Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

Unsupervised Learning of Complex Articulated Kinematic Structures combining Motion and Skeleton Information

Hyung Jin Chang Yiannis Demiris
Department of Electrical and Electronic Engineering
Imperial College London, United Kingdom
{hj.chang, y.demiris}@imperial.ac.uk

Abstract

In this paper we present a novel framework for unsupervised kinematic structure learning of complex articulated objects from a single-view image sequence. In contrast to prior motion information based methods, which estimate relatively simple articulations, our method can generate arbitrarily complex kinematic structures with skeletal topology by a successive iterative merge process. The iterative merge process is guided by a skeleton distance function which is generated from a novel object boundary generation method from sparse points. Our main contributions can be summarised as follows: (i) Unsupervised complex articulated kinematic structure learning by combining motion and skeleton information. (ii) Iterative fine-to-coarse merging strategy for adaptive motion segmentation and structure smoothing. (iii) Skeleton estimation from sparse feature points. (iv) A new highly articulated object dataset containing multi-stage complexity with ground truth. Our experiments show that the proposed method out-performs state-of-the-art methods both quantitatively and qualitatively.

1. Introduction

Learning the underlying kinematic structure of articulated objects is an active research topic in computer vision and robotics. Accurate and efficient kinematic structure estimation is beneficial to many higher level tasks such as object kinematic recognition [25], human action recognition [2, 17], body scheme learning for robotic manipulators [24], articulated objects manipulation [9, 18], etc. In this paper we focus on the specific case of complex articulated kinematic structure learning using only 2D position of interest points tracked over time.

Many algorithms which recover an articulated structure from 2D tracking data have shown automatic detection of articulated motion types (*i.e.* folding, rotation and translation) [35, 10] and a kinematic chain building [19, 35, 5].

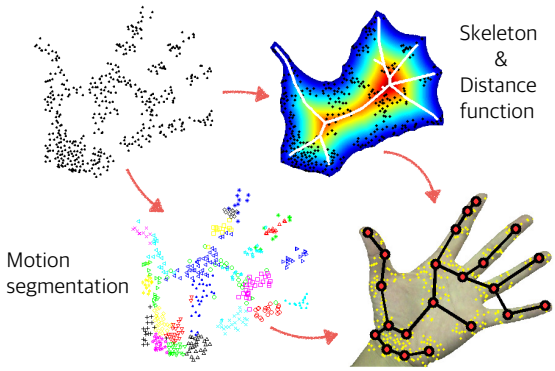


Figure 1. The proposed framework reliably learns the underlying kinematic structure of complex articulated objects from a combination of motion and skeleton information.

However they have been applied to relatively simple articulations only. Our target is to find a kinematic structure of arbitrary objects with articulated motion capabilities that range from simple structures to complex structures such as the human hand.

Furthermore, most of the existing kinematic structure generation methods [35, 5, 23, 9] use motion information only. Such techniques miss global refinement steps that enforce topological or kinematic constraints, and as such can produce highly implausible structures as output. On the other hand, articulated structure estimation methods from shape information [17, 36] have been presented. Normally, the estimated structure is a skeletal structure which represents the medial axis of body and implies topological properties, but such estimation methods cannot represent kinematic structures.

In this paper, we present a novel framework for complex articulated kinematic structure estimation from 2D feature points trajectories. We combine motion and skeleton information for generating elaborate and plausible kinematic structure (see Figure 1). We assume that an articulated ob-

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

ject is composed of a set of rigid segments and the structure represents the connections between segments. It is difficult to estimate a proper number of segments in advance when the articulation is complex and the input data is noisy. So we introduce a fine-to-coarse strategy which performs iterative merging and smoothing over segmented parts guided by skeletal topology and motion similarity. For 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 of the object, such as the number of motion segments and object category, and the learned structure can represent complex articulations using their skeletal topology. Our experiments show that the proposed method outperforms state-of-the-art methods quantitatively and qualitatively.

2. Related Work

Several approaches for the articulated structure generation of moving objects have been proposed. RGB-D sensors-based human/hand skeleton estimation methods have been successfully presented [21, 26]. However, the methods are designed for specific target skeletons featuring a computationally demanding pre-training step. The results are typically skeletons and not kinematic structures. Also RGB cameras are still more widely used for various applications, so it is necessary to develop a good algorithm for 2D sequences from a single view. Since our method relies on this type of input data, we will mainly discuss related work that uses 2D feature tracks only (not using depth information). Three main categories can be distinguished in the literature; motion segmentation and factorisation based approaches, probabilistic graphical model approaches and cost function based optimisation methods.

Motion segmentation and factorisation methods (proposed by [28, 3]) are perhaps the most popular for articulated reconstruction. Various methods for motion segmentation have been proposed such as subspace fitting (GPCA) [32], subspace clustering [4] and multiview-based approaches [6]. The GPCA [32] is widely used in papers for motion segmentation [33, 35], but it requires the number of motion segments in advance. Also it cannot be applied to more than a few subspaces as the number of required samples grows exponentially with the number of subspaces. Recently, Jung *et al.* [13] proposed a novel rigid motion segmentation algorithm based on the randomized voting (RV). They showed that it can achieve the state-of-the-art motion segmentation performance even under noisy environments within a reasonable time. However, it also requires an exact number of motion clusters as a prior for good performance, which makes it difficult to be applied to complex articulated videos.

Tresadern and Reid [30] and Yan and Pollefeys [33] developed the factorization method [28, 3] for articulated objects, showing that the rank of a matrix of feature tracks indicates the type of joint present. It is very effective to segment dependent motions but cannot deal with high degrees of articulations. Furthermore, Yan and Pollefeys [35] estimated a kinematic chain by modelling the articulated motion as a set of intersecting motion subspaces. The locations of the joints can be 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 consequently is sensitive to noise. There are also many tuning parameters in each step. Overall, this method is very difficult to apply to complex articulations. Jacquet *et al.* [10] presented a relative transformation analysis method based on linear subspaces, but it focused on detecting the type of articulated motion between two restricted motion parts.

Ross *et al.* [19] proposed probabilistic graphical model approaches to learn the articulated structure from 2D feature tracks. They could find the number of joints and their connections adaptively, but their method is sensitive to the prior and has difficulty recovering from a poor initial segmentation. Also it has difficulty escaping from local minima. Sturm *et al.* [24, 25] similarly used a probabilistic approach to learn kinematic joints especially for robot vision applications; body scheme learning and object manipulation. They required fiducial markers on each object part for noise-free input data and the number of motion segmentations had to be given as a prior. A markerless sparse feature tracking-based articulation learning was presented by Pillai *et al.* [18], which also did not require prior object models. However they required RGB-D feature data and could not handle concurrent articulated motions.

A single cost function based optimisation approach for simultaneous segmentation and 3D reconstruction was proposed by Fayad *et al.* [5]. No 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. However, by enforcing the overlap between segments, the resulting segments are smoothed such that complex structures are difficult to be estimated correctly.

3. Methodology

Our goal is to generate articulated kinematic structures via motion and skeleton information, whilst being accurate and plausible under complicated concurrent motions. To this end, we use only 2D trajectories for learning (assuming that one target subject exists in scene). To extract each rigid motion segment, we adopt the best performing motion segmentation method: randomized voting [13]. To estimate skeletal information from 2D sparse feature points, a one class data description method (support vector data descrip-

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

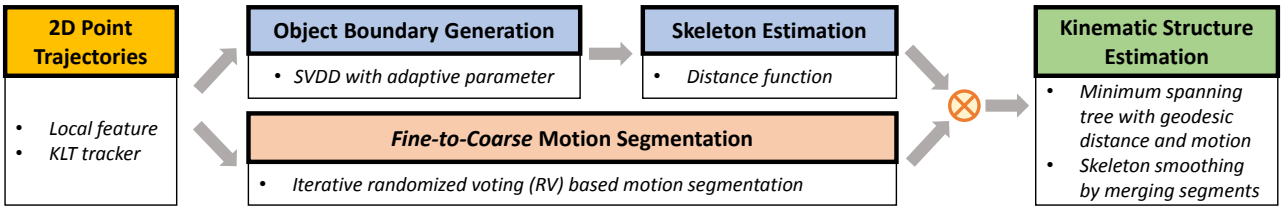


Figure 2. Overall flow of the proposed method.

tion [27]) is used for object silhouette generation, and for the skeleton extraction we utilise a distance function based contour-pruned skeletonisation[1].

The overall concept of the proposed framework is illustrated in Figure 2. In Section 3.1 we define the notations. In Section 3.2 we discuss how the adaptive motion segmentation is performed. Following this, in Section 3.3 we discuss how we generate an object boundary and a skeleton distance function from the sparse feature points. Finally, in Section 3.4, we discuss a kinematic structure generation and smoothing algorithm using the generated skeleton and motion information.

3.1. Notations

The 2D feature points are denoted as x_i where $i = 1, \ldots, N$. N is the total number of points. The point set X is defined as $X = \{x_1, x_2, \ldots, x_N\}$, with x_i represented in homogeneous coordinates. The trajectories are represented as x_i^f, with $f = 1, ..., F$ as sequence index and F as the number of frames in the input video. We are dealing with trajectory data, so we express the sequence index by a superscript. To express motion segments, we use S_k for the disjoint set of points belonging to the kth segment where $k = 1, ..., c$, and c as the total number of segments and $X = S_1 \cup S_2 \cup ... \cup S_c$, and y_k denotes a centre position of segment S_k obtained by averaging its points. We express an object region by Ω and its boundary as $\delta\Omega$. The terms *object boundary* and *silhouette* are used interchangeably.

3.2. Motion Segmentation

It is difficult to estimate the precise number of motion segments especially when the motions are highly articulated and the input data is noisy. In order to cope with these complicated cases, we present an iterative fine-to-coarse inference strategy adaptively estimates an upper-bound number of initial motion segmentation. We use the randomized voting (RV) method [13] which performs best up-to-now and is robust to noise for the motion segmentation, but requires the number of segments c in advance.

Since RV utilises an iterative fundamental matrix estimation, at least 8 points should be assigned to each segment in order to start the algorithm. Hence we estimate the initial number of segments as $\hat{c}^{init} = \lfloor N/8 \rfloor$. Even though every segment is assigned more than 8 points initially, there could be some segments having less than 8 points through the randomised voting procedure. Among the resultant segments, the number of segments with less than 8 points ($c_{<8}$) is counted, and the segment number is reset to $\hat{c}^{t+1} = \hat{c}^t - c_{<8}$. Then we perform the RV algorithm iteratively with the decreased segment number $\hat{c}^{t+1}$ until all segments have more than 8 points ($c_{<8} = 0$). The iterative fine-to-coarse segmentation procedure is described in Algorithm 1.

3.3. Skeleton from Sparse Feature Points

Using a skeleton as an abstraction of an object has major benefits; it can contain both essential shape features in a low-dimensional form and topological structures of an object. There have been numerous algorithms for skeleton estimation from a binary silhouette image of a target object. Relying on background subtraction, differential holistic features or human body detection techniques, the silhouette can be extracted out of RGB images. Unfortunately, these approaches are not suitable for producing a silhouette and a skeleton from sparse 2D feature points.

Algorithm 1 Fine-to-coarse Motion Segmentation

Input: $x_i, i = 1, ..., N$ ▷ Point trajectories
Output: $S_k, k = 1, ..., \hat{c}$

$t \leftarrow 1$
$\hat{c}^t \leftarrow \lfloor N/8 \rfloor$ ▷ Initialise the number of segments
repeat
  $S_k^t \leftarrow$ RV motion segmentation$(\{x_i\}_{i=1}^N, \hat{c}^t)$
  $c_{<8} \leftarrow 0$
  for $k = 1, ..., \hat{c}^t$ **do**
    if $|S_k^t| < 8$ **then**
      $c_{<8} \leftarrow c_{<8} + 1$
  $\hat{c}^{t+1} \leftarrow \hat{c}^t - c_{<8}$
  $t \leftarrow t + 1$
until $c_{<8} = 0$

[1]Code available at http://www.cs.smith.edu/~nhowe/research/code/

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

3.3.1 Object Boundary Generation

We now propose an adaptive object boundary ($\delta\Omega$) generation method from sparse feature points X^f based on support vector data description (SVDD) [27]. The SVDD tries to find a tight description covering all target data with minimising superfluous space. We consider the description boundary as the object boundary ($\delta\Omega$) of the points.

In order to formulate the covering description with minimum superfluous space, it is defined that the description shape is a sphere with minimum volume [27]. As a result it obtains a spherically shaped closed boundary (an *hypersphere*) enclosing all the target data. Analogous to SVM, the boundary can be made flexible by using kernel functions. The sphere is characterised by a centre $\mathbf{a}$ and radius $\mathbf{R} > 0$. The volume of the sphere is minimised by minimising $\mathbf{R}^2$. The objective function to minimise $\mathbf{R}^2$ with slack variable $\xi_i \geq 0$ and penalty parameter C is defined as:

$$F(\mathbf{R}, \mathbf{a}) = \mathbf{R}^2 + C \sum_i \xi_i \tag{1}$$

subject to the following constraints:

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

Analogous to SVM derivation, Equation (1) and Equation (2) can be combined by introducing Lagrange multipliers $\alpha_i \geq 0$;

$$\begin{aligned} L &= \sum_i \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \\ & s.t. \, 0 \leq \alpha_i \leq C. \end{aligned} \tag{3}$$

Furthermore, the non-linear boundary can be found by replacing the inner product $(x_i \cdot x_j)$ with a kernel function $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ (where Φ is an implicit mapping of the data into high dimensional feature space, and we indicate a kernel parameter as σ). We use an exponential radial basis function kernel $K(x_i, x_j) = exp(-\|x_i - x_j\|/2\sigma^2)$ which produces a tight piecewise linear solution [7].

However, unlike the two-class SVM, it is difficult to select an optimal kernel parameter σ which controls boundary tightness since there are no outliers in the data. Figure 3 shows generated boundaries with different kernel parameters. There have been several approaches to solve the kernel parameter selection problem. However, theoretical analysis approaches [20] give too loose bounds, and a heuristic approach [29] with genetic algorithm takes too much time. In the artificial outlier generation methods [16, 8], generating good outliers is an issue.

In this work, we introduce a novel optimal kernel parameter selection method using *sample margins* [15, 14]. The sample margin is a distance from a datum to a hyperplane passing through the centre of hypersphere in a kernel space [14]. Sample margins reflect the distribution of images of data in the kernel space and can be calculated by $\gamma(x_i) = \dfrac{\mathbf{a} \cdot \Phi(x_i)}{\|\mathbf{a}\|}$ for each data point x_i ($0 \leq \gamma(x_i) \leq 1, \forall i$), where $\mathbf{a} = \sum_i \alpha_i \Phi(x_i)$. Each sample margin reflects a normalised relative position between the centre and boundary of the hypersphere, so different kernel parameters give different sample margin distributions as well as different description boundary as shown in Figure 3.

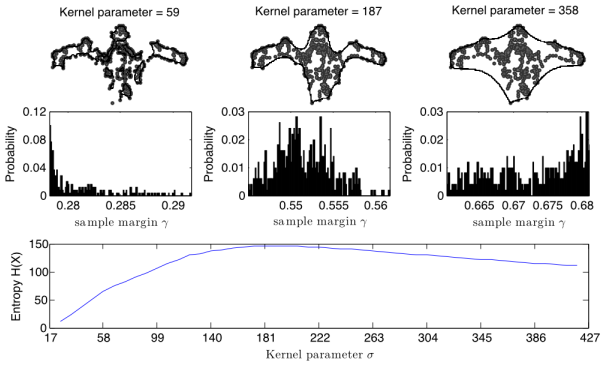


Figure 3. Object boundary generation results with various kernel parameters. A small parameter value produces over-estimated results with separated boundary regions and a large value gives an under-estimated boundary result. The distributions of the sample margin γ are shown in the middle respect to each kernel value. As we can see the most proper boundary comes from the kernel value which gives the maximum entropy.

In this paper we propose a new criterion for the kernel parameter selection by calculating the entropy of the sample margin distribution. If the description is overfitted, the sample margins are distributed toward the boundary of the hypersphere. If the boundary is underfitted, the distribution is biased to the centre. By finding a kernel parameter of the maximum entropy (*i.e.*, evenly spread), we avoid over/underfitting. Furthermore, according to the principle of maximum entropy [12], if no prior knowledge is available about a distribution, then the probability distribution with the largest entropy best represents the current state of knowledge. So the optimal kernel parameter $\hat{\sigma}^f$ of current frame feature points X^f can be estimated by

$$\begin{aligned} \hat{\sigma}^f &= \arg\max_{\sigma} H(\gamma(X^f)) \\ &= \arg\max_{\sigma} \sum_i -p_i log(p_i), \end{aligned} \tag{4}$$

where H is the entropy and p_i is a probability distribution with $p_i = Pr(\gamma(x_i^f))$.

The object boundary $\delta\Omega$ can be considered as a set of all the points of image space $\mathcal{I}$ lying on the same distance to the centre $\mathbf{a}$ of the hypersphere as the radius $\mathbf{R}$. It can be

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

generated with the selected optimal kernel parameter $\hat{\sigma}$ by

$$\delta\Omega = \{q|\forall q \in \mathcal{I}, \|q - \mathbf{a}\|^2 = 1 - 2\sum_i \alpha_i K(q, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \mathbf{R}^2\}. \quad (5)$$

In order to measure goodness of the generated boundary by the new criterion quantitatively, we follow [16] that generate uniformly distributed outliers around target data. Using the outliers, a loss function balancing classification and rejection performance is used as a measure of good fit. As a result, we find similar descriptions as [16] but $3.2 \sim 7.8$ faster.

3.3.2 Skeleton Distance Function Generation

A skeleton of an object, $\Upsilon(\Omega)$, is defined as the set of all centre points of maximal circles contained in an object Ω, which is a medial axis of an object [1]. It can be formulated as the locus of points at equal distance from at least two boundary point as [22]:

$$\Upsilon(\Omega) = \{p \in \Omega | \exists q, r \in \delta\Omega, q \neq r : dist(p, q) = dist(p, r)\}. \quad (6)$$

The skeleton contains both shape features and topological structures of the original objects. As a good representation of the skeleton, a distance transform [11] is defined as a function that returns the closest distance to the boundary for each point p. Using the obtained object boundary, the distance function ($\Psi(p)$) of Ω is defined as [22]:

$$\Psi(p) = \min_{q \in \delta\Omega}(dist(p, q)) \quad (7)$$

for all points $p \in \Omega$. 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.

3.4. Kinematic Structure Estimation and Structure Smoothing

In this section we present how to generate the kinematic structure of an articulated object using the motion segments (S) and skeleton distance function (Ψ) results. We assume that the kinematic structure is not cyclic as [35], which covers most articulated objects. We utilise a graphical model $G = (V, E)$ to determine the topological connections between motion segments. All the motion segment centres $y_1, ..., y_{\hat{c}}$ are treated as nodes V in a complete graph. The proximity $E(y_k, y_l)$ between segment y_k and y_l is defined as follows:

$$E(y_k, y_l) = \underset{f \in F}{\text{median}}\{(\zeta(y_k^f - y_l^f; \Psi^f) \times \|\dot{y}_k^f - \dot{y}_l^f\|\} \quad (8)$$

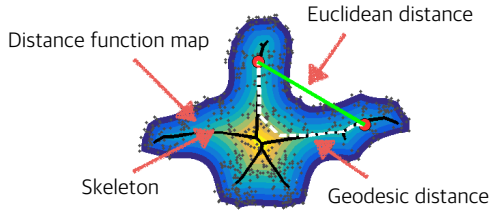


Figure 4. Geodesic distance and Euclidean distance between two points. The white dotted line is the geodesic distance ζ and the green solid line is Euclidean distance. The black solid line is a skeleton of the object. The geodesic distance represents the minimum distance following the skeleton.

which is a combination of geodesic distance in skeleton distance transform and moving velocity difference. For the final proximity estimation between two segments over all frames, we take the median value in order to be robust to outliers.

Given the distance function Ψ, a geodesic distance between two points $\mathbf{p}$ and $\mathbf{q}$ is defined as follows:

$$\zeta(\mathbf{p}, \mathbf{q}; \Psi^f) = \min_{\Gamma \in \mathcal{P}_{\mathbf{p},\mathbf{q}}} \sum_{n=1}^{l(\Gamma)} \frac{1}{\Psi^f(p_n)} \quad (9)$$

where Γ is a path connecting the two points and $\mathcal{P}_{\mathbf{p},\mathbf{q}}$ is the set of all possible paths. Thus the Equation (9) 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 velocity. 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.

However, the initially generated structure is highly contorted, because many small motion segments deviate from the median axes. So we further perform structure smoothing by an iterative merging procedure guided by the skeleton distance function. If a segment S_k largely deviates from the medial axis, then the $\Psi(y_k)$ is small (*i.e.* $\Psi(y_k) < \tau$). We set the threshold τ as the minimum distance function value of the skeleton Υ; $\tau = \min \Psi(\Upsilon)$. The deviated S_k is merged to a connected neighbour segment having larger Ψ value ($\hat{c} = \hat{c} - 1$), and then we reconstruct the structure until all the segment centres are located close to the skeleton (see Figure 5).

4. Experiments

Dataset The proposed method has been evaluated with well-known sequences such as 'arm' [31], 'toy' and 'dance' [35] [2], but the conventional data sequences are rel-

[2]Note that the same dataset as [35, 5] of 'toy' and 'dance' are not available, so we extracted the feature points and their trajectories from the

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

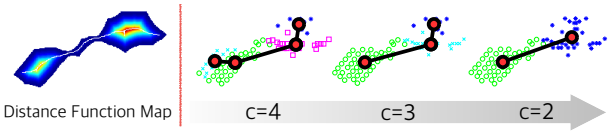


Figure 5. Structure smoothing by iterative segments merging.

atively simple. So we introduce new challenging sequences which are composed of complex articulated motions [3]. We have tried to avoid severely occluding motions because the 2D feature point trackers cannot keep tracking under occlusion. However, the new sequences still contain diverse complex motions such as articulations, concurrency, rotating, affine and scaling. We summarised the dataset properties in Table 1. Because there are many motions within a frame, we extracted feature points densely not to miss subtle movements. We have manually labelled each motion segments for ground truth.

4.1. Self Comparison

We have performed various experiments to validate the proposed framework. Our method is based on randomised voting, so the results are not exactly the same across different trials. All the experimental statistics are obtained from one hundred trials.

In order to evaluate the performance quantitatively, we design the error measurement as:

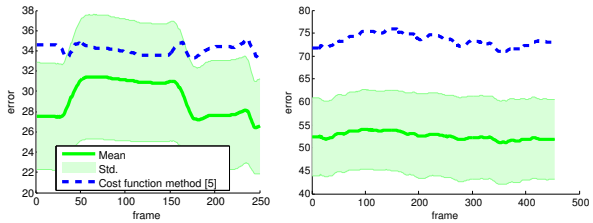
$$error = \frac{1}{\hat{c} \cdot F} \sum_{\substack{k=1 \\ f=1}}^{\hat{c},F} \left(\min_{g=1 \ldots c_{GT}} \|y_k^f - \mathbf{y}_g^f\| \right) \times \left(1 + \frac{|\hat{c} - c_{GT}|}{c_{GT}} \right), \quad (10)$$

where c_{GT} and $\mathbf{y_g}$ indicate the number of ground truth segments and their centres respectively. With this measure, we can consider structural complexity differences as well as spatial deviation of each segment.

Firstly, we validate whether the proposed fine-to-coarse iterative merging process can find the correct number of segments. As shown in Figure 6, as the iteration proceeds the resultant segments number converges closely to the ground truth value. Furthermore, we have also measured the error changes over frames. Through these experiments, we can test the kinematic accuracy of the estimated segments. As we can see in Figure 7, our method finds more accurate kinematic points than the other method [5].

presented result videos. That is why the points locations and results are different from [35, 5].

[3] We utilised two robots; iCub (http://www.icub.org) and Baxter (http://www.rethinkrobotics.com/baxter/)



(a) iCub body (b) Baxter

Figure 7. The error level comparisons across frames.

4.2. Comparisons with State-of-the-art

We have compared the proposed method with state-of-the-art methods. We implemented the RANSAC based method [34], and the factorisation based method [35] which is one of the most compared. The third method we have compared to is the cost function based optimisation approach [5], which is the best performing method up-to-now. All the methods were implemented as described in their respective papers using the noted toolboxes.

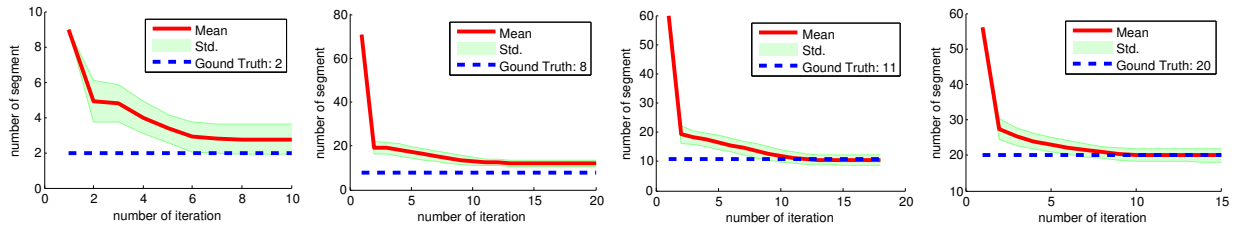
In order to get reasonable results of [34, 35], we manually tuned some parameters for each data sequence such as the number of motion segments and rank detection parameter. [5] finds the structure nodes by averaging the intersection points of two rigid segments and connects them. However, more detailed description about end nodes (having no overlaps) and the connection procedure is not mentioned. So we manually select the end nodes and apply the minimum spanning tree method for connection. For comparison, we also show the structures without manual intervention for the end joints in Figure 9.

We would like to emphasise that we did not particularly tune any parameters of the algorithm for any specific sequence from roughly defined initial values. All the comparisons are obtained through the fully adaptive approach of

Table 1. Properties of the dataset. The newly introduced datasets are more challenging because they are composed of concurrent and highly articulated motions.

Dataset	# of seg.	# of points	# of frames	motion concur.
arm	2	77	30	no
toy	3	93	69	no
dance	6	236	60	yes
robot arm	8	144	737	yes
iCub body	7	573	250	yes
iCub hand	8	154	280	yes
Baxter	11	484	454	yes
human hand	20	450	634	yes

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE



(a) arm (b) iCub body (c) Baxter (d) human hand

Figure 6. The number of motion segments converges close to the ground truth through the proposed iterative merging process.

the algorithm under fixed parameters (no manual intervention is required).

In Table 2 and Figure 8 we show the average of the joint error. Our approach achieves comparable low average error for simple articulations, and our method largely outperforms the other state-of-the-art methods throughout the complex articulated motion sequences. Additionally, in Figure 9 we present some qualitative results.

The RANSAC based method [34] and the factorisation based method [35] are very sensitive to noise and parameter setting, and noise effect increases with complex motions. In [5], the cost function balances overall model complexity and local motion errors, performing well when the structure is simple. However, if the motion complexity increases, it finds a moderate structure than an actual detailed structure. Moreover, the cost function enforces overlaps between related motion segments such that the output becomes far from what a human would normally estimate. Our fine-to-coarse procedure finds detailed structures, and the skeleton information reduces noise effect. So the learned structures are more elaborate and plausible.

Furthermore, our method runs 1.8 times faster than [5] on average (It takes 93.0±6.5 versus 180.2 seconds for the 'iCub hand' sequence). Note that here both our method and [5] are implemented in Matlab, unoptimised single threaded, without any CPU/GPU parallelism.

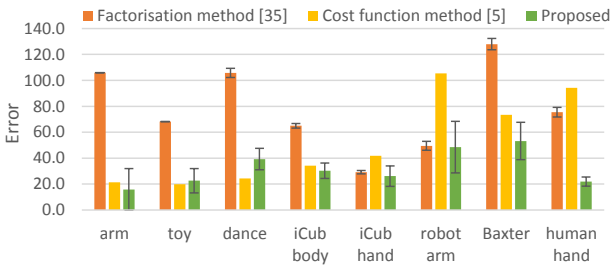


Figure 8. Quantitative error comparison graph.

5. Conclusion and Future Works

In this paper we have introduced a novel articulated kinematic structure estimation framework which can represent complex structures plausibly. We have demonstrated that the challenges can be efficiently met via the adoption of a state-of-the-art motion segmentation into iterative merging process guided by skeleton information. We employ a fine-to-coarse agglomerative merge scheme, *i.e.* we start with over-segmented motion segments. An object silhouette generation using sparse feature points is proposed and skeleton distance function is generated with the silhouette. In turn, during the structure learning, we made use of the motion segments and the skeleton distance function to build a connection tree by considering motion similarity and topology. Our method is evaluated using both public datasets and our

Table 2. Estimated joints accuracy comparison with state-of-the-arts methods. All the values are from one hundred trials except [5] as it gives a consistent results from optimisation method. The above number is a mean value and the number in parenthesis are standard deviation.

Dataset	RANSAC method [34]	Factorisation method [35]	Cost function method [5]	Proposed
arm	561.2 (176.4)	105.8 (0.0)	**21.5**	**15.7** (16.2)
toy	2357.0 (0.0)	68.2 (0.0)	**20.0**	22.6 (9.4)
dance	3041.1 (320.9)	105.8 (3.5)	**24.3**	39.3 (8.2)
iCub body	6357.4 (1482.6)	65.0 (1.8)	**34.1**	**30.4** (5.9)
iCub hand	975.6 (0.0)	29.2 (1.3)	41.8	**26.2** (7.9)
robot arm	1305.4 (120.1)	49.5 (3.5)	105.3	**48.5** (19.9)
Baxter	6606.9 (1108.7)	127.9 (4.3)	73.4	**53.2** (14.4)
human hand	3127.2 (226.2)	75.5 (3.7)	94.1	**21.9** (3.5)

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

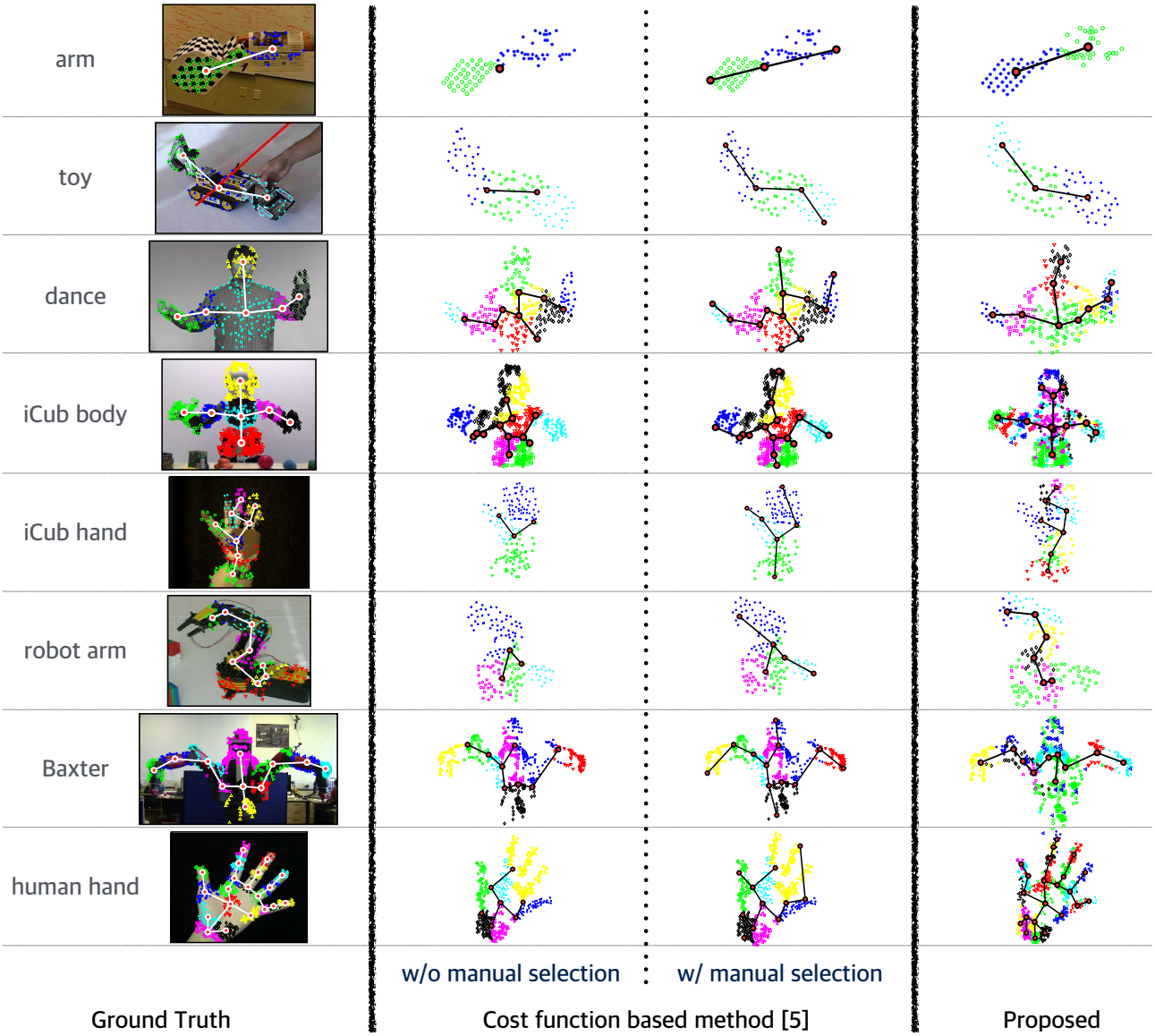


Figure 9. The cost function based method [5] requires manual selection of the end nodes. The learned structures by the proposed method are more elaborate and plausible.

new challenging complex articulated motion dataset. Previous work needed manual interventions (*e.g.*, number of segments, end joint positions), while we could find motion parts and skeletons adaptively without tuning parameters. As a result, apart from accurate motion joint detection results we can obtain a highly plausible representative structures facilitating further tasks such as object manipulation, object recognition, or robot body scheme understanding to name a few. The proposed method has a limitation in handling occlusions because the 2D feature points fail tracking when occlusion occurs. As a future work, we plan to use RGB-D camera for feature tracking, object separation and 3D silhouette generation. Also the proposed silhouette generation method can be used as a prior of object segmentation.

Acknowledgement: This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139.

References

[1] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167 – 180, 1978. 5

Preprint version; final version available at http://ieeexplore.ieee.org
CVPR (2015), pp: 3138-3146
Published by: IEEE

[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 *ICCV workshop*, 2013. 1

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998. 2

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 2

[5] J. Fayad, C. Russell, and L. Agapito. Automated articulated structure and 3D shape recovery from point correspondences. In *ICCV*, 2011. 1, 2, 5, 6, 7, 8

[6] F. Flores-Mangas and A. Jepson. Fast rigid motion segmentation via incrementally-complex local models. In *CVPR*, 2013. 2

[7] S. R. Gunn. Support vector machines for classification and regression. Technical report, University of Southhampton, School of Electronics and Computer Science, 1988. 4

[8] 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*, pages 1697–1715, 2011. 4

[9] X. Huang, I. Walker, and S. Birchfield. Occlusion-aware multi-view reconstruction of articulated objects for manipulation. *Robotics and Autonomous Systems*, 62:497–505, 2014. 1

[10] B. Jacquet, R. Angst, and M. Pollefeys. Articulated and restricted motion subspaces and their signatures. In *CVPR*, 2013. 1, 2

[11] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989. 5

[12] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957. 4

[13] H. Jung, J. Ju, and J. Kim. Rigid motion segmentation using randomized voting. In *CVPR*, 2014. 2, 3

[14] P. J. Kim. *Fast incremental learning for one-class support vector classifiers*. PhD thesis, Seoul National University, 2008. 4

[15] P. J. Kim, H. J. Chang, and J. Y. Choi. Fast incremental learning for one-class support vector classifier using sample margin information. In *ICPR*, 2008. 4

[16] 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*, 27(8):908 – 917, 2006. 4, 5

[17] M. R. Matthias Straka, Stefan Hauswiesner and H. Bischof. Skeletal graph based human pose estimation in real-time. In *BMVC*, 2011. 1

[18] S. Pillai, M. Walter, and S. Teller. Learning articulated motions from visual demonstration. In *Proceedings of Robotics: Science and Systems*, 2014. 1, 2

[19] D. Ross, D. Tarlow, and R. Zemel. Learning articulated structure and motion. *International Journal of Computer Vision*, 88(2):214–237, 2010. 1, 2

[20] 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*, 13(7):1443–1471, July 2001. 4

[21] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013. 2

[22] R. Strzodka and A. Telea. Generalized distance transforms and skeletons in graphics hardware. In *IEEE TCVG Conference on Visualization*, pages 221–230, 2004. 5

[23] J. Sturm, C. Plagemann, and W. Burgard. Unsupervised body scheme learning through self-perception. In *IEEE International Conference on Robotics and Automation*, pages 3328–3333, May 2008. 1

[24] J. Sturm, C. Plagemann, and W. Burgard. Body schema learning for robotic manipulators from visual self-perception. *Journal of Physiology-Paris*, 103(35):220 – 231, 2009. 1, 2

[25] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal on Artificial Intelligence Research (JAIR)*, 41:477–626, August 2011. 1, 2

[26] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*, 2014. 2

[27] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, Jan. 2004. 3, 4

[28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992. 2

[29] Q.-A. Tran, X. Li, and H. Duan. Efficient performance estimate for one-class support vector machine. *Pattern Recognition Letters*, 26(8):1174 – 1182, 2005. 4

[30] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *CVPR*, 2005. 2

[31] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *CVPR*, 2007. 5

[32] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005. 2

[33] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *CVPR*, 2005. 2

[34] J. Yan and M. Pollefeys. Articulated motion segmentation using ransac with priors. In *Lecture Notes in Computer Science*, 2007. 6, 7

[35] 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*, 30(5):865–877, May 2008. 1, 2, 5, 6, 7

[36] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*, 2014. 1