Research Article

Human Motion Estimation Based on Low Dimensional Space Incremental Learning

Wanyi Li and Jifeng Sun

School of Electronic and Information Engineering, South China University of Technology, Guangzhou, Guangdong 510641, China

Correspondence should be addressed to Wanyi Li; luther1212@163.com

Received 28 September 2014; Revised 5 January 2015; Accepted 9 January 2015

Academic Editor: Mohamed Djemai

Copyright © 2015 W. Li and J. Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a novel algorithm called low dimensional space incremental learning (LDSIL) to estimate the human motion in 3D from the silhouettes of human motion multiview images. The proposed algorithm takes the advantage of stochastic extremum memory adaptive searching (SEMAS) and incremental probabilistic dimension reduction model (IPDRM) to collect new high dimensional data samples. The high dimensional data samples can be selected to update the mapping from low dimensional space to high dimensional space, so that incremental learning can be achieved to estimate human motion from small amount of samples. Compared with three traditional algorithms, the proposed algorithm can make human motion estimation achieve a good performance in disambiguating silhouettes, overcoming the transient occlusion, and reducing estimation error.

1. Introduction

Human motion estimation has become a hot researching topic [1–3], but it is a challenging task. Unusually, we are very interested in estimating the human motion in 3D from the silhouettes of human motion multiview images. The challenges are mainly as follows: firstly, it is hard to build the mapping between multiview silhouettes and human motion in 3D; secondly, the matching is ambiguous between multiview silhouettes and human motion in 3D; finally, it is hard to determine the spatial position information of the human motion depicted in the multiview images. In the past few years, a number of algorithms have been proposed to estimate the human motion. In the works of Sigal et al. [4] and Deutscher and Reid [5], they use some improved particle filters to estimate the human motion in 3D. It cannot work well because of searching (sampling) in high dimensional (HD) space for many times. There are some problems if we directly search in HD space, for example, searching in large scale will get the invalid data and searching in small scale will not get the target data. Moreover, if searching many times in small scale, it also brings about the invalid

data. Li et al. use principal component analysis (PCA) to reduce the dimension of the HD data samples and simulated annealing particle swarm optimism (SAPSO) to estimate the human motion in 3D [6]. This algorithm is time-consuming and its performance is not so well because the HD data converted from corresponding low dimensional (LD) data will be quite different from original HD data. Besides, it does not consider the spatial position information of the human motion. Some traditional Monte Carlo methods [7–9] have drawbacks, which can not ensure collecting the best sample each time during searching, thus stochastic extremum memory adaptive searching (SEMAS) is proposed to solve this problem. In the work of Wang et al. [10], Gaussian process dynamical model (GPDM) can be used to reduce the dimension of the HD data to acquire the corresponding LD data and build the mapping from LD space to HD space, but GPDM can not quickly reduce the dimension of the new HD data sample and acquire the new corresponding LD data; thus incremental probabilistic dimension reduction model (IPDRM) is proposed to solve this problem based on GPDM. Some improved incremental or nonincremental learning algorithms in [11–14] can not satisfy our need.

The limitations are that output data denoting the class label or other simple information have only one dimension, which can not satisfy the description of some output data, and they can not carry out the unsupervised incremental learning of HD data. Inspired by the researches stated above, the key to estimate the human motion in 3D depends on generating the better prior information. The human motion in 3D can be estimated more accurately through searching around the better prior information in small scale only once. In this paper, we mainly focus on the regular human motion cycle (walking or running).

Our task is how to use the small amount of samples of HD data as the prior information to estimate the human motion in 3D which matches the multiview images. Based on the works of the researches above, we proposed a novel algorithm called low dimensional space incremental learning (LDSIL). The LDSIL mainly carries out through SEMAS and IPDRM to collect the new HD samples and updates the mapping from the LD space to HD space through the selection of new HD samples, thus the searching in the LD space can generate the more accurate HD data to estimate the corresponding human motion in 3D. Then, SEMAS is used to find the spatial position of human motion in 3D, and it can find the best data sample during searching more easily. IPDRM is used to reduce the dimension of the new HD data sample and acquire the new corresponding LD data, and it can help to select the new HD samples through the mapping of incremental dimension reduction. Moreover, it provides the LD space to generate the valid HD data. Based on IPDRM, the method of selecting the HD data samples for incremental learning can be achieved by comparing corresponding LD data.

The main contribution of this paper is listed as follows:

- (1) SEMAS is proposed to find the spatial position of human motion model. It can get the best data sample better than the traditional Monte Carlo methods.
- (2) IPDRM is proposed to reduce the dimension of the new HD data sample and acquire the new corresponding LD data. It can promote the incremental learning in LD space to update the mapping from the LD space to HD space. Besides, it provides the LD space to generate valid HD data through searching.
- (3) The method of selecting the HD data samples is proposed; it is used to update the mapping from LD space to HD space.

Overall, due to LDSIL being able to make use of LD data, it can solve these problems mentioned above and contribute a lot to estimating the human motion in 3D, which has the better performance than other traditional algorithms, including disambiguating silhouettes, overcoming the transient occlusion, and reducing estimation error.

The rest of this paper is organized as follows: Section 2 introduces the corresponding data and models, and they are used to estimate the human motion. Section 3 proposes SEMAS algorithm to find the spatial position of human motion model. Section 4 proposes IPDRM to achieve the incremental dimension reduction. Section 5 proposes

orthogonal least squares learning of multiple outputs (OLSLMO) to learn the mapping (HD space to LD space and LD space to HD space). Section 6 discusses LDSIL based on Sections 2–5: this section mentions how to select the new HD data sample from the estimated human motion models to achieve the incremental learning in the LD space. Section 7 proposes the method of searching in LD space to estimate the human motion model; the method is taking the advantage of the SEMAS and IPDRM. Section 8 shows the validity of proposed algorithm (LDSIL) through the experiments and evaluations. Section 9 discusses the limitation of LDSIL algorithm and the improvement in the future.

Let us give the more detailed discussion in the following sections.

2. Corresponding Data and Models

We introduce the corresponding data and models in the works [4, 5]. All image data can be found in HumanEva-I dataset [4], as shown in Figures 1(a)–1(d). Figure 1(a) shows the human motion model denoting the human motion in 3D, which is described by HD data. The model is our estimated object, which need match the limbs in the multiview images. Figure 1(b) shows the multiview images, which are depicting human motion and its spatial position. After using some image segmentation algorithms [15–17] to process the multiview images, we can get the silhouettes as shown in Figure 1(c). Then, we project the model to the corresponding views and obtain the projection images in Figure 1(d). The images in Figure 1(d) are used to compare with the images in Figure 1(c).

In the following, we give some definitions for the above-mentioned data and models. The equation is built as below:

$$w_t^j = p(m_t | {}^p y_t^j) = \prod_{k=1}^{N_v} p(m_t^k | {}^p y_t^j) \quad (1)$$

$$= \exp \left(-\frac{S}{N_v} \sum_{k=1}^{N_v} \left((1 - \alpha) \frac{B_t^k}{B_t^k + H_t^k} + \alpha \frac{R_t^k}{R_t^k + H_t^k} \right) \right),$$

where t denotes the t th frame, k denotes the k th view of camera, N_v is the number of views, α denotes the weight coefficient, p denotes the pixel of the image, M_t^f denotes the set of pixels in the silhouette image, and M_t^b denotes the set of pixels in the projection image of the human motion model. Then, $M_t^f(p)$ is statistical function of p , which is in the silhouette ($M_t^f(p) = 1, p \in M_t^f$ or $M_t^f(p) = 0, p \notin M_t^f$). $M_t^b(p)$ is statistical function of p , which is in the model projection ($M_t^b(p) = 1, p \in M_t^b$ or $M_t^b(p) = 0, p \notin M_t^b$). Thus, in the k th view, $R_t^k = \sum_p (M_t^f(p)(1 - M_t^b(p)))$ denotes the pixel number of p in the silhouette and not in the model projection, $B_t^k = \sum_p (M_t^b(p)(1 - M_t^f(p)))$ denotes the pixel number of p in the model projection and not in the silhouette, and $H_t^k = \sum_p (M_t^f(p)M_t^b(p))$ denotes the pixel number of p in the silhouette and the model projection. Then, the object image consists of multiview silhouette images, m_t^k

FIGURE 1: Data and models.

denotes the k th view feature of object image, m_t denotes multiview features of object image, ${}^p\mathbf{y}_t^i \in R^{D_1}$ and ${}^u\mathbf{y}_t^i \in R^{D_2}$, respectively, denote the HD data samples which contain spatial position information (6 dimensions in total) and no spatial position information, $D_1 - D_2 = 6$, and thus $p(m_t^k | {}^p\mathbf{y}_t^i)$ is the conditional probability of matching image feature of the k th view from the appearance of ${}^p\mathbf{y}_t^i$. Usually, we let $S = 100$ and $\alpha = 0.5$ in most cases. After dimension reduction, the corresponding LD data of the HD data samples with no spatial position information can be seen in Figure 1(e). The HD data samples are the small amount of samples which cover one cycle of human motion.

For the better description of our proposed algorithm in the following section, we need to define the symbols of some operations as follows. $\mathbf{y}(q_1 : q_2)$ is extracting the q_1 th- q_2 th dimension data of vector $\mathbf{y}$ as a subvector; ${}^1\mathbf{A}$ and ${}^1\mathbf{B}$ are matrices or vectors, ${}^1\mathbf{A} * {}^1\mathbf{B} = \mathbf{D}^1$, $\mathbf{D}_{i,j}^1 = {}^1\mathbf{A}_{i,j} {}^1\mathbf{B}_{i,j}$; $\text{ones}(n_1, n_2) \in R^{n_1 \times n_2}$ is the matrix whose elements are 1; $\text{rand}(n_1, n_2) \in R^{n_1 \times n_2}$ is the matrix whose elements are conforming the distribution $U(-1, 1)$; the set can be described by $\{m_1, m_1 + 1, \dots, m_2\} = m_1 : m_2$, $m_1 < m_2$, $m_1, m_2 \in N$; $\|\mathbf{H}\| = \|\mathbf{H}\|_2 = (\sum_i \sum_j |\mathbf{H}_{i,j}|^2)^{1/2}$ is the calculation of 2-norm (Euclidean-norm), and H can denote the vector or matrix.

3. Stochastic Extremum Memory Adaptive Searching

We will propose SEMAS algorithm in this section, which is used to find the spatial position of human motion model. The spatial position information includes the angles and coordinates of root marker in the model, which is the low dimensional data. In time t (the t th frame), let ${}^p\mathbf{y}_t^i(q_1 : q_2)$ denote the spatial position information of ${}^p\mathbf{y}_t^i \in R^{D_1}$, the set $C_q = \{q_1, q_1 + 1, \dots, q_2\} = q_1 : q_2$, $C_q \cup \bar{C}_q = \{1, 2, \dots, D_1\}$, and $\mathbf{y}_p^i = {}^p\mathbf{y}_t^i(q_1 : q_2)$ and $\mathbf{y}_a = {}^p\mathbf{y}_t^i(\bar{C}_q)$; we get

$$\begin{aligned} w_t^i &= p(m_t | {}^p\mathbf{y}_t^i) = p(m_t | {}^p\mathbf{y}_t^i(q_1 : q_2), {}^p\mathbf{y}_t^i(\bar{C}_q)) \\ &= p(m_t | \mathbf{y}_p^i, \mathbf{y}_a) = f_p(\mathbf{y}_p^i), \end{aligned} \quad (2)$$

where the SEMAS can be carried out as follows. We denote $\hat{k}_1$ as the ordinal number of maximum element in $\mathbf{W}_o = \{w_t^1, w_t^2, \dots, w_t^{N_e}\}$, $w_t^{\hat{k}_1} > w_t^i$, $i = 1, 2, \dots, N_e$, $i \neq \hat{k}_1$, and ${}_r\mathbf{y}_p^i$ is a sample of the r th time searching in time t , ${}_r\mathbf{y}_p^i \in R^d$, $d = q_2 - q_1 + 1$, $1 \leq d \leq 6$. Then, we let $\mathbf{u}_t^i = \mathbf{v} * \text{rand}(d, 1)$, $\mathbf{v} \in R^d$ is the scale parameter vector, $\eta_i = \text{rand}(1, 1)$, and L is the searching parameter of the step size; thus the $N_e - 1$ searching vectors can be given by $\bar{\mathbf{u}}_t^i = \eta_i L(\mathbf{u}_t^i / \|\mathbf{u}_t^i\|)$ or $L\mathbf{u}_t^i$, $i = 1, 2, \dots, N_e - 1$; the $N_e - 1$ new data samples can be obtained by ${}_{r+1}\mathbf{y}_p^i = {}_r\mathbf{y}_p^{\hat{k}_1} + \bar{\mathbf{u}}_t^i$, $i = 1, 2, \dots, N_e - 1$. Here, the extremum needs to be kept, let ${}_{r+1}\mathbf{y}_p^{N_e} = {}_r\mathbf{y}_p^{\hat{k}_1}$, and, in the next time searching, it is recording the extremum for ensuring obtaining the data samples which is not worse than the ones

in the last time searching. In each searching, we can get N_e data samples. After searching twice, we can adjust the value of L according to the equation as below:

$$L = \begin{cases} A_1 + \gamma_1 \left\| {}_{r-1}\mathbf{y}_p^{\hat{k}_1} - {}_{r-2}\mathbf{y}_p^{\hat{k}_1} \right\|, \\ \left| p(m_t | {}_{r-1}\mathbf{y}_p^{\hat{k}_1}, \mathbf{y}_a) - p(m_t | {}_{r-2}\mathbf{y}_p^{\hat{k}_1}, \mathbf{y}_a) \right| < \eta', \\ A_2 + \gamma_2 \left\| {}_{r-1}\mathbf{y}_p^{\hat{k}_1} - {}_{r-2}\mathbf{y}_p^{\hat{k}_1} \right\|, \\ \left| p(m_t | {}_{r-1}\mathbf{y}_p^{\hat{k}_1}, \mathbf{y}_a) - p(m_t | {}_{r-2}\mathbf{y}_p^{\hat{k}_1}, \mathbf{y}_a) \right| \geq \eta', \end{cases}$$

$$r > 1, \quad \gamma_1 \geq \gamma_2 \geq 0, \quad A_1 \geq A_2 > 0, \quad \eta' > 0, \quad (3)$$

where when the subtraction of the best object values in the past two times searching is below some value, L can be enlarged to search better data samples; on the contrary, if the subtraction is above some value, it illustrates that the value of L can be shrunk to avoid missing the best data sample. Thus, the method can make the convergence faster. After r times searching (r is large enough), when $f_p(\mathbf{y}_p^i)$ is unimodal approximately, the best sample $\mathbf{y}_p^*$ can be obtained by

$$\mathbf{y}_p^* = \arg \max_{\mathbf{y}_p^i} f_p(\mathbf{y}_p^i) \approx {}_r\mathbf{y}_p^{\hat{k}_1}, \quad (4)$$

and the derivation of (4) and the pseudocode can be seen in the appendices.

Then, we compare and analyze the performances between traditional Monte Carlo method and SEMAS as depicted in Figures 2(a) and 2(b). In Figure 2, we take 4 times searching and 3 data samples each time for example. Figure 2(a) shows that the traditional Monte Carlo method can not ensure that the searched data samples are not worse than the last time, which keeps the larger weight data samples simple. The reason is that it does not keep the best data sample each time during searching, which can not be compared with the data samples searched in the next time; moreover, the length and direction of searching vector are stochastic and uncontrolled. After several times searching, in most cases, it may appear that all of the searched data samples can not close the $f_p^* = \max(f_p)$ better, and the mean $\bar{f}_p = \sum \bar{w}_t^i {}_r\mathbf{y}_p^i$, $\bar{w}_t^i = f_p({}_r\mathbf{y}_p^i) / \sum f_p({}_r\mathbf{y}_p^i)$ can not reach f_p^* also. As shown in Figure 2(b), the SEMAS will keep the best data sample in each time searching, which will be compared with the searched data samples in the next time searching. It can ensure that the searched data samples can not be worse than the last time searching and can adjust the length of searching vector to control the searching scale, according to subtraction of the two best values of objective function f_p in the past twice searching; thus it will close the $f_p^* = \max(f_p)$ better and has more chance to get the better data samples than traditional Monte Carlo method.

FIGURE 2: The comparison of traditional Monte Carlo method and SEMAS.

4. Incremental Probabilistic Dimension Reduction Model

4.1. Probabilistic Models of GPDM. We give the probabilistic models of the GPDM in [10] as follows:

$$\begin{aligned}
 p(\mathbf{Y} | \mathbf{X}, \bar{\boldsymbol{\beta}}, \mathbf{W}) &= \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND} |\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_Y^{-1} \mathbf{Y} \mathbf{W}^2 \mathbf{Y}^T)\right), \quad (5) \\
 p(\mathbf{X} | \bar{\boldsymbol{\alpha}}) &= \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)q} |\mathbf{K}_X|^q}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_X^{-1} \mathbf{X}_{2:N} \mathbf{X}_{2:N}^T)\right), \quad (6)
 \end{aligned}$$

In (5), HD data sequence can be denoted by $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_i, \dots, \mathbf{y}_N]^T \in R^{N \times D}$, $\mathbf{y}_i \in R^D$, and LD data sequence can be denoted by $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N]^T \in R^{N \times q}$, $\mathbf{x}_i \in R^q$. Kernel matrix with parameter $\bar{\boldsymbol{\beta}} = [\beta_1, \beta_2]$ is $\mathbf{K}_Y \in R^{N \times N}$, $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j) = \exp(-(\beta_1/2)\|\mathbf{x}_i - \mathbf{x}_j\|^2) + \beta_2^{-1} \delta_{\mathbf{x}_i, \mathbf{x}_j}$; $\mathbf{W}$ is scale parameter matrix, $p(\mathbf{W}) = \prod_{m=1}^D (2/(\kappa\sqrt{2\pi})) \exp(-w_m^2/(2\kappa^2))$, $w_m > 0$, $\kappa = 10^{-3}$; in (6),

kernel matrix with parameter $\bar{\boldsymbol{\alpha}} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ is $\mathbf{K}_X \in R^{(N-1) \times (N-1)}$, $(\mathbf{K}_X)_{i,j} = k_X(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1 \exp(-(\alpha_2/2)\|\mathbf{x}_i - \mathbf{x}_j\|^2) + \alpha_3 \mathbf{x}_i^T \mathbf{x}_j + \alpha_4^{-1} \delta_{\mathbf{x}_i, \mathbf{x}_j}$, $\mathbf{X}_{1:N-1} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N-1}]^T$, $\mathbf{X}_{2:N} = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N]^T$, and $\mathbf{x}_1$ conforms Gaussian distribution of q dimensions. $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ satisfy $p(\bar{\boldsymbol{\alpha}}) \propto \prod_i \alpha_i^{-1}$ and $p(\bar{\boldsymbol{\beta}}) \propto \prod_i \beta_i^{-1}$, respectively. Then, we give the conditional distribution as follows:

$$\begin{aligned}
 p(\mathbf{Y}^* | \mathbf{X}^*, \Gamma) &= \frac{|\mathbf{W}|^M}{\sqrt{(2\pi)^{MD} |\mathbf{K}_{Y^*}|^D}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_{Y^*}^{-1} \mathbf{Z}_Y \mathbf{W}^2 \mathbf{Z}_Y^T)\right), \quad (7) \\
 p(\mathbf{X}^* | \Gamma) &= \frac{p(\mathbf{x}_1^*)}{\sqrt{(2\pi)^{(M-1)q} |\mathbf{K}_{X^*}|^q}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_{X^*}^{-1} \mathbf{Z}_X \mathbf{Z}_X^T)\right), \quad (8)
 \end{aligned}$$

where $\Gamma = \{\mathbf{X}, \mathbf{Y}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W}\}$, $\mathbf{Y}^* \in R^{M \times D}$, $\mathbf{X}^* \in R^{M \times q}$, $\mathbf{Z}_Y = \mathbf{Y}^* - \mathbf{A}^T \mathbf{K}_Y^{-1} \mathbf{Y}$, $\mathbf{K}_{Y^*} = \mathbf{B} - \mathbf{A}^T \mathbf{K}_Y^{-1} \mathbf{A}$, $\mathbf{Z}_X = \mathbf{X}_{2:N}^* - \mathbf{C}^T \mathbf{K}_X^{-1} \mathbf{X}_{2:N}$, and $\mathbf{K}_{X^*} = \mathbf{D} - \mathbf{C}^T \mathbf{K}_X^{-1} \mathbf{C}$. Kernel matrices include $\mathbf{A} \in R^{N \times M}$,

$\mathbf{A}_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j^*)$, $\mathbf{B} \in R^{M \times M}$, $\mathbf{B}_{i,j} = k_Y(\mathbf{x}_i^*, \mathbf{x}_j^*)$, $\mathbf{C} \in R^{(N-1) \times (M-1)}$, $\mathbf{C}_{i,j} = k_X(\mathbf{x}_i, \mathbf{x}_j^*)$ and $\mathbf{D} \in R^{(M-1) \times (M-1)}$, and $\mathbf{D}_{i,j} = k_X(\mathbf{x}_i^*, \mathbf{x}_j^*)$.

When training GPDM, $\mathbf{Y}$ is known; thus $p(\mathbf{Y})$ is constant, and $\min(-\ln p(\mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W} \mid \mathbf{Y})) \Leftrightarrow \min(-\ln p(\mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W}, \mathbf{Y}))$. Then, the LD data and corresponding parameters can be calculated as below:

$$\begin{aligned} & \mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W} \\ &= \arg \min_{\mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W}} (-\ln p(\mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W}, \mathbf{Y})) \\ &= \arg \min_{\mathbf{X}, \bar{\alpha}, \bar{\beta}, \mathbf{W}} (-\ln(p(\mathbf{Y} \mid \mathbf{X}, \bar{\beta}, \mathbf{W}) \\ & \quad \cdot p(\mathbf{X} \mid \bar{\alpha}) p(\bar{\alpha}) p(\bar{\beta}) p(\mathbf{W}))). \end{aligned} \quad (9)$$

Optimization of (9) can use the method of scaled conjugate gradient (SCG) [18, 19]. After the optimization, $\mathbf{X}$ can be depicted as shown in Figure 1(e). According to (7), $\mathbf{y}^* \sim \mathcal{N}(\mu_Y(\mathbf{x}^*), \sigma_Y^2(\mathbf{x}^*)\mathbf{I})$, the mapping from LD data to HD data can be given by the mean as below:

$$\begin{aligned} \mathbf{y}^* &= \mathbf{f}(\mathbf{x}^*) = \mu_Y(\mathbf{x}^*) \\ &= \mathbf{Y}^T \mathbf{K}_Y^{-1} [k_Y(\mathbf{x}_1, \mathbf{x}^*), k_Y(\mathbf{x}_2, \mathbf{x}^*), \dots, k_Y(\mathbf{x}_N, \mathbf{x}^*)]^T \\ &= \mathbf{Y}^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}^*). \end{aligned} \quad (10)$$

4.2. The Mapping of Incremental Dimension Reduction. After finishing training, GPDM can not process the new HD data sample, which means it can not acquire the corresponding LD data. From (10), $\mathbf{y}^*$ is known, the $\mathbf{x}^*$ is embedded in nonlinear kernel, and thus $\mathbf{x}^*$ is hard to be solved through this equation. Thus, we need to build the mapping from HD data to LD data and acquire the LD data of new HD data sample fast.

We denote HD data samples not containing spatial position information as $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1^{\text{up}}, \dots, \tilde{\mathbf{y}}_i^{\text{up}}, \dots, \tilde{\mathbf{y}}_{N_{\text{up}}}^{\text{up}}]^T \in R^{N_{\text{up}} \times D_2}$; then we denote the corresponding LD data as $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_i, \dots, \tilde{\mathbf{x}}_{N_{\text{up}}}]^T \in R^{N_{\text{up}} \times q}$, which can be acquired by GPDM. The mapping can be built through the training model as below:

$$\tilde{\mathbf{X}}^T = \mathbf{W}_D \Phi + \mathbf{E}, \quad (11)$$

where $\mathbf{W}_D \in R^{q \times Nk}$ is the weight matrix and $\Phi \in R^{Nk \times N_{\text{up}}}$ is the matrix of Gaussian basis function; thus we have $\Phi_{k_1, k_2} = \phi(\text{up} \tilde{\mathbf{y}}_{k_2}, \mathbf{c}_{k_1}) = \exp(b \|\text{up} \tilde{\mathbf{y}}_{k_2} - \mathbf{c}_{k_1}\|^2)$, $\mathbf{E} \in R^{q \times N_{\text{up}}}$ is the bias matrix, and $Nk \leq N_{\text{up}}$. b is known; after finishing training of (11), $\mathbf{c}_{k_1}$, $\mathbf{W}_D$, and $\mathbf{E}$ can be confirmed, here, $\widehat{\mathbf{W}}_D$ and $\widehat{\mathbf{E}} = \widehat{\mathbf{B}}_p \text{ones}(1, N_{\text{up}})$ denote the trained values of $\mathbf{W}_D$ and $\mathbf{E}$, respectively, $\widehat{\mathbf{B}}_p \in R^{q \times 1}$. Then, we have the mapping of incremental dimension reduction as below:

$$\mathbf{x}_t^* = \mathbf{g}(\text{up} \mathbf{y}_t^*) = \widehat{\mathbf{W}}_D \Phi(\text{up} \mathbf{y}_t^*) + \widehat{\mathbf{E}}, \quad (12)$$

where as to one HD new samples $\text{up} \mathbf{y}_t^*$, let $N_{\text{up}} = 1$, $\Phi(\text{up} \mathbf{y}_t^*) = [\phi(\text{up} \mathbf{y}_t^*, \mathbf{c}_1), \phi(\text{up} \mathbf{y}_t^*, \mathbf{c}_2), \dots, \phi(\text{up} \mathbf{y}_t^*, \mathbf{c}_{Nk})]^T$, and $\widehat{\mathbf{E}} = \widehat{\mathbf{B}}_p \text{ones}(1, 1)$. The mapping as (12) can acquire the corresponding LD data of the new HD data sample; furthermore, it can acquire the multiple corresponding ones of the multiple HD data samples, which only need to add the corresponding column in Φ and $\widehat{\mathbf{E}}$. Equations (5)–(12) constitute IPDRM; however, the learning method of (11) is the key to estimate the human motion in 3D; thus we will discuss it in the next section.

In the following, we need to discuss the several advantages of IPDRM. Firstly, the model is used to build the LD space through the known HD data samples, acquire the LD data of the new HD data sample, and provide the human motion global prior information in the LD space. The searching position in the LD space can be confirmed to disambiguate silhouettes through clustering. This is because the mapping from HD space to LD space is one-to-one. Secondly, it can reduce the estimation error and overcome the transient occlusion through linear and nonlinear searching in the LD space via the global prior information; moreover, the HD data samples generated through the large scale nonlinear searching between the two neighbours LD data in the LD space are better and more valid than the large scale nonlinear searching between the two neighbours HD data in the HD space directly, which can be tested as shown in Figure 3. Let us see the two experiments (walking and running models); we find that the HD data which are generated through nonlinear searching in the LD space (Figure 3(b)) and converted to human motion models are more like human motion shape than the ones which are generated through the nonlinear searching in the HD space (Figure 3(a)). Finally, searching in the LD space can have lower computation cost than searching in the HD space obviously.

5. Orthogonal Least Squares Learning of Multiple Outputs

In this section, we propose a method called orthogonal least squares learning of multiple outputs (OLSLMO). In the LD space or HD space, the data are multiple dimensions, for which the feature or model can be described better. Let us denote the model as follows:

$$\mathbf{D} = \mathbf{P} \tilde{\boldsymbol{\theta}} + \mathbf{e}. \quad (13)$$

Here, matrices of the output and input vectors are, respectively, $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k, \dots, \mathbf{d}_N]^T = [\mathbf{d}'_1, \dots, \mathbf{d}'_c, \dots, \mathbf{d}'_{Q_1}] \in R^{N \times Q_1}$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_k, \dots, \mathbf{v}_N]^T \in R^{N \times Q_2}$; $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_k, \dots, \mathbf{p}_M] \in R^{N \times M}$, $\mathbf{P}_{k,k'} = f_e(\mathbf{v}_k, \mathbf{z}_{k'})$, $\mathbf{z}_{k'} \in \{\mathbf{v}_k \mid k = 1, 2, \dots, N\}$, and $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{k'}, \dots, \boldsymbol{\theta}_M]^T = [\boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_c, \dots, \boldsymbol{\theta}'_{Q_1}] \in R^{M \times Q_1}$; error matrix is $\mathbf{e} = [\mathbf{e}_1, \dots, \mathbf{e}_k, \dots, \mathbf{e}_N]^T = [\mathbf{e}'_1, \dots, \mathbf{e}'_c, \dots, \mathbf{e}'_{Q_1}] \in R^{N \times Q_1}$. Then, the matrix $\mathbf{P}$ can be decomposed into this form $\mathbf{P} = \widehat{\mathbf{W}} \mathbf{A}$, and $\widehat{\mathbf{W}} = [\mathbf{w}_1, \dots, \mathbf{w}_{k'}, \dots, \mathbf{w}_M] \in R^{N \times M}$ is the matrix with orthogonal vector $\mathbf{w}_{k'}$ ($\mathbf{w}_{k'}^T \mathbf{w}_{\bar{k}'} = 0$, $k' \neq \bar{k}'$; $\mathbf{w}_{k'}^T \mathbf{w}_{\bar{k}'} \neq 0$, $k' = \bar{k}'$; $\bar{k}', k' = 1, 2, \dots, M$), $\widehat{\mathbf{W}}^T \widehat{\mathbf{W}} \in R^{M \times M}$ is a diagonal

FIGURE 3: Comparison of searching in the HD space and LD space among neighbour data samples.

matrix; $\mathbf{A} \in R^{M \times M}$ is the invertible matrix with 1 on the diagonal given by

$$\mathbf{A}_{i,j} = \begin{cases} a_{i,j}, & i < j, \\ 1, & i = j, \\ 0, & i > j; \end{cases} \quad (14)$$

let least squares (LS) estimator $\tilde{\boldsymbol{\theta}} = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{D}$, $\tilde{\mathbf{g}} = (\tilde{\mathbf{W}}^T \tilde{\mathbf{W}})^{-1} \tilde{\mathbf{W}}^T \mathbf{D}$, and we can give the derivation as below:

$$\begin{aligned} \mathbf{P} \tilde{\boldsymbol{\theta}} &= \tilde{\mathbf{W}} \mathbf{A} \tilde{\boldsymbol{\theta}} = \tilde{\mathbf{W}} \mathbf{A} (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{D} \\ &= \tilde{\mathbf{W}} \mathbf{A} (\mathbf{A}^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \mathbf{A})^{-1} \mathbf{A}^T \tilde{\mathbf{W}}^T \mathbf{D} \\ &= \tilde{\mathbf{W}} (\tilde{\mathbf{W}}^T \tilde{\mathbf{W}})^{-1} \tilde{\mathbf{W}}^T \mathbf{D} = \tilde{\mathbf{W}} \tilde{\mathbf{g}}. \end{aligned} \quad (15)$$

Equation (13) can be written by

$$\mathbf{D} = \tilde{\mathbf{W}} \tilde{\mathbf{g}} + \mathbf{e}, \quad (16)$$

and let $\tilde{\mathbf{g}} = [\mathbf{g}_1, \dots, \mathbf{g}_{k'}, \dots, \mathbf{g}_M]^T = [\tilde{\mathbf{g}}_1, \dots, \tilde{\mathbf{g}}_c, \dots, \tilde{\mathbf{g}}_{Q_1}] \in R^{M \times Q_1}$; according to property of LS estimation, we have

$$\begin{aligned} \min \left(\frac{\|\mathbf{D} - \mathbf{P} \tilde{\boldsymbol{\theta}}\|^2}{(N \times Q_1)} \right) \\ = \min \left(\frac{\|\mathbf{D} - \tilde{\mathbf{W}} \tilde{\mathbf{g}}\|^2}{(N \times Q_1)} \right) = \min \left(\frac{\text{tr}(\mathbf{e}^T \mathbf{e})}{(N \times Q_1)} \right) \end{aligned}$$

$$\begin{aligned} &= \min \left(\frac{1}{Q_1} \sum_{c=1}^{Q_1} \left(\frac{1}{N} (\mathbf{e}'_c)^T \mathbf{e}'_c \right) \right) \\ &= \min \left(\frac{1}{Q_1} \sum_{c=1}^{Q_1} \left(\frac{1}{N} (\mathbf{d}'_c)^T \mathbf{d}'_c - \frac{1}{N} \sum_{k'=1}^M \mathbf{w}_{k'}^T \mathbf{w}_{k'} \tilde{\mathbf{g}}_c^2(k') \right) \right), \end{aligned} \quad (17)$$

and let $S_w = \{\mathbf{w}_1, \dots, \mathbf{w}_{k'}, \dots, \mathbf{w}_M\}$, $S_{w'} = \{\mathbf{w}'_1, \dots, \mathbf{w}'_{k'}, \dots, \mathbf{w}'_N\}$; minimizing (17) can be achieved by

$$\begin{aligned} \max_{S_w} \left(\sum_{c=1}^{Q_1} \sum_{k'=1}^M \left(\frac{\mathbf{w}_{k'}^T \mathbf{w}_{k'} \tilde{\mathbf{g}}_c^2(k')}{(\mathbf{d}'_c)^T \mathbf{d}'_c} \right)^{\tilde{n}} \right), \\ 0 < M \leq N, \quad S_w \in S_{w'}, \quad \tilde{n} = 1, 2, \end{aligned} \quad (18)$$

where $g_{k'} = \mathbf{w}_{k'}^T \mathbf{D} / (\mathbf{w}_{k'}^T \mathbf{w}_{k'})$; let $\mathbf{P}' = [\mathbf{p}'_1, \dots, \mathbf{p}'_{k'}, \dots, \mathbf{p}'_N] \in R^{N \times N}$, $\mathbf{p}'_{k,k'} = f_e(\mathbf{v}_{k'}, \mathbf{z}_{k'})$, and $S_{w'}$ is corresponding orthogonal vector set from vector set $\{\mathbf{p}'_1, \dots, \mathbf{p}'_{k'}, \dots, \mathbf{p}'_N\}$. The value of M can be confirmed according to the condition $\|\mathbf{D} - \mathbf{P} \tilde{\boldsymbol{\theta}}\|^2 / (N \times Q_1) = \|\mathbf{D} - \tilde{\mathbf{W}} \tilde{\mathbf{g}}\|^2 / (N \times Q_1) = \text{tr}(\mathbf{e}^T \mathbf{e}) / (N \times Q_1) < \varepsilon_1$, $\varepsilon_1 > 0$. When this condition is satisfied, we let

$$\mathbf{D} = \mathbf{P} \hat{\boldsymbol{\theta}} + \mathbf{E}' + \mathbf{e}' = [\mathbf{P}, \mathbf{R}_I] \begin{bmatrix} \hat{\boldsymbol{\theta}}^T, \mathbf{B}_p^T \end{bmatrix}^T + \mathbf{e}' = \mathbf{G} \hat{\mathbf{F}} + \mathbf{e}', \quad (19)$$

where $\mathbf{R}_I = \text{ones}(N, 1) \in R^{N \times 1}$ and error matrix is $\mathbf{e}' \in R^{N \times Q_1}$. According to the equation and updating method [20] of LS, $\hat{\mathbf{D}}$ can denote the LS estimator of $\mathbf{D}$, and we let

$\hat{\mathbf{F}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{D}$, $\mathbf{P}_G = \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T$, $\mathbf{P}_\theta = \mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$, and $\mathbf{P}_{1/\theta} = \mathbf{I} - \mathbf{P}_\theta$ and get

$$\begin{aligned} \hat{\mathbf{D}} &= \mathbf{G}\hat{\mathbf{F}} = \mathbf{P}_G \mathbf{D} = \mathbf{P}_\theta \mathbf{D} + \mathbf{P}_{1/\theta} \mathbf{R}_I (\mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I)^{-1} \mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{D} \\ &= \tilde{\mathbf{P}}\theta + \mathbf{P}_{1/\theta} \mathbf{R}_I (\mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I)^{-1} \mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{D}, \end{aligned} \quad (20)$$

where if condition $\|\mathbf{D} - \tilde{\mathbf{P}}\theta\|^2 / (N \times Q_1) = \|\mathbf{D} - \tilde{\mathbf{W}}\mathbf{g}\|^2 / (N \times Q_1) = \text{tr}(\mathbf{e}^T \mathbf{e}) / (N \times Q_1) < \varepsilon_1$, $\varepsilon_1 > 0$ is satisfied, and if condition $\|\mathbf{D} - \mathbf{G}\hat{\mathbf{F}}\|^2 / (N \times Q_1) = \|\mathbf{D} - \mathbf{P}_G \mathbf{D}\|^2 / (N \times Q_1) = \text{tr}((\mathbf{e}')^T \mathbf{e}') / (N \times Q_1) < \varepsilon_2 < \varepsilon_1$, $\varepsilon_2 > 0$ can also be satisfied; because $\mathbf{P}_\theta$ is determined, $\mathbf{D}$ can get further approximation based on $\tilde{\mathbf{P}}\theta$ which is playing the leading role during training. The more detailed deviation of the condition can be seen in appendices. Then, we have $\hat{\theta} = \hat{\mathbf{F}}_{1:M,:} \in R^{M \times Q_1}$, $\mathbf{B}_p = \hat{\mathbf{F}}_{M+1,:} \in R^{1 \times Q_1}$, and $\hat{\mathbf{E}}' = \mathbf{R}_I \mathbf{B}_p \in R^{N \times Q_1}$.

Then, we summarize the algorithm of OLSLMO as in Algorithm 1.

The learning of (11) can use Algorithm 1, we let input parameters $\mathbf{D} = \tilde{\mathbf{X}}$, $\mathbf{V} = \tilde{\mathbf{Y}}$, $Q_1 = q$, and $Q_2 = D_2$ and functions $f_e(\mathbf{v}_k, \mathbf{z}_{k'}) = \phi(\text{up}\tilde{\mathbf{y}}_{k_2}, \mathbf{c}_{k_1})$, $\mathbf{v}_k = \text{up}\tilde{\mathbf{y}}_{k_2}$, $k = k_2 = 1, 2, \dots, N_{\text{up}}$, and $N = N_{\text{up}}$, and then get corresponding output parameters $\mathbf{c}_{k_1} = \mathbf{z}_{k'}$, $Nk = M$, $k_1 = k' = 1, 2, \dots, Nk$, $\tilde{\mathbf{W}}_D = \hat{\theta}^T$, and $\hat{\mathbf{E}} = [\text{ones}(1, 1) \mathbf{B}_p]^T$.

6. Low Dimensional Space Incremental Learning

After using low dimensional space searching (LDSS) to estimate some human motion models of corresponding frames in 3D (about one motion cycle), the new HD data samples can be got from the models. The LDSS algorithm is given in the framework of LDSIL shown in Section 7.2. Then, continuing to use the mapping of (10) to generate HD data sample will be less accurate, thus incremental learning is needed to update the mapping from LD space to HD space. Equation (10) is so complicated that its parameters can not be updated easily, due to the need of calculation in (9). For this consideration, we still use the model like (11) to build the mapping, which can take advantage of new HD data samples to learn through the method of Algorithm 1. The details of incremental learning will be discussed as follows.

The key of incremental learning is how to select the HD data samples to update the mapping from LD space to HD space. The better new samples used to training the mapping can be selected from the estimated HD data with spatial position information removed. The estimated HD data samples contain errors, which can convert into human motion models of corresponding frames. After we get corresponding new HD data samples $\tilde{\mathbf{Y}}_{\text{new}}^* = [\text{up}\tilde{\mathbf{y}}_1^*, \dots, \text{up}\tilde{\mathbf{y}}_{n^*}^*, \dots, \text{up}\tilde{\mathbf{y}}_{N^*}^*]^T \in R^{N^* \times D_2}$, we can use (12) to obtain the corresponding LD data, $(\tilde{\mathbf{X}}_{\text{new}}^*)^T = \mathbf{g}((\tilde{\mathbf{Y}}_{\text{new}}^*)^T) = [\mathbf{g}(\text{up}\tilde{\mathbf{y}}_1^*), \dots, \mathbf{g}(\text{up}\tilde{\mathbf{y}}_{n^*}^*), \dots, \mathbf{g}(\text{up}\tilde{\mathbf{y}}_{N^*}^*)]$, and get $\tilde{\mathbf{X}}_{\text{new}}^* = [\tilde{\mathbf{x}}_1^*, \dots, \tilde{\mathbf{x}}_{n^*}^*, \dots, \tilde{\mathbf{x}}_{N^*}^*]^T \in R^{N^* \times q}$. We know the LD data is representing the feature of HD data; thus the distance

Input: $\mathbf{D}$, $\mathbf{V}$, $\tilde{n}$, b ;

Output: $\mathbf{z}_{k'}, k' = 1, 2, \dots, M$, $\hat{\theta}$, $\mathbf{B}_p$;

For $k = 1 : 1 : N$

If $k = 1$

$\mathbf{w}_k^s = \mathbf{p}'_s$, $s = 1, 2, \dots, N$;

Else If $k \geq 2$

$\alpha_{ik}^s = \frac{\mathbf{w}_i^T \mathbf{p}'_s}{\mathbf{w}_i^T \mathbf{w}_i}$, $s = 1, 2, \dots, N$, $s \notin \{In_{1:k-1}\}$, $1 \leq i < k$;

$\mathbf{w}_k^s = \mathbf{p}'_s - \sum_{i=1}^{k-1} \alpha_{ik}^s \mathbf{w}_i$;

End If

$\mathbf{g}_k^s = \frac{(\mathbf{w}_k^s)^T \mathbf{D}}{(\mathbf{w}_k^s)^T \mathbf{w}_k^s}$;

$EV^s = \sum_{c=1}^{Q_1} \left(\frac{(\mathbf{w}_k^s)^T \mathbf{w}_k^s (\mathbf{g}_1^s(c))^2}{(\mathbf{d}'_c)^T \mathbf{d}'_c} \right)^{\tilde{n}}$;

$s^* = \arg \max_s EV^s$; $\mathbf{w}_k = \mathbf{w}_k^{s^*}$;

$In_k = s^*$; $\mathbf{z}_k = \mathbf{v}_{In_k}$; $\mathbf{G} = [\mathbf{P}'_{:,In_{1:k}}, \mathbf{R}_I]$;

$\hat{\mathbf{F}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{D}$; $\hat{\theta} = \hat{\mathbf{F}}_{1:k,:}$;

$\mathbf{B}_p = \hat{\mathbf{F}}_{k+1,:}$; $\hat{\mathbf{E}}' = \mathbf{R}_I \mathbf{B}_p$;

If $\|\mathbf{D} - \mathbf{G}\hat{\mathbf{F}}\|^2 / (N \times Q_1) < \varepsilon_2$

Break;

End If

End For

$M = k$;

ALGORITHM 1

among LD data can describe the similarity of HD data, and we need to select the LD data from $\tilde{\mathbf{X}}_{\text{new}}^*$, which are well-distributed in the LD space for getting the corresponding HD data. The selecting rule can be carried out as below. We denote ordinal number as below:

$$\begin{aligned} f_L &= i, \quad 1 \leq i \leq N_{\text{up}}, \\ b_L &= i + 1, \quad 1 \leq i < N_{\text{up}}, \\ b_L &= 1, \quad i = N_{\text{up}}, \end{aligned} \quad (21)$$

and then, we have

$$\begin{aligned} D_{n^*}^* &= \min D_{\text{se}}, \quad D_{\text{se}} \in \{\|\tilde{\mathbf{x}}_{n^*}^* - \tilde{\mathbf{x}}_{f_L}\|, \|\tilde{\mathbf{x}}_{n^*}^* - \tilde{\mathbf{x}}_{b_L}\|\}, \\ n^* &= 1, 2, \dots, N^*, \quad N^* > N_{\text{up}}, \end{aligned} \quad (22)$$

and the LD data and the corresponding HD data can be selected by

$$\begin{aligned} \mathbf{X}_{\text{se}}^* &= (\mathbf{X}_{\text{new}}^*)_{n_{\text{se}}^*}, \quad \mathbf{Y}_{\text{se}}^* = (\mathbf{Y}_{\text{new}}^*)_{n_{\text{se}}^*}, \\ n_{\text{se}} &= \{n^* \mid D_{n^*}^* < c_s \|\mathbf{x}_{f_L} - \mathbf{x}_{b_L}\|, n^* = 1, 2, \dots, N^*\}, \\ 0 &< c_s \leq 1, \end{aligned} \quad (23)$$

where we let $\tilde{\mathbf{X}}_{se}^* = [\tilde{\mathbf{x}}_1^*, \dots, \tilde{\mathbf{x}}_{i_s}^*, \dots, \tilde{\mathbf{x}}_{N_s}^*]^T \in R^{N_s \times q}$ and $\tilde{\mathbf{Y}}_{se}^* = [\tilde{\mathbf{y}}_1^*, \dots, \tilde{\mathbf{y}}_{i_s}^*, \dots, \tilde{\mathbf{y}}_{N_s}^*]^T \in R^{N_s \times D_2}$ and then get

$$i_s^* = \arg \min_{i_s} \left\| \tilde{\mathbf{x}}_{i_s}^* - \tilde{\mathbf{x}}_{f_L} \right\| - \left\| \tilde{\mathbf{x}}_{i_s}^* - \tilde{\mathbf{x}}_{b_L} \right\|, \quad i_s = 1, 2, \dots, N_s. \quad (24)$$

Equation (24) means selecting the best well-distributed LD data. Thus, the training data samples of input and output are given by

$$\tilde{\mathbf{X}}_{n_{even}}^{tra} = bs \tilde{\mathbf{X}}^* = [bs \tilde{\mathbf{x}}_1^*, \dots, bs \tilde{\mathbf{x}}_{i_{bs}}^*, \dots, bs \tilde{\mathbf{x}}_{N_{up}}^*]^T \in R^{N_{up} \times q},$$

$$bs \tilde{\mathbf{x}}_{i_{bs}}^* = s \tilde{\mathbf{x}}_{i_s}^*,$$

$$\tilde{\mathbf{Y}}_{n_{even}}^{tra} = bs \tilde{\mathbf{Y}}^* = [bs \tilde{\mathbf{y}}_1^*, \dots, bs \tilde{\mathbf{y}}_{i_{bs}}^*, \dots, bs \tilde{\mathbf{y}}_{N_{up}}^*]^T \in R^{N_{up} \times D_2},$$

$$bs \tilde{\mathbf{y}}_{i_{bs}}^* = s \tilde{\mathbf{y}}_{i_s}^*,$$

$$\tilde{\mathbf{X}}_{n_{odd}}^{tra} = \tilde{\mathbf{X}} \in R^{N_{up} \times q}, \quad \tilde{\mathbf{Y}}_{n_{odd}}^{tra} = \tilde{\mathbf{Y}} \in R^{N_{up} \times D_2},$$

$$i_{bs} = i, \quad n_{odd} = 1, 3, \dots, 2N_{up} - 1, \quad n_{even} = 2, 4, \dots, 2N_{up}, \quad (25)$$

where we let $\tilde{\mathbf{X}}^{tra} = [\tilde{\mathbf{x}}_1^{tra}, \dots, \tilde{\mathbf{x}}_{n_{tra}}^{tra}, \dots, \tilde{\mathbf{x}}_{N_{up}}^{tra}]^T \in R^{N_{up} \times q}$ and $\tilde{\mathbf{Y}}^{tra} = [\tilde{\mathbf{y}}_1^{tra}, \dots, \tilde{\mathbf{y}}_{n_{tra}}^{tra}, \dots, \tilde{\mathbf{y}}_{N_{up}}^{tra}]^T \in R^{N_{up} \times D_2}$, $bs \tilde{\mathbf{X}}^*$, $bs \tilde{\mathbf{Y}}^*$ are the samples of incremental learning, the $\tilde{\mathbf{X}}^{tra}$ can be seen in the LD space as depicted in Figure 4, and the green ones denote $bs \tilde{\mathbf{X}}^*$; the new mapping can be built through the training model as below:

$$\begin{aligned} (\tilde{\mathbf{Y}}^{tra})^T &= [\tilde{\mathbf{y}}_1^{tra}, \tilde{\mathbf{y}}_2^{tra}, \dots, \tilde{\mathbf{y}}_{N_{up}}^{tra}] \\ &= [\text{up} \tilde{\mathbf{y}}_1, bs \tilde{\mathbf{y}}_1^*, \text{up} \tilde{\mathbf{y}}_2, bs \tilde{\mathbf{y}}_2^*, \dots, \text{up} \tilde{\mathbf{y}}_{N_{up}}, bs \tilde{\mathbf{y}}_{N_{up}}^*] \\ &= \mathbf{W}_D^* \Phi^* + \mathbf{E}^* \\ &= \mathbf{W}_D^* [\Phi^*(\tilde{\mathbf{x}}_1^{tra}), \Phi^*(\tilde{\mathbf{x}}_2^{tra}), \dots, \Phi^*(\tilde{\mathbf{x}}_{N_{up}}^{tra})] + \mathbf{E}^* \\ &= \mathbf{W}_D^* [\Phi^*(\tilde{\mathbf{x}}_1), \Phi^*(bs \tilde{\mathbf{x}}_1^*), \Phi^*(\tilde{\mathbf{x}}_2), \\ &\quad \Phi^*(bs \tilde{\mathbf{x}}_2^*), \dots, \Phi^*(\tilde{\mathbf{x}}_{N_{up}}), \Phi^*(bs \tilde{\mathbf{x}}_{N_{up}}^*)] \\ &\quad + \mathbf{E}^*, \end{aligned} \quad (26)$$

where $\mathbf{W}_D^* \in R^{D_2 \times Nk}$, $\Phi^* \in R^{Nk \times N_{up}^*}$, $\Phi_{k_1, k_2}^* = \phi^*(\tilde{\mathbf{x}}_{k_2}^{tra}, \mathbf{c}_{k_1}^*) = \exp(b \|\tilde{\mathbf{x}}_{k_2}^{tra} - \mathbf{c}_{k_1}^*\|^2)$, $\Phi(\mathbf{x}_E) = [\phi(\mathbf{x}_E, \mathbf{c}_1^*), \phi(\mathbf{x}_E, \mathbf{c}_2^*), \dots, \phi(\mathbf{x}_E, \mathbf{c}_{Nk}^*)]^T$, $\mathbf{E}^* \in R^{D_2 \times N_{up}^*}$, $N_{up}^* = 2N_{up}$, $\widehat{\mathbf{W}}_D^*$ and $\widehat{\mathbf{E}}^* = \widehat{\mathbf{B}}_p^* \text{ones}(1, N_{up}^*)$ denote the trained values of $\mathbf{W}_D^*$ and $\mathbf{E}^*$, respectively, and $\widehat{\mathbf{B}}_p^* \in R^{D_2 \times 1}$. After finishing the training of (26), the incremental learning is completed. Then, the mapping is updated, so that the new HD data can be generated through the updated mapping:

$$\text{up} \mathbf{y}_t^* = \mathbf{f}^{\text{new}}(\mathbf{x}_t^*) = \widehat{\mathbf{W}}_D^* \Phi^*(\mathbf{x}_t^*) + \widehat{\mathbf{E}}^*, \quad (27)$$

FIGURE 4: The low dimensional data through increment learning.

where, similarly, as to the one $\mathbf{x}_t^*$, let $N_{up}^* = 1$, $\Phi(\mathbf{x}_t^*) = [\phi(\mathbf{x}_t^*, \mathbf{c}_1^*), \phi(\mathbf{x}_t^*, \mathbf{c}_2^*), \dots, \phi(\mathbf{x}_t^*, \mathbf{c}_{Nk}^*)]^T$, and $\widehat{\mathbf{E}}^* = \widehat{\mathbf{B}}_p^* \text{ones}(1, 1)$. When we have multiple $\mathbf{x}_t^*$, we can also get the corresponding multiple $\text{up} \mathbf{y}_t^*$, and it only needs to add the column in Φ^* and $\widehat{\mathbf{E}}^*$.

Training equation (26) can use Algorithm 1; we let input parameters $\mathbf{D} = \tilde{\mathbf{Y}}^{tra}$, $\mathbf{V} = \tilde{\mathbf{X}}^{tra}$, $Q_1 = D_2$, and $Q_2 = q$ and functions $f_e(\mathbf{v}_k, \mathbf{z}_{k'}) = \phi^*(\tilde{\mathbf{x}}_{k_2}^{tra}, \mathbf{c}_{k_1}^*)$, $\mathbf{v}_k = \text{up} \tilde{\mathbf{x}}_{k_2}$, $k = k_2 = 1, 2, \dots, N_{up}$, and $N = N_{up}$ and then get corresponding output parameters $\mathbf{c}_{k_1}^* = \mathbf{z}_{k'}$, $Nk = M$, $k_1 = k' = 1, 2, \dots, Nk$, $\widehat{\mathbf{W}}_D^* = \widehat{\boldsymbol{\theta}}^T$, and $\widehat{\mathbf{E}}^* = [\text{ones}(1, 1) \mathbf{B}_p]^T$. Let us summarize the incremental learning algorithm as in Algorithm 2.

The mapping of (27) is incremental learn mapping, which replaces (10) of IPDRM; it can promote the second advantage of IPDRM mentioned above, due to the mapping from LD data to HD data being updated to get more accurate. Worthy of attention is that at least one $\tilde{\mathbf{x}}_{n_s}^*$ must distribute between $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_{i+1}$, so that effectiveness of incremental learning can be better, thus using LDSS to generate HD data samples should follow this.

7. Human Motion Estimation via Searching in Low Dimensional Space

7.1. The Method of Searching. We estimate the human motion model via searching in low dimensional space. The method of searching needs to combine linear searching and nonlinear searching in order to generate the better corresponding HD data sample, which is depicted as Figure 5. In Figure 5, the hollow dots denote the new LD data acquired by linear searching and nonlinear searching in the LD space; the solid dots denote the known LD data acquired by dimension reduction of known HD data samples. Let us discuss more details as follow.

Input: $\tilde{\mathbf{X}}_{\text{new}}^*$, $\tilde{\mathbf{Y}}_{\text{new}}^*$, b , c_s ;
Output: $\mathbf{c}_{k_1}^*$, $k_1 = 1, 2, \dots, Nk$, $\widehat{\mathbf{W}}_D^*$, $\widehat{\mathbf{E}}^*$, $\tilde{\mathbf{X}}^{\text{tra}}$;
For $i = 1 : 1 : N_{\text{up}}$
If $i = N_{\text{up}}$
 $f_L = i$; $b_L = 1$;
Else
 $f_L = i$; $b_L = i + 1$;
End If
 $k_D = 1$;
For $n^* = 1 : 1 : N^*$
 $D_{n^*}^* = \min D_{\text{se}}, D_{\text{se}} \in \{\|\tilde{\mathbf{x}}_{n^*}^* - \tilde{\mathbf{x}}_{f_L}\|, \|\tilde{\mathbf{x}}_{n^*}^* - \tilde{\mathbf{x}}_{b_L}\|\}$;
If $D_{n^*}^* < c_s \|\mathbf{x}_{f_L} - \mathbf{x}_{b_L}\|$
 $IN(k_D) = n^*$;
End If
 $k_D = k_D + 1$;
End For
 $\mathbf{X}_{\text{se}}^* = (\mathbf{X}_{\text{new}}^*)_{IN(1:k_D),:}$; $\mathbf{Y}_{\text{se}}^* = (\mathbf{Y}_{\text{new}}^*)_{IN(1:k_D),:}$;
For $i_s = 1 : 1 : N_s$
 $D_{\text{gap}}(i_s) = \|\|\tilde{\mathbf{x}}_{i_s}^* - \tilde{\mathbf{x}}_{f_L}\| - \|\tilde{\mathbf{x}}_{i_s}^* - \tilde{\mathbf{x}}_{b_L}\|\|$;
End For
 $i_s^* = \arg \min_{i_s} D_{\text{gap}}(i_s)$; $i_{\text{bs}} = i_s$; $\text{bs}\tilde{\mathbf{x}}_{i_{\text{bs}}}^* = \tilde{\mathbf{x}}_{i_s^*}^*$; $\text{bs}\tilde{\mathbf{y}}_{i_{\text{bs}}}^* = \tilde{\mathbf{y}}_{i_s^*}^*$;
End For
 $\tilde{\mathbf{X}}_{\text{even}}^{\text{tra}} = \text{bs}\tilde{\mathbf{X}}$; $\tilde{\mathbf{Y}}_{\text{even}}^{\text{tra}} = \text{bs}\tilde{\mathbf{Y}}^*$; $\tilde{\mathbf{X}}_{\text{odd}}^{\text{tra}} = \tilde{\mathbf{X}}$; $\tilde{\mathbf{Y}}_{\text{odd}}^{\text{tra}} = \tilde{\mathbf{Y}}$;
 $\mathbf{D} = \tilde{\mathbf{Y}}^{\text{tra}}$; $\mathbf{V} = \tilde{\mathbf{X}}^{\text{tra}}$; $f_e(\mathbf{v}_k, \mathbf{z}_{k'}) = \phi^*(\tilde{\mathbf{x}}_{k_2}^{\text{tra}}, \mathbf{c}_{k_1}^*)$;
 $Q_1 = D_2$; $Q_2 = q$; $\mathbf{v}_k = \tilde{\mathbf{x}}_{k_2}^{\text{tra}}$; $\mathbf{z}_{k'} = \mathbf{c}_{k_1}^*$; $k = k_2$; $k' = k_1$;
Use Algorithm 1 to get $\mathbf{z}_{k'}, k' = 1, 2, \dots, M, \hat{\boldsymbol{\theta}}$ and $\mathbf{B}_p$;
 $\mathbf{c}_{k_1}^* = \mathbf{z}_{k'}$, $k_1 = k' = 1, 2, \dots, Nk$, $Nk = M$; $\widehat{\mathbf{W}}_D^* = \hat{\boldsymbol{\theta}}^T$;
 $\widehat{\mathbf{E}}^* = [\text{ones}(1, 1) \mathbf{B}_p]^T$;

ALGORITHM 2

FIGURE 5: The method of searching.

We take $\tilde{\mathbf{X}}' = [\tilde{\mathbf{x}}'_1, \dots, \tilde{\mathbf{x}}'_i, \dots, \tilde{\mathbf{x}}'_{N'_{\text{up}}}]^T \in R^{N'_{\text{up}} \times q}$ ($\tilde{\mathbf{X}}' = \tilde{\mathbf{X}}$ or $\tilde{\mathbf{X}}^{\text{tra}}$), $\tilde{\mathbf{Y}}' = [\tilde{\mathbf{y}}'_1, \dots, \tilde{\mathbf{y}}'_i, \dots, \tilde{\mathbf{y}}'_{N'_{\text{up}}}]^T \in R^{N'_{\text{up}} \times D_2}$ ($\tilde{\mathbf{Y}}' = \tilde{\mathbf{Y}}$ or $\tilde{\mathbf{Y}}^{\text{tra}}$), and the mapping $\mathbf{f}_s$ ($\mathbf{f}_s = \mathbf{f}$ or $\mathbf{f}^{\text{new}}$), for example, to discuss the method. As to the new HD data sample ${}^{\text{up}}\mathbf{y}_t^*$, we use mapping $\mathbf{g}$ to obtain the $\mathbf{x}_t^*$ ($\mathbf{x}_t^* = \mathbf{g}({}^{\text{up}}\mathbf{y}_t^*)$). Then, the clustering of

the $\mathbf{x}_t^*$ can be executed through the comparison of distance to find the nearest known LD data:

$$i^* = \arg \min_i \|\mathbf{x}_t^* - \tilde{\mathbf{x}}'_i\|, \quad i = 1, 2, \dots, N'_{\text{up}}. \quad (28)$$

After finding the $\tilde{\mathbf{x}}'_{i^*}$, N_{ls} denotes the amount of new searched LD data; we need linear searching to get the new LD data between the two known neighbours LD data near $\tilde{\mathbf{x}}'_{i^*}$, like this

$$\begin{aligned} \mathbf{s}_l^1 &= \tilde{\mathbf{x}}'_{B_2} + \frac{l(\tilde{\mathbf{x}}'_{B_1} - \tilde{\mathbf{x}}'_{B_2})}{(N_{ls} + 1)}, \\ \mathbf{s}_l^2 &= \tilde{\mathbf{x}}'_{B_1} + \frac{l(\tilde{\mathbf{x}}'_{i^*} - \tilde{\mathbf{x}}'_{B_1})}{(N_{ls} + 1)}, \\ \mathbf{s}_l^3 &= \tilde{\mathbf{x}}'_{i^*} + \frac{l(\tilde{\mathbf{x}}'_{F_1} - \tilde{\mathbf{x}}'_{i^*})}{(N_{ls} + 1)}, \\ \mathbf{s}_l^4 &= \tilde{\mathbf{x}}'_{F_1} + \frac{l(\tilde{\mathbf{x}}'_{F_2} - \tilde{\mathbf{x}}'_{F_1})}{(N_{ls} + 1)}, \end{aligned} \quad (29)$$

$$l = 1, 2, \dots, N_{ls};$$

here, $F_{ns} = i^* + ns$, $B_{ns} = i^* - ns$, and $ns \in N^+$; let $ns = 1, 2$; then we have $F_{ns} = F_{ns} - N'_{\text{up}}$, $F_{ns} > N'_{\text{up}}$, and $B_{ns} = B_{ns} + N'_{\text{up}}$, $B_{ns} < 1$. Besides, we also need nonlinear searching to get them similarly, and N_{lns} denotes the number of new ones

$$\begin{aligned} \tilde{\mathbf{s}}_{l'}^1 &= \tilde{\mathbf{x}}'_{B_1} + \tilde{\mathbf{u}}_{l'}^1, \\ \tilde{\mathbf{s}}_{l'}^2 &= \tilde{\mathbf{x}}'_{i^*} + \tilde{\mathbf{u}}_{l'}^2, \\ \tilde{\mathbf{s}}_{l'}^3 &= \tilde{\mathbf{x}}'_{F_1} + \tilde{\mathbf{u}}_{l'}^3, \end{aligned} \quad (30)$$

$$l' = 1, 2, \dots, N_{lns},$$

where $\tilde{\mathbf{u}}_{l'}^{sv} = \tilde{\eta}_{l'} \tilde{L}(\tilde{\mathbf{u}}_{l'}^{sv} / \|\tilde{\mathbf{u}}_{l'}^{sv}\|)$, $sv = 1, 2, 3, \dots, \tilde{L} = \arg \max \tilde{D}$, $\tilde{D} \in \{\|\tilde{\mathbf{x}}'_{i^*} - \tilde{\mathbf{x}}'_{B_1}\|, \|\tilde{\mathbf{x}}'_{B_1} - \tilde{\mathbf{x}}'_{B_2}\|, \|\tilde{\mathbf{x}}'_{i^*} - \tilde{\mathbf{x}}'_{F_1}\|, \|\tilde{\mathbf{x}}'_{F_1} - \tilde{\mathbf{x}}'_{F_2}\|\}$, $\tilde{\eta}_{l'} = \text{rand}(1, 1)$, and $\tilde{\mathbf{u}}_{l'}^{sv} = \text{rand}(q, 1)$. From (29) and (30), the set containing searched LD data can be

$$\begin{aligned} S_L &= \{\tilde{\mathbf{x}}'_{B_2}, \mathbf{s}_1^1, \dots, \mathbf{s}_{N_{ls}}^1, \tilde{\mathbf{x}}'_{B_1}, \mathbf{s}_1^2, \dots, \mathbf{s}_{N_{ls}}^2, \tilde{\mathbf{x}}'_{i^*}, \mathbf{s}_1^3, \dots, \mathbf{s}_{N_{ls}}^3, \\ &\quad \tilde{\mathbf{x}}'_{F_1}, \mathbf{s}_1^4, \dots, \mathbf{s}_{N_{ls}}^4, \tilde{\mathbf{x}}'_{F_2}, \mathbf{s}_1^1, \dots, \mathbf{s}_{N_{lns}}^1, \\ &\quad \tilde{\mathbf{s}}_1^2, \dots, \tilde{\mathbf{s}}_{N_{lns}}^2, \tilde{\mathbf{s}}_1^3, \dots, \tilde{\mathbf{s}}_{N_{lns}}^3\}; \end{aligned} \quad (31)$$

use the mapping $\mathbf{f}_s$ to get another set as below:

$$S_{\text{LH}} = \{{}^{\text{up}}\mathbf{y}_t^*, \mathbf{y}_a' = \mathbf{f}_s(\mathbf{x}_a) \mid \mathbf{x}_a \in S_L\}; \quad (32)$$

according to (2), we have

$$\hat{\mathbf{y}}_a^* = \arg \max_{\hat{\mathbf{y}}_a} p(m_t \mid \hat{\mathbf{y}}_p^*, \hat{\mathbf{y}}_a), \quad \hat{\mathbf{y}}_a \in S_{\text{LH}}, \quad (33)$$

where the optimal prior information of the pose $\bar{\mathbf{y}}_a^* \in R^{D_2}$ can be obtained, and then the optimal prior HD data can be $\mathbf{y}_{\text{pri}} = [\bar{\mathbf{y}}_p^{*T}, \bar{\mathbf{y}}_a^{*T}]^T \in R^{D_1}$, and $\bar{\mathbf{y}}_p^*$ is the optimal spatial position information which can be obtained by SEMAS.

On the basis of work above, we can estimate the human motion model from multiview silhouettes through Bayesian theory. When $\mathbf{y}_{\text{pri}}$ is known, $p(\mathbf{y}_t^i | m_t)$ is unknown and N'_e is large enough; we let $p(\mathbf{y}_t^i) = q(\mathbf{y}_t^i | \bar{\mathbf{y}} = \mathbf{y}_{\text{pri}})$, and the estimation can be achieved by

$$\begin{aligned} w_b^i &= \frac{p(\mathbf{y}_t^i, m_t)}{q(\mathbf{y}_t^i | \bar{\mathbf{y}} = \mathbf{y}_{\text{pri}})} \\ &= \frac{(p(m_t | \mathbf{y}_t^i) p(\mathbf{y}_t^i))}{q(\mathbf{y}_t^i | \bar{\mathbf{y}} = \mathbf{y}_{\text{pri}})} = p(m_t | \mathbf{y}_t^i) = w_t^i, \end{aligned} \quad (34)$$

$$\begin{aligned} p\mathbf{y}_t^* &= \mathbf{E}(\mathbf{y}_t^i | m_t) = \int \mathbf{y}_t^i p(\mathbf{y}_t^i | m_t) d\mathbf{y}_t^i \\ &\approx \sum_{i=1}^{N'_e} p\mathbf{y}_t^i p(\mathbf{y}_t^i | m_t) = \frac{(\sum_{i=1}^{N'_e} p\mathbf{y}_t^i p(\mathbf{y}_t^i, m_t))}{p(m_t)} \\ &= \frac{(\sum_{i=1}^{N'_e} p\mathbf{y}_t^i p(\mathbf{y}_t^i, m_t))}{(\sum_{i=1}^{N'_e} p(\mathbf{y}_t^i, m_t))} \\ &= \frac{(\sum_{i=1}^{N'_e} p\mathbf{y}_t^i w_t^i q(\mathbf{y}_t^i | \bar{\mathbf{y}} = \mathbf{y}_{\text{pri}}))}{(\sum_{i=1}^{N'_e} w_t^i q(\mathbf{y}_t^i | \bar{\mathbf{y}} = \mathbf{y}_{\text{pri}}))} \\ &\approx \frac{((1/N'_e) \sum_{i=1}^{N'_e} p\mathbf{y}_t^i w_t^i)}{((1/N'_e) \sum_{i=1}^{N'_e} w_t^i)} = \sum_{i=1}^{N'_e} \bar{w}_t^i p\mathbf{y}_t^i, \end{aligned} \quad (35)$$

and, after getting $p\mathbf{y}_t^*$, the human motion model can be drawn in the space. Worthy of attention is that $q(\mathbf{y}_t^i | \mathbf{y}_{\text{pri}})$ is multivariate normal distribution density function, which takes $\bar{\mathbf{y}} = \mathbf{y}_{\text{pri}}$ as the mean, N'_e is the number of generating $p\mathbf{y}_t^i$, $\mathbf{y}_{\text{pri}}$ is contained as one of N'_e samples, which can let $p\mathbf{y}_t^{N'_e} = \mathbf{y}_{\text{pri}}$, and $\bar{w}_t^i = w_t^i / \sum_{i=1}^{N'_e} w_t^i$ is normalized weight. The method of searching before or after incremental learning is also like this; it only needs to replace some variables to be carried out.

The poses of human motion are continuous, which are shown by the human motion models. The pose estimated last time is close to the one of this time, thus the new HD sample ${}^{\text{up}}\mathbf{y}_t^*$ can be seen as the pose estimated last time, and the corresponding LD data $\mathbf{x}_t^*$ can be acquired by IPDRM. Then, we find the other LD data close to $\mathbf{x}_t^*$ through clustering to confirm the searching position in the LD space, so that the HD data samples closing to the true data this time can be generated efficiently through searching in the nearby area of this position. For these advantages, the proposal algorithm (LDSIL) can disambiguate silhouettes, overcome the transient occlusion, and reduce estimation error. Furthermore,

the performance can be promoted through its incremental learning which is used to update the mapping from the LD space to the HD space.

As to the estimation of Bayesian theory above, we use SEMAS and the proposed method of searching to find the optimal prior HD data $\mathbf{y}_{\text{pri}}$, which can make $p(\mathbf{y}_t^i, m_t) / q(\mathbf{y}_t^i | \bar{\mathbf{y}}) = p(m_t | \mathbf{y}_t^i)$ large enough. When $p\mathbf{y}_t^i = \mathbf{y}_{\text{pri}}$, according to (34), somehow, $p(\mathbf{y}_t^i, m_t)$ also become large enough. Then, it means that $\mathbf{y}_{\text{pri}}$ can be seen as the sample generating from the unimodal distribution density $p(\mathbf{y}_t^i | m_t)$, due to $p(\mathbf{y}_t^i, m_t) \propto p(\mathbf{y}_t^i | m_t) = p(\mathbf{y}_t^i, m_t) / p(m_t)$, and $p(m_t)$ is known. Moreover, the true HD data can be seen as the mean relative to the distribution density $p(\mathbf{y}_t^i | m_t)$, and thus its value of distribution density will be larger than others. On the whole, it also means that $\mathbf{y}_{\text{pri}}$ is the sample which is close to the true HD data; thus taking $\mathbf{y}_{\text{pri}}$ as the mean ($\bar{\mathbf{y}} = \mathbf{y}_{\text{pri}}$) with the small scale variance to search the HD data in the HD space for only once can generate the HD data samples which will also have the large value of distribution density $p(\mathbf{y}_t^i | m_t)$; then the mean of these samples can be close to mean relative to distribution density $p(\mathbf{y}_t^i | m_t)$; in other words, it will be close to the true HD data. The work above can avoid the small scale searching (sampling) the HD data for many times or the large scale searching (sampling) the HD data directly and generating the invalid HD data to estimate the human motion in 3D.

Then, we will discuss more details about the whole procedure of human motion estimation in the following section.

7.2. The Procedure of Algorithm for Human Motion Estimation.

The work from Sections 2–7 can be summarized as the complete algorithm procedure of LDSIL to estimate human motion in 3D. The framework of LDSIL can be seen as shown in Figure 6. Let us give more detailed description for the framework.

From Figure 6, we can see the framework of LDSIL. Firstly, LDSIL is using the LDSS to estimate the human motion in 3D based on the small amount of HD data samples and get the estimated HD data covering about one motion cycle. The framework of LDSS consists of SEMAS, linear and nonlinear searching in LD space, IPDRM, and Bayesian theory. The estimated HD data with the spatial position information removed can be selected as the new HD data samples and used for updating the mapping from LD space to HD space through IPDRM. Secondly, the IPDRM is used to obtain the corresponding LD data of the new HD data samples through its mapping $\mathbf{g}$ of incremental dimension reduction. Selecting the new HD data samples can be achieved through comparing the distance among the corresponding LD data. After the comparison is finished, the LD data and the corresponding HD data can be obtained and used to update the mapping from LD space to HD space for the incremental learning of LD space. Thirdly, the new mapping $\mathbf{f}^{\text{new}}$ updated by the selected HD data and the corresponding LD data can generate the HD data

FIGURE 6: Framework of LDSIL.

more accurately; thus the estimation can achieve a better performance with the help of new mapping $\mathbf{f}^{\text{new}}$. Finally, the estimation of human motion in 3D from the multiview silhouettes is carried out through the framework including SEMAS, linear and nonlinear searching in LD space, IPDRM with new mapping $\mathbf{f}^{\text{new}}$, and Bayesian theory. However, when LDSIL is used to estimate the human motion, these data are known as follows.

We have the sequence of the small amount of HD data samples $\tilde{\mathbf{Y}}$ (no spatial position information) covering one motion cycle and the HD data of initial frame (1st frame) containing the spatial position information (6 dimensions in total) ${}^p\mathbf{Y}_0 = {}^p\mathbf{Y}_1$. Ft can denote the number of estimated-frames without incremental learning, which are used for the sample selection of incremental learning, and Ft^* can denote the number of estimated frames with incremental learning. Then, we begin the procedure as follows.

LDSIL for Human Motion Estimation

- (1) Use (9) to reduce the dimension of $\tilde{\mathbf{Y}}$, obtain $\tilde{\mathbf{X}}$, build the mapping $\mathbf{f}$, and let ${}^p\mathbf{Y}_0^* = {}^p\mathbf{Y}_1$, $t = 1$.
- (2) Use Algorithm 1 to training equation (11), build the mapping $\mathbf{g}$, get IPDRM, and let mapping $\mathbf{f}_s = \mathbf{f}$.
- (3) When estimating the t th frame, use SEMAS and set relevant parameters to find the spatial position information. Let ${}_0\mathbf{y}_p^{k_1} = {}^p\mathbf{y}_{t-1}^*(1 : 2)$, select $\tilde{\mathbf{u}}_t^i = \eta_i L(\mathbf{u}_t^i / \|\mathbf{u}_t^i\|)$, and get the optimal planimetric position information ${}_1\mathbf{y}_p^* = \mathbf{y}_p^* \in R^2$. Then, let ${}_0\mathbf{y}_p^{k_1} = {}^p\mathbf{y}_{t-1}^*(3 : 6)$, select $\tilde{\mathbf{u}}_t^i = L\mathbf{u}_t^i$, and get the optimal height and rotation angle information ${}_2\mathbf{y}_p^* = \mathbf{y}_p^* \in R^4$. At last, obtain $\tilde{\mathbf{y}}_p^* = [{}_1\mathbf{y}_p^{*T}, {}_2\mathbf{y}_p^{*T}]^T$.

- (4) Let ${}^{\text{up}}\mathbf{y}_t^* = {}^p\mathbf{y}_{t-1}^*(7 : D_1)$, obtain the LD data of the t th frame $\mathbf{x}_t^* = \mathbf{g}({}^{\text{up}}\mathbf{y}_t^*)$, and calculate $i^* = \arg \min_i \|\mathbf{x}_t^* - \tilde{\mathbf{x}}_i\|$, $i = 1, 2, \dots, N_{\text{up}}$. Search the data in the LD space, get the set $S_{\text{LH}} = \{\mathbf{y}_a^* | \mathbf{y}_a^* = \mathbf{f}_s(\mathbf{x}_a) \mid \mathbf{x}_a \in S_L\}$, and calculate $\tilde{\mathbf{y}}_a^* = \arg \max_{\tilde{\mathbf{y}}_a^*} p(m_t \mid \tilde{\mathbf{y}}_a^*, \tilde{\mathbf{y}}_a^*)$, $\tilde{\mathbf{y}}_a^* \in S_{\text{LH}}$, and then get $\mathbf{y}_{\text{pri}} = [\tilde{\mathbf{y}}_p^{*T}, \tilde{\mathbf{y}}_a^{*T}]^T$.
- (5) Generate ${}^p\mathbf{y}_t^i$, $i = 1, 2, \dots, N_e'$, through $q({}^p\mathbf{y}_t^i \mid \tilde{\mathbf{y}} = \mathbf{y}_{\text{pri}})$, let ${}^p\mathbf{y}_t^{N_e'} = \mathbf{y}_{\text{pri}}$, calculate $\tilde{w}_t^i = w_t^i / \sum_{i=1}^{N_e'} w_t^i$, $i = 1, 2, \dots, N_e'$, and get ${}^p\mathbf{y}_t^* = \sum_{i=1}^{N_e'} \tilde{w}_t^i {}^p\mathbf{y}_t^i$. Then, human motion model can be outputted through ${}^p\mathbf{y}_t^*$, and let ${}^{\text{up}}\tilde{\mathbf{y}}_n^* = {}^p\mathbf{y}_t^*(7 : D_1)$, $n^* = t$. If $t \leq Ft$, $t = t + 1$, return to step (3); otherwise, get the estimated HD data $\tilde{\mathbf{Y}}_{\text{new}}^*$ (new HD data samples) and go to next step.
- (6) If needing incremental learning, go to next step; otherwise end this algorithm.
- (7) After getting $\tilde{\mathbf{Y}}_{\text{new}}^*$, use mapping $\mathbf{g}$ to obtain $(\tilde{\mathbf{X}}_{\text{new}}^*)^T = \mathbf{g}((\tilde{\mathbf{Y}}_{\text{new}}^*)^T) = [\mathbf{g}({}^{\text{up}}\tilde{\mathbf{y}}_1^*), \dots, \mathbf{g}({}^{\text{up}}\tilde{\mathbf{y}}_{n^*}^*), \dots, \mathbf{g}({}^{\text{up}}\tilde{\mathbf{y}}_{N^*}^*)]$. Then, get the training samples $\tilde{\mathbf{Y}}^{\text{tra}}$ and $\tilde{\mathbf{X}}^{\text{tra}}$ selected from $\tilde{\mathbf{Y}}_{\text{new}}^*$ and $\tilde{\mathbf{X}}_{\text{new}}^*$ by the method proposed above, and use Algorithm 2 to carry out incremental learning to train (26). After finishing the training, get updated mapping $\mathbf{f}^{\text{new}}$. Let $\tilde{\mathbf{X}}' = \tilde{\mathbf{X}}^{\text{tra}}$, $Ft = Ft^*$, ${}^p\mathbf{Y}_0^* = {}^p\mathbf{Y}_1$, and mapping $\mathbf{f}_s = \mathbf{f}^{\text{new}}$, and return to step (3).

8. Experiments and Evaluations

We tested the performance of our proposed algorithm (LDSIL) in three views ($Nv = 3$) by comparison with other

three traditional algorithms, which included annealed particle filter [4] (APF), Gaussian particle filter [21] (GPF), and particle filter [22] (PF). The tested performance included disambiguating silhouettes, overcoming the transient occlusion, and estimation error. Besides, we compared LDSIL with LDSS to test the validity of incremental learning independently, and the LDSS was similar to LDSIL somehow, so it had the performance of LDSIL, but it was without the incremental learning, and the mapping from LD space to HD space was not updated, which still used (10) to generate HD data. The performance of LDSIL and LDSS could be seen by the comparison of the mean error and maximum error.

Firstly, we were testing disambiguating silhouettes and taking walking and running motion as the test cases. Then, we saw in the subfigures of Figure 7, after estimating several frames, APF, GPF, and PF could not disambiguate silhouettes to estimate the human motion in 3D; on the contrary, LDSIL could disambiguate silhouettes to estimate the one that was close to the true data. The ambiguity meant a set of silhouettes such that the pictures of Figure 1(c) could not distinguish positions of the limbs from the human body; for example, the silhouettes could not depict which one of the legs was in front or in back. Thus, we could see in the subfigures of Figures 7(a) and 7(b) that the estimated human motion projections in all views from APF, GPF, and PF showed that the positions of left leg (white line) and right leg (gray line) were, respectively, opposite by the comparison of true data. The corresponding human motion models were reflecting the same phenomenon, and the left leg (gray one) and right leg (black one) of the model were also opposite.

Secondly, we tested overcoming transient occlusion. The transient occlusion was equal to estimate the adjacent and discontinuous frame, due to transient occlusion of cameras from all the views, which would lead to the result that the silhouettes from all the views could not be obtained. The two frames (initial data and true data) between which the interval of 10 frames existed were used to test all the algorithms. We could find in the subfigures of Figures 8(a) and 8(b) that APF, GPF, and PF could not estimate the human motion models accurately, which showed that the limbs from the human body mismatched the images of all the views. However, LDSIL could estimate the models more accurate than APF, GPF, and PF, which were depicted in the corresponding subfigures, and the limbs from the human body matched the images of all the views better like the true data.

Thirdly, we used 50 continuous frames to test the estimation error of each algorithm. We used walking and running motion sequences (walking 1–3, running 1–2) to test, whose spatial positions had obvious change. The subfigures of Figure 9 showed the experiment results, the mean error and standard deviation from LDSIL were the smallest among these algorithms, and the maximum error shown in the identifications of the corresponding subfigures was also the smallest; besides, the errors of most frames from LDSIL were smallest. The error of each frame could be computed as below [23]:

$$Er = \frac{1}{Mar} \sum_{ma=1}^{Mar} \|\mathbf{Bx}_{ma} - \mathbf{Bx}'_{ma}\|, \quad \mathbf{Bx}_{ma}, \mathbf{Bx}'_{ma} \in R^3, \quad (36)$$

where $\mathbf{Bx}_{ma}$ and $\mathbf{Bx}'_{ma}$ were, respectively, true position and estimated position of the joint marker in the model and Mar was the number of the markers. From Figures 9(b)–9(f), we found that the errors of all the tested algorithms were close in the estimation of the first 20 frames, but the error from LDSIL could be kept smaller than APF, GPF, and PF after the 20th frame. However, it was reasonable that the errors from APF, GPF, and PF might be close to LDSIL in some frame estimation, because initial data was close to the true data and the probability was stochastic, which were still larger than the LDSIL on the whole.

Finally, we tested the validity of incremental learning of LDSIL. Compared with LDSIL, LDSS had no incremental learning. We selected 3 motion sequences to test the estimation error for 6 times arbitrarily, which included known and unknown spatial position information. We were setting the same parameters in the experiments, and the performance of LDSIL was better than LDSS as shown in the Table 1, according to the comparison of the mean errors and maximum error. The experiment results indicated that the incremental learning played the key role, because LDSIL had the smaller mean error and maximum error in the tests.

9. Conclusions

We could see the experiments above, which indicated that the proposed algorithm (LDSIL) could contribute a lot to estimate the human motion containing the spatial position information in 3D from the multiview silhouettes. The results showed that LDSIL had the best performance including disambiguating silhouettes, overcoming the transient occlusion, and reducing estimation error by the comparison of the other three algorithms (APF, GPF, and PF). Meanwhile, the feasibility and performance of incremental learning in the LDSIL were also validated by the experiments. In addition, the segmentation of multiview images must achieve the high quality; otherwise the results in the experiments would be affected. Our work in this paper had some limitation as follows. Firstly, the initialized frame of the human motion model (the 1st frame) was known, which contained the spatial position information; secondly, the spatial position information and pose of the estimated human motion in 3D were changing regularly, and the human motion model had markers. In the future, these limitations would be improved. Our work would solve the problem that initialized frame needed to be known and focus on complex human motion estimation. Moreover, the human motion model would have no marker [24, 25].

Appendices

A. Derivation of (4)

Due to $\tilde{\mathbf{u}}_t^i = [\tilde{\mathbf{u}}_t^i(1), \dots, \tilde{\mathbf{u}}_t^i(d'), \dots, \tilde{\mathbf{u}}_t^i(d)]^T$, $i = 1, \dots, N_e - 1$, $\tilde{\mathbf{u}}_t^i(d') \sim U(-\eta_i L, \eta_i L)$; thus, in the space of d dimensions, each direction has the same chance to generate $\tilde{\mathbf{u}}_t^i$. When objective function $w^i = f_p(\mathbf{y}_p^i) = p(m_i | \mathbf{y}_p^i, \mathbf{y}_a)$ is unimodal

LDSIL

APF

GPF

PF

True data

(a) Walking

FIGURE 7: Continued.

FIGURE 7: Several algorithms were disambiguating silhouettes to estimate the human motion in 3D.

LDSIL

APF

GPF

PF

Initial data

True data

(a) Walking

FIGURE 8: Continued.

FIGURE 8: Several algorithms were overcoming transient occlusion to estimate the human motion in 3D.

(a) Mean error and standard deviation

(b) The error and maximum error in walking 1

(c) The error and maximum error in running 1

(d) The error and maximum error in walking 2

(e) The error and maximum error in running 2

(f) The error and maximum error in walking 3

FIGURE 9: The estimation errors of several algorithms.

TABLE 1: The test of incremental learning.

Algorithm	Sequence (total: 30 frames)	Mean error (mm)	Maximum error (mm)	Spatial position information
LDSS	1	54.87	77.24	Unknown
	2	60.89	109.90	Unknown
	3	34.23	44.02	Unknown
	1	54.01	146.93	Known
	2	28.08	39.63	Known
	3	28.23	37.79	Known
LDSIL	1	49.11	73.47	Unknown
	2	43.03	74.06	Unknown
	3	31.52	39.60	Unknown
	1	25.54	43.42	Known
	2	24.36	35.74	Known
	3	21.24	29.84	Known

approximately, $\widehat{k}_1$ is the ordinal number of maximum element in $\mathbf{W}_o = \{w_t^1, w_t^2, \dots, w_t^{N_e}\}$, $w_t^{\widehat{k}_1} > w_t^i$, $i = 1, 2, \dots, N_e$, $i \neq \widehat{k}_1$, $\mathbf{y}_p^* = \arg \max_{\mathbf{y}_p^i} f_p(\mathbf{y}_p^i)$, in the r th time searching, we have ${}_r\mathbf{y}_p^i = {}_{r-1}\mathbf{y}_p^{\widehat{k}_1} + \tilde{\mathbf{u}}_t^i$, $i = 1, \dots, N_e - 1$, ${}_r\mathbf{y}_p^{N_e} = {}_{r-1}\mathbf{y}_p^{\widehat{k}_1}$, and get $w_t^i = f_p({}_r\mathbf{y}_p^i)$, $i = 1, 2, \dots, N_e$, $w_t^{\widehat{k}_1} = f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) > w_t^i = f_p({}_r\mathbf{y}_p^i)$, $i \neq \widehat{k}_1$. Then

$$\begin{aligned} f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) &> f_p({}_r\mathbf{y}_p^{N_e}) = f_p({}_{r-1}\mathbf{y}_p^{\widehat{k}_1}), \quad \widehat{k}_1 \neq N_e, \\ f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) &= f_p({}_r\mathbf{y}_p^{N_e}) = f_p({}_{r-1}\mathbf{y}_p^{\widehat{k}_1}), \quad \widehat{k}_1 = N_e; \end{aligned} \quad (\text{A.1})$$

thus, we get

$$\begin{aligned} |f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| &\leq |f_p({}_{r-1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)|, \\ |f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| &< |f_p({}_r\mathbf{y}_p^i) - f_p(\mathbf{y}_p^*)|, \end{aligned} \quad (\text{A.2})$$

$$i \neq \widehat{k}_1, \quad i = 1, 2, \dots, N_e;$$

let ${}_{r+1}\mathbf{y}_p^{N_e} = {}_r\mathbf{y}_p^{\widehat{k}_1}$, in the $(r+1)$ th time searching, and we also have ${}_{r+1}\mathbf{y}_p^i = {}_r\mathbf{y}_p^{\widehat{k}_1} + \tilde{\mathbf{u}}_t^i$, $i = 1, \dots, N_e - 1$, similarly, and get $w_t^i = f_p({}_{r+1}\mathbf{y}_p^i)$, $i = 1, 2, \dots, N_e$, $w_t^{\widehat{k}_1} = f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) > w_t^i = f_p({}_{r+1}\mathbf{y}_p^i)$, $i \neq \widehat{k}_1$. Then,

$$\begin{aligned} f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) &> f_p({}_{r+1}\mathbf{y}_p^{N_e}) = f_p({}_r\mathbf{y}_p^{\widehat{k}_1}), \quad \widehat{k}_1 \neq N_e, \\ f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) &= f_p({}_{r+1}\mathbf{y}_p^{N_e}) = f_p({}_r\mathbf{y}_p^{\widehat{k}_1}), \quad \widehat{k}_1 = N_e, \end{aligned}$$

$$\begin{aligned} |f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| &\leq |f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| \\ &\leq |f_p({}_{r-1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)|, \end{aligned}$$

$$\begin{aligned} |f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| &< |f_p({}_{r+1}\mathbf{y}_p^i) - f_p(\mathbf{y}_p^*)|, \\ i &\neq \widehat{k}_1, \quad i = 1, 2, \dots, N_e. \end{aligned} \quad (\text{A.3})$$

If $|f_p({}_{r+1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| = |f_p({}_r\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| = |f_p({}_{r-1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)|$, the ${}_{r-1}\mathbf{y}_p^{\widehat{k}_1}$ can be kept, and then ${}_{r+1}\mathbf{y}_p^{N_e} = {}_r\mathbf{y}_p^{\widehat{k}_1} = {}_{r-1}\mathbf{y}_p^{\widehat{k}_1}$, and let $r' = r + 1$, and continue to search until finding ${}_{r'+1}\mathbf{y}_p^{\widehat{k}_1}$ to make $f_p({}_{r'+1}\mathbf{y}_p^{\widehat{k}_1}) > f_p({}_{r'+1}\mathbf{y}_p^{N_e}) = f_p({}_{r'}\mathbf{y}_p^{\widehat{k}_1})$, $\widehat{k}_1 \neq N_e$, and $|f_p({}_{r'+1}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)| < |f_p({}_{r'}\mathbf{y}_p^{\widehat{k}_1}) - f_p(\mathbf{y}_p^*)|$. Due to $\tilde{\mathbf{u}}_t^i$ being stochastic searching vector in the space of d dimensions, it will be possible to satisfy this, according to theory of probability. Thus, when r is large enough, $\mathbf{y}_p^* = \arg \max_{\mathbf{y}_p^i} f_p(\mathbf{y}_p^i) \approx {}_r\mathbf{y}_p^{\widehat{k}_1}$.

SEMAS can run as in Algorithm 3.

B. Derivation of the Condition in Section 5

B.1. Condition. If condition $\|\mathbf{D} - \mathbf{P}\tilde{\boldsymbol{\theta}}\|^2/(N \times Q_1) = \|\mathbf{D} - \widetilde{\mathbf{W}}\widehat{\mathbf{g}}\|^2/(N \times Q_1) = \text{tr}(\mathbf{e}^T \mathbf{e})/(N \times Q_1) < \varepsilon_1$, $\varepsilon_1 > 0$ is satisfied, and if condition $\|\mathbf{D} - \mathbf{G}\widehat{\mathbf{F}}\|^2/(N \times Q_1) = \|\mathbf{D} - \mathbf{P}_G \mathbf{D}\|^2/(N \times Q_1) = \text{tr}((\mathbf{e}')^T \mathbf{e}')/(N \times Q_1) < \varepsilon_2 < \varepsilon_1$, $\varepsilon_2 > 0$ can also be satisfied.

B.2. Derivation. According to (19), (20), and property of LS, $\mathbf{P}_\theta = \mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$, $\mathbf{P}_{1/\theta} = \mathbf{I} - \mathbf{P}_\theta$, $\mathbf{e}^T \mathbf{P} = \mathbf{P}^T \mathbf{e} = \mathbf{0}$, $\mathbf{e}^T \mathbf{P}_\theta = \mathbf{P}_\theta^T \mathbf{e} = \mathbf{0}$, and $\mathbf{R}_I^T \mathbf{P}_{1/\theta} \mathbf{R}_I \in \mathbb{R}^{1 \times 1}$; we can see

$$\begin{aligned} \mathbf{e}' &= \mathbf{D} - \mathbf{G}\widehat{\mathbf{F}} = \mathbf{D} - \mathbf{P}\tilde{\boldsymbol{\theta}} - \mathbf{P}_{1/\theta} \mathbf{R}_I (\mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I)^{-1} \mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{D} \\ &= \mathbf{e} - \mathbf{P}_{1/\theta} \mathbf{R}_I (\mathbf{R}_I^T \mathbf{P}_{1/\theta} \mathbf{R}_I)^{-1} \mathbf{R}_I^T \mathbf{e} \end{aligned}$$

Input: $\mathbf{y}_p^{\hat{k}_1}$, R , L , $\mathbf{v}$, A_1 , A_2 , γ_1 , γ_2 , η' ;
Output: $\mathbf{y}_p^*$;
For $r = 1 : 1 : R$
 $\mathbf{u}_t^i = \mathbf{v} * \text{rand}(d, 1)$; $\tilde{\mathbf{u}}_t^i = \eta_i L(\mathbf{u}_t^i / \|\mathbf{u}_t^i\|)$ or $L\mathbf{u}_t^i$;
 ${}_r\mathbf{y}_p^i = {}_{r-1}\mathbf{y}_p^{\hat{k}_1} + \tilde{\mathbf{u}}_t^i$, $i = 1, \dots, N_e - 1$;
 ${}_r\mathbf{y}_p^{N_e} = {}_{r-1}\mathbf{y}_p^{\hat{k}_1}$; $w_t^i = f_p({}_r\mathbf{y}_p^i)$, $i = 1, 2, \dots, N_e$;
 $\hat{k}_1 = \arg \max_i (w_t^i)$, $w_t^i \in \mathbf{W}_o = \{w_t^1, w_t^2, \dots, w_t^{N_e}\}$;
If $r > 1$ and $|f_p({}_{r-1}\mathbf{y}_p^{\hat{k}_1}) - f_p({}_{r-2}\mathbf{y}_p^{\hat{k}_1})| < \eta'$
 $L = A_1 + \gamma_1 \left\| {}_{r-1}\mathbf{y}_p^{\hat{k}_1} - {}_{r-2}\mathbf{y}_p^{\hat{k}_1} \right\|$;
Else If $r > 1$ and $|f_p({}_{r-1}\mathbf{y}_p^{\hat{k}_1}) - f_p({}_{r-2}\mathbf{y}_p^{\hat{k}_1})| \geq \eta'$
 $L = A_2 + \gamma_2 \left\| {}_{r-1}\mathbf{y}_p^{\hat{k}_1} - {}_{r-2}\mathbf{y}_p^{\hat{k}_1} \right\|$;
End If
End For
 $\mathbf{y}_p^* = {}_r\mathbf{y}_p^{\hat{k}_1}$;

ALGORITHM 3: SEMAS.

$$= \mathbf{e} - \frac{\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{\mathbf{R}_I^T \mathbf{P}_{1/\theta} \mathbf{R}_I} = \mathbf{e} - \frac{\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{\mathbf{R}_I^T \mathbf{R}_I - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$= \mathbf{e} - \frac{\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I};$$

(B.1)

then we get

$$(\mathbf{e}')^T \mathbf{e}' = \mathbf{e}^T \mathbf{e} - \frac{(\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})^T \mathbf{e}}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I} - \frac{\mathbf{e}^T (\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$+ \frac{(\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})^T (\mathbf{P}_{1/\theta} \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})}{(N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I)^2}$$

$$= \mathbf{e}^T \mathbf{e} - \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{P}_\theta^T \mathbf{e}}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$- \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{P}_\theta (\mathbf{R}_I \mathbf{R}_I^T \mathbf{e})}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$+ \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{(N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I)^2}$$

$$= \mathbf{e}^T \mathbf{e} - \frac{2\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$+ \frac{N\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{(N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I)^2}$$

$$= \mathbf{e}^T \mathbf{e} + (-2(N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I) \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} + N\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})$$

$$\cdot (N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I)^{-2}$$

$$= \mathbf{e}^T \mathbf{e} + (-N\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} + 2(\mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I) \mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e} - \mathbf{e}^T \mathbf{R}_I (\mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I) \mathbf{R}_I^T \mathbf{e})$$

$$\cdot (N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I)^{-2}$$

$$= \mathbf{e}^T \mathbf{e} - \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{N - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I} = \mathbf{e}^T \mathbf{e} - \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{\mathbf{R}_I^T \mathbf{R}_I - \mathbf{R}_I^T \mathbf{P}_\theta \mathbf{R}_I}$$

$$= \mathbf{e}^T \mathbf{e} - \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{\mathbf{R}_I^T \mathbf{P}_{1/\theta} \mathbf{R}_I} = \mathbf{e}^T \mathbf{e} - \frac{\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e}}{\mathbf{R}_I^T \mathbf{P}_{1/\theta} \mathbf{P}_{1/\theta} \mathbf{R}_I},$$

(B.2)

and then $\text{tr}(\mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I) = \mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I \in R^{1 \times 1}$; we have

$$\text{tr}((\mathbf{e}')^T \mathbf{e}') = \text{tr}(\mathbf{e}^T \mathbf{e}) - \frac{\text{tr}(\mathbf{e}^T \mathbf{R}_I \mathbf{R}_I^T \mathbf{e})}{\text{tr}(\mathbf{R}_I^T \mathbf{P}_{1/\theta}^T \mathbf{P}_{1/\theta} \mathbf{R}_I)}$$

$$= \text{tr}(\mathbf{e}^T \mathbf{e}) - \frac{\text{tr}((\mathbf{R}_I^T \mathbf{e})^T \mathbf{R}_I^T \mathbf{e})}{\text{tr}((\mathbf{P}_{1/\theta} \mathbf{R}_I)^T \mathbf{P}_{1/\theta} \mathbf{R}_I)},$$

(B.3)

and, from the equation above, we know $\mathbf{R}_I^T \mathbf{e} \neq 0$, $\text{tr}((\mathbf{R}_I^T \mathbf{e})^T \mathbf{R}_I^T \mathbf{e}) > 0$, $\mathbf{P}_{1/\theta} \mathbf{R}_I \neq 0$, and $\text{tr}((\mathbf{P}_{1/\theta} \mathbf{R}_I)^T \mathbf{P}_{1/\theta} \mathbf{R}_I) > 0$ and get $\text{tr}((\mathbf{R}_I^T \mathbf{e})^T \mathbf{R}_I^T \mathbf{e}) / \text{tr}((\mathbf{P}_{1/\theta} \mathbf{R}_I)^T \mathbf{P}_{1/\theta} \mathbf{R}_I) > 0$; thus we let $\rho = \text{tr}((\mathbf{R}_I^T \mathbf{e})^T \mathbf{R}_I^T \mathbf{e}) / \text{tr}((\mathbf{P}_{1/\theta} \mathbf{R}_I)^T \mathbf{P}_{1/\theta} \mathbf{R}_I)$ and $\varepsilon_2 = \varepsilon_1 - \rho / (N \times Q_1)$ and get this following equation, when

$$\frac{\|\mathbf{D} - \mathbf{P}\tilde{\theta}\|^2}{(N \times Q_1)} = \frac{\|\mathbf{D} - \mathbf{W}\tilde{\mathbf{g}}\|^2}{(N \times Q_1)} = \frac{\text{tr}(\mathbf{e}^T \mathbf{e})}{(N \times Q_1)} < \varepsilon_1, \quad \varepsilon_1 > 0,$$

$$\frac{\|\mathbf{D} - \mathbf{G}\tilde{\mathbf{F}}\|^2}{(N \times Q_1)} = \frac{\|\mathbf{D} - \mathbf{P}_G \mathbf{D}\|^2}{(N \times Q_1)}$$

$$= \frac{\text{tr}((\mathbf{e}')^T \mathbf{e}')}{(N \times Q_1)} = \frac{(\text{tr}(\mathbf{e}^T \mathbf{e}) - \rho)}{(N \times Q_1)} < \varepsilon_1$$

$$- \frac{\rho}{(N \times Q_1)} = \varepsilon_2 < \varepsilon_1, \quad \varepsilon_2 > 0.$$

(B.4)

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61202292) and Guangdong Provincial Natural Science Foundation of China (no. 9151064101000037).

References

- [1] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of multiple characters using multiview image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2720–2735, 2013.
- [2] E. Yeguas-Bolivar, R. Muñoz-Salinas, R. Medina-Carnicer, and A. Carmona-Poyato, "Comparing evolutionary algorithms and particle filters for Markerless Human Motion Capture," *Applied Soft Computing Journal*, vol. 17, pp. 153–166, 2014.
- [3] U. Gündükbay, İ. Demir, and Y. Dedeoğlu, "Motion capture and human pose reconstruction from a single-view video sequence," *Digital Signal Processing*, vol. 23, no. 5, pp. 1441–1450, 2013.
- [4] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 4–27, 2010.
- [5] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 185–205, 2005.
- [6] Y. Li, Z.-X. Sun, S.-L. Chen, and Q. Li, "3D human pose analysis from monocular video by simulated annealed particle swarm optimization," *Acta Automatica Sinica*, vol. 38, no. 5, pp. 732–741, 2012.
- [7] T. Sato, "Particle relaxation method of Monte Carlo filter for structure system identification," *Journal of Civil Structural Health Monitoring*, vol. 3, no. 4, pp. 325–334, 2013.
- [8] C. A. L. Waechter, D. Pustka, and G. J. Klinker, "Real-time monocular people tracking by sequential Monte-Carlo filtering," in *Proceedings of the 6th International Conference on Computer Vision (CV '13)/Computer Graphics Collaboration Techniques and Applications (CGCTA '13)*, pp. 1–7, June 2013.
- [9] G. To and M. R. Mahfouz, "Quaternionic attitude estimation for robotic and human motion tracking using sequential monte carlo methods with von mises-fisher and nonuniform densities simulations," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 11, pp. 3046–3059, 2013.
- [10] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.
- [11] Y. W. Wong, K. P. Seng, and L.-M. Ang, "Radial basis function neural network with incremental learning for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 4, pp. 940–949, 2011.
- [12] N. Sun and Y.-F. Guo, "An improved incremental learning approach based on SVM model for network data stream," in *Advances in Computer Science and Information Engineering*, Advances in Intelligent and Soft Computing, pp. 277–282, Springer, Berlin, Germany, 2012.
- [13] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
- [14] Z. Zhang, T. Ke, N. Deng, and J. Tan, "Biased p -norm support vector machine for PU learning," *Neurocomputing*, vol. 136, pp. 265–261, 2014.
- [15] F. Dong, Z. Chen, and J. Wang, "A new level set method for inhomogeneous image segmentation," *Image and Vision Computing*, vol. 31, no. 10, pp. 809–822, 2013.
- [16] A. Kasaiezadeh and A. Khajepour, "Multi-agent stochastic level set method in image segmentation," *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 1147–1162, 2013.
- [17] B. He, G.-J. Wang, and C. Zhang, "Iterative transductive learning for automatic image segmentation and matting with RGB-D data," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1031–1043, 2014.
- [18] N. Andrei, "Scaled conjugate gradient algorithms for unconstrained optimization," *Computational Optimization and Applications*, vol. 38, no. 3, pp. 401–416, 2007.
- [19] I. T. Nabney, *Algorithms for Pattern Recognition*, Springer, Berlin, Germany, 2001.
- [20] X.-D. Zhang, *Matrix Analysis and Applications*, Tsinghua University press, Beijing, China, 2nd edition, 2013.
- [21] M. F. Bugallo and P. M. Djuric, "Gaussian particle filtering in high-dimensional systems," in *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP '14)*, pp. 129–132, Gold Coast, Australia, June-July 2014.
- [22] F. A. Ruslan, R. Adnan, A. M. Samad, and Z. M. Zain, "Parameters effect in Sampling Importance Resampling (SIR) particle filter prediction and tracking of flood water level performance," in *Proceedings of the 12th International Conference on Control, Automation and Systems (ICCAS '12)*, pp. 868–872, Jeju-do, Republic of Korea, October 2012.
- [23] L. Sigal, A. O. Balan, and M. J. Black, *Humaneva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion*, The Report of Brown University, Providence, RI, USA, 2006.
- [24] M. Sandau, H. Koblauch, T. B. Moeslund, H. Aanæs, T. Alkjær, and E. B. Simonsen, "Markerless motion capture can provide reliable 3D gait kinematics in the sagittal and frontal plane," *Medical Engineering and Physics*, vol. 39, no. 9, pp. 1168–1175, 2014.
- [25] W. Luo, T. Yamasaki, and K. Aizawa, "Cooperative estimation of human motion and surfaces using multiview videos," *Computer Vision and Image Understanding*, vol. 117, no. 11, pp. 1560–1574, 2013.