Articulated Gaussian Kernel Correlation for Human Pose Estimation

Meng Ding and Guoliang Fan*
School of Electrical and Computer Engineering,
Oklahoma State University, Stillwater, OK, USA 74074
meng.ding@okstate.edu; guoliang.fan@okstate.edu

Abstract

In this paper, we address the problem of human pose estimation through a novel articulated Gaussian kernel correlation function which is applied to human pose tracking from a single depth sensor. We first derive a unified Gaussian kernel correlation that can generalize the previous Sum-of-Gaussians (SoG)-based methods for the similarity measure between a template and the observation. Furthermore, we develop an articulated Gaussian kernel correlation by embedding a tree-structured skeleton model, which enables us to estimate the full-body pose parameters. Also, the new kernel correlation framework can easily penalize undesired body intersection which is more natural than the clamping function in previous methods. Our algorithm is general, simple yet effective and can achieve real-time performance. The experimental results on a public depth dataset are promising and competitive when compared with state-of-the-art algorithms.

1. Introduction

Human pose estimation has been intensely studied for decades in the field of computer vision due to its wide applications. Recently, the launch of low-cost RGB-D sensors (e.g. Kinect) has further triggered a large amount of research due to their good performance from extra depth information. The existing algorithms can be roughly categorized into three groups, i.e., discriminative, generative and hybrid ones. The approaches in the first group usually require a large database for querying or training [9, 11], while those in the second group need an articulated mesh/geometrical body model for template matching [19, 6] and those in the third category often involve both [5, 2, 18, 17]. To capture the human pose efficiently from multi-view video sequences, a sum of Gaussians (SoG) model was developed in [14]. This simple yet effective shape representation provides a differentiable model-to-image similarity function, allowing a fast and accurate full-body pose estimation. The SoG model was also used in [8, 12, 3] for human or hand pose estimation. Extended from SoG, a generalized SoG model (GSoG) was proposed in [4], where it encapsulated fewer anisotropic Gaussians for human shape modeling, and a similarity function between GSoG and SoG was derived in 3D space. Meanwhile, a sum of anisotropic Gaussians (SAG) model [13] shared the similar spirit with GSoG for hand pose estimation, and it provided an overlap measurement between projected SAG and SoG/SAG in 2D image.

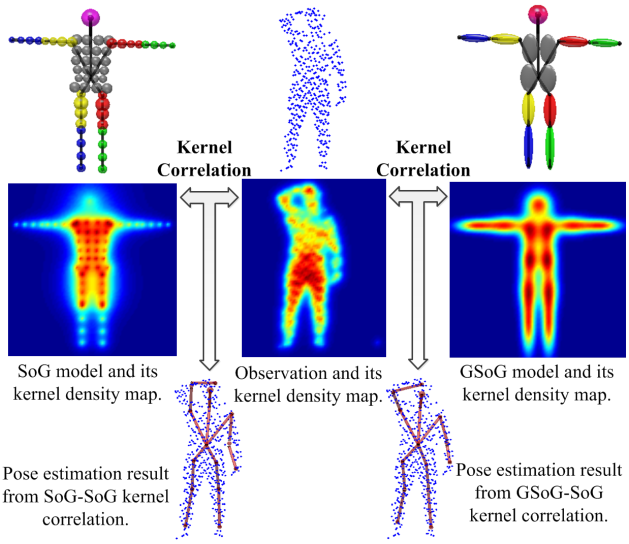

Figure 1. The illustration of SoG and GSoG human shape models and their kernel density maps, as well as the visual results of pose estimation from SoG-SoG and SoG-GSoG kernel correlation.

Although GSoG and SAG based approaches have improved the pose estimation performance with better model adaptability, their similarity functions are specifically designed for different situations/applications. Also, the clamping function which aims to handle the model intersection problem in previous SoG-based approaches [14, 8, 4] leads to a discontinuous energy function that could hinder the gradient-based optimization. In this work, inspired by

*This work is supported by the Oklahoma Center for the Advancement of Science and Technology (OCAST) under grant HR12-30 and the National Science Foundation (NSF) under grant NRI-1427345.

the classical *Kernel Correlation*-based algorithm [16], we generalize previous SoG-based methods and derive a unified similarity function from the perspective of Gaussian kernel correlation. More importantly, we embed a kinematical skeleton into the kernel correlation which enables us to achieve a fast articulated pose estimation.

There are mainly three contributions in this work compared with [4]. First, we treat human pose estimation as an *articulated kernel correlation problem* and have a more general similarity function, where the human shape model and the input data from a depth image can be any pairwise combination, including SoG↔SoG, SoG↔GSoG, GSoG↔GSoG or even mixed model↔mixed model (shown in Fig. 1 and Fig. 2). Second, we embed a kinematical skeleton into the continuous and differentiable kernel correlation function to achieve an efficient articulated pose estimation using a gradient-based optimization. The third contribution is that our generalized kernel correlation framework naturally deduces a new continuous intersection penalty term to deal with the model self-intersection problem. This intersection penalty term can replace the artificial clamping function in the previous SoG-based methods, leading to an optimization-friendly constraint. Our algorithm is simple and efficient and can run at about 20 FPS on a desktop PC without GPU acceleration. We evaluate our pose tracking algorithm on a challenging benchmark dataset [5], which shows that our pose estimation accuracy is competitive compared to the best results reported so far [15, 18, 19].

2. Articulated Gaussian Kernel Correlation

2.1. Multivariate Gaussian Kernel Correlation

In this paper, we focus on the Gaussian kernel correlation and extend the univariate Gaussian case to the multivariate Gaussian one. With this generalization, all the previous SoG-based methods can be unified in one framework. Given two points $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^n$, their Gaussian kernel correlation is defined as the integral of the product of two Gaussian kernels over the whole space [16],

$$KC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \int_{\mathbb{R}^n} G(\mathbf{x}, \boldsymbol{\mu}_1) \cdot G(\mathbf{x}, \boldsymbol{\mu}_2) d\mathbf{x}, \qquad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, and $G(\mathbf{x}, \boldsymbol{\mu}_1), G(\mathbf{x}, \boldsymbol{\mu}_2)$ represent the Gaussian kernels centered at the data point $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$, respectively. The non-normalized univariate Gaussian kernel defined in [10] has the form,

$$G_u(\mathbf{x}, \boldsymbol{\mu}) = \exp(-\frac{||\mathbf{x} - \boldsymbol{\mu}||^2}{2\sigma^2}), \qquad (2)$$

where σ^2 is the variance. Replacing (1) with (2), it is straightforward to have the univariate Gaussian kernel correlation of $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ as,

$$KC_u(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \left(2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^{\frac{n}{2}} \exp\left(-\frac{||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||^2}{2(\sigma_1^2 + \sigma_2^2)}\right). \qquad (3)$$

If the variance σ^2 is extended to the covariance matrix Σ, we have the non-normalized multivariate Gaussian kernel form,

$$G_m(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right). \qquad (4)$$

We re-write (1) using (4). Then, we can derive the multivariate Gaussian kernel correlation of two kernels which are centered at points $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}$ and are modeled by the covariance matrices Σ_1, Σ_2 respectively,

$$KC_m(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sqrt{\frac{(2\pi)^n}{|\Sigma_1^{-1} + \Sigma_2^{-1}|}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right). \qquad (5)$$

2.2. Sum of Gaussian Kernels Correlation

Several Gaussian kernels which are centered at a set of points $\Omega = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$ can be combined as a sum of Gaussian kernels $\mathcal{K}$,

$$\mathcal{K} = \sum_{i=1}^{k} G(\mathbf{x}, \boldsymbol{\mu}_i). \qquad (6)$$

Given two collections of Gaussian kernels $\mathcal{K}_A$ and $\mathcal{K}_B$, composed by M and N Gaussian kernels respectively, their kernel correlation is defined as,

$$\begin{aligned} KC(\mathcal{K}_A, \mathcal{K}_B) &= \int_{\mathbb{R}^n} \sum_{i=1}^{M} \sum_{j=1}^{N} G(\mathbf{x}, \boldsymbol{\mu}_i) G(\mathbf{x}, \boldsymbol{\mu}_j) d\mathbf{x} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} KC_m(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j), \end{aligned} \qquad (7)$$

where $KC_m(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ has been derived in (5). It worth noting that $\mathcal{K}_A$ and $\mathcal{K}_B$ can be composed by univariate Gaussian kernels (SoG in Fig. 2 (a)) or multivariate Gaussian kernels (GSoG in Fig. 2 (c)) or both of them (mixed model in Fig. 2 (d)). Consequently, we obtain a unified kernel correlation function in (7) to evaluate the similarity between any pairwise combination of SoG, GSoG and mixed model, as shown in Fig. 2. When the covariance matrices in $\mathcal{K}_B$ degenerate to variances in the 3D space, the degenerated equation (7) will be equivalent to the similarity function between SoG and GSoG in [4]. Further, if the covariance matrices in $\mathcal{K}_A$ degrade to variances, (7) will become the SoG-SoG similarity in [3, 8, 12]. Both degenerations imply that our kernel correlation function generalizes the previous SoG-based methods.

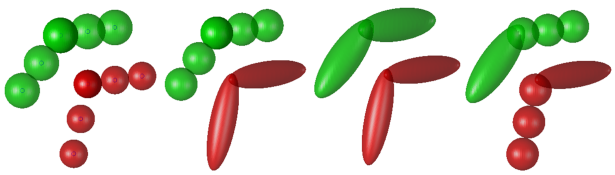

Figure 2. The illustration of the sum of Gaussian kernels $\mathcal{K}_A$ (red) and $\mathcal{K}_B$ (green) in four cases: (a) SoG-SoG, (b) SoG-GSoG, (c) GSoG-GSoG, (d) mixed model-mixed model.

2.3. Kernel Correlation for an Articulated Model

Now, we aim to embed a collection of Gaussian kernels into an articulated structure and explicitly derive the articulated Gaussian kernel correlation. For human pose estimation in this work, the body template comprises a kinematic skeleton and a GSoG shape model $\mathcal{K}_A$ composed by M anisotropic Gaussians. We denote $\mathcal{K}_A^0$ as a standard T-pose template as shown in Fig. 1 (right). The benefits of GSoG model compared with SoG model has been demonstrated in [4]. The kinematic skeleton is constructed by a hierarchical tree structure, as illustrated in Fig. 3. Each rigid body segment defines a local coordinate system that can be transformed to the world coordinate system via a 4×4 transformation matrix T_l, as

$$T_l = T_{par(l)} R_l, \tag{8}$$

where R_l denotes the local transformation from segment l to its parent $par(l)$. If l is the root (the hip joint), T_{root} is the global transformation of the whole body. In this way, the center of each Gaussian kernel in the segment l at the T-pose $\boldsymbol{\mu}_i^0$ can be transferred to its new position in the world coordination through the transformation matrix T_l,

$$\boldsymbol{\mu}_i = T_l \boldsymbol{\mu}_i^0. \tag{9}$$

The R_l of all body segments and T_{root} are the pose parameters to be estimated. In this work, we express a 3D joint rotation as a normalized quaternion which facilitates gradient-based optimization. Here, we have U joints ($U = 10$, marked as red stars in Fig. 3 (b)), each of which allows a 3 DoF rotation represented by a quaternion vector of four elements. Also, there is a global translation at the hip (root) joint. As a result, we totally have 43 pose parameters in the variable $\boldsymbol{\Theta}$. Similar to (9), given the shape model at T-pose $\mathcal{K}_A^0$, we can obtain the deformed model under pose $\boldsymbol{\Theta}$ as,

$$\begin{aligned} \mathcal{K}_A &= \mathcal{K}_A^0(\boldsymbol{\Theta}) \\ &= \sum_{i=1}^{M} G(\mathbf{x}, \boldsymbol{\mu}_i^0(\boldsymbol{\Theta})), \end{aligned} \tag{10}$$

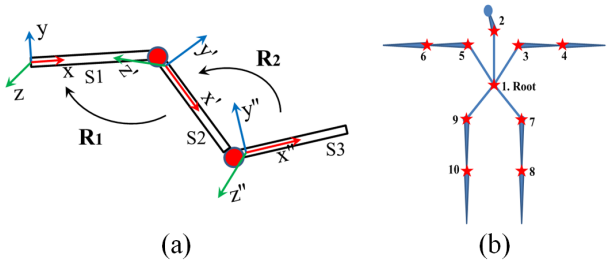

Figure 3. (a) The illustration of a kinematical chain structure and the coordination transformation from the child segment to its parent segment, i.e., $S_3 \rightarrow S_2$ via R_2 and $S_2 \rightarrow S_1$ via R_1. (b) The articulated human skeleton model where the red stars represent the body joints to be estimated.

Re-writing (7) using (10), we explicitly obtain the articulated Gaussian kernel correlation as,

$$KC(\mathcal{K}_A^0(\boldsymbol{\Theta}), \mathcal{K}_B) = \sum_{i=1}^{M}\sum_{j=1}^{N} KC_m(\boldsymbol{\mu}_i^0(\boldsymbol{\Theta}), \boldsymbol{\mu}_j), \tag{11}$$

where $KC_m(\boldsymbol{\mu}_i^0(\boldsymbol{\Theta}), \boldsymbol{\mu}_j)$ can be calculated in (5). Consequently, the Gaussian kernel correlation is embedded into an articulated skeleton and controlled by the pose variable $\boldsymbol{\Theta}$. To represent the raw 3D point cloud data with a SoG model $\mathcal{K}_B$, we employ the same method in [4], where an Octree is used to directly partition the 3D point cloud in terms of the standard deviation of the points in a Octree node along the depth direction. More details for the SoG-based point cloud representation can be found in [4].

The analytical representation of our articulated kernel correlation in (11) constructs the main part of the objective function. As a result, the pose estimation problem is converted to finding the maximum kernel correlation of the human template $\mathcal{K}_A^0(\boldsymbol{\Theta})$ and the point cloud observation $\mathcal{K}_B$. We use the same subject-specific body shape modeling algorithm in [4] to obtain a subject-specific body template. Next, we will present the objective function based on our articulated kernel correlation in details.

3. Proposed Pose Estimation Algorithm

In this section, we will present our objective function, where a balanced kernel correlation specially for the articulated model is proposed. Also, we enhance the energy function with a new intersection penalty term that is a by-product of our generalized Gaussian kernel correlation in (11), and additional two constraints (visibility and continuity). Last, we provide the gradient of the objective function with respect to pose parameters.

3.1. Objective Function

Our pose tracking algorithm is to estimate the pose parameters $\boldsymbol{\Theta}$ at time t from an observed point cloud con-

verted from the corresponding depth image, by optimizing an objective function. We define our objective function that includes the articulated Gaussian kernel correlation $KC(\mathcal{K}_A^0(\mathbf{\Theta}), \mathcal{K}_B)$ defined in (11) along with three additional terms. The first is a visibility term Vis to cope with the incomplete data problem from occlusion; The second one is an intersection penalty $E_{int}(\mathbf{\Theta})$ to penalize the body segments self-intersection; The third one is a continuity term $E_{con}(\mathbf{\Theta})$ to enforce smooth pose transition during tracking. Then pose estimation is formulated as an optimization problem with the objective function,

$$\begin{aligned}\hat{\mathbf{\Theta}} &= \arg\min_{\mathbf{\Theta}} \Big\{ \sum_{i=1}^{M} -KC(\boldsymbol{\mu}_i^0(\mathbf{\Theta}), \mathcal{K}_B) \cdot Vis(i) \\ &\quad + \lambda E_{int}(\mathbf{\Theta}) + \gamma E_{con}(\mathbf{\Theta}) \Big\}, \end{aligned} \tag{12}$$

where $\boldsymbol{\mu}_i^0(\mathbf{\Theta})$ means the center of the i_{th} Gaussian kernel in the body model at pose $\mathbf{\Theta}$, λ, γ are the weights to balance the intersection penalty and continuity terms and Vis is defined as,

$$Vis(i) = \begin{cases} 0 & \text{if the } i\text{th Gaussian is invisible,} \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

Balanced Kernel Correlation The main part of the energy function is the Gaussian kernel correlation $KC(\mathcal{K}_A^0(\mathbf{\Theta}), \mathcal{K}_B)$, which can be evaluated according to (11) and (5). In practice, we find that the kernel correlation from larger segments (e.g. torso in the human body or palm in the hand) could dominate the energy function, overshadowing contributions from small segments. This dominance may trap the optimizer in a wrong local minimum, since the direction of the gradient is also mostly effected by the body segments that achieve relatively considerable energy. To equalize the energy contributions from different segments, we propose a simple yet effective modification to (11) to balance the influence of each articulated segment, referred as "Balanced Kernel Correlation". Specifically, the kernel correlation from each body segment is normalized by a coefficient, which is formulated as,

$$KC_b(\mathcal{K}_A^0(\mathbf{\Theta}), \mathcal{K}_B) = \sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{i=1}^{m_l} \sum_{j=1}^{N} KC_m(\boldsymbol{\mu}_{li}^0(\mathbf{\Theta}), \boldsymbol{\mu}_j), \tag{14}$$

where m_l is the number of Gaussian kernels in the l_{th} body segment (totally we have L segments with the equality $m_1 + \cdots + m_l + \cdots + m_L = M$), and $\frac{1}{\omega_l}$ means the weight of the corresponding segment. Without loss of generality, we calculate ω_l as the integral of all the Gaussian kernels in the l_{th} segment,

$$\begin{aligned}\omega_l &= \int_{\mathbb{R}^n} \sum_{i=1}^{m_l} G(\mathbf{x}, \boldsymbol{\mu}_i^0) d\mathbf{x} \\ &= \sum_{i=i}^{m_l} \sqrt{\frac{(2\pi)^n}{|\Sigma_i^{-1}|}}, \end{aligned} \tag{15}$$

where ω_l means a volumetric measure of the l_{th} segment, revealing that the larger body segment, the greater value of ω_l, but less weight it has. In this way, we balance the contribution of every body segment to the kernel correlation using a given subject-specific body shape. Meanwhile, the value of ω_l can be calculated off-line without reducing the efficiency.

Intersection Penalty Term In previous SoG-based pose estimation methods, to avoid the situation that two or more body segments intersect and overlap with each other, an artificial clamping function was used to constrain the energy contribution of each Gaussian kernel in $\mathcal{K}_B$. However, this energy clamping causes a discontinuity of the energy function, which may hinder the performance of the local optimizer. In this paper, we present an intersection penalty term to replace the artificial clamping function. Interestingly, the new intersection penalty is straightforwardly deduced from our derived Gaussian kernel correlation framework in (11). Our idea is that two body segments are treated as a model $\mathcal{K}_a$ and a target $\mathcal{K}_b$, respectively. Consequently, their kernel correlation measure is equivalent to their intersection penalty. In practice, we consider five self-intersection cases, i.e., head-torso, forearm-arm, upper limb-torso, shank-thigh and lower limb-torso.

$$E_{int}(\mathbf{\Theta}) = \sum_{s=1}^{S} KC^{(s)}(\mathcal{K}_a^0(\mathbf{\Theta}), \mathcal{K}_b), \tag{16}$$

where s in $KC^{(s)}$ represent the s_{th} intersection case (S would be 5 in this work), and $\mathcal{K}_a^0, \mathcal{K}_b$ is the model and target parts separated from a full-body template. When any two body segments intersect with each other, E_{int} will be triggered as a soft constraint, which is still continuous and differentiable.

Visibility Term To address the occlusion problem in a non-frontal view, we use the same strategy that was developed in [4]. The idea is that a relatively large overlap among multiple Gaussian kernels in the projection plane may indicate an occlusion. The pose in the previous frame is used to detect the visible parts. First, each Gaussian component of the SoG model is orthographically projected to a 2D image plane along the depth direction, resulting in a set of circles. Then, we compute the overlap area between every two circles. if the overlap area of any two circles is larger than

certain percentage (e.g. $\frac{1}{3}$) of the area of the smaller one, we declare an occlusion. The Gaussian component which is closer to the camera is remained. The occluded ones are excluded during optimization. More details about occlusion handling can be found in [4].

Continuity Term To encourage continuous sequential pose tracking, we augment the energy function with a continuity term to smooth pose estimation,

$$E_{con}(\mathbf{\Theta}_t) = \sum_{k=1}^{N_k} \left[\left(\mathbf{\Theta}_t^{(k)} - \mathbf{\Theta}_{t-1}^{(k)} \right) - \left(\mathbf{\Theta}_{t-1}^{(k)} - \mathbf{\Theta}_{t-2}^{(k)} \right) \right]^2, \tag{17}$$

where $\mathbf{\Theta}_t$ is the pose parameter at time t and $\mathbf{\Theta}_{t-1}, \mathbf{\Theta}_{t-2}$ are the previous two poses; k is the index of the dimension of the parameter $\mathbf{\Theta}$ (totally $N_k = 43$ in this work). As a regularizer, the continuity term penalizes a large deviation from previous poses, ensuring a smooth pose transition.

3.2. Gradient-based Optimization over Pose Θ

Due to the differentiable energy function and the benefits of quaternion-based rotation representation, we can explicitly derive the derivative of the objective function E with respect to $\mathbf{\Theta}$ and employ a gradient-based optimizer. Different with a variant of steepest descent used in [14, 8], we employ a Quasi-Newton method (L-BFGS [1]) because of its faster convergence. For simplicity, we ignore the visibility term in (12) and has the following form:

$$\begin{aligned} \frac{\partial E(\mathbf{\Theta})}{\partial \mathbf{\Theta}} &= -\frac{\partial KC(\mathcal{K}_A^0(\mathbf{\Theta}), \mathcal{K}_B)}{\partial \mathbf{\Theta}} \\ &\quad + \lambda \frac{\partial E_{int}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \gamma \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} \\ &= -\sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{i=1}^{m_l} \sum_{j=1}^{N} \frac{KC_m(\boldsymbol{\mu}_{li}^0(\mathbf{\Theta}), \boldsymbol{\mu}_j)}{\partial \mathbf{\Theta}} \\ &\quad + \lambda \frac{\partial E_{int}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \gamma \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}}. \end{aligned} \tag{18}$$

We denote $\mathbf{r} = [r_1, r_2, r_3, r_4]^T$ as an un-normalized quaternion, which is normalized to $\mathbf{p} = [x, y, z, w]^T$ according to $\mathbf{p} = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. We represent the pose $\mathbf{\Theta}$ as $[\mathbf{t}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(U)}]$, where $\mathbf{t} \in \mathbb{R}^3$ defines a global translation, U is the number of joints to be estimated, and each normalized quaternion $\mathbf{p}^{(u)}$ from $\mathbf{r}^{(u)} \in \mathbb{R}^4$ defines the relative rotation of the u_{th} joint. Defined in (5), $\boldsymbol{\mu}_{li}$ is the center of i_{th} Gaussian kernel in the l_{th} segment which is transformed from its local coordinate $\boldsymbol{\mu}_{li}^0$ through transformation T_l in (9) and the corresponding covariance matrix $\mathbf{\Sigma}_i^{(l)}$ is approximated and updated from the previous pose under the assumption that is adjacent poses should be close to each other. We explicitly represent every pairwise kernel correlation using (5) and take derivative with respect to each pose parameter, which will be sum over to obtain the gradient vector of our kernel correlation:

$$\frac{\partial KC_m}{\partial t_n} = \frac{\partial KC_m}{\partial \boldsymbol{\mu}_i} \frac{\partial \boldsymbol{\mu}_i}{\partial t_n}, \ (n = 1, 2, 3) \tag{19}$$

$$\frac{\partial KC_m}{\partial r_m^{(u)}} = \frac{\partial KC_m}{\partial \boldsymbol{\mu}_i} \frac{\partial \boldsymbol{\mu}_i}{\partial T_l} \frac{\partial T_l}{\partial \mathbf{p}^{(u)}} \frac{\partial \mathbf{p}^{(u)}}{\partial r_m^{(u)}}, \ (m = 1, ..., 4) \tag{20}$$

which are straightforward to calculate. The derivative of $E_{int}(\mathbf{\Theta})$ is the same with above two equations (19), (20). Since $E_{con}(\mathbf{\Theta}_t)$ in (17) is a standard quadratic form, we have its gradient expression directly as:

$$\frac{\partial E_{con}(\mathbf{\Theta}_t)}{\partial \mathbf{\Theta}_t^{(k)}} = 2 \left[\left(\mathbf{\Theta}_t^{(k)} - \mathbf{\Theta}_{t-1}^{(k)} \right) - \left(\mathbf{\Theta}_{t-1}^{(k)} - \mathbf{\Theta}_{t-2}^{(k)} \right) \right]. \tag{21}$$

Again, k is the index of the dimension of the parameter $\mathbf{\Theta}$. The initialization of $\mathbf{\Theta}$ is the estimated pose in previous frame and the pose in the first frame is assumed to be close to a standard pose, similar to the treatment in many other algorithms.

4. Experimental Results

4.1. Experiment Setup

Test Database: We evaluate our algorithm and compare with a few recent algorithms using the same benchmark dataset SMMC-10 [5]. The ground truth data are the 3D marker positions recorded by the optical tracker. The large amounts of noise and outliers in this dataset makes it challenging yet proper for evaluating algorithm robustness and noise tolerance.

Evaluation Metrics: We adopt two evaluation metrics in our experiments. One evaluation metric is to directly measure the averaged Euclidean distance error between the ground-truth markers and estimated ones over all markers across all frames,

$$\bar{e} = \frac{1}{N_f} \frac{1}{N_m} \sum_{k=1}^{N_f} \sum_{i=1}^{N_m} \|\mathbf{p}_{ki} - \mathbf{v}_{disp}^{(i)} - \hat{\mathbf{p}_{ki}}\|, \tag{22}$$

where N_f and N_m are the number of frames and markers; $\mathbf{p}_{ki}$ and $\hat{\mathbf{p}_{ki}}$ are the ground-truth location of the i_{th} marker and the estimated one in the k_{th} frame, respectively; $\mathbf{v}_{disp}^{(i)}$ is the displacement vector of the i_{th} marker. Because the definitions of marker location across different body models are different, the inherent and constant displacement $\mathbf{v}_{disp}$ should be subtracted from the error, as a routine in most recent methods. In this paper, we have improved the displacement calculation method used in [4] to make the $\mathbf{v}_{disp}$ independent with poses by projecting the markers on each centerline of the segment and computing a local offset for each segment individually. The other evaluation metric is the percentage of correctly estimated joints whose errors are less than $10cm$.

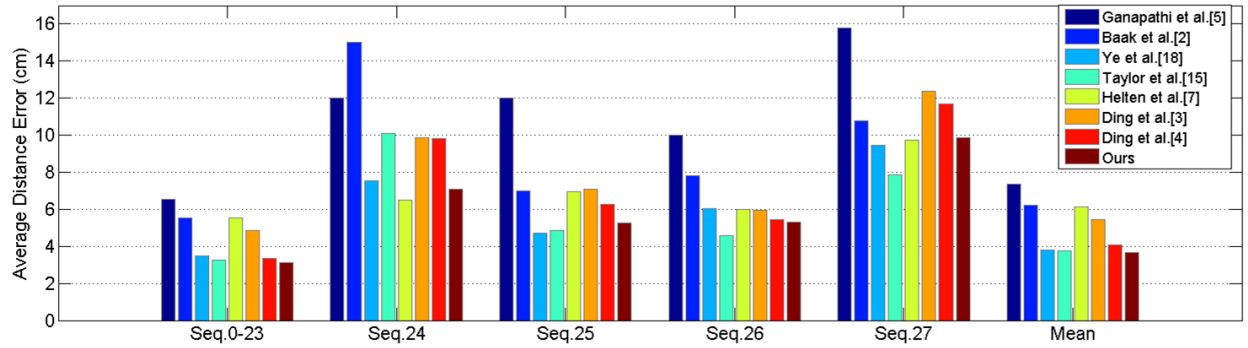

Figure 4. The accuracy comparison with the state-of-the-art methods [5, 2, 18, 15, 7, 3, 4] in terms of the joint distance error (cm). Except [3, 4] and ours, all the others use a large scale database and a mesh model.

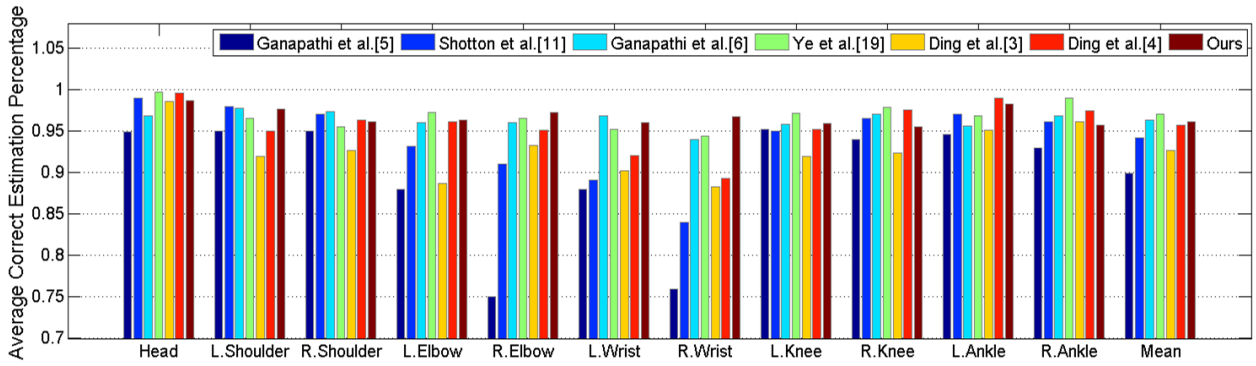

Figure 5. The prediction precision comparison with the state-of-the-art methods [5, 11, 6, 19, 3, 4].

4.2. Quantitative Results

Tracking accuracy comparison

In Fig. 4 and Fig. 5, we compare our algorithm with the state-of-the-arts in two metrics. Our approach achieves the accuracy of average $3.65cm$ on SMMC-10 dataset and it is close to the best results so far (around $3.4cm$) [18, 15, 19] where a database and/or a detailed mesh model are involved. We notice that our results are much more accurate than the original SoG algorithm (implemented in [3]) and [7] where extra inertial sensors were used. It also outperforms the GSoG method [4], which should owe to our proposed balanced kernel correlation and the continuous intersection penalty term, as well as the new displacement calculation. Compared with others (except [4]), our algorithm is simpler and the computational complexity is lower. Furthermore, we compare the accuracy by the correct rate of joint estimation (less than 10cm) in Fig. 5. It shows that our algorithm is still comparable with the best algorithms [19, 6], revealing the accuracy and robustness of our pose tracking algorithm.

Efficiency Analysis

Among all mesh-based generative methods, the computational complexity is expressed as $O(MN)$, where M is the number of vertices in a surface model and N is the number of points in observation. Due to the GSoG body shape representation, M and N in our approach are much less than those in the state-of-the-art methods and M in GSoG is only about a quarter as that in SoG, leading to a lower computational cost. We implement our tracking algorithm in $C++$ with the L-BFGS optimization library [1]. Currently, the efficiency is evaluated on a PC without GPU acceleration. We allow maximum 30 iterations in the first frame and then 15 iterations in the following frames, and we ignore the computing time of background segmentation using a depth threshold and the Octree partitioning which are very efficient. The average processing rate of our algorithm is about 20 frames per second without the code optimization. Our algorithm is also suitable for GPU-based parallel computing for further speed-up.

4.3. Qualitative Results

Some pose estimation results are shown in Fig. 6 to illustrate the performance of our method. While the estimated poses are visually correct in most frames and the visibility term can solve the incomplete data problem to some extent, our tracker may fail when deal with some complex motions with significant occlusion problems. One possible reason is the visibility term employs an approximation

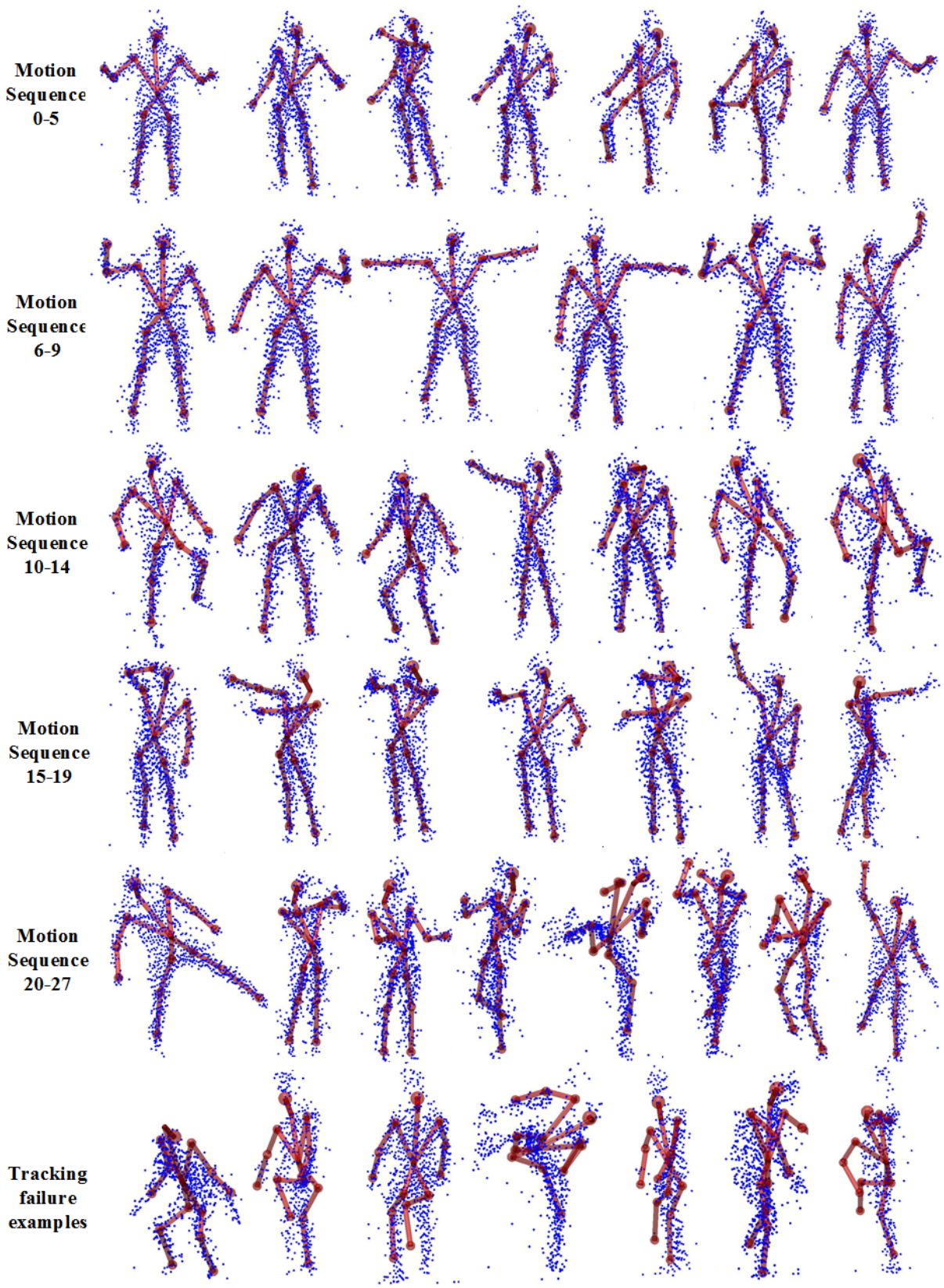

Figure 6. Some pose tracking results from the depth motion sequences 0-27 and some tracking failure examples (the bottom row). The blue points are the input point cloud data and the red skeleton represents the estimated pose.

from previous result and a coarse orthographical projection assumption when to identify the invisible components, and the other possible reason is that the optimizer may be stuck into local minima and cannot be recovered automatically. Both possible reasons will guide our future work.

5. Conclusion

We have developed a generalized Gaussian kernel correlation framework that provides a continuous and differentiable similarity measure between two point sets, both of which are represented by a collection of univariate or multivariate Gaussians or even a mixed model. Based on the unified kernel correlation function, we have presented an efficient human pose tracking algorithm, where the observed point cloud is represented by a SoG and the human template is developed by embedding a GSoG in a quaternion-based articulated skeleton. Consequently, the human pose is estimated by maximizing a new articulated kernel correlation along with three additional constraints to ensure valid and smooth pose estimation. The new articulated kernel correlation function naturally supports a penalty term to discourage undesired body segment intersection which is more natural than the clamping function used before. We evaluate our proposed tracker on a public depth dataset, and the experimental results are encouraging and promising compared with the state-of-the-art algorithms, especially considering its simplicity and efficiency. Our algorithm can achieve real-time human pose tracking with competitive accuracy and robustness. Moreover, the generalized Gaussian kernel correlation framework could be applied to other applications involving an articulated structure and complex multi-segment structure.

References

[1] libLBFGS, http://www.chokkan.org/software/liblbfgs/.

[2] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. ICCV*, 2011.

[3] M. Ding and G. Fan. Fast human pose tracking with a single depth sensor using sum of Gaussians models. In *Advances in Visual Computing*, volume 8887 of *Lecture Notes in Computer Science*, pages 599–608. 2014.

[4] M. Ding and G. Fan. Generalized sum of Gaussians for real-time human pose tracking from a single depth sensor. In *Proc. WACV*, 2015.

[5] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Proc. CVPR*, 2010.

[6] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time human pose tracking from range data. In *Proc. ECCV*, 2012.

[7] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt. Real-time body tracking with one depth camera and inertial sensors. In *Proc. ICCV*, 2013.

[8] D. Kurmankhojayev, N. Hasler, and C. Theobalt. Monocular pose capture with a depth camera using a Sums-of-Gaussians body model. In *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 415–424. 2013.

[9] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Proc. ICRA*, 2010.

[10] B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. Computational Molecular Biology. MIT Press, Cambridge, MA, USA, Aug. 2004.

[11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*, 2011.

[12] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proc. ICCV*, 2013.

[13] S. Sridhar, H. Rhodin, H. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic Gaussians model. In *Proc. International Conference on 3D Vision (3DV)*, 2014.

[14] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of Gaussians body model. In *Proc. ICCV*, 2011.

[15] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. CVPR*, 2012.

[16] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proc. ECCV*. 2004.

[17] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 31(6), Nov. 2012.

[18] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D pose estimation from a single depth image. In *Proc. ICCV*, 2011.

[19] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects with a single depth camera. In *Proc. CVPR*, June 2014.