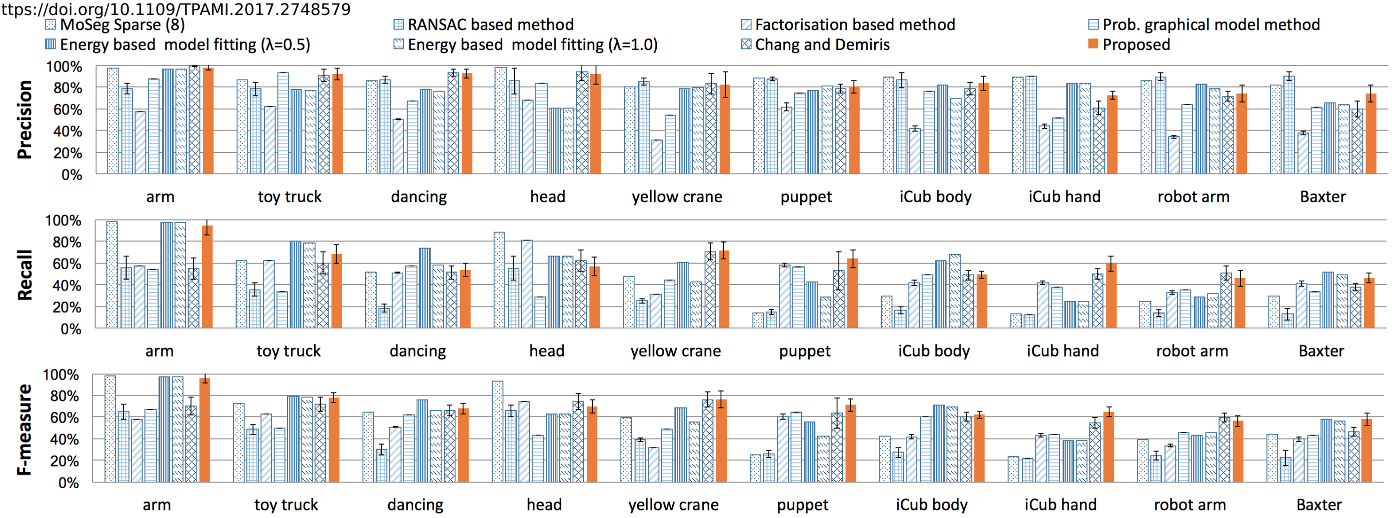
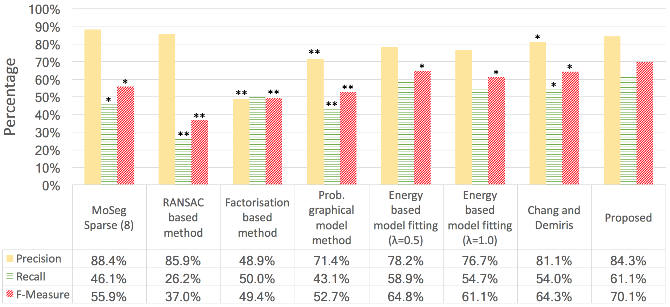


Fig. 9. Comparison with other kinematic structure estimation methods. All values are based on one hundred trials except for probabilistic method [24] (ten trials instead of one hundred as each run takes too much time), and MoSeg Sparse [38] and energy based method [8] (results are consistent). The number in parentheses of MoSeg [38] indicates a sampling rate.



	MoSeg Sparse (8)	RANSAC based method	Factorisation based method	Prob. graphical model method	Energy based model fitting (λ=0.5)	Energy based model fitting (λ=1.0)	Chang and Demiris	Proposed
▇ Precision	88.4%	85.9%	48.9%	71.4%	78.2%	76.7%	81.1%	84.3%
☰ Recall	46.1%	26.2%	50.0%	43.1%	58.9%	54.7%	54.0%	61.1%
▨ F-Measure	55.9%	37.0%	49.4%	52.7%	64.8%	61.1%	64.3%	70.1%

Fig. 10. Average values of metrics for all data sequences. $*$ (p-value $<$ 0.05) and $**$ (p-value < 0.005) indicate statistically significant difference with the proposed method. The proposed method achieves better accuracies than the other methods.

TABLE 4
Summary of the kinematic structure estimation algorithms. The proposed method efficiently estimates more accurate structures without sequence-by-sequence parameter tuning, but the result is not consistent.

Method	Structure consistency	Param. tuning per sequence	Ave. time per point (sec)	Ave. time per segm. (sec)
MoSeg Sparse (8) [38]	Y	Y	19.2	508.1
RANSAC method [63]	N	Y	0.2	9.4
Factorisation method [6]	N	Y	0.02	0.9
Probabilistic graphical method [24]	N	Y	250.0	10975.9
Energy based model fitting ($\lambda = 0.5$) [8]	Y	Y	53.6	1856.5
Energy based model fitting ($\lambda = 1.0$) [8]	Y	Y	56.9	1892.7
Chang and Demiris [1]	N	N	1.5	51.0
Proposed	N	N	1.8	56.5

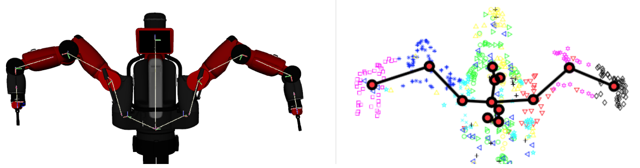
regions of overlap as described in [8]. In the non-overlap allowing condition ($\lambda = 1.0$), the joint centres cannot be defined on the overlap regions, so we define the joint centres as the centres of the rigid body parts similarly to the proposed method. We show the performance of both conditions.

In order to obtain reasonable results for the methods of [6], [8], [24], [63], we manually tuned some parameters for each data sequence (such as the number of motion segments, the rank detection parameter [6], and the number of nearest neighbours [8]).

Quantitative comparison: In Figure 9, we show the accuracy values (precision, recall and F-measure [38]) on the dataset of Table 1. The state-of-the-art 2D motion segmentation method as used in [38] is a direct alternative for the proposed method. In [38] feature points are extracted from all image regions which do not have ground truth labels. When we evaluated the performances in Figure 9 and Figure 12, the background regions were removed and measured only for the foreground object regions, which have ground truth labels. As we can see in Figure 9 and Figure 12, [38] showed very good performance for simple structure sequences but showed a relatively rapid performance degradation as the motion complexity increased. Similarly, the RANSAC based method [63] and the factorisation based method [6] were very sensitive to parameter settings and noise, and the effect of noise increases largely for complex motions. In particular, the Moseg Sparse [38] and the RANSAC based method [63] showed large precisions but low recalls, and the gaps between the two metrics become large as the structure complexity increases. This shows that the methods tend to underestimate segments for complex structure estimation problems. The factorisation method [6] runs faster than the other methods but is of low accuracy.

The probabilistic graphical model method [24] is based on a greedy search based expectation-maximisation (EM) method, which makes it very slow. Even processing simple data sequences involves a significant time constraint (in the orders of hours), although the implementation was optimised and a multi-core process was used. The high computational load hampers tuning parameters (although good parameters are critical for a reasonable performance). We performed the test in Figure 9 with parameter settings producing moderate accuracy obtained within a reasonable computational time. Another drawback we encountered is that the method easily falls into local minima, which could only be avoided using precise initialisation settings. The method updates variables by considering all data points at each step, and as the number of articulations increases, the computation time increases exponentially. This makes it very hard for the algorithm to be applied to complex motion analysis because the motion is generally composed of many feature points and articulations (‘arm’ took 496.5 seconds but ‘Baxter’ took 7.7 days).

The method of [8] generally performs well, as the energy function balances overall model complexity and local motion errors. We expected that the without-overlap condition ($\lambda = 1.0$) would give more flexibility to the model and would perform better. However, as shown in Figure 12, the original condition of [8] which allows segments overlap ($\lambda = 0.5$) shows better performances resulting more detailed segments.



(a) Real physical kinematic structure (b) Estimated kinematic structure

Fig. 11. Comparison between real physical kinematic structure obtained from its simulator and estimated kinematic structure of the Baxter humanoid. The physical kinematic structure joints are located in the motor joints, and the proposed method places the joints to the centre of each body part. Although there are some erroneous joints in the torso because no motion information is available from the non-moving part, the proposed method could generate a similar arm kinematic structure to the physical structure.

Figures 9 and 10 show that our approach achieves comparably high average accuracy for simple articulations, and our method outperformed the other methods for many complex articulated motion sequences. Based on the paired-Wilcoxon test, the proposed method outperforms all other methods base on the F-measure (p-value < 0.05) and shows comparable performances in Precision and Recall. In Table 4 we summarised important properties and numerical results of all compared methods. The proposed method accurately estimates structures in a relatively short time, and critical parameter tuning is not required for different sequences. The proposed method achieves 9.0% F-measure accuracy enhancement with 28.5% less computational time on average by adopting the simulated annealing (SA) search and density weighted skeleton map when comparing to Chang and Demiris [1]. Furthermore, the SA search reduces the computational time variances between trials. However, the randomised voting based motion segmentation produced slightly different segmentation results each time, and consequently, the motion values based on the segmentation results affected to the final kinematic structure building outputs. In addition, our method produced finer segments than other segmentation methods but sometimes produced over-segmented results so the Precision value was relatively small.

In addition, Figure 11 shows a comparison between a real physical kinematic structure of Baxter humanoid obtained from its simulator and the estimated kinematic structure from visual data. There are some redundant joints in the torso part as the torso does not move at all in the sequence and thus motion segmentation is not successful. However, both arms are accurately estimated similar to the real kinematic structure.

Qualitative comparison: In Figure 12 we present some qualitative kinematic structure estimation results. The factorisation based method [6] suffered from high noise levels and complexity, resulting in incorrectly estimated segments. For the energy based model fitting method [8], we showed three different results: a) allowing segment overlaps and placing the joints on the overlap (following the setting of [8]), b) allowing segment overlaps but placing the joints on the centre of each segment (same as the proposed method's joint location setting), and c) not allowing overlaps and placing the joints on the centre of each segment. The results of the condition b) showed the most accurate estimation performances with better structure representations for many sequences, especially for 'toy', 'dance', 'robot arm' and 'Baxter'.

Comparably we can see that the fine-to-coarse segmentation procedure detailed structures while keeping structural topology, and the skeleton information reduces the noise effect. The density weighted skeleton map preserves shape details to some degree. Hence, there are performance improvements in describing relatively small and narrow parts such as the thumb of the 'iCub hand'. However, in the result of 'puppet' in Figure 12, the motion segmentation results are correct but the obtained kinematic structure is not perfect. In the process of generating a structure graph using the minimum spanning tree, the result did not include multiple branch nodes. Finding a method to accurately represent such complex structures with several branches will be considered in a future research topic.

5 Conclusion and Future Work

In this paper, we introduced a novel articulated kinematic structure estimation framework using motion and skeleton information. We have demonstrated that the challenges are effectively met by applying the iterative fine-to-coarse segment merging process followed by a refinement step guided by skeleton information. The adaptive object silhouette generation from sparse feature points was proposed with a new effective parameter selection method. Our method was evaluated using public datasets and our complex articulated motion dataset. Experimental results showed that the newly proposed iterative fine-to-course scheme, object boundary generation with adaptive parameter selection, as well as the kinematic structure generation method are valid.

The proposed method has some limitations. Firstly, the method cannot build a structure under severe occlusions because the 2D feature point tracker fails when occlusions occur. Secondly, the RV motion segmentation needs complete feature trajectories, and thus prohibits online learning. Finally, randomisation of motion segmentation affects the consistency of the resultant structure, and improving the structure consistency is an important future work. Another future work would be to apply the proposed framework to the 3D reconstructed shape might be beneficial in generating the object silhouette and its density map. We also plan to use a depth camera for object separation and 3D shape generation.

Acknowledgement: This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139. The authors wish to thank Chris Russell for helpful comments on the energy based method implementation and providing the dataset. The authors gratefully acknowledge the support from the members of the Personal Robotics Laboratory at Imperial College London.

References

[1] H. J. Chang and Y. Demiris, "Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3138–3146.

[2] A. A. Chaaraoui, J. R. Padilla-Lopez, and F. Florez-Revuelta, "Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2013, pp. 91–97.

[3] J. Sturm, C. Plagemann, and W. Burgard, "Body schema learning for robotic manipulators from visual self-perception," *Journal of Physiology-Paris*, vol. 103, no. 35, pp. 220 – 231, 2009.

[4] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown 3D articulated objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[5] H. J. Chang, T. Fischer, M. Petit, M. Zambelli, and Y. Demiris, "Kinematic structure correspondences via hypergraph matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4216–4425.

Preprint version; final version available at http://ieeexplore.ieee.org/document/8025409/
IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
DOI: https://doi.org/10.1109/TPAMI.2017.2748579

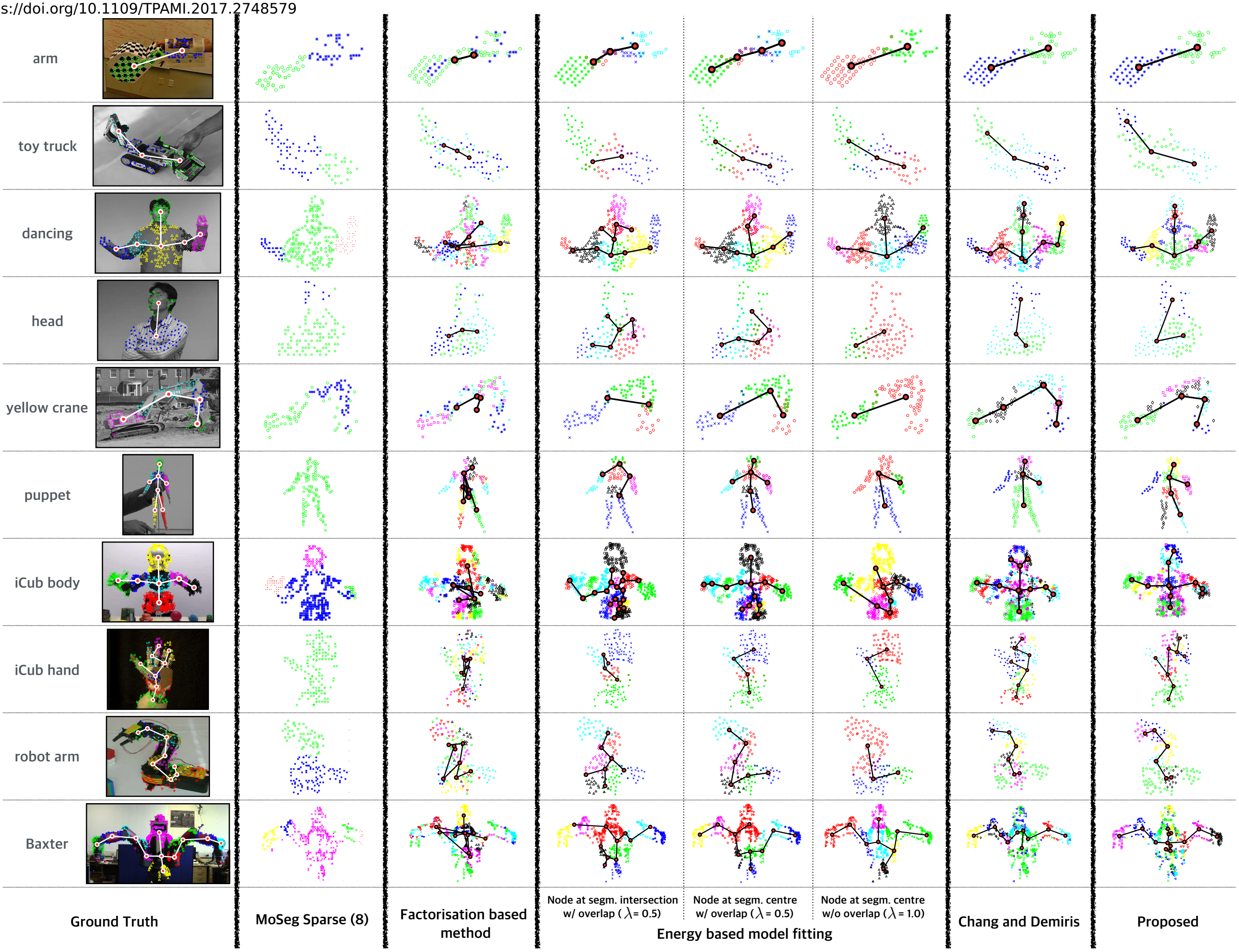


Fig. 12. Articulated kinematic structure estimation results on the conventional data sequences and our new sequences. The leftmost column shows original images and segmented feature points with the ground truth annotated kinematic structure. We can see that the proposed method estimates the most similar kinematic structures when compared to the ground truth in many sequences. Best viewed in colour.

[6] J. Yan and M. Pollefeys, "A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, no. 5, pp. 865–877, May 2008.

[7] B. Jacquet, R. Angst, and M. Pollefeys, "Articulated and restricted motion subspaces and their signatures," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 1506–1513.

[8] J. Fayad, C. Russell, and L. Agapito, "Automated articulated structure and 3D shape recovery from point correspondences," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 431–438.

[9] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 3328–3333.

[10] X. Huang, I. Walker, and S. Birchfield, "Occlusion-aware multi-view reconstruction of articulated objects for manipulation," *Robotics and Autonomous Systems*, vol. 62, pp. 497–505, 2014.

[11] M. R. Matthias Straka, Stefan Hauswiesner and H. Bischof, "Skeletal graph based human pose estimation in real-time," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2011, pp. 69.1–69.12.

[12] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 2353–2360.

[13] T. Boult and L. Brown, "Factorization-based segmentation of motions," in *Proc. IEEE Workshop Visual Motion*, 1991, pp. 179–186.

[14] J. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision (IJCV)*, vol. 29, no. 3, pp. 159–179, 1998.

[15] C. Gear, "Multibody grouping from motion images," *International Journal of Computer Vision (IJCV)*, vol. 29, no. 2, pp. 133–150, 1998.

[16] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision (IJCV)*, vol. 9, pp. 137–154, 1992.

[17] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1–8.

[18] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no. 12, pp. 1945–1959, 2005.

[19] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 11, pp. 2765–2781, 2013.

[20] H. Jung, J. Ju, and J. Kim, "Rigid motion segmentation using randomized voting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 1210–1217.

[21] P. Tresadern and I. Reid, "Articulated structure from motion by factorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, June 2005, pp. 1110–1115 vol. 2.

[22] J. Yan and M. Pollefeys, "A factorization-based approach to articulated motion recovery," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, June 2005, pp. 815–821 vol. 2.

[23] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 282–295.

[24] D. Ross, D. Tarlow, and R. Zemel, "Learning articulated structure and

motion," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, pp. 214–237, 2010.

[25] D. A. Ross, "Learning probabilistic models for visual motion," Ph.D. dissertation, University of Toronto, 2008.

[26] J. Sturm, C. Stachniss, and W. Burgard, "A probabilistic framework for learning kinematic models of articulated objects," *Journal on Artificial Intelligence Research*, vol. 41, pp. 477–626, August 2011.

[27] S. Pillai, M. Walter, and S. Teller, "Learning articulated motions from visual demonstration," in *Robotics: Science and Systems*, July 2014.

[28] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 5, pp. 898–916, 2011.

[29] A. Spoerri, "The early detection of motion boundaries," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences, 1991.

[30] M. Black and D. Fleet, "Probabilistic detection and tracking of motion discontinuities," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 1999, pp. 551–558 vol.1.

[31] H. Wang, A. Klser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision (IJCV)*, vol. 103, no. 1, pp. 60–79, 2013.

[32] D. Sun, S. Roth, and M. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision (IJCV)*, vol. 106, no. 2, pp. 115–137, 2014.

[33] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing (TIP)*, vol. 3, no. 5, pp. 625–638, Sep 1994.

[34] M. Unger, M. Werlberger, T. Pock, and H. Bischof, "Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1878–1885.

[35] P. Ochs and T. Brox, "Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions," in *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 1583–1590.

[36] P. Ochs, J. Malik, and T. Brox, "Higher order motion models and spectral clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 614–621.

[37] K. Fragkiadaki, G. Zhang, and J. Shi, "Video segmentation by tracing discontinuities in a trajectory embedding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[38] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 6, pp. 1187–1200, June 2014.

[39] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, December 2013, pp. 1841–1848.

[40] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Learning to detect motion boundaries," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[42] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, July 2001.

[43] Q.-A. Tran, X. Li, and H. Duan, "Efficient performance estimate for one-class support vector machine," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1174 – 1182, 2005.

[44] T. C. Landgrebe, D. M. Tax, P. Paclk, and R. P. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 908 – 917, 2006.

[45] Q. Guo, W. Li, Y. Liu, and D. Tong, "Predicting potential distributions of geographic events using one-class data: concepts and methods," *International Journal of Geographical Information Science*, vol. 25, pp. 1697–1715, 2011.

[46] P. J. Kim, "Fast incremental learning for one-class support vector classifiers," Ph.D. dissertation, Seoul National University, 2008.

[47] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[48] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transaction on Pattern Recognition and Machine Intelligence (TPAMI)*, vol. 19, no. 6, pp. 580–593, Jun 1997.

[49] G. Ben-Artzi, T. Halperin, M. Werman, and S. Peleg, "Two points fundamental matrix," *CoRR*, vol. abs/1604.04848, 2016. [Online]. Available: http://arxiv.org/abs/1604.04848

[50] H. Blum and R. N. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition*, vol. 10, no. 3, pp. 167 – 180, 1978.

[51] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[52] J. Ding, Y. Wang, and L. Yu, "Extraction of human body skeleton based on silhouette images," in *International Workshop on Education Technology and Computer Science*, March 2010, pp. 71–74.

[53] R. Strzodka and A. Telea, "Generalized distance transforms and skeletons in graphics hardware," in *IEEE TCVG Conference on Visualization*, 2004, pp. 221–230.

[54] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[55] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2001, pp. 409–415.

[56] S. R. Gunn, "Support vector machines for classification and regression," University of Southhampton, School of Electronics and Computer Science, Tech. Rep., 1988.

[57] P. J. Kim, H. J. Chang, and J. Y. Choi, "Fast incremental learning for one-class support vector classifier using sample margin information," in *International Conference on Pattern Recognition (ICPR)*, Dec 2008.

[58] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, pp. 620–630, May 1957.

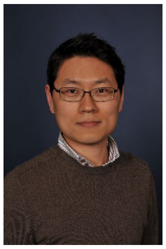
[59] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *The Annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, 2010.

[60] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, pp. 1–8.

[61] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[62] D. M. Tax, R. P. Duin, N. Cristianini, J. Shawe-taylor, and B. Williamson, "Uniform object generation for optimizing one-class classifiers," *Journal of Machine Learning Research (JMLR)*, vol. 2, pp. 155–173, 2001.

[63] J. Yan and M. Pollefeys, "Articulated motion segmentation using RANSAC with priors," in *Lecture Notes in Computer Science*, 2007, pp. 75–85.



Hyung Jin Chang received his B.S. and Ph.D. degree from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Republic of Korea. He is a post doctoral researcher with the Department of Electrical and Electronic Engineering at Imperial College London. His current research interests include articulated structure learning, human robot interaction, object tracking, human action understanding, and user modelling.

Yiannis Demiris (SM03) received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Artificial Intelligence, University of Edinburgh, Edinburgh, U.K. He is a Professor with the Department of Electrical and Electronic Engineering at Imperial College London, where he heads the Personal Robotics Laboratory. His current research interests include human robot interaction, machine learning, user modelling, and assistive robotics; he has published more than 130 journal and peer reviewed conference papers on these topics. Dr Demiris was the Chair of the IEEE International Conference on Development and Learning in 2007 and the Program Chair of the ACM/IEEE International Conference on HumanRobot Interaction in 2008. He was a recipient of the Rectors Award for Teaching Excellence in 2012 and the FoE Award for Excellence in Engineering Education in 2012. He is a senior member of the IEEE, and a Fellow of the IET, BCS and the Royal Statistical Society.