Fast Human Pose Tracking with a Single Depth Sensor Using Sum of Gaussians Models*

Meng Ding and Guoliang Fan

School of Electrical and Computer Engineering
Oklahoma State University

Abstract. We introduce a simple yet effective 3D human pose tracking from a single depth sensor by using the Sum of Gaussians (SoG) models. Both the human body model and the point cloud converted from a depth map are represented by two different SoG models, which allow us to compute and optimize their similarity analytically. We have two main contributions in this work. The first is we extend the SoG-based similarity by integrating two additional terms to enhance the robustness and accuracy of 3D pose tracking. One is a visibility term to handel the incomplete data problem and the other is a continuity term to smooth the motion estimation. Second, we develop a validation and re-initialization strategy to detect and recover tracking failures. Our algorithm is practically promising that neither involves training data nor a detailed mesh or complicated 3D model. The experimental results are impressing and competitive when compared with state-of-the-art algorithms on a benchmark dataset considering the efficiency and simplicity of our method.

1 Introduction

Human pose estimation from images is a highly active research topic in the field of computer vision, due to its wide applications. Recently, the popularity of low-cost RGB-D sensors (Kinect) have further triggered a large body of research due to their cost-effectiveness and great performance. The existing approaches can be roughly categorized into three groups, i.e., discriminative, generative and hybrid ones. Discriminative approaches extract features in a depth map and detect the best pose by either searching in a database or directly predicting the location of body parts according to training data, e.g. [1]. This kind of methods rely on a large training dataset. Generative methods aim to estimate the parameters of a human model that best explains the observation. Most generative methods involve correspondence estimation between the model and the observation, and then iteratively update the pose and correspondence [2, 3]. One exception is [4], where a Gaussian Mixture Model based energy function along with an articulated structure was developed. While most of the generative approaches are capable of achieving high accuracy, they normally require a good initialization and the

* This work is supported by Oklahoma Center for the Advancement of Science and Technology (OCAST) under grants HR09-030 and HR12-30.

G. Bebis et al. (Eds.): ISVC 2014, Part I, LNCS 8887, pp. 599–608, 2014.

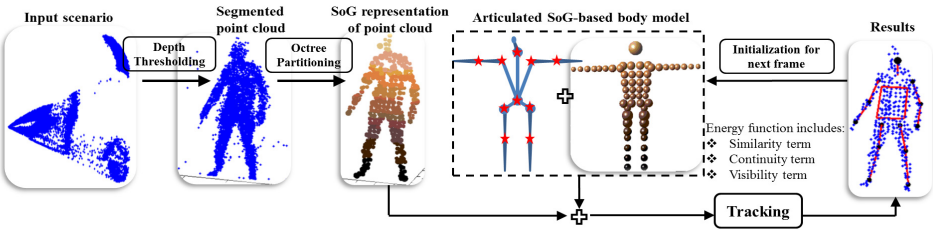

Fig. 1. The framework of our tracking system

computational complexity is usually high. The hybrid approaches combines both discriminative and generative approaches due to their complementary benefits [5, 6], however, the systems are usually complex. For fast pose estimation, a sum of Gaussians (SoG) model was developed in [7, 8] where both the human body model and the image are represented by two different SoG models. Similarly, this strategy was applied in [9] for a hand motion tracking. The SoG-based method is succinct and efficient with a gradient-based optimization. However, it has never been evaluated on any benchmark depth dataset for human pose estimation. Also, due to the incomplete data problem and multiple local minima of the objective function, pose tracking is not reliable, which inspires this work.

In this paper, we propose a novel SoG-based 3D pose tracking framework, which has several advantages over those in [7, 8]. First, we directly partition the point cloud date using Octree (instead of quad-tree) for SoG representation. Secondly, we incorporate a visibility term to handle the incomplete data problem and a continuity term to penalize large motion variation during tracking. Third, we develop a validation and re-initialization strategy to detect and recover tracking failures. Fourth, to speed up the convergence, we use the Quasi-Newton optimization over the joint angles represented by quaternion. Compared with the algorithms using database or a detailed mesh model, our method is simple yet effective and efficient. We evaluate our proposed algorithm on a public depth dataset [10] and compare it with state-of-the-art methods. The experimental results are competitive and promising considering the efficiency and simplicity of our method. Our system is shown in Fig. 1. After simply segmenting the target, we represent the noisy point cloud as a SoG model using Octree. Then, the SoG-based body model is fitted into the SoG-represented observation for tracking the articulated motion by minimizing an energy function. In the following sections, we will introduce each step in detail.

2 SoG Representation of Human Body and Point Cloud

2.1 Sum of Gaussians Preliminaries

A single un-normalized 3D Gaussian G has the form:

$$G(\mathbf{x}) = \exp\left(-\frac{||\mathbf{x}-\mu||^2}{2\sigma^2}\right), \tag{1}$$

where $\mathbf{x}$ is a vector in the 3D space $\mathbb{R}^3$, σ^2 and $\mu \in \mathbb{R}^3$ are the variance and the mean respectively. Several spatial Gaussians are combined as a Sum of Gaussians $\mathscr{K}$ in [7] to describe a volumetric model:

$$\mathscr{K}(\mathbf{x}) = \sum_{i=1}^{n} G_i(\mathbf{x}), \tag{2}$$

Given two SoG representations $\mathscr{K}_a$ and $\mathscr{K}_b$, a similarity of the two models is defined as the integral of the product of $\mathscr{K}_a$ and $\mathscr{K}_b$ over the 3D space Ω:

$$\begin{aligned} E(\mathscr{K}_a, \mathscr{K}_b) &= \int_{\Omega} \sum_{i \in \mathscr{K}_a} \sum_{j \in \mathscr{K}_b} G_i(\mathbf{x}) G_j(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i \in \mathscr{K}_a} \sum_{j \in \mathscr{K}_b} E_{ij}, \end{aligned} \tag{3}$$

where E_{ij} is the similarity measurement of two Gaussian components:

$$E_{ij} = \left(2\pi \frac{\sigma_i^2 \sigma_j^2}{\sigma_i^2 + \sigma_j^2} \right)^{\frac{3}{2}} \exp\left(-\frac{||\mu_i - \mu_j||^2}{2(\sigma_i^2 + \sigma_j^2)} \right), \tag{4}$$

where μ_i and μ_j are the centers of Gaussians G_i and G_j, σ_i^2 and σ_j^2 are the corresponding variances. The Equ. (3) and (4) explain that the more similarity of two SoG models over 3D space means larger value E, resulting in an energy function. We notice that Equ. (4) is a continuous and differentiable function, which allows an analytical derivative computation for fast optimization.

2.2 Quaternion-Based Articulated Human Model

Our human body model comprises a skeleton and a SoG model $\mathscr{K}_M$ attached on it. Similar to [7, 8], we simplify the body model using 57 3D isotropic Gaussian components which is much less than the number of vertices in a mesh model, as shown in the middle part of Fig. 1 (Articulated SoG-based body model). The skeleton is constructed by a tree structure, where each rigid segment is defined in its local coordinate system and can be transformed to the world coordinate system via a 4×4 matrix T_l:

$$T_l = T_{par(l)} R_l, \tag{5}$$

where R_l denotes a relative transformation from segment l to its parent, $par(l)$ indicates the parent of segment l. If l is the root joint, T_{root} is the global transformation. In fact, the rotation in each R_l constructs the pose parameters.

We use quaternion to represent the 3D rotation considering its benefit on the gradient-based optimization due to its continuousness and less constraints. We have L joints ($L = 10$ marked as red stars in Fig. 1), each of which allows a 3 DoF rotation represented by a four elements quaternion vector. Integrating one global translation at the hip (root) joint, we totally have 43 elements in the pose parameters $\mathbf{\Theta}$. Because the estimation of subject-specific skeleton is beyond the scope of this work, we use a standard adult skeleton and roughly scale it to the size of observations in a pre-processing step.

2.3 SoG Representation of Point Cloud with Octree

We approximate the raw point cloud with a SoG representation. In [7, 8], a quad-tree is used to cluster the image pixels with similar depth into a larger square, and then each of them is approximated by a 2D Gaussian. To measure the similarity between the SoG body model $\mathscr{K}_M$ and the SoG depth image $\mathscr{K}_I$, $\mathscr{K}_M$ has to be projected into 2D image or $\mathscr{K}_I$ has to be converted into 3D space, complicating the system in pre-processing step. In this paper, we employ the Octree to directly partition the point cloud in 3D space .

Octree is a useful shape representation tool to partition a 3D space by recursively subdividing it into eight octants. We further develop our own partition criterion in the Octree to adapt our algorithm. If the standard deviation in depth direction of the points in a Octree node is larger than a threshold η_{depth}, we subdivide the node into eight sub-nodes, up to a maximum Octree level of typically n_{level}. Then, each cube (leaf note of Octree) is represented by an isotropic Gaussian G_i, where μ_i is the mean of all the points in one cube and σ^2 is set to be the square of half-length of a side of the cube. Consequently, we have the SoG representation $\mathscr{K}_P$ of point cloud. Adjusting the maximum level of the Octree n_{level} and depth threshold η_{depth} can control the number of leaf notes. The SoG representation of point cloud after Octree partitioning is shown in Fig. 2, where we can observe that large number of points are clustered into small number of isotropic Gaussians and the noise are restrained, which promote the efficiency and robustness of our system.

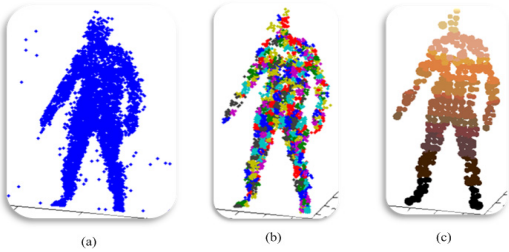

Fig. 2. An illustration of a SoG representation of point cloud. (a) point cloud; (b) the partitioning results (points in one color have similar depth); (c) a SoG representation.

3 Proposed Tracking Algorithm

Our tracking algorithm is to estimate the pose parameters $\mathbf{\Theta}$ from a set of consecutive point cloud by minimizing an energy function, which is mainly based on the SoG similarity with a visibility and a continuity terms. We employ a gradient-based optimization over parameters $\mathbf{\Theta}$ for fast pose estimation. To recover tracking failures, we develop a validation and re-initialization procedure.

3.1 Objective Function

Similarity Term. The main part of our energy function is measuring the similarity $E_{sim}(\mathbf{\Theta})$ between the body model in pose $\mathbf{\Theta}$ denoted as $\mathscr{K}_M(\mathbf{\Theta})$ and the SoG-based point cloud $\mathscr{K}_P$. Given two SoG models, it is straightforward to calculate $E_{sim}(\mathbf{\Theta})$ with Equ. (3) and (4). One possible situation is two or more body segments overlap on the same part of the observation and thereby some Gaussians in the observation could contribute several times to the energy function, resulting a wrong similarity. To avoid this, we modify Equ. (3) to clamp the energy of each Gaussian in observation:

$$E_{sim}(\mathbf{\Theta}) = \sum_{i \in \mathscr{K}_P} min \left(\left(\sum_{j \in \mathscr{K}_M} E_{ij}(\mathbf{\Theta}) \right), \omega E_{ii} \right), \tag{6}$$

where E_{ii} is the maximum energy of a Gaussian in observation, ω is a constant (≥ 1) to scale E_{ii}. It is worth mentioning that the modified function $E_{sim}(\mathbf{\Theta})$ is still continuous, but not differentiable everywhere, i.e. the derivative at exactly the point where $\sum_{j \in \mathscr{K}_M} E_{ij}(\mathbf{\Theta}) = \omega E_{ii}$ does not exist. However, the chance of evaluating the derivative at exactly that point is nearly zero so that the modified $E_{sim}(\mathbf{\Theta})$ can still be regarded as derivable in practice.

Visibility Term. To handel incomplete data like Fig. 3 (a), we develop a visibility term to identify which Gaussian components in body model are invisible so that they will not be involved in the similarity computation. We develop a visibility detector based on the projection overlap area using previous estimated pose. We first orthographically projected Gaussian components of the body into a 2D image along depth direction. As shown in Fig. 3 (b), we obtain a set of circles whose radii are the standard deviation of the Gaussians. Then we compute the overlap area of each circle pair. If the overlap area of any two circles is larger than a percentage (e.g. $\frac{1}{3}$) of the area of the smaller circle, we regard it as a occlusion. The Gaussian component which is closer to the camera is remained, thereby the occluded ones are excluded during the calculation of similarity.

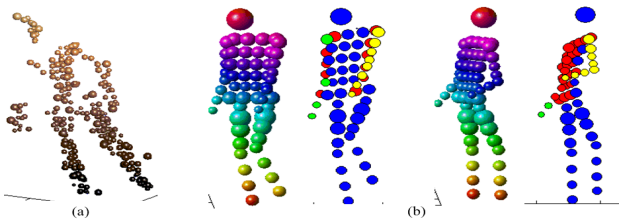

Fig. 3. (a) An example of incomplete observation. (b) Two body models and their 2D projections, where the red circles on torso and right arm denote the occluded body components; the left arm in yellow and part of right arm in green are remained.

Continuity Term. To penalize large motion change, we augment the energy function with a continuity term for smoothing the parameters estimation:

$$E_{con}(\mathbf{\Theta}_t) = \sum_{l=1}^{N_l} \left[\left(\mathbf{\Theta}_t^{(l)} - \mathbf{\Theta}_{t-1}^{(l)} \right) - \left(\mathbf{\Theta}_{t-1}^{(l)} - \mathbf{\Theta}_{t-2}^{(l)} \right) \right]^2, \tag{7}$$

where $\mathbf{\Theta}_t$ is the pose in current frame and $\mathbf{\Theta}_{t-1}, \mathbf{\Theta}_{t-2}$ are previous last two poses; N_l is the number of elements in vector $\mathbf{\Theta}$.

Full Objective Function. Maximizing the similarity is equivalent to minimizing its negative. The continuity term should be minimized to penalize the large motion change. Consequently, we have a full objective function as:

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \left\{ \sum_{i \in \mathcal{K}_M} -E_{sim}^{(i)}(\mathbf{\Theta}) \cdot Visibility(i) + \lambda_{con} E_{con}(\mathbf{\Theta}) \right\}, \tag{8}$$

where i represents the index of body Gaussian components, λ_{con} is a weight to balance the terms and the $Visibility$ is defined by:

$$Visibility(i) = \begin{cases} 0 & \text{if } i_{th} \text{ Gaussian is invisible,} \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

3.2 Gradient-Based Optimization

Due to our derivable SoG-based energy function and the beneficial features of quaternion-based rotation, we can analytically derive the derivatives and employ a gradient-based optimizer. Different with a variant of steepest descent in [7][8], we employ a Quasi-Newton optimization (L-BFGS) because of its faster convergence. Below, we provide the derivative of E with respect to parameters $\mathbf{\Theta}$ in details. For simplicity, we ignore the visibility term and have,

$$\begin{aligned} \frac{\partial E(\mathbf{\Theta})}{\partial \mathbf{\Theta}} &= -\frac{\partial E_{sim}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \lambda_{con} \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} \\ &= -\sum_{i \in \mathcal{K}_a} \sum_{j \in \mathcal{K}_b} \frac{\partial E_{ij}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \lambda_{con} \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}}, \end{aligned} \tag{10}$$

We denote an un-normalized quaternion $\mathbf{r} = (r_1, r_2, r_3, r_4)^T$, which is normalized to a unit quaternion $\mathbf{p} = (x, y, z, w)^T$ according to $\mathbf{p} = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. We explicitly represent the pose $\mathbf{\Theta}$ as $\{\mathbf{t}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(L)}\}$, where $\mathbf{t} \in \mathbb{R}^3$ defines a global translation and each normalized quaternion $\mathbf{p}^{(l)} \in \mathbb{R}^4$ defines the relative rotation of joint l. We can explicitly expand Equ. (4) and derive $\frac{\partial E_{ij}}{\partial \mathbf{t}}$ and $\frac{\partial E_{ij}}{\partial \mathbf{r}^{(l)}} = \frac{\partial E_{ij}}{\partial \mathbf{p}^{(l)}} \frac{\partial \mathbf{p}^{(l)}}{\partial \mathbf{r}^{(l)}}$. Since $E_{con}(\Theta_t)$ in Equ. (7) is a standard quadratic form, we have its gradient expression directly:

$$\frac{\partial E_{con}(\mathbf{\Theta}_t)}{\partial \mathbf{\Theta}_t^{(l)}} = 2 \left[\left(\mathbf{\Theta}_t^{(l)} - \mathbf{\Theta}_{t-1}^{(l)} \right) - \left(\mathbf{\Theta}_{t-1}^{(l)} - \mathbf{\Theta}_{t-2}^{(l)} \right) \right], \tag{11}$$

The initialization of $\mathbf{\Theta}_t$ is the estimated pose in previous frame and we assume the pose in the first frame is close to a pre-defined pose as many systems use.

3.3 Validation and Re-initialization

One limitation of the local optimizer is the tracking could get stuck in a local minimum and cannot recover automatically. This motivates us to develop a validation process to supplement our tracking with a re-initialization. The key issue is how to detect the tracking fails. To this end, we propose a method to measure how well the reconstructed pose match the observation by evaluating the similarity defined in Equ. (4). Specifically, when a certain percentage of adjacent Gaussians in observation are not overlapped by any part of the body model, it indicates that the tracking is trapped into a local minimum. The procedure is shown in Fig. 4. We first compute the energy of each Gaussian in observation with all the body model Gaussians with Equ. (4). Then, we collect those Gaussians whose energy are smaller than a threshold γ, which means they may not be overlapped. If the number of these Gaussians is larger than a percentage η of the total number of Gaussians in observation, a re-initialization will be triggered. Many re-initialization strategies could be used. In this work, we simply use the mean value of previous last two poses, which has been proved to be valid in our experiments. Another solution is using a linear auto regression to predict a re-initialization pose from previous estimation.

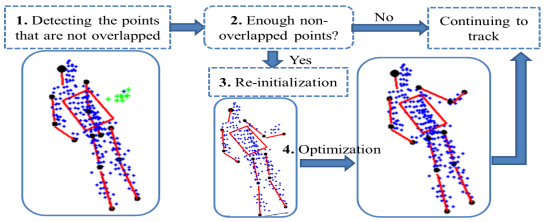

Fig. 4. We mark the Gaussians which are not overlapped by any body parts in green. Once there are certain number of these green points, a re-initialization will be triggered.

4 Experimental Results

4.1 Experiment Setup

Test Database. We use a benchmark dataset SMMC-10 [10] to evaluate our algorithm and compare with other state-of-the-art methods. SMMC-10 dataset captures 28 motion sequences, including various motion types. The ground truth is the marker positions which are recorded by an optical tracker.

Error Metrics. One error metrics is to directly measure the average Euclidean distance error, which is calculated per marker per frame:

$$\bar{e} = \frac{1}{N_f}\frac{1}{N_m}\sum_{k=1}^{N_f}\sum_{i=1}^{N_m}\|\mathbf{p}_i - \mathbf{v}_{disp}^{(i)} - \hat{\mathbf{p}}_i\|, \tag{12}$$

where N_f and N_m are the number of frames and markers; $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$ are i_{th} marker location of ground truth and the estimation, respectively; $\mathbf{v}_{disp}^{(i)}$ is i_{th} marker displacement vector. Because the definitions of marker location across different body model are diverse, a inherent displacement $\mathbf{v}_{disp}$ should be subtracted from the error. To obtain the displacement, similar to many papers, we manually define our marker locations in 30 frames from Sequence #6 and compute the average differences between our marker system and ground truth marker definition. Another error metrics is the percentage of correctly estimated joints whose Euclidean distance errors are less than $0.1m$.

Parameters. Some empirical parameters we used throughout our experiments is listed. In Octree partitioning, the threshold η_{depth} and maximum Octree level n_{level} are set to $20mm$ and 6, respectively. The weight λ_{con} in Equ. (8) is set to 0.2. In validation and re-initialization, the energy threshold γ and the percentage η are 0.2 and 5%, respectively.

4.2 Quantitative Results

The Effect of Different Terms. To exhibit the effect of each term developed in our tracking algorithm, we prepare four sets of experiment, where the continuity term, validation and re-initialization and the visibility term are incorporated into the similarity energy function successively. Their distance errors are shown in Fig. 5 (a), where we find that the tracking accuracy gradually promote with the extra terms. Especially, in Sequence 24-27, where the motions are more complex, these terms make a lot contribution for the accuracy improvement. Fig.5 (b) and (c) illustrate the error of left elbow in Sequence 24 and the error of right shoulder in Sequence 27 respectively. It is clear to observe that using extra terms (in red) achieves smaller error than without them (in blue), which demonstrates the effectiveness of our extra terms.

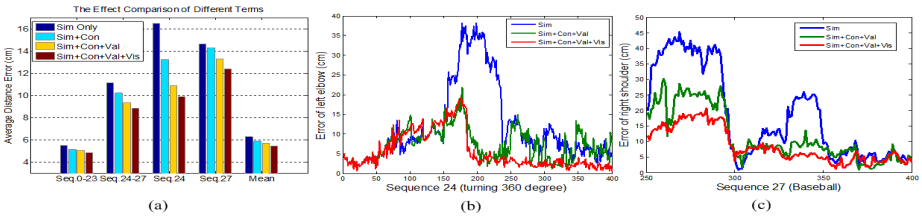

Fig. 5. The effect of different terms in distance error (cm). "Sim", "Con", "Val" and "Vis" denote similarity, continuity, validation and visibility terms, respectively.

The Accuracy Comparison with State-of-the-Art Methods. In Fig. 6, we exhibit the accuracy comparison by average distance error metric and correct percentage metric within several state-of-the-art algorithms. Our approach

achieves the accuracy of 5.4 centimeter testing on SMMC-10 dataset and it even outperforms [11] where extra inertial sensors were used. While our method is not the most accurate one, it is much simpler and lower computational complexity than other methods, where a detailed mesh model and a large scale dataset are necessary. Also, in Sequences 24-27 where the motions are more complicated, our tracker can still achieve comparable accuracy. In Fig. 6 (b), the accuracy of our algorithm is relatively low at the left elbow and two wrist joints. The main reason is our simple and rigid SoG body model is less representative to handle very complex and detailed motion in Sequence 27.

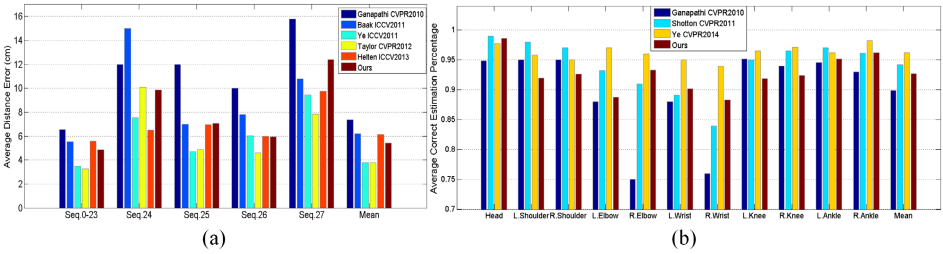

Fig. 6. (a) The accuracy comparison with state-of-the-art methods [5, 6, 10–12] in distance error (cm). Except [10] and ours, all the others use a database and a mesh model. (b) The correct percentage comparison with state-of-the-art methods [1, 4, 10].

Efficiency Analysis. In generative methods, the computational complexity is expressed as $O(MN)$, where M is the number of vertices in a mesh model and N is the number of points in observation. Due to the SoG representation, the M and N in our approach is much less than them in other methods, leading to a lower computational complexity. Currently, the efficiency is evaluated on the *Matlab* platform using a standard desktop computer. We allow a maximum 50 iterations in the first frame and then 20 iterations in the following frames, which has been proved sufficient in all the experimental dataset. The average processing rate is 5 fps in *Matlab* without code optimization.

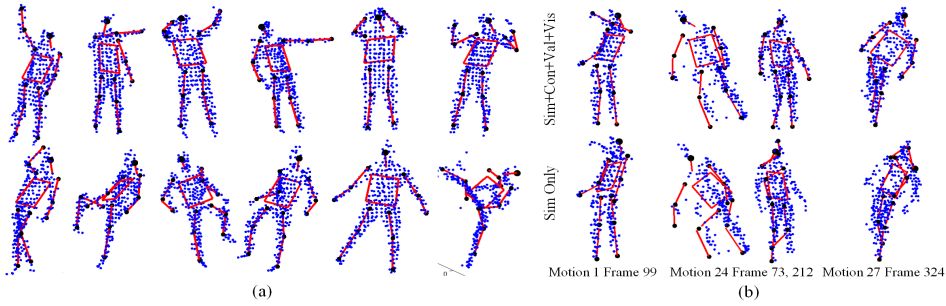

Fig. 7. (a) Estimation results. (b) The effect of additional terms.

4.3 Qualitative Results

We first show some pose estimation results to illustrate the performance of our method using the stick man in Fig. 7 (a) and visually compare the effect of extra terms in Fig. 7 (b). We can observe that the developed terms can help to solve the incomplete data problem and recover tracking failures.

5 Conclusion

We have introduced an efficient, accurate and robust human pose tracking algorithm based on a simple yet effective Sum of Gaussians model. To enhance our tracking algorithm, we build up a visibility term to handel the incomplete data problem. Also, a validation and re-initialization has been developed to recover tracking failures. We evaluate our proposed tracker on a public dataset. The experimental results are impressing considering neither a database nor a mesh model is involved. Our method well balances the accuracy and the systematic complexity for a fast motion capture system and has potentials in mobile devices.

References

1. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proceedings of CVPR (2011)
2. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 738–751. Springer, Heidelberg (2012)
3. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., et al.: Motion capture using joint skeleton tracking and surface estimation. In: Proceedings of CVPR (2009)
4. Ye, M., Yang, R.: Real-time simultaneous pose and shape estimation for articulated objects with a single depth camera. In: Proceedings of CVPR (2014)
5. Baak, A., Muller, M., Bharaj, G., et al.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: Proceedings of ICCV (2011)
6. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3D pose estimation from a single depth image. In: Proceedings of ICCV (2011)
7. Stoll, C., Hasler, N., Gall, J., Seidel, H.P., Theobalt, C.: Fast articulated motion tracking using a sums of Gaussians body model. In: Proceedings of ICCV (2011)
8. Kurmankhojayev, D., Hasler, N., et al.: Monocular pose capture with a depth camera using a Sums-of-Gaussians body model. In: Pattern Recognition (2013)
9. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using RGB and depth data. In: Proceedings of ICCV (2013)
10. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real time motion capture using a single time-of-flight camera. In: Proceedings of CVPR (2010)
11. Helten, T., Muller, M., Seidel, H.P., Theobalt, C.: Real-time body tracking with one depth camera and inertial sensors. In: Proceedings of ICCV (2013)
12. Taylor, J., Shotton, J., et al.: The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In: Proceedings of CVPR (2012)