Computer Vision and Image Understanding
journal homepage: www.elsevier.com

3D Articulated Skeleton Extraction Using a Single Consumer-Grade Depth Camera

Xuequan Lu[a,**], Zhigang Deng[b], Jun Luo[c], Wenzhi Chen[d], Sai-Kit Yeung[e], Ying He[c]

[a]Deakin University, Australia
[b]University of Houston, USA
[c]Nanyang Technological University, Singapore
[d]Zhejiang University, China
[e]Hong Kong University of Science and Technology, China

ABSTRACT

Articulated skeleton extraction or learning has been extensively studied for 2D (e.g., images and video) and 3D (e.g., volume sequences, motion capture, and mesh sequences) data. Nevertheless, robustly and accurately learning 3D articulated skeletons from point set sequences captured by a single consumer–grade depth camera still remains challenging, since such data are often corrupted with substantial noise and outliers. Relatively few approaches have been proposed to tackle this problem. In this paper, we present a novel unsupervised framework to address this issue. Specifically, we first build one-to-one point correspondences among the point cloud frames in a sequence with our non-rigid point cloud registration algorithm. We then generate a skeleton involving a reasonable number of joints and bones with our skeletal structure extraction algorithm. We lastly present an iterative Linear Blend Skinning based algorithm for accurate joints learning. At the end, our method can learn a quality articulated skeleton from a single 3D point sequence possibly corrupted with noise and outliers. Through qualitative and quantitative evaluations on both publicly available data and in-house Kinect-captured data, we show that our unsupervised approach soundly outperforms state of the art techniques in terms of both quality (i.e., visual) and accuracy (i.e., Euclidean distance error metric) . Moreover, the poses of our extracted skeletons are even comparable to those by KinectSDK, a well-known supervised pose estimation technique; for example, our method and KinectSDK achieves similar distance errors of 0.0497 and 0.0521.

1. Introduction

[1]3D skeleton learning or extraction (Rossi and Torsello, 2014) from articulated objects has a variety of applications in computer vision and graphics, such as action recognition (Ghorbel et al., 2018; Zhang et al., 2018; Jalal et al., 2017), continuous skeleton tracking (Ye and Yang, 2014), tele-immersion streaming (Raghuraman et al., 2013), computer animation (Le and Deng, 2014), or other tasks (Han et al., 2017). Also, body tracking and skeleton modeling play a vital role in several applications including multimedia contexts (Yang et al., 2008; Jalal and Kamal, 2014), complex object movements (Song et al., 2014), video streaming (Jalal et al., 2014), healthcare systems (Jalal et al., 2014), and smart indoor security systems (Jalal and Kamal, 2014). 3D point motion sequences can be easily captured by a single off-the-shelf depth sensor, thanks to the fast development of depth cameras nowadays. However, the collected motion data is noisy, incomplete, and lacks one-to-one point correspondences among frames. As a result, without any prior knowledge on the captured objects, robustly and accurately extracting an articulated skeleton directly from such a point set sequence remains to be an unresolved research challenge to date.

Skeleton extraction techniques are mostly proposed for either images/video (Tresadern and Reid, 2005; Ramanan et al., 2006; Yan and Pollefeys, 2008; Ross et al., 2010; Chang and Demiris, 2015) or 3D motion (e.g., volume sequences (Chun

[**]Corresponding author.
e-mail: `xuequan.lu@deakin.edu.au` (Xuequan Lu)

[1] A preliminary version of this manuscript appears in a conference (Lu et al., 2018).

et al., 2003), silhouette data (Cheung et al., 2003), marker-based motion capture data (Kirk et al., 2005), and mesh sequences (Schaefer and Yuksel, 2007; Le and Deng, 2014)). By contrast, to date relatively few approaches, except (Kirk et al., 2005; Zhang et al., 2013), have been proposed to extract skeletons directly from point set sequences without any prior knowledge on the captured objects. Nevertheless, the technique in Zhang et al. (2013) suffers from the following major limitations: (1) the matching accuracy is limited due to several-to-one matches, leading to sparse point clouds and further affecting body clustering; (2) a joint is naively selected from either of two neighboring body segments. The approach in Kirk et al. (2005) can extract 3D skeletons from motion capture data but it depends greatly on the quality of the inputted mocap markers. Both of the techniques (Kirk et al., 2005; Zhang et al., 2013) do not consider either joint constraints or mixed bone-point impacts (e.g., *Linear Blend Skinning (LBS)* (Magnenat-Thalmann et al., 1988)) when solving joint locations between two body parts. Also, they often require non-trivial and time-consuming parameter tuning, in particular the number of segment clusters, to achieve the desired skeletal structure with reasonable numbers of joints and bones. As a result, these state of the art methods have limited accuracy and robustness.

In this paper, we present an *unsupervised* approach to robustly and accurately learn articulated skeletons directly from point set sequences collected by a single low-cost depth camera. As suggested by existing works (Zhang et al., 2013; Kirk et al., 2005), part clustering is a necessary initial step to utilize point correspondence among frames. Establishing point correspondences makes motion-based part clustering feasible. Thus, we first present a non-rigid point set registration algorithm to robustly build one-to-one point correspondences among frames. We then extract a skeletal structure from the new sequence that is outputted from the first step. The skeletal structure usually involves inaccurate joints. Finally, on top of the LBS model, we use the dual data source (the original input and the registered point sets) for accurate joints learning. We also analyzed the effects of different energy terms and tested different parameter settings. Through extensive experiments on publicly available data and our in-house Kinect-captured data, we demonstrate the effectiveness of our approach. Through qualitative and quantitative comparisons, we show that our method can significantly outperform the state of the art methods (Kirk et al., 2005; Zhang et al., 2013). In addition, the poses of the learned skeletons by our method are even comparable to those by KinectSDK, a well-known supervised pose tracking technique (Microsoft, 2017).

2. Related Work

In this section, we first review recent techniques on skeleton extraction, and then describe recent efforts on non-rigid point set registrations. Finally, we review some recent related techniques on pose estimation and tracking.

2.1. Skeleton Extraction

A variety of methods have been proposed for skeleton extraction, which can be roughly classified into three types according to the data source.

Skeleton extraction from a static model. To extract skeletons from a single static model (2D or 3D), researchers have proposed various approaches (Au et al., 2008; Tagliasacchi et al., 2009; Livny et al., 2010; Huang et al., 2013). However, without motion-related cues, the extracted skeletons are the medial axes of a single shape in theory. As a result, the extracted skeleton, which is the abstract shape of the static model, can hardly be applied to other applications (e.g., pose estimation).

Skeleton extraction from images/videos. Many techniques have also been proposed to learn skeletons from image-based or video data (Tresadern and Reid, 2005; Yan and Pollefeys, 2006; Ramanan et al., 2006; Ross et al., 2008; Yan and Pollefeys, 2008; Ross et al., 2010; Chang and Demiris, 2015). However, these techniques suffer greatly from the quality of feature points, illumination variations, occlusions, and other environmental factors.

Skeleton extraction from 3D motion. A substantial amount of research efforts have been focused on extracting skeletons from 3D motion. Based on a volumetric sequence captured by multiple cameras, Chun et al. (2003) use the generated underlying nonlinear axes from each frame to derive a kinematic model. Cheung et al. (2003) introduced a Shape-from-Silhouette algorithm for articulated objects, to recover the motion, shape, and joints from silhouette and color images. An unsupervised approach has been proposed to learn skeletons from marker-based motion capture data collected by multiple cameras (Kirk et al., 2005). Recently, based on the deformable matching among different frames, an unsupervised method has been presented by Zhang et al. (2013), to learn articulated skeletons from point motion sequences collected by a Kinect device (Microsoft, 2017). Other previous works (Anguelov et al., 2004; Schaefer and Yuksel, 2007; De Aguiar et al., 2008; Hasler et al., 2010; Le and Deng, 2012, 2014) were introduced to extract bone transformations or skeletons from mesh sequences, where correct one-to-one vertex correspondences among various frames have been provided. Since the input has accurate vertex correspondences and is noise-free, these methods can often produce quality skeletons.

2.2. Non-rigid Point Set Registration

Non-rigid point set registration, a fundamental problem in computer vision and graphics, has been studied for decades. Researchers have focused on the fundamental modeling between two single point sets (source and target). For example, Chui and Rangarajan (2003) developed the TPS-RPM algorithm with the thin-plate spline (TPS) as the parameterization of non-rigid spatial mapping and soft-assignments for the correspondences. Recently, a variety of techniques (Myronenko and Song, 2010; Ma et al., 2016; Qu et al., 2017) have been proposed for non-rigid point-based registrations, and they can generate promising results. Additional research efforts have been centered on motion registrations, i.e., motion tracking for sequential frames. For example, Li et al. (2009) presented a framework for the motion reconstruction of complex deforming shapes, with the assistance of a smooth mesh template and an embedded deformation model. Recently, researchers proposed a number of motion registration approaches (Cagniart et al., 2010; Ye and Yang, 2014; Guo et al., 2015; Dou et al., 2016) to produce increasingly better results. Readers are referred to Tam et al. (2013) for a comprehensive review.

2.3. Pose Estimation

Many techniques have been developed to estimate the poses of humans, hands, and other articulated objects from various data inputs including RGB images/video and depth data. Readers are referred to Sarafianos et al. (2016); Zhang et al. (2016); Erol et al. (2007) for comprehensive reviews on this specific topic.

RGB-based pose estimation. At the early time, edge-based histograms have been used for human pose estimation (Mori and Malik, 2002), and image database indexing techniques have also been used to estimate hand poses (Athitsos and Sclaroff, 2003). Subsequently, part-based models were developed to produce more accurate pose estimations (Pishchulin et al., 2013a,b; Sapp and Taskar, 2013). Recently, with the increasing popularity of deep neural networks, researchers also explored deep neural networks to estimate full-body poses (Toshev and Szegedy, 2014; Carreira et al., 2016). Puwein et al. (2015) proposed a framework for joint camera pose estimation and 3D human pose estimation in a multi-cameras setup.

Depth-based pose estimation. Existing depth-based pose estimation techniques can be generally classified into three categories: *generative*, *discriminative*, and *hybrid*. Generative methods (Gall et al., 2011; Athitsos and Sclaroff, 2003; Ganapathi et al., 2012; Ye and Yang, 2014; Iason Oikonomidis and Argyros, 2011; Tkach et al., 2016; Sinha et al., 2016) estimate poses by fitting a template to the observed data. Discriminative techniques (Shotton et al., 2011; Girshick et al., 2011; Jung et al., 2015, 2016) directly estimate body joint positions, without the assumption of a generative template. Hybrid methods (Ye et al., 2011; Wei et al., 2012; Helten et al., 2013) combine the features of both generative and discriminative methods for pose estimations. Jalal and Kim (2014) presented a real-time tracking system for the pose recognition of body parts, by utilizing the ridge data of depth maps. Oh et al. Oh et al. (2014) proposed to reconstruct 3D full-body poses using wireless camera sensor networks. Another work attempted to solve tracking and recognition from RGB-D video sequences using a feature structured framework Farooq et al. (2015).

Research efforts have also been conducted for articulated objects such as doors or drawers. For example, Sturm et al. (2010) learned the articulated models of cabinet doors and drawers with rectangle detection. Michel et al. (2015) proposed a technique for the pose estimation of kinematic chain instances from RGB-D images.

Pose estimation versus skeleton extraction. It is noteworthy that pose estimation and skeleton extraction are two different research problems. The former aims at estimating poses, usually with the aid of template priors (e.g., a skeleton or body) or supervised learning. However, the purpose of the latter is to extract articulated skeletons consisting of joints and bones.

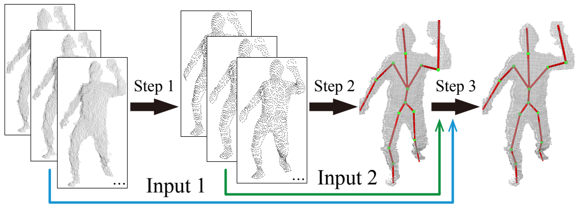


Fig. 1. Pipeline overview of our approach. The black arrows denote three steps of our method: non-rigid point set registration to establish point correspondences, skeletal structure extraction to extract a skeletal structure, and LBS-based joints learning to refine the joints. The green and cyan arrows indicate that we utilize the dual sources (the original input and the matched point sets) for accurate joints learning. See Fig. 4 to clearly observe the inaccurate joints of the output of Step 2.

3. Approach Overview

Our approach consists of the following three main steps: non-rigid point set registration, skeletal structure extraction, and LBS-based skeletal joints learning. Fig. 1 shows the pipeline overview of our approach. Note that we simply set a bounding box for each sequence to remove the interference (backgrounds, etc,.) before the three steps. We use only positions of points rather than features or anything else.

Non-rigid point set registration. To facilitate the motion-based clustering, we first establish point correspondences among frames in the input point set sequence (Section 4).

Skeletal structure extraction. We perform the motion-based clustering and extract a skeletal structure from the *new* point set sequence which is the output of the first step (Section 5). Note that the skeletal structure may contain unnecessary bones and joints which can be removed at this step.

LBS-based skeleton joints learning. To achieve accurate joints, we perform the LBS-based joints learning algorithm in an iterative way, by taking both the original and the registered point set sequences as input (Section 6).

In this work, we use $\mathbf{V}^t = \{\mathbf{v}_i^t\}$ and N^t to denote the positions and the number of the original points at frame t. Let $\mathbf{Y}^t = \{\mathbf{y}_m^t\}$ be the registered points at frame t. F is the number of frames. The total 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 by relating the embedded deformation model (Sumner et al., 2007) with a Gaussian Mixture Model (GMM). We then explain the optimization of this problem with an EM algorithm. Finally we introduce additional soft and hard constraints and present an effective optimization scheme for node transformations at the M-step.

4.1. The Probabilistic Model

Intuitively, the registered point set surface $\mathbf{Y}^t$ should approximate the original point set ($\mathbf{V}^t$) surface. To achieve this, we assume the points $\mathbf{V}^t$ follow a 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 Gaussians, as suggested in Myronenko and Song (2010).

We suppose $\mathbf{y}'_m$ follows the general embedded deformation model (Li et al., 2009; Sumner et al., 2007) which supports to reconstruct unknown complex material behavior and facilitates the registration, due to no prior knowledge on captured objects.

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

$\mathbf{y}'_m$ is the new position induced by its neighboring nodes $\mathbf{n}_j$ with different weights $\bar{\omega}(\mathbf{y}_m, \mathbf{n}_j)$. $\{\mathbf{R}_j, \mathbf{T}_j\}$ is the transformation (i.e., $D \times D$ rotation matrix and $D \times 1$ translation vector) of node $\mathbf{n}_j$. Nodes are extracted uniformly from the registered point set $\mathbf{Y}$, and thus sparsely distributed. Refer to (Sumner et al., 2007; Li et al., 2009) for more details.

The non-rigid registration problem in our work can be formulated as a parameter estimation problem under the above assumptions. We minimize the following negative log-likelihood function (i.e., maximizing the likelihood) to estimate the parameters (i.e., $\{\mathbf{R}_j\}$ and $\{\mathbf{T}_j\}$).

$$E(\{\mathbf{R}_j\}, \{\mathbf{T}_j\}, \sigma^2) = -\log\prod_{i=1}^{N} p(\mathbf{v}_i) \tag{3}$$

4.2. EM Optimization

We use the Expectation-Maximization algorithm (Dempster et al., 1977) for optimization. The E-step is to calculate the posterior probabilities using the old values $\mathbf{Y}$ and σ^2, based on the Bayes' rule. 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). These two steps are alternately called until convergence or reaching a maximum number of iterations.

E-step. Based on the Bayes' theorem, we use the "old" values to calculate the posterior probabilities $p^{old}(\mathbf{y}'_m|\mathbf{v}_i)$.

$$p^{old}(\mathbf{y}'_m|\mathbf{v}_i) = \frac{e^{\frac{-\|\mathbf{v}_i - \mathbf{y}_m\|^2}{2\sigma^2}}}{\sum_{m=1}^{M} e^{\frac{-\|\mathbf{v}_i - \mathbf{y}_m\|^2}{2\sigma^2}} + \frac{(2\pi\sigma^2)^{\frac{D}{2}}\omega M}{(1-\omega)N}} \tag{4}$$

M-step. We estimate the involved parameters ($\{\mathbf{R}_j\}$, $\{\mathbf{T}_j\}$ and σ^2) by minimizing the following complete negative log-likelihood (i.e., upper bound) of Eq. (3).

$$E_{\text{GMM}} = -\sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}\left(\log(\frac{1-\omega}{M}p(\mathbf{v}_i|\mathbf{y}'_m)) + \log\frac{\omega}{N}\right) \tag{5a}$$

$$\propto \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}\|\mathbf{v}_i - \mathbf{y}'_m\|^2 + \frac{DN_p}{2}\log\sigma^2, \tag{5b}$$

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, other constraints (soft and hard) besides E_{GMM} are necessary to be introduced, to meet different demands during registration. Specifically, inspired by Li et al. (2009), we introduce a smooth term E_{smooth} to encourage the transformation of a node to be close to its neighboring nodes.

$$E_{\text{smooth}} = \sum_{\mathbf{n}_j}\sum_{\mathbf{n}_k} \bar{\omega}_{jk}\|\mathbf{R}_j(\mathbf{n}_k - \mathbf{n}_j) + \mathbf{n}_j + \mathbf{T}_j - (\mathbf{n}_k + \mathbf{T}_k)\|^2, \tag{6}$$

where $\mathbf{n}_k$ is a neighbor of $\mathbf{n}_j$ (i.e., $\mathbf{n}_k \in N(\mathbf{n}_j)$) and $\bar{\omega}_{jk}$ is the weight. We set $\bar{\omega}_{jk} = 1$ since we do not find noticeable artifacts compared to other weight computations, which is consistent with Sumner et al. (2007).

we assume small motion changes for all the nodes at each iteration, to better regularize the solution space. Thus the third term can be naturally defined as:

$$E_{\text{small}} = \sum_j \|\mathbf{R}_j - \mathbf{R}_j^{pre}\|_F^2 + \|\mathbf{T}_j - \mathbf{T}_j^{pre}\|^2, \tag{7}$$

where $\mathbf{R}_j$ and $\mathbf{T}_j$ are the new rotation and translation which need to be solved at the current iteration. $\mathbf{R}_j^{pre}$ and $\mathbf{T}_j^{pre}$ are solved at the previous iteration.

In addition, a hard constraint has been imposed to restrict $\mathbf{R}_j$ to be in the group of special orthogonal matrices (i.e., SO(3)). Therefore, the final objective function for the M-step based on all the terms and the SO(3) constraint can be defined as:

$$E = E_{\text{GMM}} + \frac{\beta_{\text{smooth}}}{2}E_{\text{smooth}} + \frac{\beta_{\text{small}}}{2}E_{\text{small}} \tag{8a}$$

$$\text{s.t. } \mathbf{R}_j{}^T\mathbf{R}_j = \mathbf{I}, det(\mathbf{R}_j) = 1, \forall j \tag{8b}$$

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 (Eq. (5b)). Different from the previous motion reconstruction work (Guo et al., 2015; 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. Also, the techniques in (Li et al., 2009; Guo et al., 2015) tend to converge into a local minimum (Fig. 2(a-b)) as they are ICP-based.

We present an efficient scheme to solve node transformations: updating one node transformation by fixing the remaining nodes. To reduce the accumulated errors, we optimize node transformations in a dual way (forward and backward). An example is shown in Fig. 2(c-d). This optimization scheme greatly decreases the complexity of the problem. 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:

$$\mathbf{T}_{\hat{j}} = \mu_v - \mathbf{R}_{\hat{j}}\mu_y, \tag{9}$$

where μ_v and μ_y are $D \times 1$ vectors which can be easily calculated. Note that the specific formulas for such variables (μ_v,

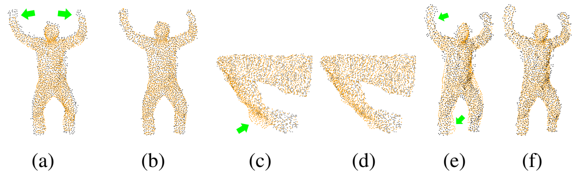


Fig. 2. (a) result of Li et al. (2009), (b) our result. The yellow point set is registered to the black point set. (c) Only forward optimization, (d) the dual-way (forward and backward) optimization; (e) result of Myronenko and Song (2010), (f) our result.

μ_y and some other variables in the following texts) are not provided since they are too lengthy. Please refer to supplemental derivation material.

After substituting Eq. (9) into Eq. (8a) and reorganizing it, we can obtain

$$E = -tr(\mathbf{H}\mathbf{R}_{\hat{j}}) + z, \tag{10}$$

where $tr()$ denotes the trace operation, $\mathbf{H}$ is a $D \times D$ matrix, and z is a scalar.

Lemma 1 (Myronenko and Song, 2009). Let $\mathbf{R}_{D\times D}$ be an unknown rotation matrix and $\mathbf{A}_{D\times D}$ be a known real square matrix. Let $\mathbf{USV}^T$ be a Singular Value Decomposition of $\mathbf{A}$, where $\mathbf{UU}^T = \mathbf{VV}^T = \mathbf{I}$ and $\mathbf{S} = diag(s_i)$, with $s_1 \geq s_2 \geq, ..., \geq s_D \geq 0$. Then the optimal rotation matrix $\mathbf{R}$ that maximizes $tr(\mathbf{A}^T\mathbf{R})$ is $\mathbf{R} = \mathbf{UCV}^T$, where $\mathbf{C} = diag(1, 1, ..., 1, det(\mathbf{UV}^T))$.

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

$$\mathbf{R}_{\hat{j}} = \mathbf{U}_{\hat{j}}\mathbf{C}_{\hat{j}}\mathbf{V}_{\hat{j}}^T, \tag{11}$$

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. (9). Taking the partial derivative with respect to σ^2 and equating it to zero, we can obtain

$$\sigma^2 = \frac{1}{DN_p}\sum_{i=1}^{N}\sum_{m=1}^{M} p_{mi}\|\mathbf{v}_i - \mathbf{y}_m\|^2 \tag{12}$$

We compute the new point positions $\{\mathbf{y}'_m\}$ after each iteration. We summarize the proposed algorithm for non-rigid point set registration in Algorithm 1. Refer to Section 6 (last paragraph) and Section 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. (4)
    M-step:
        • compute $\mathbf{R}_{\hat{j}}$ and $\mathbf{T}_{\hat{j}}$ via Eq. (11) and Eq. (9) for each node $\hat{j}$
        • update $\sigma^2$ via Eq. (12)
        • 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 point set surface from the initial frame 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. We initialize $\mathbf{Y}$ with the registration output in the previous frame. We then register the point sets at the rest frames sequentially.

This algorithm is only one step of our method, however, it is quite different from the previous research (Myronenko and Song, 2010; Ye and Yang, 2014; Cagniart et al., 2010). Specifically, the technique in Myronenko and Song (2010) addresses the motion coherent registration of two single point sets. The accumulated errors of sequential registration are sometimes remarkable (Fig. 2(e-f)). With the aid of a complete skinning mesh template embedded with skeleton, the method in Ye and Yang (2014) estimates poses using a single depth sensor. Within a Bayesian framework, the method in Cagniart et al. (2010) deforms a complete mesh template to fit mesh sequences acquired from multiple cameras. By contrast, we formulate this problem by relating 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 camera 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

At this step, we aim to learn an initial skeleton based on the point correspondence after registration. We first give an introduction on the LBS model which is used in both Section 5 and 6. Then we explain how to cluster body parts based on motion, and we describe skeletal structure generation with the achieved clusters and refinement of the skeletal structure.

5.1. 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 et al. (1988), which can be formulated as follows.

$$\mathbf{x}_m^t = \sum_{j=1}^{B} w_{mj}(\mathbf{R}_j^t\mathbf{q}_m + \mathbf{T}_j^t), \tag{13}$$

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. This model describes the deformation from the rest pose to each frame: the position $\mathbf{x}_m^t$ is controlled by the bones with corresponding weights $\{w_{mj}\}$, rotations $\{\mathbf{R}_j^t\}$ and translations $\{\mathbf{T}_j^t\}$.

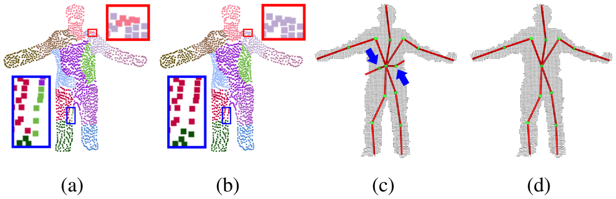


Fig. 3. (a) and (b): without and with improving local continuity, respectively. (c) and (d): without and with skeletal structure refinement.

5.2. Motion-based Clustering

We assume each body part is nearly rigid, to build an initial skeleton from the output sequence $\{\mathbf{Y}^t\}$ obtained from the first step. Precisely, the m-th point is only influenced by a single bone j (i.e., $w_{mj} = 1$). Thus, we can obtain

$$\arg\min_{\mathbf{R}_j^t, \mathbf{T}_j^t} \sum_{m \in clu(n)} \|\mathbf{y}_m^t - (\mathbf{R}_j^t \mathbf{q}_m + \mathbf{T}_j^t)\|^2, \tag{14}$$

where $clu(n)$ denotes the point index set for the n-th cluster (one cluster corresponding to one bone). Eq. (14) is the absolute orientation problem (Kabsch, 1978). To find the best bone transformation for each cluster, we present an iterative update strategy. Specifically,

(i) optimize Eq. (14) 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.

Clustering is achieved by the above strategy, the underlying rationale of which, is that points of the same cluster should have the same label across frames and the same rotation and translation from the rest-pose frame to a certain frame. Note (iii) is to improve the local continuity among points (Fig. 3(b)). Only (i) and (ii) would result in inaccurate clusters, for example, some points of a cluster are far away from the other points in this cluster (Fig. 3(a)). We empirically perform a few iterations (e.g., 10) of the above update scheme.

We employ the K-means clustering as initialization. This is because it is more efficient than other clustering methods (e.g., spectral clustering (Ng et al., 2002) and mean shift clustering (Georgescu et al., 2003)) for such an initialization. Note that it is unnecessary to determine the exact number of clusters at this initialization step, because insignificant bones would be removed later (Section 5.4).

5.3. Skeletal Structure Generation

With the achieved clusters, we generate a graph $\mathbb{G}$ where bones are viewed as nodes. A small edge weight indicates the large probability of two bones sharing a joint (Le and Deng, 2014). Since two connected bones typically have more similar transformations than two unconnected bones, the residuals should be small after interchanging the bone transformations if two bones have a real joint. Specifically, we define the edge weight e_{ij} between bone i and j as

$$\begin{aligned} e_{ij} = & \frac{1}{|clu(i)|} \sum_{t=1}^{F} \sum_{k \in clu(i)} \|\mathbf{y}_k^t - (\mathbf{R}_j^t \mathbf{q}_k + \mathbf{T}_j^t)\|^2 \\ & + \frac{1}{|clu(j)|} \sum_{t=1}^{F} \sum_{k' \in clu(j)} \|\mathbf{y}_{k'}^t - (\mathbf{R}_i^t \mathbf{q}_{k'} + \mathbf{T}_i^t)\|^2, \end{aligned} \tag{15}$$

where $|clu(i)|$ and $|clu(j)|$ are the numbers of points in clusters i and j, respectively.

We compute the minimum spanning tree $\mathbb{S}$ of $\mathbb{G}$ to determine which two bones share a joint. We can easily infer a skeleton tree $\mathbb{S}'$, given $\mathbb{S}$ and the root joint of this skeleton. For visualization and rendering purposes, we set the root joint to be the cluster center which is the shortest to the center of the *rest pose*. We need to compute the initial joint locations by minimizing Eq. (20) in Section 6, to visualize the current skeleton.

5.4. Skeletal Structure Refinement

We refine the produced skeletal structure by removing the unnecessary joints and bones (Fig. 3 (c)-(d)), which is unlike some previous methods (Kirk et al., 2005; Zhang et al., 2013). To obtain a desired skeletal structure, we empirically present the following criteria:

- if a joint connects more than one joints, we search for each next joint, and remove this joint and the associated bone if it is a leaf node;
- since a skeletal structure should not include any loops, we remove the loops if there exist some in the skeleton;
- we merge two neighboring joints if they are very close.

6. LBS-based Skeleton Joints Learning

In this section, we first analyze the issues that lead to inaccurate joints. Then we explain how to formulate the joints learning problem based on the LBS and GMM, and we show how to solve this problem using an EM algorithm. Finally, we introduce new energy terms and describe how to estimate the involved parameters at the M-step.

6.1. Inaccurate Joints

Based on the LBS model (Eq. (13)), 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 leads to inaccurate bone transformations. As a result, the acquired initial joints can be noticeably inaccurate, such as the examples shown in Fig. 4 (a)-(c). As illustrated in Fig. 4(d), it may not be sufficient using only the registered points $\{\mathbf{y}_m^t\}$ for joints learning, as certain useful information may be overlooked by the lacking of the original input $\{\mathbf{v}_i^t\}$.

To overcome these two issues, we present an iterative LBS-based algorithm, which combines both the original input ($\{\mathbf{v}_i^t\}$) and the registered data ($\{\mathbf{y}_m^t\}$) (Section 4) for accurate joints learning.

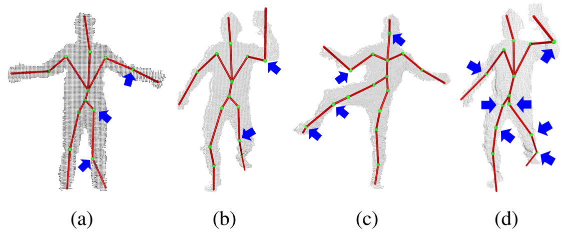


Fig. 4. (a)-(c): several learned skeleton examples after the second step (Section 5). (d): joints learning using only $\{\mathbf{y}_m^t\}$ (Section 6).

6.2. GMM-based Formulation and EM Optimization

The deformed points $\{\mathbf{x}_m^t\}$ should be close to the underlying original point set $\{\mathbf{v}_i^t\}$ at each frame t. We assume 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

$$p(\mathbf{v}_i^t) = (1-\omega')\sum_{m=1}^{M} p(\mathbf{x}_m^t)p(\mathbf{v}_i^t|\mathbf{x}_m^t) + \omega'\frac{1}{N^t} \tag{16}$$

Please refer to Section 4 for the interpretations of similar variables. The approximation problem over all the frames can be regarded as a parameter estimation problem by minimizing the negative log-likelihood function as follows.

$$\mathbb{E}(\mathbf{W},\mathbf{R},\mathbf{T},\tau) = -\log\Big(\prod_{t=1}^{F}\prod_{i=1}^{N^t} p(\mathbf{v}_i^t)\Big), \tag{17}$$

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 Section 4, the EM procedure is also adopted to minimize $\mathbb{E}$. At the E-step, we compute $p^{old}(\mathbf{x}_m^t|\mathbf{v}_i^t)$ using the same form as Eq. (4). For the first iteration, we initialize the deformed points $\{\mathbf{x}_m^t\}$ using the bone transformations in Section 5. At the M-step, the involved parameters are updated. Suppose each frame is independent, $\mathbb{E}_{\text{GMM}}$ can be derived from $\mathbb{E}$.

$$\mathbb{E}_{\text{GMM}} = \sum_{t=1}^{F}\Big(\frac{1}{2\tau^t}\sum_{i=1}^{N^t}\sum_{m=1}^{M} p_{mi}^t\|\mathbf{v}_i^t - \mathbf{x}_m^t\|^2 + \frac{DN_p^t}{2}\log\tau^t\Big), \tag{18}$$

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

We introduce a registration term $\mathbb{E}_{\text{Register}}$ involving $\{\mathbf{y}_m^t\}$ to utilize the registered data (Section 4). We also present a joint term $\mathbb{E}_{\text{Joint}}$ involving joint locations to learn accurate joints.

$$\mathbb{E}_{\text{Register}} = \sum_{t=1}^{F}\sum_{m=1}^{M}\|\mathbf{y}_m^t - \mathbf{x}_m^t\|^2 \tag{19}$$

$$\mathbb{E}_{\text{Joint}} = \eta\sum_{<j,k>\in\mathbb{S}}\|\mathbf{c}_{jk} - \tilde{\mathbf{c}}_{jk}\|^2 + \sum_{t=1}^{F}\sum_{<j,k>\in\mathbb{S}}\|(\mathbf{R}_j^t\mathbf{c}_{jk} + \mathbf{T}_j^t) - (\mathbf{R}_k^t\mathbf{c}_{jk} + \mathbf{T}_k^t)\|^2 \tag{20}$$

where $\tilde{\mathbf{c}}_{jk}$ is the centroid of boundary points between clusters j and k. We include a data constraint (i.e., $\sum_{<j,k>\in\mathbb{S}}\|\mathbf{c}_{jk} - \tilde{\mathbf{c}}_{jk}\|^2$) in $\mathbb{E}_{\text{Joint}}$, because optimizing only the second term in $\mathbb{E}_{\text{Joint}}$ would possibly generate multiple solutions when solving joint positions.

Remarks. 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 (Schaefer and Yuksel, 2007) encourages the deformed points to be close to the registered points, and the latter (Le and Deng, 2014) favors each joint approaching the nearly same deformed positions after two neighboring transformations. Like Section 4, we also assume small motion changes at each iteration.

Thus the final energy $\mathbb{E}_{\text{Total}}$ for the M-step is the weighted sum of $\mathbb{E}_{\text{GMM}}$ (Eq. (18)), $\mathbb{E}_{\text{Register}}$, $\mathbb{E}_{\text{Joint}}$ and $\sum_{t=1}^{F} E_{\text{small}}^t$ (E_{small}^t defined in Eq. (7)).

$$\mathbb{E}_{\text{Total}} = \mathbb{E}_{\text{GMM}} + \frac{\zeta}{2}\mathbb{E}_{\text{Register}} + \frac{\alpha}{2}\mathbb{E}_{\text{Joint}} + \frac{\gamma}{2}\sum_{t=1}^{F} E_{\text{small}}^t \tag{21a}$$

$$\text{s.t. } w_{mj} \geq 0, \sum_{j=1}^{B} w_{mj} = 1, \|\mathbf{W}_{m,:}\|_0 \leq 4, \forall m \tag{21b}$$

$$\mathbf{R}_j^{t\,T}\mathbf{R}_j^t = \mathbf{I}, det(\mathbf{R}_j^t) = 1, \forall t, j \tag{21c}$$

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. (21b)), and the orthogonal constraint is added to bone rotations (Eq. (21c)).

6.4. Parameters Estimation

We now explain how to estimate the involved parameters ($\{\mathbf{c}_{jk}\}$, $\mathbf{W}$, $\mathbf{R}$, $\mathbf{T}$ and τ) at 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 obtain accurate joints, it is also necessary to estimate other involved parameters. To minimize the above total energy $\mathbb{E}_{\text{Total}}$, we present an optimization strategy at the M-step. Specifically, we fix the other parameters when estimating one class of parameters. We employ the optimization scheme presented in Section 4 for bone transformations ($\{\mathbf{R}_j^t, \mathbf{T}_j^t\}$).

Point weights estimation. The weights of a point are independent of those of the other points. Thus, the objective function for the m-th point is

$$\mathbb{E}(\mathbf{W}_{\hat{m},:}) = \sum_{t=1}^{F}\frac{1}{2\tau^t}\sum_{i=1}^{N^t} p_{\hat{m}i}^t\|\mathbf{v}_i^t - \mathbf{x}_{\hat{m}}^t\|^2 + \frac{\zeta}{2}\sum_{t=1}^{F}\|\mathbf{y}_{\hat{m}}^t - \mathbf{x}_{\hat{m}}^t\|^2, \tag{22}$$

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 first 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. (21b)).

Bone transformations estimation. Because of the independence of bone transformations at each frame, we can obtain the following objective function for bone $\hat{j}$ at frame t.

$$\begin{aligned}\mathbb{E}(\mathbf{R}^t_{\hat{j}}, \mathbf{T}^t_{\hat{j}}) = \frac{\zeta}{2}\sum_{m=1}^{M}||\mathbf{y}^t_m - \mathbf{u}^t_{m\hat{j}} - w_{m\hat{j}}(\mathbf{R}^t_{\hat{j}}\mathbf{q}_m + \mathbf{T}^t_{\hat{j}})||^2 + \\ \frac{1}{2\tau^t}\sum_{i=1}^{N^t}\sum_{m=1}^{M} p^t_{mi}||\mathbf{v}^t_i - \mathbf{u}^t_{m\hat{j}} - w_{m\hat{j}}(\mathbf{R}^t_{\hat{j}}\mathbf{q}_m + \mathbf{T}^t_{\hat{j}})||^2 \\ + \frac{\alpha}{2}\sum_{<\hat{j},k>\in\mathbb{S}} ||(\mathbf{R}^t_{\hat{j}}\mathbf{c}_{\hat{j}k} + \mathbf{T}^t_{\hat{j}}) - (\mathbf{R}^t_k\mathbf{c}_{\hat{j}k} + \mathbf{T}^t_k)||^2 \\ + \frac{\gamma}{2}(||\mathbf{R}^t_{\hat{j}} - \mathbf{R}^{pre}_{\hat{j}}||^2_F + ||\mathbf{T}^t_{\hat{j}} - \mathbf{T}^{pre}_{\hat{j}}||^2)\end{aligned} \tag{23}$$

Here, $\mathbf{u}^t_{m\hat{j}} = \sum_{j=1,j\neq\hat{j}}^{B} w_{mj}(\mathbf{R}^t_j\mathbf{q}_m + \mathbf{T}^t_j)$, and $\mathbf{U}^t_{\hat{j}} = \{\mathbf{u}^t_{m\hat{j}}\}$. Taking the partial derivative of $\mathbb{E}(\mathbf{R}^t_{\hat{j}}, \mathbf{T}^t_{\hat{j}})$ with respect to $\mathbf{T}^t_{\hat{j}}$ and equating it to zero, we can obtain

$$\mathbf{T}^t_{\hat{j}} = \mu^t_{u\hat{j}} - \mathbf{R}^t_{\hat{j}}\mu^t_{q\hat{j}}, \tag{24}$$

where $\mu^t_{u\hat{j}}$ and $\mu^t_{q\hat{j}}$ are $D \times 1$ vectors. Substituting Eq. (24) into Eq. (23), we can obtain the objective function involving only $\mathbf{R}^t_{\hat{j}}$, shown as follows.

$$\mathbb{E}(\mathbf{R}^t_{\hat{j}}) = -tr(\mathbf{Z}^t_{\hat{j}}\mathbf{R}^t_{\hat{j}}) + b, \tag{25}$$

where $\mathbf{Z}^t_{\hat{j}}$ is a $D \times D$ matrix and b is a scalar. Similar to Section 4, we yield

$$\mathbf{R}^t_{\hat{j}} = \mathbf{U}^t_{\hat{j}}\mathbf{C}^t_{\hat{j}}\mathbf{V}^{t\,T}_{\hat{j}}, \tag{26}$$

where $\mathbf{U}^t_{\hat{j}}\mathbf{S}^t_{\hat{j}}\mathbf{V}^{t\,T}_{\hat{j}} = svd(\mathbf{Z}^{t\,T}_{\hat{j}})$ and $\mathbf{C}^t_{\hat{j}} = diag(1, 1, ..., det(\mathbf{U}^t_{\hat{j}}\mathbf{V}^{t\,T}_{\hat{j}}))$. $\mathbf{T}^t_{\hat{j}}$ can be computed via Eq. (24).

Joint locations estimation. To estimate joint locations, we minimize $\mathbb{E}_{\text{Total}}$ with respect to $\mathbf{c}_{\hat{j}k}$, which amounts to minimizing the least squares problem (Eq. (20)).

Covariances estimation. We estimate the covariances in a similar way to Section 4 (Eq. (12)).

Deformed points update. The deformed points $\{\mathbf{x}^t_m\}$ are updated using the estimated point weights and bone transformations via the LBS model (Eq. (13)).

We summarize this algorithm in Algorithm 2. We stop the EM procedure for both Algorithm 1 and 2 when the number of iterations is more than 20 or the difference of the total energy between two consecutive iterations is smaller than a threshold. We found that our algorithms typically converge within 20 iterations.

Notice that we extend the GMM to both Section 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 articulated LBS model. We also introduced new energy terms and presented effective optimization schemes for the M-step of both sections.

7. Experimental Results

We have three parts in this section. First, we analyze the effects of energy terms at Step 1 and Step 3. Second, we test different parameter values, and give empirical parameter settings for our approach. Finally, we compare our method with state of the art techniques, both qualitatively and quantitatively.

```
ALGORITHM 2: LBS-based Skeleton Joints Learning
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. (4)
    M-step:
        • estimate weights point by point (Eq. (22))
        • compute $\mathbf{R}^t_{\hat{j}}$ and $\mathbf{T}^t_{\hat{j}}$ via Eq. (26) and Eq. (24) for each
          bone $\hat{j}$ at each frame $t$
        • estimate joint locations (Eq. (20))
        • estimate covariances similarly as Eq. (12)
        • update the deformed points (Eq. (13))
until convergent OR maximum iterations are reached;
```

7.1. *Effects of Energy Terms*

As aforementioned, multiple energy terms have been used at the M-step of both Section 4 and Section 6. There are three terms in Eq. (8) in Section 4: E_{GMM}, E_{smooth} and E_{small}. E_{GMM} can be viewed as the data term, thus we conduct four experiments: (1) the effect of E_{GMM}, (2) the effect of E_{GMM} and E_{smooth}, (3) the effect of E_{smooth} and E_{small}, (4) the effect of E_{GMM}, E_{smooth} and E_{small}. Figure 5 shows that: (1) E_{smooth} controls the smoothness on the point set surface (Fig. 5(b-d)) and its over-contribution would limit the registration (Fig. 5(d)); (2) E_{small} constrains the scale of the transformation in each iteration (Fig. 5(e-g)); (3) combining these terms generates a much better result (Fig. 5(h)).

Eq. (21) in Section 6 has four energy terms: $\mathbb{E}_{\text{GMM}}$, $\mathbb{E}_{\text{Register}}$, $\mathbb{E}_{\text{Joint}}$, and $\sum_{t=1}^{F} E^t_{\text{small}}$. We design the following tests: (1) $\mathbb{E}_{\text{GMM}}$, (2) $\mathbb{E}_{\text{GMM}}$ and $\mathbb{E}_{\text{Register}}$, (3) $\mathbb{E}_{\text{GMM}}$, $\mathbb{E}_{\text{Register}}$ and $\sum_{t=1}^{F} E^t_{\text{small}}$, (4) all terms. We can observe from Figure 8 (a-d) that: (i) only the data terms ($\mathbb{E}_{\text{GMM}}$, $\mathbb{E}_{\text{Register}}$) cannot produce desired joint positions (Fig. 8(a-b)); (ii) introducing $\sum_{t=1}^{F} E^t_{\text{small}}$ is helpful as it restricts small motion at each iteration (Fig. 8(c)); (iii) the incorporation of all terms produces desired results (Fig. 8(d)). The effect of $\mathbb{E}_{\text{Register}}$ has been shown in Figure 4(d). Here we did not provide the effects of $\mathbb{E}_{\text{GMM}}$, $\mathbb{E}_{\text{Register}}$ and $\mathbb{E}_{\text{Joint}}$, which will be tested more in the following subsection.

7.2. *Parameter Tests and Settings*

Eq. (8) in Section 4 involves two parameters: β_{smooth} and β_{small}. Figure 6 illustrates the results when fixing β_{smooth}, and demonstrates the result is becoming smoother with increasing β_{small}. However, the smoothness is undesired when β_{smooth} is too small (Fig. 6(a-d)). Excessive β_{small} means little motion changes (node transformations) at each iteration so that the registration could be inaccurate (6(d,h)). Fig. 7 shows smoother results when β_{smooth} increases. Excessive β_{smooth} also produces poor results (Fig. 7(d)).

Three parameters (ζ, α and γ) are involved in Eq. (21) in Section 6. Fig. 8(e-h) shows that a great α is crucial to favor bones rotating more rigorously around joints. Thus, we devise a

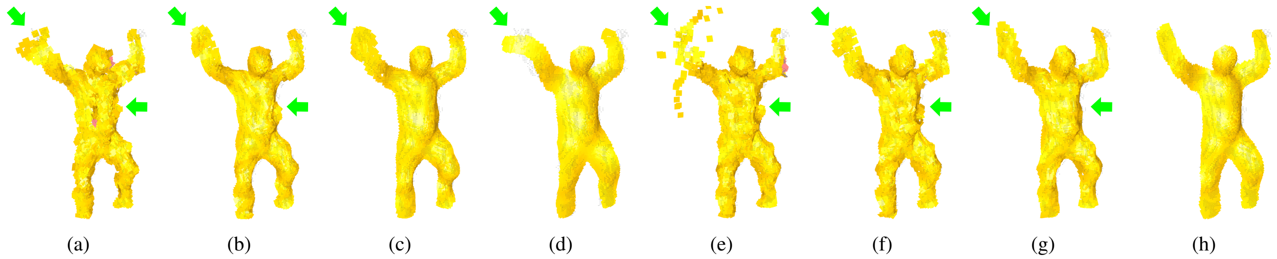


Fig. 5. Effects of energy terms in Eq. (8) **in Section 4.3. (a)** $\beta_{\text{smooth}} = 0, \beta_{\text{small}} = 0$; **(b)** $\beta_{\text{smooth}} = 6 \times 10^2, \beta_{\text{small}} = 0$; **(c)** $\beta_{\text{smooth}} = 6 \times 10^3, \beta_{\text{small}} = 0$; **(d)** $\beta_{\text{smooth}} = 6 \times 10^4, \beta_{\text{small}} = 0$; **(e)** $\beta_{\text{smooth}} = 0, \beta_{\text{small}} = 10^2$; **(f)** $\beta_{\text{smooth}} = 0, \beta_{\text{small}} = 10^3$; **(g)** $\beta_{\text{smooth}} = 0, \beta_{\text{small}} = 10^4$; **(h)** $\beta_{\text{smooth}} = 6 \times 10^3, \beta_{\text{small}} = 10^4$. **Point set surface rendering was used to show the appearance difference.**

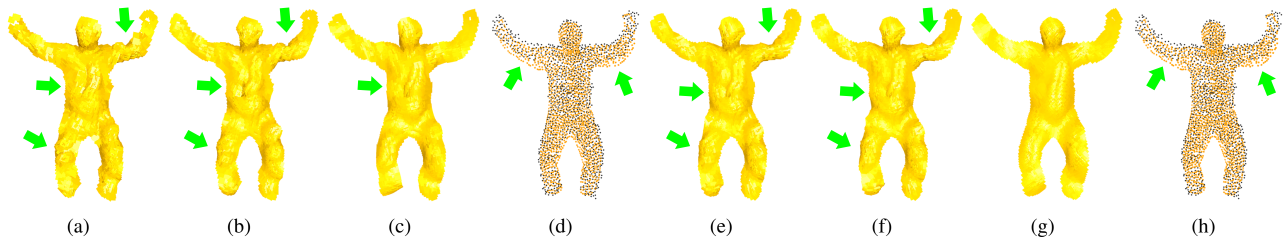


Fig. 6. The tests of β_{small} **in Eq.** (8) **in Section 4.3. (a-d)** $\beta_{\text{smooth}} = 10^2, \beta_{\text{small}} = 0, 10^2, 10^4, 10^6$. **(e-h)** $\beta_{\text{smooth}} = 10^3, \beta_{\text{small}} = 0, 10^2, 10^4, 10^6$. **We render points for (d) and (h) to easily observe the difference.**

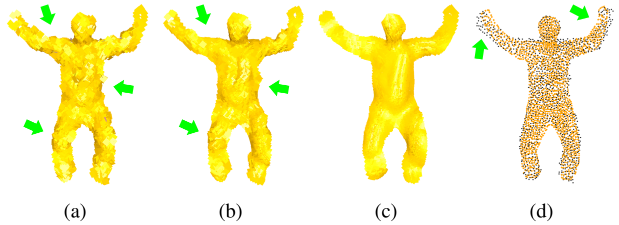


Fig. 7. The tests of β_{smooth} **in Eq.** (8) **in Section 4.3. (a-d):** $\beta_{\text{small}} = 1$, $\beta_{\text{smooth}} = 0, 10^2, 10^4, 10^5$.

strategy by increasing α at each iteration (by default, initialized with ζM and multiplied by 1.45). The justification is that bone transformations are becoming more and more accurate with increased optimization iterations so that $\frac{\alpha}{2}\mathbb{E}_{\text{Joint}}$ should account for the increased impacts of the total energy. Figs. 9 and 10 indicate that α set by the above strategy makes the results insensitive to ζ and γ, due to the increasing significance of $\frac{\alpha}{2}\mathbb{E}_{\text{Joint}}$. Fig. 9(a-e) and Fig. 10(a-c) illustrate different results when setting α to 0. This does not mean that $\frac{\zeta}{2}\mathbb{E}_{\text{Register}}$ and $\frac{\gamma}{2}\sum_{t=1}^{F} E_{\text{small}}^{t}$ are unimportant, since they indeed contribute to the total energy (e.g., Fig. 9(f)) and α is not always large in all iterations.

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

Eq. (8)	$\omega = 0.01, \beta_{\text{smooth}} = 10^5, \beta_{\text{small}} = 1$
Eq. (21)	$\omega' = 0.01, \eta = 1, \zeta = 10^4, \gamma = 0.1$

Parameter settings. To show the robustness of our method, the involved parameters except α (see the above paragraph) in all our experiments are empirically fixed (Table 1). Like Ye and Yang (2014), all σ^2 (Section 4) and $\{\tau^t\}$ are initialized with the same fixed value, 6×10^{-4}. As σ^2 and $\{\tau^t\}$ are generally smaller than 10^{-3}, some regularized weights (β_{smooth} and ζ) are typically large so that they play their respective roles in optimization (see the above parameter tests). We found the results are not sensitive to β_{small} and γ (usually in the range $[0.1, 10^3]$). We suspect it is because of reasonable motion gaps between two continuous frames, and thus simply set them according to Table 1. They can be tuned to be larger when it appears to be common that large motion gaps exist in two neighboring frames.

7.3. Comparison with Existing Methods

7.3.1. 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), as well as the captured Kinect data by ourselves. Besides, we qualitatively and quantitatively compared our approach with the state of the art techniques, respectively labeled as *Method I* (KinectSDK) (Microsoft, 2017), *II* (Kirk et al., 2005) and *III* (Zhang et al., 2013) for simplicity. Ground truth joints are provided by all datasets. PDT and EVAL provide marker data input for Method II. A skeleton pose for each frame of our data (only full body and upper body) is estimated by KinectSDK (Method I), as it is designed only for human poses estimation. Therefore, regarding PDT and EVAL, experiments are conducted using Method II, III and our approach, both qualitatively and quantitatively. We compare Method I, III and our method both qualitatively and quantitatively using our captured data. For fair comparisons, skeletons are learned and rendered by following their works (Method II and III) .

We did not choose to compare our method with (Schaefer and Yuksel, 2007; Hasler et al., 2010; Le and Deng, 2014), because

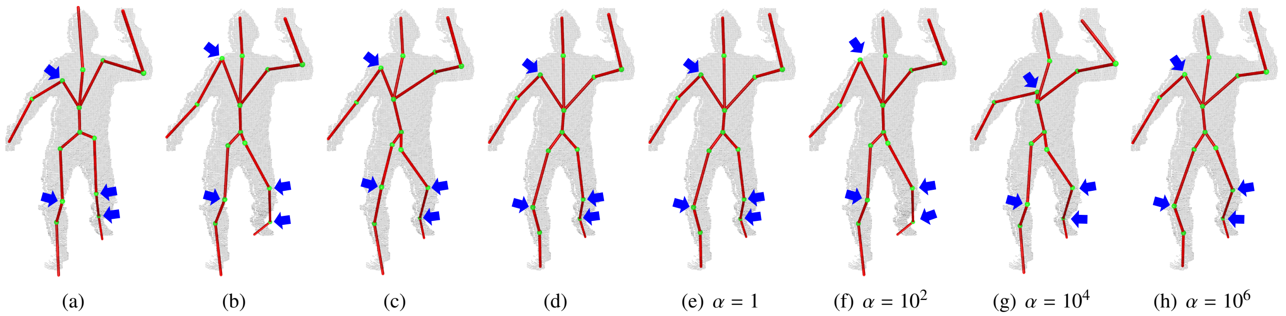

(a) (b) (c) (d) (e) $\alpha = 1$ (f) $\alpha = 10^2$ (g) $\alpha = 10^4$ (h) $\alpha = 10^6$

Fig. 8. (a-d) Effects of energy terms in Eq. (21) **in Section 6.3. (a)** $\zeta = \gamma = \alpha = 0$; **(b)** $\zeta = 10^4$, $\gamma = \alpha = 0$; **(c)** $\zeta = 10^4$, $\gamma = 6 \times 10^3$, $\alpha = 0$; **(d)** $\zeta = 10^4$, $\gamma = 0.1$, α **is set according to Section 7.2. (e-h) The tests of** α **in Eq.** (21). $\zeta = 10^4$ **and** $\gamma = 0.1$.

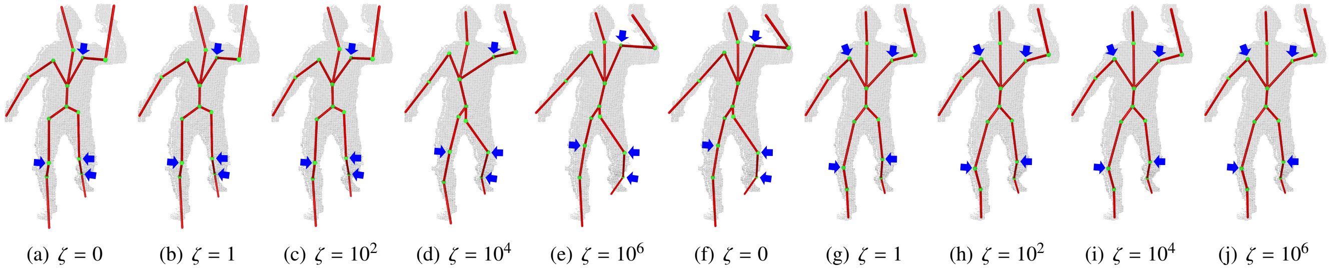

(a) $\zeta = 0$ (b) $\zeta = 1$ (c) $\zeta = 10^2$ (d) $\zeta = 10^4$ (e) $\zeta = 10^6$ (f) $\zeta = 0$ (g) $\zeta = 1$ (h) $\zeta = 10^2$ (i) $\zeta = 10^4$ (j) $\zeta = 10^6$

Fig. 9. The tests of ζ **in Eq.** (21) **in Section 6.3. (a-e)** $\gamma = 6 \times 10^3$, $\alpha = 0$; **(f-j)** $\gamma = 0.1$, α **is set according to Section 7.2.**

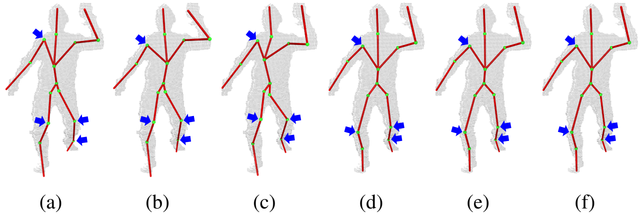

(a) (b) (c) (d) (e) (f)

Fig. 10. The tests of γ **in Eq.** (21) **in Section 6.3. (a-c):** $\zeta = 10^4$, $\alpha = 0$, $\gamma = 10^3, 3 \times 10^3, 10^4$. **(d-f):** $\zeta = 10^4$, α **is set according to Section 7.2 and** $\gamma = 1, 10^2, 10^4$.

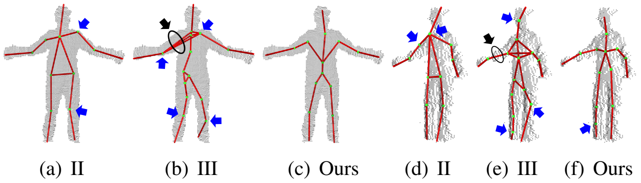

(a) II (b) III (c) Ours (d) II (e) III (f) Ours

Fig. 11. Learned skeletons on two sequences (seq_a_7 and seq_c_1) of EVAL. Blue and black arrows indicate inaccurate and missing joints, respectively. II refers to (Kirk et al., 2005) and III indicates (Zhang et al., 2013).

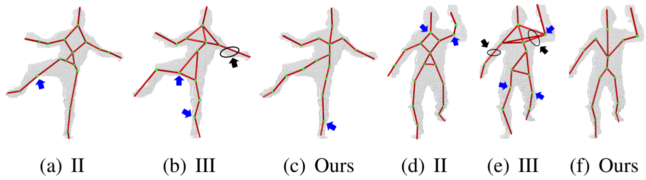

(a) II (b) III (c) Ours (d) II (e) III (f) Ours

Fig. 12. Learned skeletons on two sequences of PDT. II and III refer to (Kirk et al., 2005) and (Zhang et al., 2013), respectively.

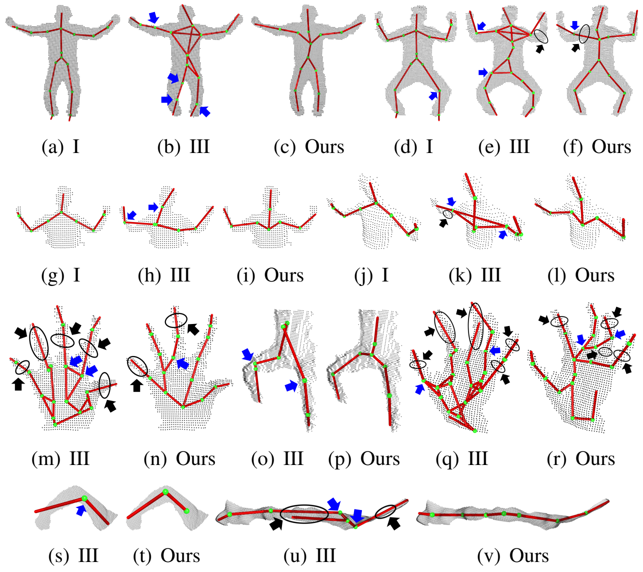
they are designed for mesh sequences which have prior connectivity and correspondence information. We did not compare our method with the methods in EVAL and PDT since they target at the pose tracking of depth images, by parameterizing human poses through the deformation of a given template model (mesh or capsule). We compared, however, our method with KinectSDK (Method I) that uses a prior human skeleton template and is *designed for supervised pose estimation (tracking) rather than skeleton learning or extraction*. The comparison with KinectSDK is to show that the poses of the learned skeletons by our method are even comparable to it.

7.3.2. *Qualitative Comparisons*

We show the visual comparisons between our approach and the state of the art techniques (Method I-III) on various objects (full body: Fig. 11, 12 and 13(a-f), upper body: Fig. 13(g-l), hand: 13(m-n,q-r), lower body: 13(o-p), arm: 13(s-t), and fish: 13(u-v)). Our approach produces substantially higher quality skeletons, compared with Method III (Zhang et al., 2013). Our method can even learn better skeletons (Figs. 11 and 12) than Method II (Kirk et al., 2005), despite their good results are probably mainly due to quality marker input. The poses of our learned skeletons are even comparable with those estimated by Method I (Microsoft, 2017) (see Fig. 13). Note that a few joints in our results are inaccurate, which is normally caused by the relatively small-scale motion of the involved clusters. Please refer to the supplemental document for results of more views.

7.3.3. *Quantitative Comparisons*

With the availability of the ground-truth joints, we compared the accuracies of all the methods. Since different approaches



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

(g) I (h) III (i) Ours (j) I (k) III (l) Ours

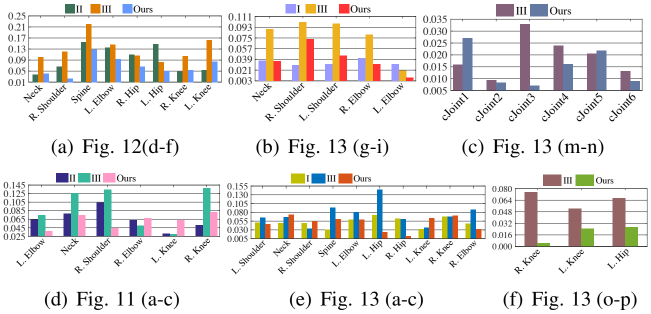
(m) III (n) Ours (o) III (p) Ours (q) III (r) Ours

(s) III (t) Ours (u) III (v) Ours

Fig. 13. Learned skeletons on our Kinect data. I and III denote (Microsoft, 2017) and (Zhang et al., 2013), respectively.

learned different sets of joints, the accuracy evaluations are performed for each tested sequence based on the common subset of joints which are close to semantic positions (e.g., elbows, shoulder) in the body. To estimate the accuracy of the learned skeletons, we use the Euclidean distance error metric used in (Helten et al., 2013; Ye and Yang, 2014). Specifically, we measure the average error over all the joints or the distance error per joint. For all tested sequences, we follow the normalization process (Helten et al., 2013) for each joint, in which the average local displacement relative to the corresponding ground-truth joint was subtracted from the location of the generated joint.

As shown in Table 2, our method outperforms state of the art techniques (Method II (Kirk et al., 2005) and III (Zhang et al., 2013)). It is even comparable to Method I (KinectSDK) (Microsoft, 2017) in terms of pose accuracy, in spite of the obvious benefit from its supervised learning and pre-embedded human skeleton. The distance error per joint for some sequences is demonstrated in Figure 14. On average our method is substantially better than the existing methods, though the per-joint errors by our method are not always smaller than other methods. However, as a major limitation of our approach, it is generally slower than other techniques because of the iterative EM optimization of steps 1 and 3 (e.g., Figure 13(h-i): the runtime for Method III and ours are 110s and 720s, respectively). Our speed depends on the number of frames and the numbers of the original and registered points per frame. It is noteworthy that we did not perform any optimizations to speed up the efficiency of the implementation of our method. GPU-based acceleration and fast Gauss transform (Greengard and Strain, 1991) can be potentially used to facilitate the efficiency of our approach significantly. See further discussion in Section 8.



(a) Fig. 12(d-f) (b) Fig. 13 (g-i) (c) Fig. 13 (m-n)

(d) Fig. 11 (a-c) (e) Fig. 13 (a-c) (f) Fig. 13 (o-p)

Fig. 14. Distance errors (per joint) on some sequences. The unit is meter.

Table 2. The statistics of Euclidean distance errors. The unit is meter. NA denotes "Not Applicable". I, II and III indicate (Microsoft, 2017), (Kirk et al., 2005) and (Zhang et al., 2013), respectively.

Methods Sequences	I	II	III	Ours
Fig. 11(a-c)	NA	0.0657	0.0922	**0.0617**
Fig. 11(d-f)	NA	0.1030	0.1088	**0.0864**
Fig. 12(a-c)	NA	0.1566	0.1205	**0.1124**
Fig. 12(d-f)	NA	0.0780	0.1221	**0.0659**
Fig. 13(a-c)	0.0521	NA	0.0739	**0.0497**
Fig. 13(d-f)	**0.0595**	NA	0.0892	0.0669
Fig. 13(g-i)	**0.0341**	NA	0.0781	0.0389
Fig. 13(j-l)	0.0359	NA	0.0842	**0.0306**
Fig. 13(m-n)	NA	NA	0.0193	**0.0149**
Fig. 13(o-p)	NA	NA	0.0647	**0.0186**
Fig. 13(q-r)	NA	NA	0.0202	**0.0124**
Fig. 13(s-t)	NA	NA	0.0866	**0.0198**
Fig. 13(u-v)	NA	NA	0.0378	**0.0178**

8. Discussion

Besides the experimental results shown above (Section 7), it is necessary to discuss each step of our approach.

First step. As the first step of our method, it aims to build one-to-one point correspondences among the frames in a sequence. Finding correspondences over frames is also a necessity for existing techniques (Zhang et al., 2013; Kirk et al., 2005). Specifically, the method in Zhang et al. (2013) achieves it with the 3D non-rigid matching which is based on the Markov Random Field Deformation Model. However, this method suffers from sparse point clouds and limited matching accuracy led by several-to-one matching. The other technique in Kirk et al. (2005) only needs to find marker correspondences when using passive optical motion capture systems. This is achieved by clustering virtual markers into actual markers. For active systems, a marker's identity is consistent over frames. It relies on the marker position data which can be reliably and accurately generated. We perform non-rigid registration by relating the embedded deformation model (Li et al., 2009; Sumner et al., 2007) with GMM and introducing new constraints. It is designed for point set sequences captured by a single consumer-level depth camera. Our algorithm does not need complete surface or skeleton templates. Please refer to Section 4.3 for the differences between existing related registration techniques and our algorithm. We did not model sequential information (except consecutive frames) which usually increases complexity and computation overhead. The motion differences between consecutive frames are generally reasonable so that our algorithm can produce sufficient output for the later steps. While our ultimate goal is to extract skeletons, a more delicate model that embeds temporal information may be needed for other issues like reconstruction (Livny et al., 2010).

Intermediate step. The goal of this step is to extract a skeletal structure with reasonable amounts of joints and bones. We assume each part is nearly rigid and relate clustering with the LBS model (Magnenat-Thalmann et al., 1988). Then we define a new edge weight function for the cluster graph, and infer the skeleton tree by computing the minimum spanning tree of the cluster graph. We eventually refine the skeletal structure according to a few criteria. By contrast, existing techniques (Zhang et al., 2013; Kirk et al., 2005) simply used a previous clustering method (e.g., spectral clustering (Ng et al., 2002)) and generate the minimum spanning tree to infer the ultimate skeleton. They do not take skeletal structure refinement or accurate joints estimation (Section 6) into account. As a result, the extracted skeleton could involve unreasonable numbers of joints and bones, as well as inaccurate joint locations. We discuss the issue of inaccurate joints in the next paragraph.

Last step. Our final step is for accurate joints estimation, which also uses the LBS model (Magnenat-Thalmann et al., 1988) similar to Step 2. The effects are different: (1) we simplify the LBS model in Step 2 by assuming nearly rigid parts, which results in our motion-based clustering; (2) we use the complete LBS model in Step 3, since the joint locations achieved by Step 2 are not accurate by assuming neither point-bone weight blending in the LBS model nor joint energy terms. For accurate joints estimation, we introduce new energy terms and present an effective optimization scheme (Sections 6.3 and 6.4). By comparison, existing techniques (Zhang et al., 2013; Kirk et al., 2005) simply achieve joint locations based on the clustered points, without considering either specific part transformations or bone-point blending effects (e.g., Linear Blend Skinning). Moreover, the method in Zhang et al. (2013) only uses the matched point sequences to calculate joint positions, without utilizing the original data as complement. The technique in Kirk et al. (2005) depends greatly on the accurate but low spatial-resolution marker data. As the result, the joint locations of their extracted skeletons are limited in both visual quality and accuracy.

Minimizing negative log-likelihood. Both the first and third steps need to minimize the negative log-likelihood, i.e., maximizing the likelihood. Taking Step 1 as example, we show an instance of the changing values of different functions with increasing EM iterations. Fig. 15(a) shows that Eq. (3), Eq. (5a), Eq. (5b) and Eq. (8a) have a similar trend, though their values are different. This is because Eq. (5a) is the complete negative log-likelihood (upper bound) of Eq. (3), and Eq. (5b) is positively proportional to (i.e., ∝) Eq. (5a), and Eq. (5b) is a main component of Eq. (8a).

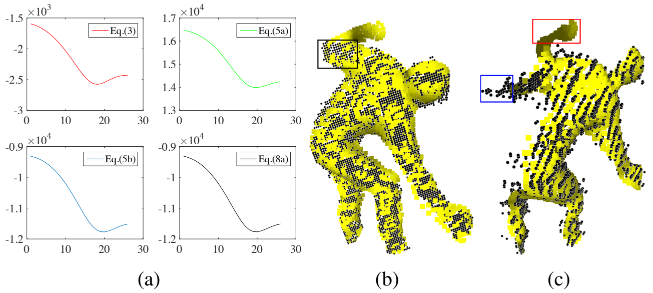


Fig. 15. (a) Different function values with increasing EM iterations in Step 1. (b) A frame with severe arm occlusions. (c) Unsuccessful point set registration in a frame with full arms when a few frames passed after (b). The yellow point set is registered to the black point set.

Pose estimation and skeleton extraction. Pose estimation/tracking from depth data generally fits a template to the observed data or estimates joint locations directly or combines both of them. Since it should be capable of estimating pose from a single frame, existing techniques often depend on either surface/skeleton templates or supervised machine learning. KinectSDK (Microsoft, 2017) is a popular technique that is designed for human pose estimation by relating a prior human skeleton template with the supervised joint prediction (Shotton et al., 2011). In our work, the purpose of comparing to KinectSDK is to demonstrate that the pose of the extracted skeleton by our method is still competitive, although our approach is designed for unsupervised skeleton extraction.

Despite the demonstrated robustness and accuracy of our approach, there still exist a few limitations.

- The non-rigid point set registration algorithm (Section 4) and the LBS-based skeleton joints learning algorithm (Section 6), involve considerable amount of optimization and calculation, and are thus time-consuming. In particular, the Algorithm 1 has been performed sequentially between two consecutive frames, which accounts for a dominating percentage of the total calculation. By contrast, the existing techniques (Zhang et al., 2013; Kirk et al., 2005) need additional time to tune parameters, which is also boring. Microsoft (2017) is for single-frame skeleton based pose estimation and Kirk et al. (2005) operates only on markers which are much fewer than points in point clouds. Thus they can be fast even real-time.

- We have tested our method on different articulated objects and demonstrated its applicability and generality. However, similar to existing techniques, the severe occlusions involved in some data (e.g., quadrupled animals) captured by a single depth sensor also pose extra challenges to our method. Fig. 15(b-c) shows such an example: the severe occlusion of one arm produces inaccurate point correspondences which may lead to poor or unsuccessful skeletons.

9. Conclusion

We present an unsupervised approach for 3D articulated skeleton learning directly from point cloud sequences collected by a single depth camera. This approach, without relying any priors on the captured objects, is robust and accurate in extracting articulated skeletons. We investigated the effects of the energy terms at Step 1 and Step 3. We also tested different values of the used parameters and provided empirical parameter settings. Experimental comparisons show that our method outperforms the state of the art approaches (Kirk et al., 2005; Zhang et al., 2013), in terms of visual quality and quantitative accuracy. Furthermore, the poses of our extracted skeletons are comparable with those by supervised pose estimation techniques like KinectSDK (Microsoft, 2017).

In the future, we could improve the computational efficiency of our approach through fast Gauss transform (Greengard and Strain, 1991) and GPU-based acceleration. We would like to explore how to accurately extract skeletons from more technically challenging data, for example, quadrupled animal data captured by a single depth camera. We would also like to investigate new techniques for efficient and accurate skeleton extraction from articulated objects.

Acknowledgements

Zhigang Deng is in part supported by NSF IIS-1524782. Jun Luo is supported in part by AcRF Tier 2 Grant MOE2016-T2-2-022. Ying He is supported by MOE RG26/17. Sai-Kit Yeung is supported by an internal grant from HKUST (R9429).

References

Anguelov, D., Koller, D., Pang, H.C., Srinivasan, P., Thrun, S., 2004. Recovering articulated object models from 3d range data, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, Virginia, United States. pp. 18–26.

Athitsos, V., Sclaroff, S., 2003. Estimating 3d hand pose from a cluttered image, in: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, IEEE. pp. II–432.

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

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

Cagniart, C., Boyer, E., Ilic, S., 2010. Probabilistic deformable surface tracking from multiple videos, in: Proceedings of the 11th European Conference on Computer Vision, Springer-Verlag, Berlin, Heidelberg. pp. 326–339.

Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J., 2016. Human pose estimation with iterative error feedback, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4733–4742.

Chang, H.J., 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 (CVPR), pp. 3138–3146.

Cheung, G.K.M., Baker, S., 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, IEEE Computer Society, Washington, DC, USA. pp. 77–84.

Chui, H., Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding 89, 114 – 141.

Chun, C.W., Jenkins, O.C., 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, IEEE. pp. II–475.

De Aguiar, E., Theobalt, C., Thrun, S., Seidel, H.P., 2008. Automatic conversion of mesh animations into skeleton-based animations. Computer Graphics Forum 27, 389–397.

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

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S.R., Kowdle, A., Escolano, S.O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., Izadi, S., 2016. Fusion4d: Real-time performance capture of challenging scenes. ACM Trans. Graph. 35, 114:1–114:13.

Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X., 2007. Vision-based hand pose estimation: A review. Computer Vision and Image Understanding 108, 52 – 73.

Farooq, A., Jalal, A., Kamal, S., 2015. Dense rgb-d map-based human tracking and activity recognition using skin joints features and self-organizing map. doi:10.3837/tiis.2015.05.017.

Gall, J., Fossati, A., van Gool, L., 2011. Functional categorization of objects using real-time markerless motion capture, in: CVPR 2011, pp. 1969–1976.

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

Georgescu, B., Shimshoni, I., Meer, P., 2003. Mean shift based clustering in high dimensions: A texture classification example, in: ICCV.

Ghorbel, E., Boonaert, J., Boutteau, R., Lecoeuche, S., Savatier, X., 2018. An extension of kernel learning methods using a modified log-euclidean distance for fast and accurate skeleton-based human action recognition. Computer Vision and Image Understanding URL: http://www.sciencedirect.com/science/article/pii/S1077314218302649, doi:https://doi.org/10.1016/j.cviu.2018.09.004.

Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A., 2011. Efficient regression of general-activity human poses from depth images, in: Proceedings of the 2011 International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA. pp. 415–422.

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

Guo, K., Xu, F., Wang, Y., Liu, Y., Dai, Q., 2015. Robust non-rigid motion tracking and surface reconstruction using l0 regularization, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3083–3091.

Han, F., Reily, B., Hoff, W., Zhang, H., 2017. Space-time representation of people based on 3d skeletal data: A review. Computer Vision and Image Understanding 158, 85 – 105. URL: http://www.sciencedirect.com/science/article/pii/S1077314217300279, doi:https://doi.org/10.1016/j.cviu.2017.01.011.

Hasler, N., Thormählen, T., Rosenhahn, B., Seidel, H.P., 2010. Learning skeletons for shape and pose, in: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ACM, New York, NY, USA. pp. 23–30.

Helten, T., Baak, A., Bharaj, G., Muller, M., Seidel, H.P., 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, IEEE Computer Society, Washington, DC, USA. pp. 279–286.

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

Iason Oikonomidis, N.K., Argyros, A., 2011. Efficient model-based 3d tracking of hand articulations using kinect, in: Proceedings of the British Machine Vision Conference, BMVA Press. pp. 101.1–101.11. Http://dx.doi.org/10.5244/C.25.101.

Jalal, A., Kamal, S., 2014. Real-time life logging via a depth silhouette-based human activity recognition system for smart home services, in: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 74–80. doi:10.1109/AVSS.2014.6918647.

Jalal, A., Kamal, S., Kim, D., 2014. A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. Sensors 14, 11735–11759.

Jalal, A., Kim, Y., 2014. Dense depth maps-based human pose tracking and recognition in dynamic scenes using ridge data, in: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 119–124. doi:10.1109/AVSS.2014.6918654.

Jalal, A., Kim, Y.H., Kim, Y.J., Kamal, S., Kim, D., 2017. Robust human activity recognition from depth video using spatiotemporal multi-fused features. Pattern Recognition 61, 295 – 308. URL: http://www.sciencedirect.com/science/article/pii/S0031320316302126, doi:https://doi.org/10.1016/j.patcog.2016.08.003.

Jung, H.Y., Lee, S., Heo, Y.S., Yun, I.D., 2015. Random tree walk toward instantaneous 3d human pose estimation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2467–2474.

Jung, H.Y., Suh, Y., Moon, G., Lee, K.M., 2016. A Sequential Approach to 3D Human Pose Estimation: Separation of Localization and Identification of Body Joints. Springer International Publishing, Cham. pp. 747–761.

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, 827–828.

Kirk, A.G., O'Brien, J.F., 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 (CVPR'05), IEEE Computer Society, Washington, DC, USA. pp. 782–788.

Le, B.H., Deng, Z., 2012. Smooth skinning decomposition with rigid bones. ACM Transactions on Graphics (TOG) 31, 199.

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

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

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., El-Sana, J., 2010. Automatic reconstruction of tree skeletal structures from point clouds. ACM Trans. Graph. 29, 151:1–151:8.

Lu, X., Chen, H., Yeung, S.K., Deng, Z., Chen, W., 2018. Unsupervised articulated skeleton extraction from point set sequences captured by a single depth camera, in: AAAI.

Ma, J., Zhao, J., Yuille, A.L., 2016. Non-rigid point set registration by preserving global and local structures. IEEE Transactions on Image Processing 25, 53–64.

Magnenat-Thalmann, N., Laperrière, R., Thalmann, D., 1988. Joint-dependent local deformations for hand animation and object grasping, in: Proceedings on Graphics Interface '88, Canadian Information Processing Society, Toronto, Ont., Canada, Canada. pp. 26–33.

Michel, F., Krull, A., Brachmann, E., Yang, M.Y., Gumhold, S., Rother, C., 2015. Pose estimation of kinematic chain instances via object coordinate regression, in: Proceedings of the British Machine Vision Conference (BMVC), BMVA Press. pp. 181.1–181.11.

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

Mori, G., Malik, J., 2002. Estimating Human Body Configurations Using Shape Context Matching. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 666–680.

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

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

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

Oh, H., Cha, G., Oh, S., 2014. Samba: A real-time motion capture system using wireless camera sensor networks. Sensors 14, 5516–5535.

Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B., 2013a. Poselet conditioned pictorial structures, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595.

Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B., 2013b. Strong appearance and expressive spatial models for human pose estimation, in: 2013 IEEE International Conference on Computer Vision, pp. 3487–3494.

Puwein, J., Ballan, L., Ziegler, R., Pollefeys, M., 2015. Joint camera pose estimation and 3d human pose estimation in a multi-camera setup, in: Cremers, D., Reid, I., Saito, H., Yang, M.H. (Eds.), Computer Vision – ACCV 2014, Springer International Publishing, Cham. pp. 473–487.

Qu, H.B., Wang, J.Q., Li, B., Yu, M., 2017. Probabilistic model for robust affine and non-rigid point set matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 371–384.

Raghuraman, S., Venkatraman, K., Wang, Z., Prabhakaran, B., Guo, X., 2013. A 3d tele-immersion streaming approach using skeleton-based prediction, in: Proceedings of the 21st ACM International Conference on Multimedia, ACM, New York, NY, USA. pp. 721–724. URL: http://doi.acm.org/10.1145/2502081.2502188, doi:10.1145/2502081.2502188.

Ramanan, D., Forsyth, D.A., Barnard, K., 2006. Building models of animals from video. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1319–1334.

Ross, D.A., Tarlow, D., Zemel, R.S., 2008. Unsupervised Learning of Skeletons from Motion. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 560–573.

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

Rossi, L., Torsello, A., 2014. Coarse-to-fine skeleton extraction for high resolution 3d meshes. Computer Vision and Image Understanding 118, 140 – 152. URL: http://www.sciencedirect.com/science/article/pii/S1077314213001938, doi:https://doi.org/10.1016/j.cviu.2013.10.006.

Sapp, B., Taskar, B., 2013. Modec: Multimodal decomposable models for human pose estimation, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3674–3681.

Sarafianos, N., Boteanu, B., Ionescu, B., Kakadiaris, I.A., 2016. 3d human pose estimation: A review of the literature and analysis of covariates. Computer Vision and Image Understanding 152, 1 – 20.

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

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., 2011. Real-time human pose recognition in parts from single depth images, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA. pp. 1297–1304.

Sinha, A., Choi, C., Ramani, K., 2016. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4150–4158.

Song, Y., Tang, J., Liu, F., Yan, S., 2014. Body surface context: A new robust feature for action recognition from depth videos. IEEE Transactions on Circuits and Systems for Video Technology 24, 952–964. doi:10.1109/TCSVT.2014.2302558.

Sturm, J., Konolige, K., Stachniss, C., Burgard, W., 2010. Vision-based detection for learning articulation models of cabinet doors and drawers in household environments, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 362–368.

Sumner, R.W., Schmid, J., Pauly, M., 2007. Embedded deformation for shape manipulation. ACM Trans. Graph. 26.

Tagliasacchi, A., Zhang, H., Cohen-Or, D., 2009. Curve skeleton extraction from incomplete point cloud. ACM Trans. Graph. 28, 71:1–71:9.

Tam, G.K.L., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L., 2013. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. IEEE Transactions on Visualization and Computer Graphics 19, 1199–1217.

Tkach, A., Pauly, M., Tagliasacchi, A., 2016. Sphere-meshes for real-time hand modeling and tracking. ACM Trans. Graph. 35, 222:1–222:11.

Toshev, A., Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660.

Tresadern, P., Reid, I., 2005. Articulated structure from motion by factorization, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE Computer Society, Washington, DC, USA. pp. 1110–1115.

Wei, X., Zhang, P., Chai, J., 2012. Accurate realtime full-body motion capture using a single depth camera. ACM Trans. Graph. 31, 188:1–188:12.

Yan, J., Pollefeys, M., 2006. Automatic kinematic chain building from feature trajectories of articulated objects, in: CVPR.

Yan, J., 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, 865–877.

Yang, A.Y., Iyengar, S., Sastry, S., Bajcsy, R., Kuryloski, P., Jafari, R., 2008. Distributed segmentation and classification of human actions using a wearable motion sensor network, in: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8. doi:10.1109/CVPRW.2008.4563176.

Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M., 2011. Accurate 3d pose estimation from a single depth image, in: 2011 International Conference on Computer Vision, pp. 731–738.

Ye, M., 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, IEEE Computer Society, Washington, DC, USA. pp. 2353–2360.

Zhang, H.B., Lei, Q., Zhong, B.N., Du, J.X., Peng, J., 2016. A survey on human pose estimation. Intelligent Automation & Soft Computing 22, 483–489.

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

Zhang, S., Yang, Y., Xiao, J., Liu, X., Yang, Y., Xie, D., Zhuang, Y., 2018.

Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. IEEE Transactions on Multimedia 20, 2330–2343. doi:10.1109/TMM.2018.2802648.