Regularity Guaranteed Human Pose Correction

Wei Shen, Rui Lei, Dan Zeng, Zhijiang Zhang

School of Communication and Information Engineering, Shanghai University, 149 Yanchang Road, Shanghai 200072, P.R. China
{wei.shen,cici_lr,dzeng,zjzhang}@shu.edu.cn

Abstract. Benefited from the advantages provided by depth sensors, 3D human pose estimation has become feasible. However, the current estimation systems usually yield poor results due to severe occlusion and sensor noise in depth data. In this paper, we focus on a post-process step, pose correction, which takes the initial estimated poses as the input and deliver more reliable results. Although the regression based correction approach [1] has shown its effectiveness in decreasing the estimated errors, it cannot guarantee the regularity of corrected poses. To address this issue, we formulate pose correction as an optimization problem, which combines the output of the regression model with a pose prior model learned on a pre-captured motion data set. By considering the complexity and the geometric property of the pose data, the pose prior is estimated by von Mises-Fisher distributions in subspaces following divide-and-conquer strategies. By introducing the pose prior into our optimization framework, the regularity of the corrected poses is guaranteed. The experimental results on a challenging data set demonstrate that the proposed pose correction approach not only improves the accuracy, but also outputs more regular poses, compared to the-state-of-the-art.

1 Introduction

With the invention of the low-cost high-speed depth sensor, such as Microsoft Xbox Kinect [2], marker-less human pose estimation from depth images becomes an increasingly active research topic in recent years [3]. By taking the advantages provided by depth data, including color and texture invariance and easy background subtraction, many reliable human pose estimation systems have been built, which output the locations of a certain number of joints to form the human skeleton. The estimated skeletons have been put into service in many real applications, especially the gaming industry. Although considerable progress and success have been achieved [4–9], pose estimation from a single depth image remains a challenging task. Various uncontrolled factors, such as occlusion, sensor noise and large articulation variation, may result in serious failures when estimating poses.

To achieve more robust pose estimation, several researchers perform an additional pose correction step which aims to recover poses from failure. The research devoted to this particular step usually makes use of the pose priors modeled from a motion capture data set and can be grouped into two broad categories: nearest

neighbor (NN) search based approaches [6, 10] and regression based approaches [1, 11]. The former refines an initial estimated skeleton by directly merging it with the body configurations of its nearest neighbors in the motion capture data set. Such approaches can ensure that the corrected skeleton is regular. However, due to the significant differences between estimated skeletons and motion capture data, the merging procedure may decrease the accuracy of pose estimation. While the latter learns a regression function mapping a initial estimated skeleton to a corrected one by considering the systematic bias existing in the estimation step. Although regression based approaches are effective to improve the accuracy of pose estimation and efficient for both training and testing, they cannot guarantee the regularity of the corrected skeleton, as the errors across each joint are not homogeneous. For a better demonstration, we illustrate several pose examples belonging to a specific action domain, golf swing [1], in Fig. 1, in each of which we see the depth data, the ground truth skeleton, the initial estimated skeleton, the corrected skeleton obtained by a NN search based approach and the one obtained by a regression based approach [1]. Note that, the estimated skeletons obtained by the NN search based method are somewhat regular poses belonging to the action domain of golf swing. However, they can barely match their ground truths due to the serious errors existing in the initial estimated skeletons (as shown in the first row in Fig. 1) or the differences between the body configurations of different individuals (as shown in the second row in Fig. 1). On contrast, although the accuracy of the corrected skeletons obtained by the regression based approach (measured by the sum of joint errors) is obviously higher, they are quite weird w.r.t the action domain of golf swing. Consequently, neither of these two schemes are satisfactory in all aspects.

In this paper, we are concerned with the step of pose correction and try to address the problems discussed above. Toward this end, following hybrid strategies we formulate pose correction as an optimization problem by combining the output of the regression model proposed in [1] with a pose prior model learned on a pre-captured motion data set. Formulating problems by hybrid terms and inferring by optimization is a principal way for many vision tasks, such as object discovery [12] and manifold denoising [13]. In our case, the regression model provides the distribution of corrected skeletons on the searching space, while the pose prior model introduces a constraint to guarantee the regularity of the corrected poses. The data pose belonging to the action domain with high-speed motion, such as golf swing, is associated with complex data manifolds. Therefore, to learn a reliable pose prior model, data partition is first performed, and then the distribution of the pose data is modeled in each partitioned clusters. This is a divide-and-conquer strategy [14]. In order to eliminate the differences between the body configurations of different individuals, our pose prior is not modeled in the world coordinate space, but in a normalized skeleton feature space instead. By considering the intrinsic geometrical property of the skeleton feature, a new similarity measure is defined to cluster the skeleton data and a generative model based on the von Mises-Fisher distribution [15, 16] is proposed to compute the pose prior. Consequently, by integrating a regression model, e.g., the random

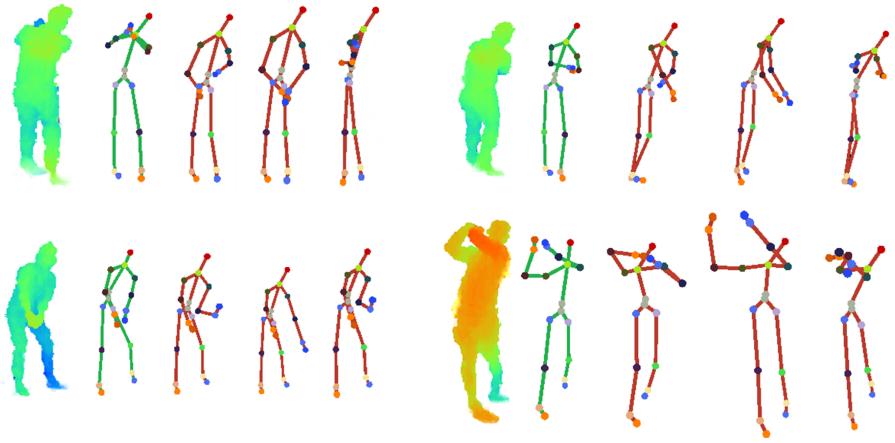

Fig. 1. Several pose examples selected from a golf swing data set [1]. In each example, we see the depth image, the ground truth skeleton, the initial estimated skeleton, the corrected skeleton obtained by a NN search based approach and the one obtained by a regression based approach. Note that, although the former corrected skeletons are regular, they are quite different from the ground truths. Conversely, the latter ones are more close to the ground truths, but they are not regular poses.

regression forest [1], with our prior model, the accuracy and regularity of the corrected poses can be guaranteed simultaneously.

Our contributions can be summarized into two aspects as follow. First, we propose an optimization framework for pose correction. Unlike the others who often take the temporal constraint into account, we consider the pose prior as a regularized term in our framework. This is attributed to the reason that the temporal constraint is usually not reliable in the actions with high-speed motion. Second, following the spirit of divide-and-conquer strategies, we propose a distribution estimation method by considering the intrinsic geometrical property of the skeleton data to properly model the pose prior.

The remainder of this paper is organized as follows. Sec. 2 reviews the related work to human pose estimation and correction from depth images. Sec. 3 introduces the skeleton data and ground truth used in our work as well as the procedure of data preprocessing. Sec. 4 describes the proposed approach to pose correction in detail, including the formulation and the inference. Experimental results on a challenge data set are given in Sec. 5. Finally, we draw the conclusion in Sec. 6.

2 Related Work

As one of the most active topics in computer vision, human pose estimated from depth images has attracted a lot of attention from the community [4–9]. A comprehensive survey on this topic can be found in [3]. Plagemann *et al.* [17] present

a novel interesting point detector for depth data, which provides the candidate proposals for body parts, such as hand, foot and head, as they think the detected interesting points coincide with the salient points of the body. Then, the body parts can be identified and localized by applying a boosted classifier learned on the local shape descriptors extracted on the detected interesting points. Shotton *et al.* [4] present a remarkable work, which formulates pose estimation from a single depth image as a per-pixel body part classification problem. They apply randomized decision trees [18, 19] to effectively inference the distribution of body parts, followed by estimating hypotheses of body joint positions by seeking the modes in the distribution. In order to handle the obstacle of occlusion, Girshick *et al.* [8] propose to predict the offset between each pixel and each joint by regression. The body joint positions are then estimated by aggregating of the weighted offset votes offered by relative pixels. Sun *et al.* [9] improve Girshick's method by learning the regression model conditioned on several global parameters, such as height and torso orientation. Hybrid strategies that combine generative and discriminative methods have proven to be a suitable methodology for pose estimation [5, 7, 20]. For example, Baak *et al.* [7] propose a data-driven hybrid strategy to optimize two hypotheses to yield the final pose, where the first one is retrieved from a 3D pose data set using sparse features extracted from depth data and the second one is generated based on the previous frame. Although we also follow hybrid strategies, our method differs from those in objective (pose estimation *vs* pose correction) and formulation (temporal constraint *vs* pose prior).

As the initial estimated pose usually yields poor results, the step of pose correction is vital [3]. Ye *et al.* [6] propose to perform pose correction through non-rigid registration between an estimated pose and its nearest neighbors searched from a motion capture data set. Shum *et al.* [10] also propose a NN search based pose correction method, in which a reliability confidence for each body part is defined to reweight the distances between poses. Shen *et al.* [1] propose to learn the systematic bias existing in the pose estimation stage by random forest regression to perform pose correction. Afterwards, they show that the performance can be further improved by learning the regression function conditioned on well-partitioned pose subspaces [11]. Although we also use the regression model, our pose prior is devoted for pose regularization and learned in an unsupervised manner. While in [11], the pose subspaces are obtained according to human annotated pose tags.

Pose prior models are quite general in 2D human pose estimation. Most of recent methods based on pictorial structure model [21] use a pairwise term that evaluates how each estimated pose fits with the pose prior model acquired by training data [22–27]. The proposed pose prior model is different from them. First, unlike those methods, e.g. Bayesian Network Prior model [28], which model a specific pose, such as wave, we model the distribution of the poses belonging to a domain-specific action which includes many types of poses, e.g., the action golf swing includes swing, hitting, standing etc. Therefore, we design a divide-and-conquer strategy for proper modeling. Second, we prefer using von Mises-Fisher

distribution to model the prior on body part configuration rather than Gaussian distribution [24] by considering the geometric property of the skeleton feature.

3 Acquisition and Data Preparation

Skeleton estimation. The Kinect camera is able to construct 640×480 depth images at 30 frames per second. In addition, the algorithm proposed in [4] has been ported into the XBOX Kinect SDK [2], which offers an advanced skeleton estimator for depth images in realtime. The human skeleton obtained by the XBOX Kinect SDK is the direct input for our approach and is called ES (**E**stimated **S**keleton) for short in the rest of this paper. As shown in Fig. 2(a), a skeleton consists of rigid bones connecting a certain number of body joints. In our experiments, we are concerned with 20 body joints: hips, spine, shoulders, head, elbows, wrists, hands, knees, ankles, and feet.
Ground truth. The recent works [4, 8] suggest that the ground truth body joint positions can be recorded by marker-based motion capture (mocap) systems. By calibrating the mocap data to match the Kinect sensor, the **G**round truth **S**keleton (GS) for the ES from the Kinect depth data is obtained. Several examples of triplets of the depth image, the ES and the GS are shown in Fig. 1.
Skeleton feature computation. To remove the global translation and the variation caused by individual body differences from skeleton data, we adopt the normalized coordinates proposed in [1] as the skeleton feature. We briefly review this skeleton feature computation process here. Given a human skeleton $\mathcal{S} = (\mathbf{x}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$, where $\mathbf{x}_i \in \mathbb{R}^3$ is the world coordinates of the j-th joints and n is the number of joints in the skeleton ($n = 20$ in this paper). Through forward kinematics [29], the skeleton joint C. Hip is selected as the root, and then the bones between the root and any other one joint form a kinematics chain. Such a chain determines an order for skeleton joints. For any two joints connected by a rigid bone, the one which is closer to the root in the kinematics chain is called the predecessor of the other, e.g., ankles and elbows are the predecessors of feet and wrists, respectively. The skeleton feature of $\mathcal{S}$ is defined according to kinematics, denoted by $h(\mathcal{S}) = (\mathbf{r}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$, where

$$\mathbf{r}_j = \begin{cases} (0,0,0)^{\mathrm{T}}, & j = 1 \\ \dfrac{\mathbf{x}_j - \mathbf{x}_{j_o}}{\|\mathbf{x}_j - \mathbf{x}_{j_o}\|_2}, & j = 2, \ldots, n \end{cases}, \tag{1}$$

In Eq. 1, $\mathbf{r}_1$ is the normalized coordinates of the root which is always on the origin (so that we drop it in our implementation) and j_o is the joint index of the predecessor of j-th joint. For notation simplicity, we define $h_j(\mathcal{S}) = \mathbf{r}_j$.

The computation of the skeleton feature $h(\mathcal{S})$ is mapping the skeleton joints to a 3-dimensional unit sphere, as shown in Fig. 2(b), which actually represents the directions of the rigid bones in the skeleton.

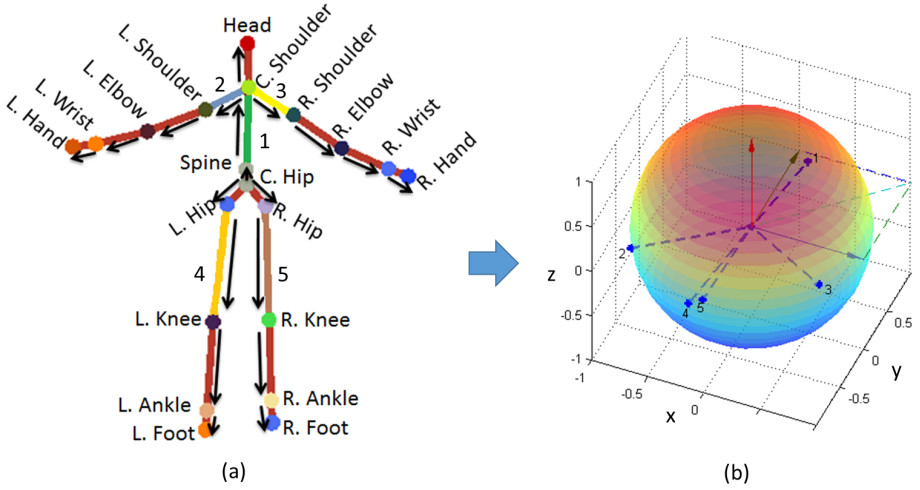

Fig. 2. (a) A skeleton and its joints. The C. Hip is the root of the kinematics chains formed by joints (marked by arrows). (b) An illustration for the skeleton feature computation. For better visualization, only 5 joints are shown in the unit sphere.

4 Problem Formulation

In this section, we give the formulation of our approach. Given a training data set $\{(\mathcal{E}_i, \mathcal{G}_i)\}_{i=1}^{N}$, where $\mathcal{G}_i$ is the ground truth of $\mathcal{E}_i$, our input is an ES $\mathcal{E} = (\hat{\mathbf{x}}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$ with n skeleton joints estimated from a single depth image by using XBOX Kinect SDK [2], and the goal is to predict the true position $\hat{\mathbf{x}}_i \rightarrow \mathbf{x}_i \in \mathbb{R}^3$ of each joint and obtain the GS $\mathcal{G} = (\mathbf{x}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$.

4.1 A Naive Bayesian Formulation

Given a input ES $\mathcal{E}$, to predict its true skeleton GS $\mathcal{G}$ is not easy, as the bias between the ES and the GS is non-linear. A solution to address this problem is to find the **C**orrected **S**keleton (CS) $\mathcal{C} = (\mathbf{z}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$ that maximize the probability of $\mathcal{C}$ given $\mathcal{E}$:

$$\hat{\mathcal{C}} = \arg\max_{\mathcal{C}} p(\mathcal{C}|\mathcal{E}) = \arg\max_{\mathcal{C}} p(\mathcal{E}|\mathcal{C})p(\mathcal{C}), \tag{2}$$

where $p(\mathcal{E}|\mathcal{C})$ is the pose likelihood and $p(\mathcal{C})$ is the pose prior according to the Bayesian inference framework. Modeling the pose likelihood $p(\mathcal{E}|\mathcal{C})$ is quite intractable. Although a naive solution could model it by directly computing the similarity between $\mathcal{E}$ and $\mathcal{C}$, as $\mathcal{C}$ is refined from $\mathcal{E}$, not only should they be somewhat similar, but the the errors existing in $\mathcal{E}$ should be decreased in $\mathcal{C}$. Consequently, no existing generative models are able to guarantee this property. We therefore choose an alternative perspective which attempts to directly approximate the posterior probability $p(\mathcal{C}|\mathcal{E})$.

4.2 An Alternative Perspective

We can express the log posterior distribution $-\log p(\mathcal{C}|\mathcal{E})$ as an energy function $E(\mathcal{C};\mathcal{E})$, which is defined by

$$E(\mathcal{C};\mathcal{E}) = E_m(\mathcal{C};\mathcal{E},\{(\mathcal{E}_i,\mathcal{G}_i)\}_{i=1}^N) + \lambda E_r(\mathcal{C};\{\mathcal{G}_i\}_{i=1}^N), \tag{3}$$

where $E_m(\mathcal{C};\mathcal{E},\{(\mathcal{E}_i,\mathcal{G}_i)\}_{i=1}^N)$ is the mapping energy function between $\mathcal{C}$ and $\mathcal{E}$, $E_r(\mathcal{C};\{\mathcal{G}_i\}_{i=1}^N)$ defines a pose prior to guarantee the regularity of $\mathcal{C}$, and λ is a weight factor between these two terms. Both of these two terms are leaned from the given training data set $\{(\mathcal{E}_i,\mathcal{G}_i)\}_{i=1}^N$. Next, we will explicitly describe how to learn them respectively.

Learning Mapping Energy Instead of direct learning the mapping function from $\mathcal{E}$ to $\mathcal{C}$, we model the bias between them: $f:\mathcal{E}\rightarrow\Delta$[1], where $\Delta=\mathcal{C}-\mathcal{E}$. The reason lies in two folds: first, the biases between ESs and GSs are somewhat systematical, so that they are predicable; second, such systematical biases naturally map ESs into certain clusters on the data manifold, while directly approaching a GS requires exploring all possible ESs in the data space. Thus, we can rewrite $E_m(\mathcal{C};\mathcal{E},\{(\mathcal{E}_i,\mathcal{G}_i)\}_{i=1}^N) = E_m(\Delta;\mathcal{E},\{(\mathcal{E}_i,\Delta_i)\}_{i=1}^N)$, where $\Delta_i=\mathcal{G}_i-\mathcal{E}_i$. The mapping function f can be obtained by minimizing the following objective function:

$$\min_f \sum_i \|\Delta_i - f(h(\mathcal{E}_i))\|_2^2. \tag{4}$$

We use a randomized regression tree [1] to learn the mapping function f. The tree is learned recursively by splitting the training sample into left and right subsets under maximum information gain criterion. After training, each leaf node stores a bias vector that is the average of all the training samples falling into it. During testing, a sample $\mathcal{E}$ traverses the tree until it reaches a leaf node. Then the tree outputs the stored bias vector in the leaf node: $f(h(\mathcal{E}))$.

Independently training T randomized regression trees to form a regression forest could give us a pseudo distribution:

$$P(\Delta|h(\mathcal{E})) = \frac{1}{T}\sum_{t=1}^{T}\exp(-\|\frac{\Delta - f_t(h(\mathcal{E}))}{\sigma}\|_2^2), \tag{5}$$

where σ is a learned bandwidth. Now, we can define the mapping energy by

$$E_m(\mathcal{C};\mathcal{E},\{(\mathcal{E}_i,\mathcal{G}_i)\}_{i=1}^N) = E_m(\Delta;\mathcal{E},\{(\mathcal{E}_i,\Delta_i)\}_{i=1}^N) = -\log(P(\Delta|h(\mathcal{E}))). \tag{6}$$

[1] The bias Δ should be normalized by the method proposed in [1] to eliminate the scale variances between individuals. While, for denotational simplicity, we do not involve the normalization factor in this paper.

Learning Pose Prior The pose prior measures how likely a CS $\mathcal{C}$ can be generated from the ground truth sets $\{\mathcal{G}_i\}_{i=1}^N$. Learning the pose prior is actually a distribution estimation (generative model learning) problem. As the variance in the skeleton data may be quite large, especially those captured from actions with high-speed motion, fitting a single model to them is not easy. To estimate the distribution of the skeleton data properly, a divide-and-conquer strategy is adopted here: cluster the training data first, followed by learning the distribution model in each cluster.

A key step leading to a reliable cluster result is to define a faithful distance/similarity measure in the data space. The most commonly used measure is Euclidean distance. However, as we have shown in Sec. 3, the skeleton feature consists of the directions of bones. Therefore, angle based measures are more suitable for our case. We define a cosine similarity based measure for two skeleton feature $h(\mathcal{S}_a)$ and $h(\mathcal{S}_b)$:

$$s(h(\mathcal{S}_a), h(\mathcal{S}_b)) = \frac{1}{n-1}\sum_{j=2}^{n} < h_j(\mathcal{S}_a), h_j(\mathcal{S}_b) >, \tag{7}$$

where $< \cdot, \cdot >$ denotes the dot product of two vectors. As both $h_j(\mathcal{S}_a)$ and $h_j(\mathcal{S}_b)$ are unit vectors, $< h_j(\mathcal{S}_a), h_j(\mathcal{S}_b) >$ is cosine of the included angle between them.

Based on the proposed similarity measure (Eq. 7), Normalized Cut [30] is applied to partition the ground truth sets $\{\mathcal{G}_i\}_{i=1}^N$ into K clusters. Let $\mathbb{1}(h(\mathcal{S}))$ denote the cluster index assigned to a skeleton $\mathcal{S}$. In each cluster $\{\mathcal{G}_i|\mathbb{1}(h(\mathcal{G}_i)) = k\}_{i=1}^N$, we consider each bone direction $h_j(\mathcal{G}_i)$ as a von Mises-Fisher distribution [15], which is a probability distribution on the $(p-1)$-dimensional sphere in $\mathbb{R}^p$:

$$p(h_j(\mathcal{G}_i), \mu_j^k, \kappa_j^k) \propto \exp(\kappa_j^k < h_j(\mathcal{G}_i), \mu_j^k >), \tag{8}$$

which μ_j^k and κ_j^k are the mean and a measure of concentration of all $h_j(\mathcal{G}_i)$s in the k-th cluster, respectively. κ_j^k characterizes how strongly the unit vector $h_j(\mathcal{G}_i)$ drawn according to $p(h_j(\mathcal{G}_i), \mu_j^k, \kappa_j^k)$ are concentrated around the mean μ_j^k. When $\kappa_j^k = 0$, $p(h_j(\mathcal{G}_i), \mu_j^k, \kappa_j^k)$ reduces to the uniform density, and if $k \to \infty$, $p(h_j(\mathcal{G}_i), \mu_j^k, \kappa_j^k)$ tends to a point density peaking at μ_j^k. By assuming each bone is independent, the parameter μ_j^k and κ_j^k can be estimated as follows. Let $\mathcal{I}_\nu$ denote the modified Bessel function of the first kind and order ν and define $\mathcal{A}_p\left(\kappa_j^k\right) = \frac{\mathcal{I}_{p/2}\left(\kappa_j^k\right)}{\mathcal{I}_{p/2-1}\left(\kappa_j^k\right)}$, then the maximum likelihood estimation of μ_j^k and κ_j^k is given by [31]:

$$\begin{aligned} \mu_j^k &= \frac{\sum_{i=1}^N h_j(\mathcal{G}_i)\delta(\mathbb{1}(h(\mathcal{G}_i)) = k)}{\| \sum_{i=1}^N h_j(\mathcal{G}_i)\delta(\mathbb{1}(h(\mathcal{G}_i)) = k)\|} \\ \kappa_j^k &= \mathcal{A}_p^{-1}\left(\bar{R}\right), \end{aligned} \tag{9}$$

where $\delta(\cdot)$ is an indicator function and

$$\bar{R} = \frac{\| \sum_{i=1}^N h_j(\mathcal{G}_i)\delta(\mathbb{1}(h(\mathcal{G}_i)) = k)\|}{\sum_{i=1}^N \delta(\mathbb{1}(h(\mathcal{G}_i)) = k)}. \tag{10}$$

A simple approximation to κ_j^k is

$$\hat{\kappa}_j^k = \frac{\bar{R}\left(p - \bar{R}^2\right)}{1 - \bar{R}^2}. \tag{11}$$

In our case, $p = 3$, as our feature is obtained by mapping the skeleton joints to a 3-dimensional sphere.

After the estimation of μ_j^k and κ_j^k, the pose prior of a CS $\mathcal{C}$ is given by

$$P(\mathcal{C}) = \sum_{k=1}^{K} \prod_{j=2}^{n} p(h_j(\mathcal{C}), \mu_j^k, \kappa_j^k) \delta(\mathbb{1}(h(\mathcal{G}_i)) = k). \tag{12}$$

Thus, we obtain the regularization term by

$$E_r(\mathcal{C}; \{\mathcal{G}_i\}_{i=1}^N) = -\log(P(\mathcal{C})). \tag{13}$$

Energy Function Optimization Given an input ES $\mathcal{E}$, we search the CS $\mathcal{C}$ which minimizes the energy function defined in Eq. 3, leading to an optimization problem:

$$\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} E(\mathcal{C}; \mathcal{E}) \tag{14}$$

By combining Eq. 3, Eq. 6 and Eq. 13, we have

$$\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} -\log(P(\mathcal{C} - \mathcal{E}|h(\mathcal{E}))) - \lambda \log(P(\mathcal{C})). \tag{15}$$

As Eq. 15 does not have close-formed solutions, here we use coordinate descend [32] to infer an approximate solution for it.

We start the optimization process with an initial skeleton $\mathcal{C}^0 \in \mathbb{R}^{n\times 3}$, which can be obtained by $\mathcal{C}^0 = \mathcal{E} + \bar{\Delta} = \mathcal{E} + \sum_t^T f_t(h(\mathcal{E}))/T$. Then we generate a sequence of skeletons $\left\{\mathcal{C}^k\right\}_{k=0}^{\infty}$ by two-level iterations. We refer to the process from $\mathcal{C}^{k-1}$ to $\mathcal{C}^k$ as an outer iteration. In each outer iteration we have $n \times 3$ inner iterations, during which each dimension of $\mathcal{C}^{k-1}$ is sequentially updated: $c_{(i)}^k \leftarrow c_{(i)}^{k-1} (i = 1, \ldots, n \times 3)$. Thus, such an outer iteration generates skeleton $\mathcal{C}^k = \sum_i^{n\times 3} c_{(i)}^k \mathbf{e}_i$, where $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ is a basis vector, in which only the i-th element is one and others are zero. More specifically, to update $\mathcal{C}^{k-1}$ to $\mathcal{C}^k$, we solve the following one-variable sub-problem sequentially:

$$\begin{aligned}
c_{(1)}^k &= \arg\min_{c_{(1)}} E(c_{(1)}\mathbf{e}_1 + c_{(2)}^{k-1}\mathbf{e}_2 + \ldots + c_{(n\times 3)}^{k-1}\mathbf{e}_{n\times 3}; \mathcal{E}) \\
c_{(2)}^k &= \arg\min_{c_{(2)}} E(c_{(1)}^k\mathbf{e}_1 + c_{(2)}\mathbf{e}_2 + \ldots + c_{(n\times 3)}^{k-1}\mathbf{e}_{n\times 3}; \mathcal{E}) \\
&\vdots \\
c_{(i)}^k &= \arg\min_{c_{(i)}} E(c_{(1)}^k\mathbf{e}_1 + c_{(2)}^k\mathbf{e}_2 + \ldots + c_{(i)}\mathbf{e}_i + \ldots + c_{(n\times 3)}^{k-1}\mathbf{e}_{n\times 3}; \mathcal{E}) \\
&\vdots \\
c_{(n\times 3)}^k &= \arg\min_{c_{(n\times 3)}} E(c_{(1)}^k\mathbf{e}_1 + c_{(2)}^k\mathbf{e}_2 + \ldots + c_{(n\times 3)}\mathbf{e}_{n\times 3}; \mathcal{E}).
\end{aligned} \tag{16}$$

As shown in Eq. 16, in each inner iteration, we optimize one dimension $c_{(i)}^{k-1}$ and update its value once for the next inner iteration. The outer iteration will be stopped when the bias between two sequentially obtained energies is no larger than a threshold: $\|E(\mathcal{C}^{k-1};\mathcal{E}) - E(\mathcal{C}^{k};\mathcal{E})\|_2 \leqslant \xi$. As the regression model usually provides a good initialization and the search space for each joint point is restricted within the range of the votes of the random forest, the optimization converges fast. It takes about 36 ms per frame.

5 Experimental Results

In this section, we show the experimental results and give the comparisons between alternative approaches. In the remainder of this section, unless otherwise specified, the parameters introduced in our method are set as follows: the weight factor $\lambda = 1.0$, the threshold for the stopping criterion $\xi = 0.001$, the number of clusters $K = 16$. We select optimal λ and K by grid search and the detail will be discussed in in Sec. 5.3. For the parameters involved in the random forest regression model, we adopt the default setting given in [1]: The number of trees in random forest $T = 50$ and the bandwidth $\sigma = 0.01$m.

In order to assess the performance of our algorithm, we evaluate our method on the data set constructed in [1]. This data set is quite challenge, which contains $15,815$ poses belong to golf swing action. The estimated skeleton and ground truth skeleton of each pose are obtained by XBOX Kinect SDK [2] and a mopcap system, respectively. We use the same experimental setup as [1]: $3,720$ poses are used for training and the rest $12,095$ poses are used for testing. Several example poses of this data set are shown in Fig. 1.

5.1 Evaluation Measurement

Following the evaluation protocol proposed in [1], we use the mean of the sum of joint error as the quality assessment factor. Given a testing data set $\{(\mathcal{E}_i, \mathcal{G}_i)\}_{i=1}^{M}$, we obtain the corrected skeletons $\{\mathcal{C}_i\}_{i=1}^{M}$. We measure the accuracy of each CS $\mathcal{C} = (\mathbf{z}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}}$ by the sum of joint errors (sJE) to its GS $\mathcal{G} = (\mathbf{x}_j^{\mathrm{T}}; j = 1, \ldots, n)^{\mathrm{T}} : \varepsilon = \sum_j \|z_j - x_j\|_2$. To quantify the average accuracy on the whole testing data set, we report the mean sJE (msJE) across all testing skeletons: $\frac{\sum_i \varepsilon_i}{M}$ (unit:meter).

5.2 Comparisons

In this section, we compare the proposed method to other competing ones, including the NN search based and regression based. We also compare the proposed pose prior model with the one under Gaussian distribution assumption [24].

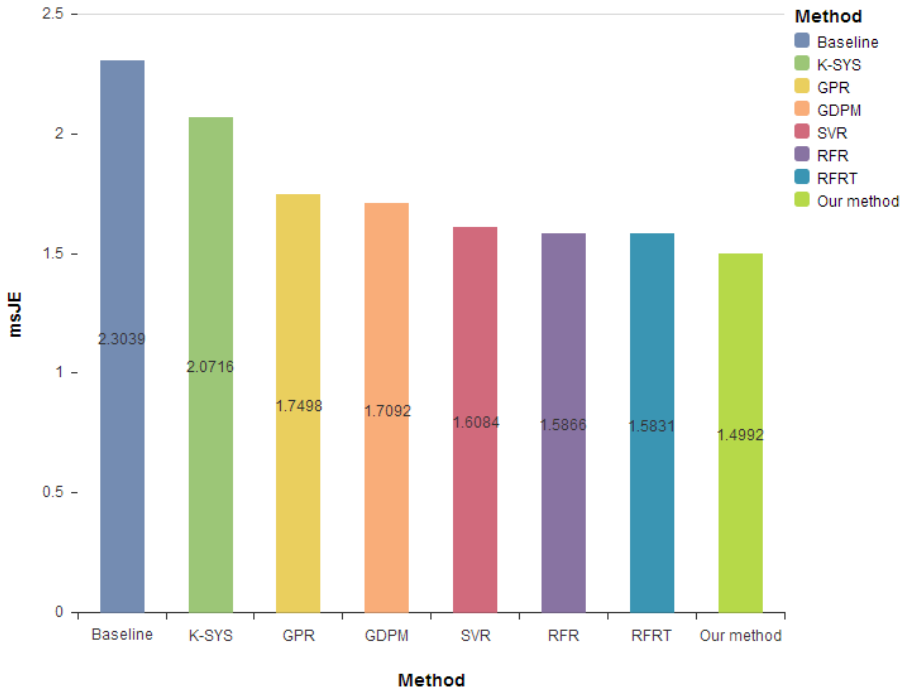

Fig. 3. Comparison with several methods on the golf swing data set [1].

Current Kinect Approach The current Kinect system for pose correction is complex, which employs several strategies such as temporal constraint and filtration. The main idea of this approach is actually nearest neighbor search. The approach finds the nearest neighbor of an input ES in the training set. The corrected skeleton is obtained by merging the corresponding GS of the nearest neighbor to the ES. We refer to the approach used in the current Kinect system as K-SYS. On the whole data set, K-SYS only achieves 2.0716 msJE, as the merging operation may damage the performance.

Regression Based Approaches To show the significance of the introduced pose prior, we compared the random forest regression (RFR) based approaches [1]. As shown in Fig. 3, our method achieves better performance than RFR (1.499 *vs* 1.586). A interesting observation is that the introduction of the temporal constraint (RFRT) only leads to a tiny performance improvement (1.583 *vs* 1.586). This is because the temporal constraint makes use of the consistency between the current and previous predictions; While the errors in the ESs from the poses with severe occlusion or high-speed motion are usually quite large; In this case, the reliability of the previous prediction cannot be guaranteed. We also show that the proposed method outperforms other regression

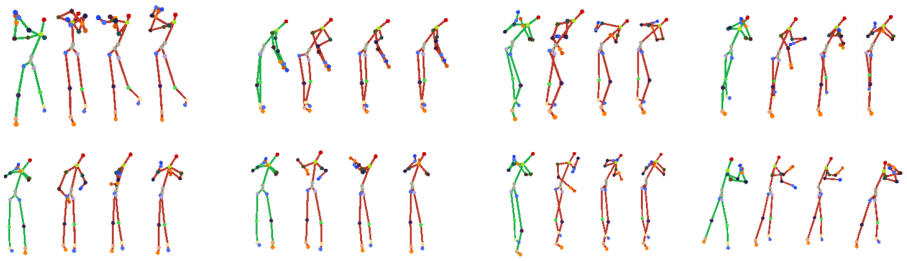

Fig. 4. Examples of corrected skeletons. In each example, we see the GS, the ES, the CS obtained by random forest regression [1], and the CS obtained by the proposed method. Note that, compared to the regression based method, the CSs obtained by our method are more similar to the GSs, and moreover, they are more regular.

based methods, including Gaussian process regressor(GPR) [33], support vector regressor(SVR) [34].

In addition to the improvement in accuracy, more importantly, we emphasize that our method can generate more regular poses. As illustrated in Fig. 4, unlike the weird poses obtained by the regression based method, benefited from the constraint introduced by the well-learned pose prior, the poses obtained by our method are regularized and more similar to the ground truths.

Gaussian Distribution based Prior Model (GDPM) Dantone *et al.* [24] proposed a prior model, which models pairwise part configuration by Gaussian distribution under the classical pictorial structure model [21]. To show the advantage of the proposed prior model, we instead use this Gaussian distribution based prior model in our pose correction framework for comparison. As shown in Fig. 3, our pose prior model achieves better performance than Dantone's (1.499 *vs* 1.709).

5.3 Parameter discussion

We thoroughly investigate the effects of two involved parameters: the weight factor λ and the number of clusters K. As shown in Fig. 5(a), when λ become quite large, our method tends to a NN search based method, which only achieves 2.010 msJE. While, when λ is small, our method reduces to a regression based method. We select a "good" λ to balance the output of the regression model and the pose prior. As shown in Fig. 5(b), the performance of our method would increase when the number of clusters become larger. The reason may be the poses in a compact cluster are more proper to fit the distribution model. But when the K is very large, the performance will decrease (e.g. K = 100, 1.7361msJE). The reliability of the estimated distribution will be damaged due to the lack of samples in each cluster.

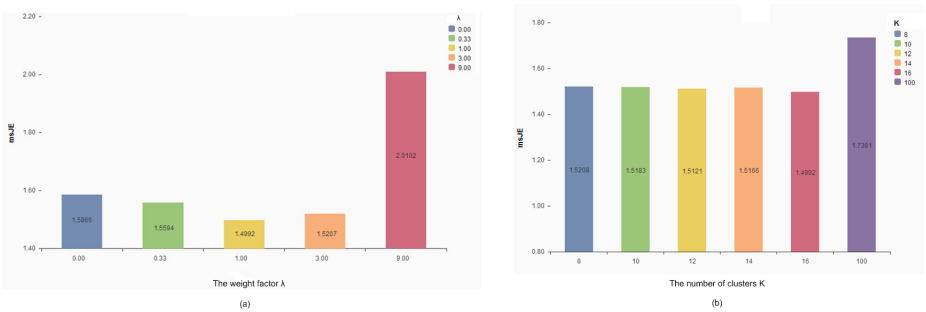

Fig. 5. Parameters versus performance. (a) The weight factor λ. (b) The number of clusters K.

6 Conclusion and Future Work

We have presented an optimization framework for correcting the human poses estimated from Kinect depth images, which combines the output of a random forest regression model with a pose prior model learned on a motion capture data set. By considering the complexity and the geometric property of the pose data, the pose prior is modeled by von Mises-Fisher distributions learned on well-partitioned subspaces separately. The experimental results verify the superiority of the proposed pose correction framework and demonstrate that the introduction of the pose prior indeed generates more regular poses, compared to the current state-of-the-art approach.

In this paper, we only demonstrated the effectiveness of the proposed framework on the problem of correcting the human poses estimated from Kinect depth images. This idea is actually general and can be extended to other problems, such as correcting the human poses estimated from still images [24] and aligning the face images [35]. Besides, to port the proposed algorithm to mobile robots [36] equipped with kinect cameras is also our future work.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 61303095, in part by Research Fund for the Doctoral Program of Higher Education of China under Grant 20133108120017, in part by Innovation Program of Shanghai Municipal Education Commission under Grant 14YZ018, in part by Innovation Program of Shanghai University under Grant SDCX2013012 and in part by Cultivation Fund for the Young Faculty of Higher Education of Shanghai under Grant ZZSD13005. We thank Microsoft Corporation for providing the skeleton data set used in our experiments.

References

1. Shen, W., Deng, K., Bai, X., Leyvand, T., Guo, B., Tu, Z.: Exemplar-based human action pose correction and tagging. In: Proc. CVPR. (2012)

2. Microsoft Corp. Kinect for XBOX 360. Redmond WA.
3. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with microsoft kinect sensor: A review. IEEE Trans. Cybernetics **43** (2013) 1318–1334
4. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: Proc. CVPR. (2011)
5. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real time motion capture using a single time-of-flight camera. In: Proc. CVPR. (2010) 755–762
6. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3d pose estimation from a single depth image. In: Proc. ICCV. (2011)
7. Baak, A., Müller, M., Bharaj, G., Seidel, H.P., Theobalt, C.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: Proc. ICCV. (2011)
8. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. In: Proc. ICCV. (2011)
9. Sun, M., Kohli, P., Shotton, J.: Conditional regression forests for human pose estimation. In: Proc. CVPR. (2012)
10. Shum, H.P.H., Ho, E.S.L., Jiang, Y., Takagi, S.: Real-time posture reconstruction for microsoft kinect. IEEE Trans. Cybernetics **43** (2013) 1357–1369
11. Shen, W., Deng, K., Bai, X., Leyvand, T., Guo, B., Tu, Z.: Exemplar-based human action pose correction. IEEE Trans. Cybernetics (2014)
12. Wang, X., Zhang, Z., Ma, Y., Bai, X., Liu, W., Tu, Z.: Robust subspace discovery via relaxed rank minimization. Neural computation **26** (2014) 611–635
13. Wang, B., Tu, Z.: Sparse subspace denoising for image manifolds. In: Proc. CVPR. (2013) 468–475
14. Bentley, J.L.: Multidimensional divide-and-conquer. Commun. ACM **23** (1980) 214–229
15. Fisher, N.I., Lewis, T., Embleton, B.J.J.: Statistical analysis of spherical data. Cambridge: Cambridge University Press (1993)
16. Wang, X., Bai, X., Ma, T., Liu, W., Latecki, L.J.: Fan shape model for object detection. In: Proc. CVPR. (2012) 151–158
17. Plagemann, C., Ganapathi, V., Koller, D., Thrun, S.: Real-time identification and localization of body parts from depth images. In: Proc. ICRA. (2010) 3108–3113
18. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
19. Quinlan, J.R.: Induction of decision trees. Machine Learning **1** (1986) 81–106
20. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: Proc. ECCV. (2012) 738–751
21. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision **61** (2005) 55–79
22. Ren, X., Berg, A.C., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: Proc. ICCV. (2005) 824–831
23. Ramanan, D.: Learning to parse images of articulated bodies. In: Proc. NIPS. (2006) 1129–1136
24. Dantone, M., Gall, J., Leistner, C., Gool, L.J.V.: Human pose estimation using body parts dependent joint regressors. In: Proc. CVPR. (2013) 3041–3048
25. Ladicky, L., Torr, P.H.S., Zisserman, A.: Human pose estimation using a joint pixel-wise and part-wise formulation. In: Proc. CVPR. (2013) 3578–3585
26. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: Proc. CVPR. (2011) 1385–1392
27. Yao, C., Bai, X., Liu, W., Latecki, L.J.: Human detection using learned part alphabet and pose dictionary. In: Proc. ECCV. (2014) 251–266

28. Lehrmann, A.M., Gehler, P.V., Nowozin, S.: A non-parametric bayesian network prior of human pose. In: Proc. ICCV. (2013) 1281–1288
29. Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press (1994)
30. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 888–905
31. Sra, S.: A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of $i_s(x)$. Computational Statistics **27** (2011) 177C190
32. Z. Q. Luo, P.T.: On the convergence of the coordinate descent method for convex differentiable minimization. Journal of Optimization Theory and Applications **72** (1992) 7–35
33. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press (2006)
34. Schölkopf, B., Smola, A., Williamson, R., Bartlett, P.L.: New support vector algorithms. Neural Computation **12** (2000) 1207–1245
35. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. In: Proc. CVPR. (2012) 2887–2894
36. Zhou, Y., Yang, Y., Yi, M., Bai, X., Liu, W., Latecki, L.J.: Online multiple targets detection and tracking from mobile robot in cluttered indoor environments with depth camera. IJPRAI **28** (2014)