Unsupervised Articulated Skeleton Extraction from Point Set Sequences Captured by a Single Depth Camera

Xuequan Lu,[1] Honghua Chen,[2] Sai-Kit Yeung,[1] Zhigang Deng,[3] Wenzhi Chen[4]
[1]Singapore University of Technology and Design, [2]Nanjing Normal University, [3]University of Houston, [4]Zhejiang University
luxuequan@yeah.net, chenhonghuacn@gmail.com, saikit@sutd.edu.sg, zdeng4@uh.edu, chenwz@zju.edu.cn

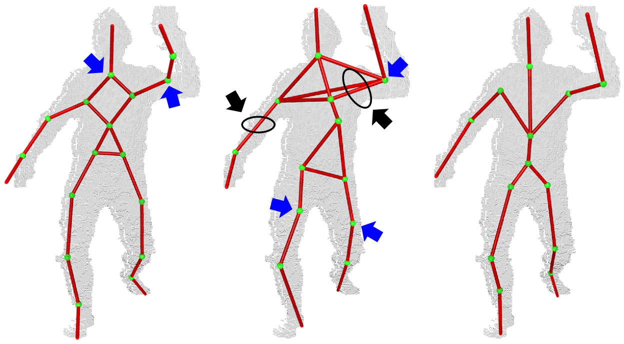
Abstract

How to robustly and accurately extract articulated skeletons from point set sequences captured by a single consumer-grade depth camera still remains to be an unresolved challenge to date. To address this issue, we propose a novel, unsupervised approach consisting of three contributions (steps): (i) a non-rigid point set registration algorithm to first build one-to-one point correspondences among the frames of a sequence; (ii) a skeletal structure extraction algorithm to generate a skeleton with reasonable numbers of joints and bones; (iii) a skeleton joints estimation algorithm to achieve accurate joints. At the end, our method can produce a quality articulated skeleton from a single 3D point sequence corrupted with noise and outliers. The experimental results show that our approach soundly outperforms state of the art techniques, in terms of both visual quality and accuracy.

1 Introduction

Automatic skeleton extraction from articulated objects is a fundamental yet unresolved research problem, despite its widely-documented applications in robotics, vision, and graphics, including learning from demonstration (Chalodhorn et al. 2007), human-robot interaction (Chrungoo, Manimaran, and Ravindran 2014), action recognition (Song et al. 2017), skeleton tracking (Zhou et al. 2016; Ye and Yang 2014), and computer animation (Le and Deng 2014). With the increasing availability of consumer-grade depth cameras, 3D point motion sequences can be easily captured by one single off-the-shelf depth sensor. However, the collected motion data has the following characteristics: noisy, incomplete, and the lacking of one-to-one point correspondences among frames. As a result, *without any prior knowledge on the captured objects*, how to robustly and accurately extract an articulated skeleton from such a point set sequence remains to be an unresolved research challenge to date.

Most of existing skeleton extraction approaches have been focused on either images/video (Tresadern and Reid 2005; Ramanan, Forsyth, and Barnard 2006; Yan and Pollefeys 2008; Ross, Tarlow, and Zemel 2010; Chang and Demiris 2015) or 3D motion (Chun, Jenkins, and Mataric 2003; Cheung, Baker, and Kanade 2003; Kirk, O'Brien, and Forsyth 2005; Schaefer and Yuksel 2007; Le and Deng 2014). By contrast, so far relatively few methods, except (Kirk, O'Brien, and Forsyth 2005; Zhang et al. 2013), have been proposed to extract skeletons directly from point set sequences without any prior knowledge on the captured objects. However, the method (Zhang et al. 2013) has the following major caveats: (1) the matching accuracy is limited due to several-to-one matching; and (2) a joint is simply selected from either of two neighboring body segments. The approach (Kirk, O'Brien, and Forsyth 2005) can extract 3D skeletons from motion capture data but it largely relies on the *high quality* marker input. They both do not consider either joint constraints or mixed bone-point impacts (e.g., Linear Blend Skinning (Magnenat-Thalmann, Laperri'ere, and Thalmann 1988)) when solving joint locations between two body parts. Also, they require nontrivial parameter tuning, in particular the number of segment clusters, to achieve reasonable numbers of joints and bones. As the result, these state of the art methods are limited in terms of both accuracy and robustness. Fig. 1 shows a comparison example.



(a) (Kirk, O'Brien, and Forsyth 2005) (b) (Zhang et al. 2013) (c) Ours

Figure 1: An example of the extracted skeletons by state of the art techniques and our method. Blue and black arrows indicate inaccurate and missing joints, respectively.

To tackle this challenge, we propose a novel unsupervised approach that is robust and accurate. Building point correspondences is a necessary initial step for almost all related methods since it makes motion-based part clustering feasible. Thus, as illustrated in Fig. 2, we first build one-to-one point correspondences among frames through non-rigid registration, and then extract a skeletal structure from the *new* sequence output by the first step, and finally use both the original input and the registered point sets to achieve accurate joints. The visual comparisons and quantitative evaluations on publicly available datasets and the Kinect data captured by ourselves show that our method significantly outperforms the state of the art methods (Kirk, O'Brien, and Forsyth 2005; Zhang et al. 2013), in terms of both quality and accuracy. The poses of the skeletons extracted by our method are even comparable to those by the supervised pose estimation (tracking) method–KinectSDK (Microsoft 2017).

2 Related Work

We only review previous research on skeleton extraction that is most related to our work. Interested readers are referred to (Tam et al. 2013) for a review on point set registration.

Skeleton extraction from a static model. Researchers have proposed various approaches to extract skeletons from a single static 2D or 3D model (Attali and Lachaud 2001; Au et al. 2008; Huang et al. 2013). However, the extracted skeletons are theoretically the medial axes of a single shape and can hardly be applied to other applications, as motion-related cues are not provided.

Skeleton extraction from images/video. Many image-based techniques have also been proposed to deal with skeletons. Some methods (Tresadern and Reid 2005; Ramanan, Forsyth, and Barnard 2006; Yan and Pollefeys 2008; Ross, Tarlow, and Zemel 2010; Chang and Demiris 2015) extract skeletons from images or video data. Nevertheless, such methods suffer greatly from the quality of feature points, illumination variations, or other factors.

Skeleton extraction from 3D motion. Some research efforts have been focused on extracting skeletons from 3D motion. Chun et al. (2003) use the generated underlying nonlinear axes from each frame to derive a kinematic model, based on a volumetric sequence captured by multiple cameras. Their method does not track points among frames, thus making the tracking or identification difficult. A Shape-from-Silhouette algorithm for articulated objects was proposed to recover the motion, shape, and joints from silhouette and color images (Cheung, Baker, and Kanade 2003). However, this method models the joints one by one, by moving one body part while keeping the rest of the body fixed. Kirk et al. (2005) extracted skeletons from marker-based MoCap data collected by multiple cameras. Recently, Zhang et al. (2013) proposed a method to extract skeletons from 3D point set sequences acquired by the Kinect device (Microsoft 2017), through deformable matching among different frames. Other previous works (Schaefer and Yuksel 2007; Le and Deng 2014) were introduced to extract skeletons from mesh sequences. Since the input is often high-quality (i.e., accurate vertex correspondences and nearly zero noise), quality skeletons can be usually generated.

3 Approach Overview

Fig. 2 shows an overview of the proposed approach. It consists of three steps (see last paragraph in Sec. 1): non-rigid point set registration (Sec. 4), skeletal structure extraction (Sec. 5) and skeleton joints estimation (Sec. 6).

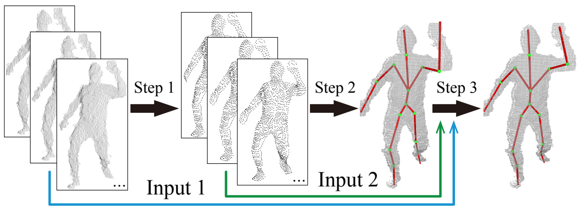


Figure 2: Overview of our approach. Input 1 and 2 indicate the original and registered point set sequences, respectively. See Fig. 5 to clearly observe the inaccurate joints of the output of Step 2.

Let $\mathbf{V}^t = \{\mathbf{v}_i^t\}$ and N^t be the positions and the number of the original points in frame t. Let $\mathbf{Y}^t = \{\mathbf{y}_m^t\}$ be the registered points at frame t. F is the number of frames. The number of points in $\mathbf{Y}^t$ is denoted as M. The data dimension, D, is 3. $\mathbf{V}^t$ and $\mathbf{Y}^t$ are $D \times N^t$ and $D \times M$ matrices, respectively.

4 Non-rigid Point Set Registration

In this section, we first formulate the non-rigid point set registration problem under a probabilistic framework. We then show how to optimize this problem by employing an Expectation-Maximization (EM) algorithm. We finally introduce new constraints and present an effective minimization scheme for the M-step.

4.1 The Probabilistic Model

Ideally, the registered points $\mathbf{Y}^t$ should approximate the original point set ($\mathbf{V}^t$) surface. To achieve this, we assume the points $\mathbf{V}^t$ follow a Gaussian Mixture Model (GMM) that takes the points $\mathbf{Y}^t$ as centroids. For simplicity, we omit the frame number (i.e., t) for the variables in this section. Then the probability of each point $\mathbf{v}_i$ is

$$p(\mathbf{v}_i) = (1-\omega)\sum_{m=1}^{M} p(\mathbf{y}'_m)p(\mathbf{v}_i|\mathbf{y}'_m) + \omega\frac{1}{N}, \tag{1}$$

where $p(\mathbf{v}_i|\mathbf{y}'_m) = \frac{1}{(2\pi\sigma^2)^{D/2}} e^{\frac{-\|\mathbf{v}_i-\mathbf{y}'_m\|^2}{2\sigma^2}}$. The uniform distribution $\frac{1}{N}$ (with its corresponding weight ω) is added to account for noise and outliers. We use the same covariance σ^2 and probability $p(\mathbf{y}'_m) = \frac{1}{M}$ for all the Guassians, as suggested by Myronenko and Song (2010).

Without prior knowledge on captured objects, we thus assume $\mathbf{y}'_m$ follows the general embedded deformation model, which supports to reconstruct unknown complex material behavior and facilitates the registration (Li et al. 2009).

$$\mathbf{y}'_m = \sum_{\mathbf{n}_j} \bar{\omega}(\mathbf{y}_m, \mathbf{n}_j)[\mathbf{R}_j(\mathbf{y}_m - \mathbf{n}_j) + \mathbf{n}_j + \mathbf{T}_j], \tag{2}$$

$\mathbf{y}'_m$ is the new position induced by its neighboring nodes $\mathbf{n}_j$ with different weights $\bar{\omega}(\mathbf{y}_m, \mathbf{n}_j)$. $\mathbf{R}_j$ and $\mathbf{T}_j$ are the rotation ($D \times D$ matrix) and translation ($D \times 1$ vector) of node $\mathbf{n}_j$. Refer to the work (Li et al. 2009) for more details. Under the above model assumptions, the non-rigid registration problem in our work can be cast as a parameter estimation problem. To estimate the parameters (i.e., $\{\mathbf{R}_j\}$ and $\{\mathbf{T}_j\}$), we need to minimize the following negative log-likelihood function: $E(\{\mathbf{R}_j\},\{\mathbf{T}_j\},\sigma^2) = -\log\prod_{i=1}^{N} p(\mathbf{v}_i)$.

4.2 EM Optimization

The Expectation-Maximization algorithm (Dempster, Laird, and Rubin 1977) is utilized for optimization. Based on the Bayes' rule, the E-step is to calculate posterior probabilities using the old values $\mathbf{Y}$ and σ^2. Given the posterior probabilities, the M-step is to estimate the involved parameters ($\{\mathbf{R}_j\}$, $\{\mathbf{T}_j\}$ and σ^2) by minimizing the expectation of the complete negative log-likelihood function (Bishop 1995).

E-step. We use the "old" parameters to compute the posterior probabilities $p^{old}(\mathbf{y}'_m|\mathbf{v}_i)$ based on the Bayes' theorem.

$$p^{old}(\mathbf{y}'_m|\mathbf{v}_i) = \frac{e^{\frac{-\|\mathbf{v}_i-\mathbf{y}_m\|^2}{2\sigma^2}}}{\sum_{m=1}^{M} e^{\frac{-\|\mathbf{v}_i-\mathbf{y}_m\|^2}{2\sigma^2}} + \frac{(2\pi\sigma^2)^{\frac{D}{2}}\,\omega M}{(1-\omega)N}}. \tag{3}$$

M-step. To estimate the involved parameters ($\{\mathbf{R}_j\}$, $\{\mathbf{T}_j\}$ and σ^2) we minimize the upper bound of E, which is: $E_{\text{GMM}} = \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}\|\mathbf{v}_i - \mathbf{y}'_m\|^2 + \frac{DN_p}{2}\log\sigma^2$, where $p_{mi} = p^{old}(\mathbf{y}'_m|\mathbf{v}_i)$, $N_p = \sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}$ and $\mathbf{P} = \{p_{mi}\}$.

4.3 Other Constraints and Minimization

At the M-step, we also introduce other constraints (soft and hard), to meet different demands during registration. Specifically, inspired by the work (Li et al. 2009), a smooth term E_{smooth} is introduced to encourage the transformation of a node to be close to its neighbors. To better regularize the solution space, we assume small motion changes for each node at each iteration and thus define: $E_{\text{small}} = \sum_j \|\mathbf{R}_j - \mathbf{R}_j^{pre}\|_F^2 + \|\mathbf{T}_j - \mathbf{T}_j^{pre}\|^2$, where $\mathbf{R}_j^{pre}$ and $\mathbf{T}_j^{pre}$ are solved at the previous iteration.

Besides, we impose a hard constraint to restrict $\mathbf{R}_j$ to be in SO(3). Based on all the terms and the SO(3) constraint, the final objective function for the M-step is therefore:

$$E = E_{\text{GMM}} + \frac{\beta_{\text{smooth}}}{2} E_{\text{smooth}} + \frac{\beta_{\text{small}}}{2} E_{\text{small}} \tag{4a}$$

$$\text{s.t. } \mathbf{R}_j{}^T\mathbf{R}_j = \mathbf{I}, det(\mathbf{R}_j) = 1, \forall j. \tag{4b}$$

β_{smooth} and β_{small} are the weights for the smooth and small motion terms, respectively. *Dividing by 2 is to be consistent with the E_{GMM} term.* Different from the motion reconstruction work (Li et al. 2009), we (i) extend GMM to non-rigid registration; (ii) introduce the term E_{small}; and (iii) impose a hard constraint to replace their soft rigid term to reduce nonlinear complexity. The technique (Li et al. 2009) is prone to converge into a local minimum (Fig. 3(a)) as it is ICP-based.

To efficiently solve node transformations, we minimize Eq. (4) in the following way: updating one node transformation by fixing the remaining nodes. This optimization scheme largely reduces the complexity of the problem and ensures the non-positive growth of the objective function.

We first take the partial derivative of E with respect to $\mathbf{T}_{\hat{j}}$ of a specific node $\hat{j}$ and equate it to zero, and then obtain:

$$\mathbf{T}_{\hat{j}} = \mu_v - \mathbf{R}_{\hat{j}}\mu_y, \tag{5}$$

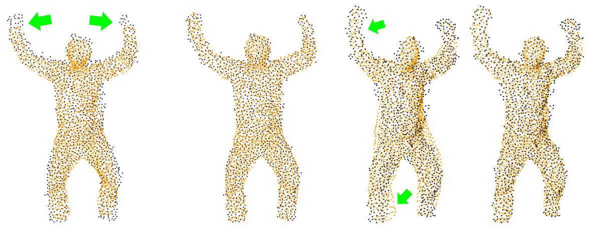

(a) (b) (c) (d)

Figure 3: (a) and (c): results of (Li et al. 2009) and (Myronenko and Song 2010); (b) and (d): our results. The yellow point set ($\mathbf{Y}$) is registered to the black point set ($\mathbf{V}$).

where μ_v and μ_y are $D \times 1$ vectors which can be easily calculated. After substituting Eq. (5) into Eq. (4a) and reorganizing it, we can obtain: $E = -tr(\mathbf{H}\mathbf{R}_{\hat{j}}) + z$, where $tr()$ indicates the trace operation, $\mathbf{H}$ is a $D \times D$ matrix and z is a scalar.

Minimizing E is equivalent to maximizing $-E$. We apply the *Lemma 1* (Myronenko and Song 2009) to achieve a closed-form solution for $\mathbf{R}_{\hat{j}}$.

$$\mathbf{R}_{\hat{j}} = \mathbf{U}_{\hat{j}}\mathbf{C}_{\hat{j}}\mathbf{V}_{\hat{j}}^T, \tag{6}$$

where $\mathbf{U}_{\hat{j}}\mathbf{S}_{\hat{j}}\mathbf{V}_{\hat{j}}^T = svd(\mathbf{H}^T)$ and $\mathbf{C}_{\hat{j}} = diag(1,1,...,det(\mathbf{U}_{\hat{j}}\mathbf{V}_{\hat{j}}^T))$. We then compute $\mathbf{T}_{\hat{j}}$ via Eq. (5). In a similar way of taking partial derivative, we can obtain

$$\sigma^2 = \frac{1}{DN_p}\sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}\|\mathbf{v}_i - \mathbf{y}_m\|^2. \tag{7}$$

The algorithm is listed in Alg. 1. Refer to Sec. 6 (last paragraph) and Sec. 7 for termination conditions and parameter settings, respectively.

Algorithm 1 Non-rigid Point Set Registration

```
Input: original point set $\mathbf{V}$
Output: registered point set $\mathbf{Y}$
  repeat
    E-step:
      • compute posterior probabilities via Eq. (3)
    M-step:
      • compute $\mathbf{R}_{\hat{j}}$ and $\mathbf{T}_{\hat{j}}$ via Eq. (6) and Eq. (5)
        for each node $\hat{j}$
      • update $\sigma^2$ via Eq. (7)
      • update $\{\mathbf{y}'_m\}$ via Eq. (2)
  until convergent or maximum iterations are reached
```

Directly using the original point sets for registration would possibly generate poor results, since they are typically corrupted with heavy noise and outliers. To generate better registration results, we first reconstruct a surface from the initial frame and then turn it into a point set by removing its topology. This point set is chosen as the *rest pose* (i.e., the rest point set). Other frames can also be chosen as the rest pose. This way ensures both robust registration results and no priors on the articulated objects. $\mathbf{Y}$ is initialized with the registration output in the previous frame. We register the rest point set among frames sequentially.

Though being only one step of our method, our algorithm differs greatly from the previous works (Myronenko and Song 2010; Ye and Yang 2014; Cagniart, Boyer, and Ilic 2010). Specifically, Myronenko and Song (2010) proposed the motion coherent registration of two single point sets rather than more challenging sequential articulated input. The accumulated errors of sequential registration are often noticeable (Fig. 3(c)). Using a single depth camera, Ye and Yang (2014) estimated poses based on a complete skinning mesh template embedded with skeleton. Cagniart, Boyer, and Ilic (2010) deformed a complete mesh template to fit mesh sequences acquired from multiple cameras within a Bayesian framework. By contrast, we relate the embedded deformation model (Li et al. 2009) with GMM, where the deformation is represented by some sparse node transformations. It deals with point set sequences captured by a single depth sensor and does not require a complete template or skeleton priors. Also, both the formulations and optimizations between these methods and our algorithm are significantly different (see the above details).

5 Skeletal Structure Extraction

LBS model. We assume the motion of articulated objects (e.g., humans) can be approximately modeled by the widely used Linear Blend Skinning (LBS) model (Magnenat-Thalmann, Laperri‘ere, and Thalmann 1988), which can be formulated as follows.

$$\mathbf{x}_m^t = \sum_{j=1}^{B} w_{mj}(\mathbf{R}_j^t \mathbf{q}_m + \mathbf{T}_j^t), \qquad (8)$$

where $\mathbf{q}_m$ is the location ($D \times 1$) of the m-th point at the *rest pose*, w_{mj} is the weight imposed on the m-th point by the j-th bone, and B is the number of bones. $\mathbf{R}_j^t$ and $\mathbf{T}_j^t$ are the $D \times D$ rotation matrix and $D \times 1$ translation vector of the j-th bone at the t-th frame, respectively. $\mathbf{x}_m^t$ is the deformed position of the m-th point at frame t. $\mathbf{Q} = \{\mathbf{q}_m\}$ and $\mathbf{X}^t = \{\mathbf{x}_m^t\}$, both of which have M points.

Motion-based clustering. To build an initial skeleton from the output sequence $\{\mathbf{Y}^t\}$ obtained from the first step, we assume each body part is nearly rigid (Le and Deng 2014). Precisely, the m-th point is only influenced by a single bone j (i.e., $w_{mj} = 1$). Thus, we can formulate the bone transformation problem as

$$\underset{\mathbf{R}_j^t, \mathbf{T}_j^t}{\arg\min} \sum_{m \in clu(n)} \|\mathbf{y}_m^t - (\mathbf{R}_j^t \mathbf{q}_m + \mathbf{T}_j^t)\|^2, \qquad (9)$$

where $clu(n)$ denotes the point index set for the n-th cluster. To achieve the optimized transformation ($\mathbf{R}_j^t$, $\mathbf{T}_j^t$) at each frame, we need to minimize the sum of squared residuals (Eq. (9)), together with the SO(3) constraint (Eq. (4b)). Eq. (9) is the absolute orientation problem (Kabsch 1978). To find the best bone transformation for each cluster, we present an iterative update strategy: (i) optimize Eq. (9) to achieve $\{\mathbf{R}_j^t, \mathbf{T}_j^t\}$; (ii) update cluster labels for points by selecting the bone that has the smallest residual (i.e., $\|\mathbf{y}_m^t - (\mathbf{R}_j^t \mathbf{q}_m + \mathbf{T}_j^t)\|$); (iii) search the neighbors of each point within a ball, and update its cluster label with the largest number of neighbors that share the same label. Note (iii) is to improve the local continuity among points (Fig. 4(b)). Only (i) and (ii) would result in inaccurate clusters (Fig. 4(a)). In our experiments, we perform 10 iterations of this strategy.

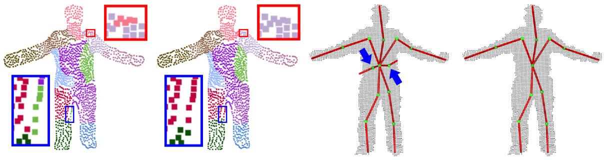


Figure 4: (a) and (b): without and with improving local continuity. (c) and (d): without and with skeletal structure refinement.

To cluster the initial parts, we employ the K-means algorithm since it is more efficient than other clustering methods (e.g., spectral clustering (Ng, Jordan, and Weiss 2002)) Note that it is unnecessary to determine the exact number of clusters at this initialization step, because insignificant bones would be removed later.

Skeletal structure generation. With the achieved clusters, we generate a graph $\mathbb{G}$ where bones are viewed as nodes. Our idea to compute edge weights is: if two bones have a real joint, the residuals should be small after interchanging the bone transformations. This is because two connected bones typically have more similar transformations than two unconnected bones. Specifically, the edge weight e_{ij} between bone i and j is computed as follows.

$$\begin{aligned} e_{ij} = & \frac{1}{|clu(i)|} \sum_{t=1}^{F} \sum_{k \in clu(i)} \|\mathbf{y}_k^t - (\mathbf{R}_j^t \mathbf{q}_k + \mathbf{T}_j^t)\|^2 \\ & + \frac{1}{|clu(j)|} \sum_{t=1}^{F} \sum_{k' \in clu(j)} \|\mathbf{y}_{k'}^t - (\mathbf{R}_i^t \mathbf{q}_{k'} + \mathbf{T}_i^t)\|^2, \end{aligned} \qquad (10)$$

where $|clu(i)|$ and $|clu(j)|$ are the numbers of points in clusters i and j, respectively. To determine which two bones share a joint, we compute the minimum spanning tree $\mathbb{S}$ of $\mathbb{G}$ (a small weight means a large probability of sharing a joint). For visualization purposes, we set the root joint to be the cluster center which is the shortest to the center of the *rest pose*. To visualize the current skeleton, we need to compute the initial joint locations by minimizing $\mathbb{E}_{\text{Joint}}$ in Sec. 6.

Skeletal structure refinement. Unlike some previous methods (Kirk, O'Brien, and Forsyth 2005; Zhang et al. 2013), we refine the produced skeletal structure by removing the unnecessary joints and bones (Fig. 4 (c)-(d)). To obtain a desired skeletal structure, we empirically present the following criteria: (1) if a joint connects more than one joints, search for each next joint, and remove this joint and the associated bone if it is a leaf node; (2) remove the loops in the skeleton; (3) merge two adjacent joints if they are very close.

6 Skeleton Joints Estimation

In this section, we first analyze the issues that lead to inaccurate joints. Then we show how to formulate the joints

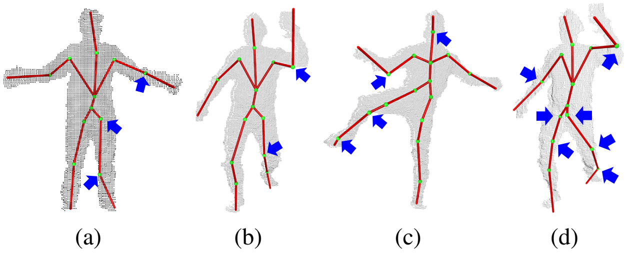


Figure 5: (a)-(c): several extracted skeleton examples after Step 2 (Sec. 5). (d): joints estimation using only $\{\mathbf{y}_m^t\}$ (Sec. 6).

estimation problem, and explain how to solve it using an EM algorithm. Finally, we introduce new energy terms and describe how to minimize the total energy in the M-step.

6.1 Inaccurate Joints

Based on the LBS model (Eq. (8)), one point is often influenced by more than one bones during motion. However, the above bone transformations are optimized by assuming neither point-bone weight blending nor joint constraints, which would lead to inaccurate bone transformations. As a result, the acquired joints can be noticeably inaccurate (Fig. 5 (a)-(c)). As illustrated in Fig. 5(d), using only the registered points $\{\mathbf{y}_m^t\}$ for joints estimation may not be sufficient, as the lacking of the original input $\{\mathbf{v}_i^t\}$ may overlook certain useful information. To overcome these two issues, we propose an iterative LBS-based algorithm, which incorporates both the original input ($\{\mathbf{v}_i^t\}$) and the output ($\{\mathbf{y}_m^t\}$) from the first step (Sec. 4) to obtain accurate joints.

6.2 GMM-based Formulation and Optimization

At each frame t, the deformed points $\{\mathbf{x}_m^t\}$ should approximate the underlying original point set $\{\mathbf{v}_i^t\}$. Suppose the deformed points $\{\mathbf{x}_m^t\}$ are the centroids of a GMM which generates the captured point cloud $\{\mathbf{v}_i^t\}$, then the probability of each point $\mathbf{v}_i^t$ is

$$p(\mathbf{v}_i^t) = (1-\omega')\sum_{m=1}^{M} p(\mathbf{x}_m^t)p(\mathbf{v}_i^t|\mathbf{x}_m^t) + \omega'\frac{1}{N^t}. \qquad (11)$$

Please refer to Sec. 4 for similar variable interpretation. The approximation problem *over all the frames* can be regarded as a parameter estimation problem by minimizing the negative log-likelihood function: $\mathbb{E}(\mathbf{W},\mathbf{R},\mathbf{T},\tau) = -\log\left(\prod_{t=1}^{F}\prod_{i=1}^{N^t} p(\mathbf{v}_i^t)\right)$, where $\mathbf{W} = \{w_{mj}\}$, $\mathbf{R} = \{\mathbf{R}_j^t\}$, $\mathbf{T} = \{\mathbf{T}_j^t\}$ and $\tau = \{\tau^t\}$ (τ^t is σ^2 at frame t).

Similar to Sec. 4, we also employ the EM procedure to minimize $\mathbb{E}$. At the E-step, $p^{old}(\mathbf{x}_m^t|\mathbf{v}_i^t)$ is computed using the same form as Eq. (3). The deformed points $\{\mathbf{x}_m^t\}$ are initialized using the bone transformations in Sec. 5 for the first iteration. At the M-step, we update the involved parameters. By assuming the independence of each frame, the upper bound of $\mathbb{E}$ is: $\mathbb{E}_{\text{GMM}} = \sum_{t=1}^{F}[\frac{1}{2\tau^t}\sum_{i=1}^{N^t}\sum_{m=1}^{M} p_{mi}^t\|\mathbf{v}_i^t - \mathbf{x}_m^t\|^2 + \frac{DN_p^t}{2}\log\tau^t]$. Here, $p_{mi}^t = p^{old}(\mathbf{x}_m^t|\mathbf{v}_i^t)$, $\mathbf{P}^t = \{p_{mi}^t\}$ and $N_p^t = \sum_{i=1}^{N^t}\sum_{m=1}^{M} p_{mi}^t$.

6.3 New Energy Terms

To utilize the registered data (Sec. 4), we introduce a registration term $\mathbb{E}_{\text{Register}}$ involving $\{\mathbf{y}_m^t\}$: $\mathbb{E}_{\text{Register}} = \sum_{t=1}^{F}\sum_{m=1}^{M}\|\mathbf{y}_m^t - \mathbf{x}_m^t\|^2$. To constrain bones rotating around joints, we also present a joint term $\mathbb{E}_{\text{Joint}}$ involving joint locations: $\mathbb{E}_{\text{Joint}} = \eta\sum_{<j,k>\in\mathbb{S}}\|\mathbf{c}_{jk} - \tilde{\mathbf{c}}_{jk}\|^2 + \sum_{t=1}^{F}\sum_{<j,k>\in\mathbb{S}}\|(\mathbf{R}_j^t\mathbf{c}_{jk} + \mathbf{T}_j^t) - (\mathbf{R}_k^t\mathbf{c}_{jk} + \mathbf{T}_k^t)\|^2$, where $\tilde{\mathbf{c}}_{jk}$ is the centroid of boundary points between clusters j and k. Since optimizing only the second term in $\mathbb{E}_{\text{Joint}}$ would possibly generate multiple solutions when solving joint positions, we include a data constraint (i.e., $\sum_{<j,k>\in\mathbb{S}}\|\mathbf{c}_{jk} - \tilde{\mathbf{c}}_{jk}\|^2$).

The GMM term favors approximating the captured point clouds with the deformed point sets. The registration and joint terms here are inspired and derived from some previous works (Schaefer and Yuksel 2007; Le and Deng 2014). The former encourages the deformed points to be close to the registered points, and the latter favors each joint approaching the nearly same deformed positions after two neighboring transformations. Like Sec. 4, we also assume small motion changes in each iteration. Thus the final energy $\mathbb{E}_{\text{Total}}$ for the M-step is

$$\mathbb{E}_{\text{Total}} = \mathbb{E}_{\text{GMM}} + \frac{\zeta}{2}\mathbb{E}_{\text{Register}} + \frac{\alpha}{2}\mathbb{E}_{\text{Joint}} + \frac{\gamma}{2}\sum_{t=1}^{F} E_{\text{small}}^t \qquad (12a)$$

$$\text{s.t. } w_{mj} \geq 0, \sum_{j=1}^{B} w_{mj} = 1, \|\mathbf{W}_{m,:}\|_0 \leq 4, \forall m \qquad (12b)$$

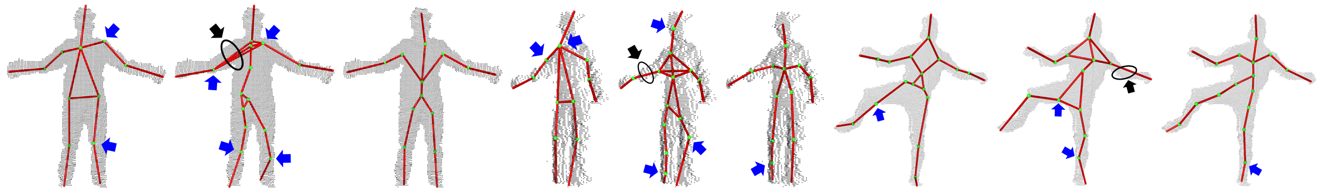
$${\mathbf{R}_j^t}^T\mathbf{R}_j^t = \mathbf{I}, det(\mathbf{R}_j^t) = 1, \forall t, j. \qquad (12c)$$

Here ζ, α and γ are the regularized weights and $\mathbf{W}_{m,:}$ is the m-th row of the weights matrix $\mathbf{W}$. The non-negative, affinity and sparse (typically set to 4) constraints are imposed to weights (Eq. (12b)), and the orthogonal constraint is added to bone rotations (Eq. (12c)).

6.4 Minimization

We now describe how to optimize the involved parameters ($\{\mathbf{c}_{jk}\}$, $\mathbf{W}$, $\mathbf{R}$, $\mathbf{T}$ and τ) in the M-step. Joint positions are closely related with other parameters ($\mathbf{W}$, $\mathbf{R}$, $\mathbf{T}$ and τ) through direct or indirect connections. To achieve accurate joints, it is also necessary to optimize other parameters. For this purpose, we present an optimization strategy to minimize $\mathbb{E}_{\text{Total}}$: the other parameters are fixed when optimizing one class of parameters. Regarding bone transformations ($\{\mathbf{R}_j^t, \mathbf{T}_j^t\}$), we employ the same scheme presented in Sec. 4.

Point weights estimation. The weights of a point are independent of the weights of the other points. Thus, the objective function for the m-th point is: $\mathbb{E}(\mathbf{W}_{\hat{m},:}) = \sum_{t=1}^{F}\frac{1}{2\tau^t}\sum_{i=1}^{N^t} p_{\hat{m}i}^t\|\mathbf{v}_i^t - \mathbf{x}_{\hat{m}}^t\|^2 + \frac{\zeta}{2}\sum_{t=1}^{F}\|\mathbf{y}_{\hat{m}}^t - \mathbf{x}_{\hat{m}}^t\|^2$, where $\mathbf{x}_{\hat{m}}^t = \sum_{j=1}^{B} w_{\hat{m}j}(\mathbf{R}_j^t\mathbf{q}_{\hat{m}} + \mathbf{T}_j^t)$. We choose 4 bones which have the smallest residuals when separately calculating the above objective function, and then solve the least squares problem on the selected 4 bones with the constraints (Eq. (12b)).



(a) Method II (b) Method III (c) Ours (d) II (e) III (f) Ours (g) Method II (h) Method III (i) Ours

Figure 6: Extracted skeletons on two sequences of EVAL (a-f) and one sequence of PDT (g-i). Blue and black arrows indicate inaccurate and missing joints, respectively.

$$\begin{aligned}\mathbb{E}(\mathbf{R}^t_{\hat{j}}, \mathbf{T}^t_{\hat{j}}) = {} & \frac{\zeta}{2}\sum_{m=1}^{M}\|\mathbf{y}^t_m - \mathbf{u}^t_{m\hat{j}} - w_{m\hat{j}}(\mathbf{R}^t_{\hat{j}}\mathbf{q}_m + \mathbf{T}^t_{\hat{j}})\|^2 \\ & + \frac{1}{2\tau^t}\sum_{i=1}^{N^t}\sum_{m=1}^{M} p^t_{mi}\|\mathbf{v}^t_i - \mathbf{u}^t_{m\hat{j}} - w_{m\hat{j}}(\mathbf{R}^t_{\hat{j}}\mathbf{q}_m + \mathbf{T}^t_{\hat{j}})\|^2 \\ & + \frac{\alpha}{2}\sum_{<\hat{j},k>\in\mathbb{S}}\|(\mathbf{R}^t_{\hat{j}}\mathbf{c}_{\hat{j}k} + \mathbf{T}^t_{\hat{j}}) - (\mathbf{R}^t_k\mathbf{c}_{\hat{j}k} + \mathbf{T}^t_k)\|^2 \\ & + \frac{\gamma}{2}(\|\mathbf{R}^t_{\hat{j}} - \mathbf{R}^{pre}_{\hat{j}}\|^2_F + \|\mathbf{T}^t_{\hat{j}} - \mathbf{T}^{pre}_{\hat{j}}\|^2).\end{aligned} \tag{13}$$

Bone transformations estimation. Since bone transformations at each frame are independent, we can obtain Eq. (13) for bone $\hat{j}$ at frame t. Here, $\mathbf{U}^t_{\hat{j}} = \{\mathbf{u}^t_{m\hat{j}}\}$ and $\mathbf{u}^t_{m\hat{j}} = \sum_{j=1,j\neq\hat{j}}^{B} w_{mj}(\mathbf{R}^t_j\mathbf{q}_m + \mathbf{T}^t_j)$. Taking the partial derivative of Eq. (13) with respect to $\mathbf{T}^t_{\hat{j}}$ and equating it to zero, we obtain

$$\mathbf{T}^t_{\hat{j}} = \mu^t_{u\hat{j}} - \mathbf{R}^t_{\hat{j}}\mu^t_{q\hat{j}}, \tag{14}$$

where $\mu^t_{u\hat{j}}$ and $\mu^t_{q\hat{j}}$ are $D \times 1$ vectors. Substituting Eq. (14) into Eq. (13), we can obtain the objective function involving only $\mathbf{R}^t_{\hat{j}}$: $\mathbb{E}(\mathbf{R}^t_{\hat{j}}) = -tr(\mathbf{Z}^t_{\hat{j}}\mathbf{R}^t_{\hat{j}}) + b$, where $\mathbf{Z}^t_{\hat{j}}$ is a $D \times D$ matrix and b is a scalar.

Similar to Sec. 4, we achieve

$$\mathbf{R}^t_{\hat{j}} = \mathbf{U}^t_{\hat{j}}\mathbf{C}^t_{\hat{j}}{\mathbf{V}^t_{\hat{j}}}^T, \tag{15}$$

where $\mathbf{U}^t_{\hat{j}}\mathbf{S}^t_{\hat{j}}{\mathbf{V}^t_{\hat{j}}}^T = svd({\mathbf{Z}^t_{\hat{j}}}^T)$ and $\mathbf{C}^t_{\hat{j}} = diag(1, 1, ..., det(\mathbf{U}^t_{\hat{j}}{\mathbf{V}^t_{\hat{j}}}^T))$. $\mathbf{T}^t_{\hat{j}}$ can be then computed via Eq. (14).

Joint locations estimation. We minimize $\mathbb{E}_{\text{Total}}$ with respect to $\mathbf{c}_{\hat{j}k}$, which amounts to minimizing $\mathbb{E}_{\text{Joint}}$.

Covariances estimation. The covariances are updated similar to Sec. 4 (Eq. (7)).

Deformed points update. We update $\{\mathbf{x}^t_m\}$ using the estimated point weights and bone transformations via Eq. (8).

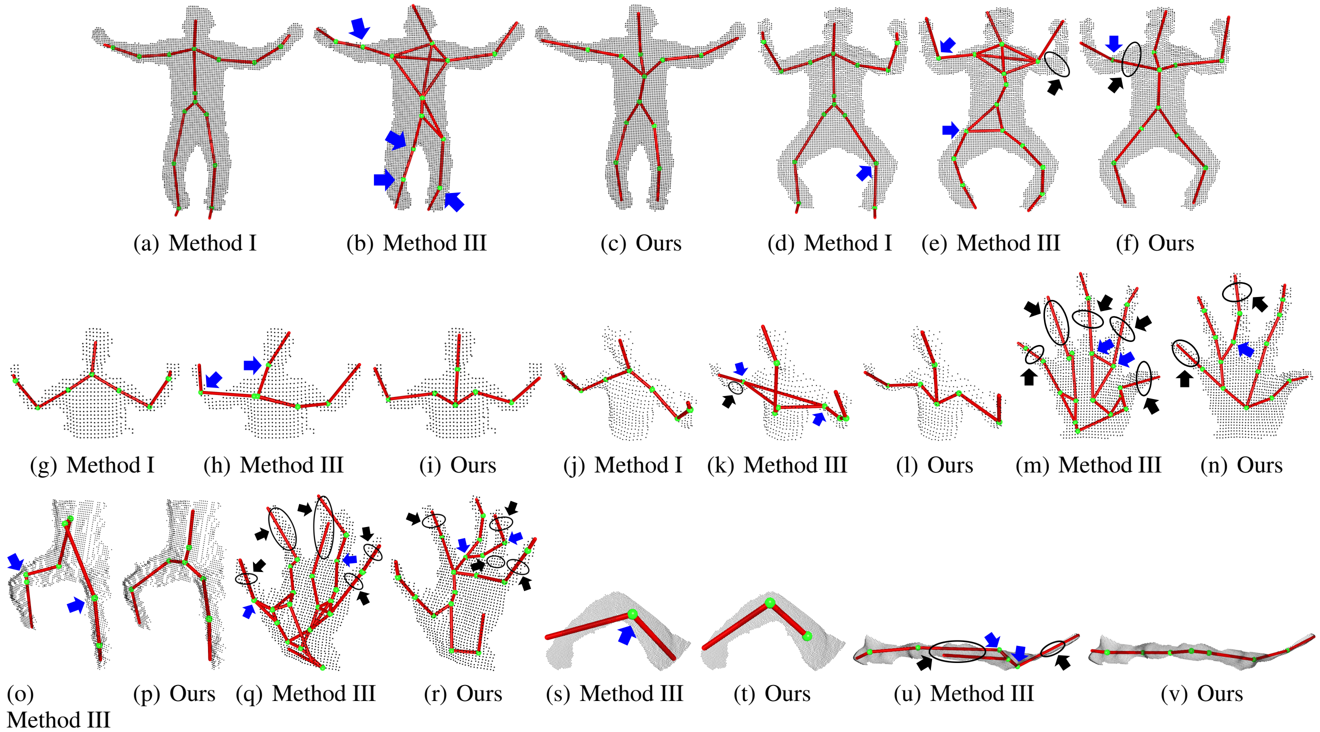
We summarize this algorithm in Alg. 2. For both Alg. 1 and 2, we stop the EM procedure when the difference of total energy between two consecutive iterations is smaller than a threshold or the number of iterations is more than 20. We found that our algorithms typically converge within 20 iterations. Notice that we extend the GMM to both Sec. 4 and 6 which involve different tasks. The former is for point registration based on the embedded deformation model, while the latter is to achieve accurate joints based on the LBS model.

Algorithm 2 Skeleton Joints Estimation

```
Input: original and registered point sets $\{\mathbf{V}^t\}$, $\{\mathbf{Y}^t\}$
Output: joint locations $\{\mathbf{c}_{jk}\}$
  repeat
     E-step:
        • compute posteriors similarly as Eq. (3)
     M-step:
        • estimate weights point by point ($\mathbb{E}(\mathbf{W}_{\hat{m},:})$)
        • compute $\mathbf{R}^t_{\hat{j}}$ and $\mathbf{T}^t_{\hat{j}}$ via Eq. (15) and Eq. (14)
          for each bone $\hat{j}$ at each frame $t$
        • estimate joint locations ($\mathbb{E}_{\text{Joint}}$)
        • estimate covariances similarly as Eq. (7)
        • update the deformed points (Eq. (8))
  until convergent OR maximum iterations are reached
```

7 Experimental Results

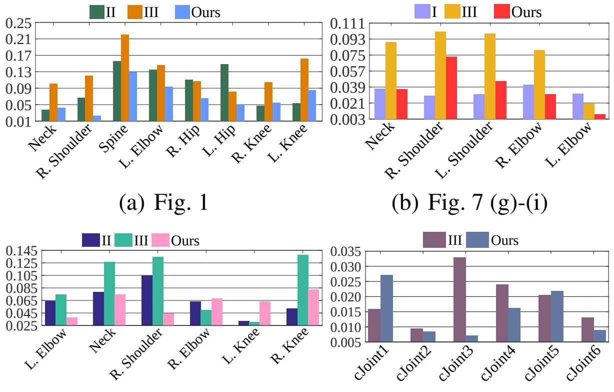
Test data. We tested our method on the sequences from three datasets: publicly available EVAL (Ganapathi et al. 2012) and PDT (Helten et al. 2013), and the Kinect data captured by ourselves. Besides, we qualitatively and quantitatively compared our approach with the state of the art techniques, respectively labeled as *Method I* (Microsoft 2017), *II* (Kirk, O'Brien, and Forsyth 2005) and *III* (Zhang et al. 2013) for simplicity. We choose to compare our method with Method II and III, because skeleton extraction from point set sequences has been sparsely treated so far (see Sec. 1). We did not compare our method with the methods in EVAL and PDT since they target at the pose tracking of depth images, by parameterizing human poses through the deformation of a given template model (mesh or capsule). Our first step (Sec. 4) plays a similar role but it does not need such a given template model. Nevertheless, we compare our method with KinectSDK (Method I) that is *designed for supervised pose estimation (tracking) rather than skeleton extraction* using a human skeleton template prior, to show the poses of the extracted skeletons by our method are even competitive. All the datasets provide ground truth joints. PDT and EVAL provide marker data input for Method II. Method I (KinectSDK) estimates a skeleton pose for each frame of our data (only full body and upper body) as it is designed *only for human poses estimation*. Therefore, regarding PDT and EVAL, we conduct experiments using Method II, III and our approach, both qualitatively and quantitatively. Method I, III and our method are compared both qualitatively and quantitatively using our captured data. For fair


(a) Method I (b) Method III (c) Ours (d) Method I (e) Method III (f) Ours

(g) Method I (h) Method III (i) Ours (j) Method I (k) Method III (l) Ours (m) Method III (n) Ours

(o) Method III (p) Ours (q) Method III (r) Ours (s) Method III (t) Ours (u) Method III (v) Ours

Figure 7: Extracted skeletons on our Kinect data. I-(Microsoft 2017), II-(Kirk, O'Brien, and Forsyth 2005) and III-(Zhang et al. 2013).



(a) Fig. 1 (b) Fig. 7 (g)-(i)

(c) Fig. 6 (a)-(c) (d) Fig. 7 (m)-(n)

Figure 8: Distance errors (per joint) on some sequences.

comparisons, skeletons are extracted and rendered by following their works (Method II and III). We did not choose to compare our method with (Le and Deng 2014), since it is for mesh sequences which have prior connectivity and correspondence information.

Table 1: The parameter values used in all our experiments.

Eq. (4)	$\omega = 0.01$, $\beta_{\text{smooth}} = 10^5$, $\beta_{\text{small}} = 1$
Eq. (12)	$\omega' = 0.01$, $\eta = 1$, $\zeta = 10^4$, $\gamma = 0.1$

Parameter settings. To show the robustness of our method, we empirically fix the parameters except α in all our experiments (Table 1). To favor bones rotating more rigorously around joints, α is initialized with ζM and multiplied by 1.45 in each iteration. Like (Ye and Yang 2014), we initialized all σ^2 (Sec. 4) and $\{\tau^t\}$ using the same fixed value, 6×10^{-4}. As σ^2 and $\{\tau^t\}$ are generally smaller than 10^{-3}, some regularized weights are large.

Qualitative comparisons. We show the visual comparisons between our approach and the state of the art techniques on various objects (full body: Fig. 1, 6 and 7(a-f), upper body: Fig. 7(g-l), hand: 7(m-n,q-r), lower body: 7(o-p), arm: 7(s-t), and fish: 7(u-v)). Compared with Method III (Zhang et al. 2013), our approach produces substantially higher quality skeletons. Our method can even generate better skeletons (Fig. 1 and 6) than Method II (Kirk, O'Brien, and Forsyth 2005), despite their good results are probably due to high quality marker input. The poses of our extracted skeletons are even comparable with those estimated by the supervised KinectSDK (Fig. 7). Note that a few joints in our results are inaccurate, which is normally caused by the relatively small-scale motion of the involved clusters.

Quantitative comparisons. We compared the accuracies of all the methods using the ground-truth joints. Since different approaches extracted different sets of joints, for each tested sequence we perform the accuracy evaluations based on the common subset of joints which are close to semantic positions (e.g., elbows). We adapt the Euclidean distance error metric used in (Helten et al. 2013; Ye and Yang 2014) to measure the accuracy of the extracted skeletons. Specifically, we measure the average error over all the joints or the distance error per joint. For all tested sequences, we follow

Table 2: The Euclidean distance errors for all the tested sequences and compared techniques. The unit is meter.

Methods Sequences	Method I KinectSDK	Method II Kirk et al.	Method III Zhang et al.	Ours
Fig. 1	NA	0.0780	0.1221	**0.0659**
Fig. 6(a-c)	NA	0.0657	0.0922	**0.0617**
Fig. 6(d-f)	NA	0.1030	0.1088	**0.0864**
Fig. 6(g-i)	NA	0.1566	0.1205	**0.1124**
Fig. 7(a-c)	0.0521	NA	0.0739	**0.0497**
Fig. 7(d-f)	**0.0595**	NA	0.0892	0.0669
Fig. 7(g-i)	**0.0341**	NA	0.0781	0.0389
Fig. 7(j-l)	0.0359	NA	0.0842	**0.0306**
Fig. 7(m-n)	NA	NA	0.0193	**0.0149**
Fig. 7(o-p)	NA	NA	0.0647	**0.0186**
Fig. 7(q-r)	NA	NA	0.0202	**0.0124**
Fig. 7(s-t)	NA	NA	0.0866	**0.0198**
Fig. 7(u-v)	NA	NA	0.0378	**0.0178**

the normalization process (Helten et al. 2013) for each joint.

Table 2 clearly shows that our method outperforms state of the art techniques (Method II and III), and is even comparable to Method I (KinectSDK) in terms of pose accuracy despite it clearly benefits from its supervised learning and pre-embedded human skeleton. Fig. 8 illustrates the distance error per joint for some sequences. Though the per-joint errors by our method are not always lower than other methods, it is on average better or at least comparable. However, as the price paid for high accuracy and robustness, our method is usually 2-15 times slower than other methods because of the iterative EM optimization in Alg. 1 and 3. It is more suitable for offline processing purpose. It is noteworthy that we did not perform any particular optimizations (e.g., GPU accelerated) to speed up the efficiency of our implementation.

8 Discussion and Conclusion

We introduced a novel approach for unsupervised skeleton extraction from point set sequences collected by a single depth camera. It is robust and accurate in extracting skeletons of various articulated objects. The extensive experiments show that our method both visually and quantitatively outperforms the state of the art approaches (Kirk, O'Brien, and Forsyth 2005; Zhang et al. 2013). The poses of our extracted skeletons are even comparable with those by the supervised pose estimation technique (Microsoft 2017). Our method can also be potentially applied to handle point sets collected by multiple cameras, or other point-based data.

The EM steps in Alg. 1 and 2 are time-consuming and could be improved via fast Gauss transform (Greengard and Strain 1991) and GPUs. Similar to existing methods, the severe occlusions involved in some data (e.g., quadrupled animals) captured by a single depth sensor also pose extra challenges to our method. In the future, we would like to explore how to extend our framework to handle this challenge. Though our optimization-based method is more accurate and robust and we provided the empirical parameter values, the optimization-based approaches often suffer from parameter tuning, quality and accuracy (e.g., (Zhang et al. 2013; Kirk, O'Brien, and Forsyth 2005)) or slow speed (e.g., our method). We plan to extend learning-based methods (supervised or unsupervised) for fast and accurate skeleton extraction in the future.

Acknowledgments

Sai-Kit Yeung was supported in part by Singapore MOE Academic Research Fund MOE2016-T2-2-154 and by a grant from the National Heritage Board of Singapore. This research was also supported in part by the NRF under its IDM Futures Funding Initiative and Virtual Singapore Award No. NRF2015VSG-AA3DCM001-014. Zhigang Deng was supported in part by US NSF IIS-1524782. We thank the authors of (Xu et al. 2017) for providing the fish data.

References

Attali, D., and Lachaud, J.-O. 2001. Delaunay conforming iso-surface, skeleton extraction and noise removal. *Computational Geometry* 19(2):175 – 189.

Au, O. K.-C.; Tai, C.-L.; Chu, H.-K.; Cohen-Or, D.; and Lee, T.-Y. 2008. Skeleton extraction by mesh contraction. *ACM Trans. Graph.* 27(3):44:1–44:10.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc.

Cagniart, C.; Boyer, E.; and Ilic, S. 2010. Probabilistic deformable surface tracking from multiple videos. In *Proceedings of the 11th European Conference on Computer Vision*, ECCV'10, 326–339. Berlin, Heidelberg: Springer-Verlag.

Chalodhorn, R.; Grimes, D. B.; Grochow, K.; and Rao, R. P. N. 2007. Learning to walk through imitation. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, 2084–2090. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Chang, H. J., and Demiris, Y. 2015. Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 3138–3146.

Cheung, G. K. M.; Baker, S.; and Kanade, T. 2003. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 77–84. Washington, DC, USA: IEEE Computer Society.

Chrungoo, A.; Manimaran, S. S.; and Ravindran, B. 2014. *Activity Recognition for Natural Human Robot Interaction*. Cham: Springer International Publishing. 84–94.

Chun, C.-W.; Jenkins, O. C.; and Mataric, M. J. 2003. Markerless kinematic model and motion capture from volume sequences. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, II–475. IEEE.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)* 1–38.

Ganapathi, V.; Plagemann, C.; Koller, D.; and Thrun, S. 2012. Real-time human pose tracking from range data. In *Proceedings of the 12th European Conference on Computer Vision*, 738–751. Berlin, Heidelberg: Springer-Verlag.

Greengard, L., and Strain, J. 1991. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing* 12(1):79–94.

Helten, T.; Baak, A.; Bharaj, G.; Muller, M.; Seidel, H.-P.; and Theobalt, C. 2013. Personalization and evaluation of a real-time depth-based full body tracker. In *Proceedings of the 2013 International Conference on 3D Vision*, 279–286. Washington, DC, USA: IEEE Computer Society.

Huang, H.; Wu, S.; Cohen-Or, D.; Gong, M.; Zhang, H.; Li, G.; and Chen, B. 2013. L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32(4):65:1–65:8.

Kabsch, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 34(5):827–828.

Kirk, A. G.; O'Brien, J. F.; and Forsyth, D. A. 2005. Skeletal parameter estimation from optical motion capture data. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 782–788. Washington, DC, USA: IEEE Computer Society.

Le, B. H., and Deng, Z. 2014. Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph.* 33(4):84:1–84:10.

Li, H.; Adams, B.; Guibas, L. J.; and Pauly, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28(5):175:1–175:10.

Magnenat-Thalmann, N.; Laperri'ere, A.; and Thalmann, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface '88*, GI '88, 26–33. Toronto, Ontario, Canada: Canadian Man-Computer Communications Society.

Microsoft. 2017. Kinectsdk. https://developer.microsoft.com/en-us/windows/kinect.

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

Myronenko, A., and Song, X. 2010. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(12):2262–2275.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In Dietterich, T. G.; Becker, S.; and Ghahramani, Z., eds., *Advances in Neural Information Processing Systems*. MIT Press. 849–856.

Ramanan, D.; Forsyth, D. A.; and Barnard, K. 2006. Building models of animals from video. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(8):1319–1334.

Ross, D. A.; Tarlow, D.; and Zemel, R. S. 2010. Learning articulated structure and motion. *International Journal of Computer Vision* 88(2):214–237.

Schaefer, S., and Yuksel, C. 2007. Example-based skeleton extraction. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 153–162. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

Song, S.; Lan, C.; Xing, J.; Zeng, W.; and Liu, J. 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'17. AAAI Press.

Tam, G. K. L.; Cheng, Z. Q.; Lai, Y. K.; Langbein, F. C.; Liu, Y.; Marshall, D.; Martin, R. R.; Sun, X. F.; and Rosin, P. L. 2013. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* 19(7):1199–1217.

Tresadern, P., and Reid, I. 2005. Articulated structure from motion by factorization. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1110–1115. Washington, DC, USA: IEEE Computer Society.

Xu, C.; Govindarajan, L. N.; Zhang, Y.; and Cheng, L. 2017. Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision* 123(3):454–478.

Yan, J., and Pollefeys, M. 2008. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(5):865–877.

Ye, M., and Yang, R. 2014. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2353–2360. Washington, DC, USA: IEEE Computer Society.

Zhang, Q.; Song, X.; Shao, X.; Shibasaki, R.; and Zhao, H. 2013. Unsupervised skeleton extraction and motion capture from 3d deformable matching. *Neurocomputing* 100:170–182.

Zhou, X.; Wan, Q.; Zhang, W.; Xue, X.; and Wei, Y. 2016. Model-based deep hand pose estimation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 2421–2427. AAAI Press.