

Research Article

Human Motion Capture Based on Incremental Dimension Reduction and Projection Position Optimization

Wanyi Li , Yuqi Zeng, Qian Zhang, Yilin Wu, and Guoming Chen

School of Computer Science, Guangdong University of Education, Guangzhou, Guangdong 510303, China

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

Received 8 February 2021; Revised 4 April 2021; Accepted 10 April 2021; Published 29 April 2021

Academic Editor: Giuseppe Piro

Copyright © 2021 Wanyi Li et al. 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.

Three-dimensional (3D) human motion capture is a hot researching topic at present. The network becomes advanced nowadays, the appearance of 3D human motion is indispensable in the multimedia works, such as image, video, and game. 3D human motion plays an important role in the publication and expression of all kinds of medium. How to capture the 3D human motion is the key technology of multimedia product. Therefore, a new algorithm called incremental dimension reduction and projection position optimization (IDRPPO) is proposed in this paper. This algorithm can help to learn sparse 3D human motion samples and generate the new ones. Thus, it can provide the technique for making 3D character animation. By taking advantage of the Gaussian incremental dimension reduction model (GIDRM) and projection position optimization, the proposed algorithm can learn the existing samples and establish the relevant mapping between the low dimensional (LD) data and the high dimensional (HD) data. Finally, the missing frames of input 3D human motion and the other type of 3D human motion can be generated by the IDRPPO.

1. Introduction

Three-dimensional (3D) human motion capture is applied for many fields, such as medical diagnosis, animation making, and 3D video game development [1–3]. How to generate the human motion in 3D becomes curial to these works. Human motion in 3D is depicted by high-dimensional (HD) data, and the motion sequence consists of poses. Each pose can be exhibited by a human motion model. One complete motion cycle is called a gait.

3D human motion capture has been developed into a hot researching topic. How to generate the human motion in 3D has various techniques. One of the hot techniques is the reconstruction of 3D human motion from the image sequence, which needs some complex preprocessing methods to extract the image feature and analyze feature sample, such as video event analysis [4] and video feature analysis [5]. Another one is 3D human motion estimation of self-supervised learning, which is learning the sparse samples of one type human motion and generating the other type human motion. Then, the method of self-supervised learning will be mainly discussed in this article. The self-supervised learning can be seen

as the special case of unsupervised learning to some extent. Recently, there are some methods which contain defects. In [6, 7], some heuristic algorithms are used to process the pre-processed image for generating the human motion. It will cost too much time, and the quality of generated human motion is susceptible to preprocessing quality of the image. The accuracy and efficiency are low. Some dimension reduction models [8–11] can process the human motion efficiently, but the HD data of the human motion can only be visualized by these models in low-dimensional (LD) space. Some improved dimension reduction models [12] have two mappings between LD space and HD space, which can generate the LD data sample for transformation of the HD data sample. These models will do great help to generate human motion, but the other type of human motion cannot be obtained. A certain improved method in [13] is proposed to fit the human motion sequence, which needs to process the LD data in LD space and increases the difficulty of generating the human motion. These methods above also cannot fast obtain one type motion from the other type directly. In summary, how to generate one type motion from the other type directly is not an easy task in a short time. The CNN [14] and its corresponding networks

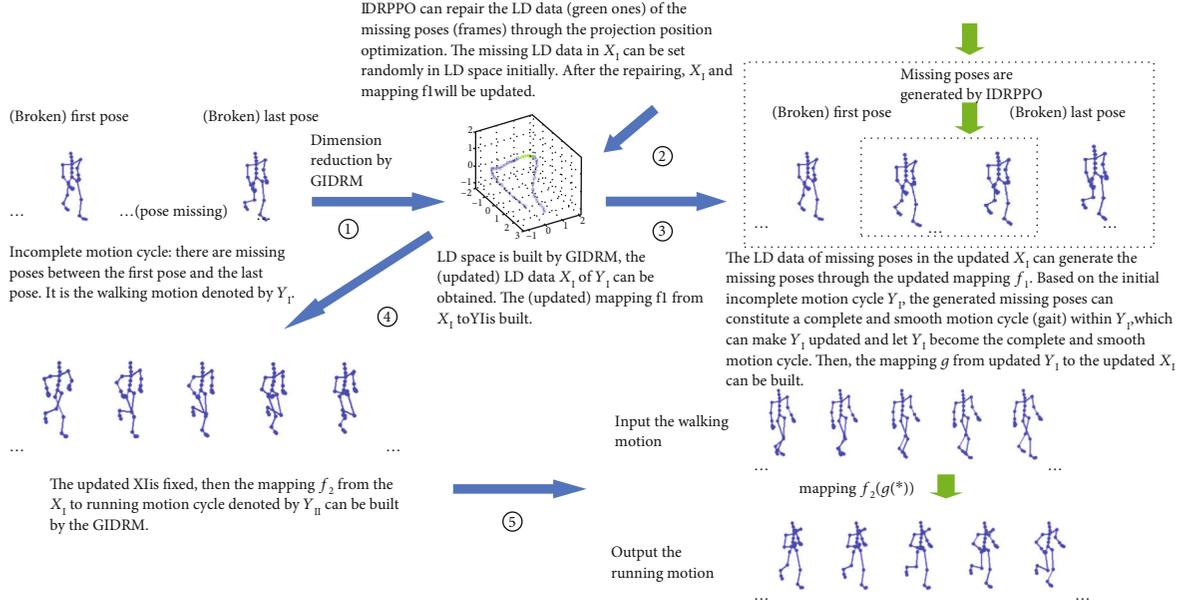


FIGURE 1: The technique framework of IDRPPPO.

are emerging in the recent years (e.g., Resnet [15], AlexNet [16], VGG [17], SqueezeNet [18], DenseNet [19], and Inception [20]), but these networks working will need much training time, a large amount of datasets, and huge budget of hardware, which will even need the high-level and costly GPU for work. Thus, a new machine learning method need to be proposed, which is suitable for fast making the animation of 3D human character. Meanwhile, the proposed method can generate the new valid train data and corresponding pseudolabel (self-encoded) data (LD data), which can be used to retrain the model and improve the prediction. In general, it can improve the self-supervised learning model. The data sequence seen as a matrix can be processed by the proposed method directly, so that it can promote performance of some frameworks of tracking and estimation to a certain extent, such as self-supervised seep correlation tracking [21] (self-SDCT). Without the artificial annotation, the proposed method can obtain the new essential samples according to the data requirement of the self-supervised learning model and let the model update the generating mapping for the improvement of tracking or estimating by the help of these samples.

In this paper, the new algorithm (method) called incremental dimension reduction and projection position optimization (IDRPPPO) is proposed to address the problems mentioned above. It can generate one type human motion from the other type. In addition, the input motion samples can be incomplete gait. IDRPPPO will show the promising performance from the experimental tests of visual effect and error. IDRPPPO will take advantage of Gaussian incremental dimension reduction model (GIDRM) [7] and projection position optimization to carry out the self-supervised learning of small-scale samples. GIDRM is similar to the bilinear analysis model of compound rank- k projections [22](CRP). Inspired by CRP, the adoption of GIDRM can process the complex HD data of the 3D human motion and make these

HD data visualized and regularized. Firstly, GIDRM can process the matrix directly without the transformation of the vectors, which is conducive to decrease the computation complexity and improve the model flexibility. The matrix can denote the HD sample sequence of human motion or the corresponding LD data sequence. Secondly, GIDRM can provide the LD space for searching and generating the optimal LD data sample, so that the corresponding 3D human motion can be reconstructed by its mappings. The two advantages are essential to the efficiency of IDRPPPO for estimating the 3D human motion. Thus, IDRPPPO with the GIDRM can learn one type incomplete gait, then the missing frames in incomplete gait and the other type motion can be output perfectly by it. Our contributions are listed as follows:

- (1) Address the problem of filling the missing frames in the incomplete motion cycle and make the motion cycle complete and smooth
- (2) Address the problem of generating the other type motion cycle from the origin incomplete motion cycle by the help of the IDRPPPO

The performance of the IDRPPPO will be tested from the experiments, and the results will indicate the IDRPPPO can help to achieve the promising visual effect and low estimating error for human motion capture. The technique framework of IDRPPPO can be seen in Figure 1. Then, the details of IDRPPPO will be discussed in the following sections.

2. Generation of Human Motion through IDRPPPO

2.1. Gaussian Incremental Dimension Reduction Model. According to the references above, the models can be given as follows [7, 12]:

$$p(\mathbf{Y} | \mathbf{X}, \bar{\boldsymbol{\beta}}, \mathbf{W}) = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND} |\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_Y^{-1} \mathbf{Y} \mathbf{W}^2 \mathbf{Y}^T)\right), \quad (1)$$

$$p(\mathbf{X} | \bar{\boldsymbol{\alpha}}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)q} |\mathbf{K}_X|^q}} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{K}_X^{-1} \mathbf{X}_{2:N} \mathbf{X}_{2:N}^T)\right). \quad (2)$$

From Equation (2) and Equation (1), HD data sequence can be denoted by $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_i, \dots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times D}$, $\mathbf{y}_i \in \mathbb{R}^D$. LD data sequence can be denoted by $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times q}$, $\mathbf{x}_i \in \mathbb{R}^q$. Kernel matrix is denoted by $\mathbf{K}_Y \in \mathbb{R}^{N \times N}$, $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j) = \exp(-(\beta_1/2) \|\mathbf{x}_i - \mathbf{x}_j\|^2) + \beta_2^{-1} \delta_{\mathbf{x}_i, \mathbf{x}_j}$. $\bar{\boldsymbol{\beta}} = [\beta_1, \beta_2]$ are the kernel parameters of \mathbf{K}_Y , and the other kernel matrix can be denoted by $\mathbf{K}_X \in \mathbb{R}^{(N-1) \times (N-1)}$, $(\mathbf{K}_X)_{i,j} = k_X(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1 \exp(-(\alpha_2/2) \|\mathbf{x}_i - \mathbf{x}_j\|^2) + \alpha_3 \mathbf{x}_i^T \mathbf{x}_j + \alpha_4^{-1} \delta_{\mathbf{x}_i, \mathbf{x}_j}$. $\bar{\boldsymbol{\alpha}} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ are the kernel parameters of \mathbf{K}_X . \mathbf{W} is the scale parameter matrix, then $p(\mathbf{W}) = \prod_{m=1}^D (2/(\kappa\sqrt{2\pi})) \exp(-w_m^2/(2\kappa^2))$, $w_m > 0$, $\kappa = 10^{-3}$. Let $\mathbf{X}_{2:N} = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N]^T$, and \mathbf{x}_1 confronts the of q -dimensional Gaussian distribution. $\bar{\boldsymbol{\alpha}}$ 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. In Equation (1) and Equation (2), \mathbf{Y} is known; thus, $p(\mathbf{Y})$ is constant, and the equivalence of $\min(-\ln p(\mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W} | \mathbf{Y})) \Leftrightarrow \min(-\ln p(\mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W}, \mathbf{Y}))$ can be got. The LD data and corresponding parameters can be obtained as follows:

$$\begin{aligned} \mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W} &= \arg \min_{\mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W}} (-\ln p(\mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W}, \mathbf{Y})) \\ &= \arg \min_{\mathbf{X}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \mathbf{W}} (-\ln (p(\mathbf{Y} | \mathbf{X}, \bar{\boldsymbol{\beta}}, \mathbf{W}) p(\mathbf{X} | \bar{\boldsymbol{\alpha}}) p(\bar{\boldsymbol{\alpha}}) p(\bar{\boldsymbol{\beta}}) p(\mathbf{W}))), \end{aligned} \quad (3)$$

where $\mathbf{y} \sim \mathbf{N}(\mu_Y(\mathbf{x}), \sigma_Y^2(\mathbf{x})\mathbf{I})$, $\mathbf{y} \in \mathbb{R}^D$, $\mathbf{x} \in \mathbb{R}^q$, the mapping from HD space to LD space can be built as follows:

$$\begin{aligned} \mathbf{y} = \mathbf{f}(\mathbf{x}) = \mu_Y(\mathbf{x}) &= \mathbf{Y}^T \mathbf{K}_Y^{-1} [k_Y(\mathbf{x}_1, \mathbf{x}), k_Y(\mathbf{x}_2, \mathbf{x}), \dots, k_Y(\mathbf{x}_N, \mathbf{x})]^T \\ &= \mathbf{Y}^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}). \end{aligned} \quad (4)$$

If two or more mappings from LD space to HD space need to be built, Equation (3) can be retrained according to the needs. After building the first mapping, the LD data from the first mapping can be fixed, which can be seen as the initial LD data of the second mapping training.

Then, the mapping of the incremental dimension reduction is built as follows:

$$\mathbf{X} = \boldsymbol{\Phi} \mathbf{W}_D, \quad (5)$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N \times Nk}$ is radial basis function, $\boldsymbol{\Phi}_{k_1, k_2} = \phi(\mathbf{y}_{k_1}, \mathbf{c}_{k_2}) = \exp(b \|\mathbf{y}_{k_1} - \mathbf{c}_{k_2}\|^2)$. $\mathbf{W}_D \in \mathbb{R}^{Nk \times q}$ is the weight matrix, $Nk \leq N$. $\widehat{\mathbf{W}}_D$ is least squares estimator, $\widehat{\mathbf{W}}_D = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{X}$. Then, $\mathbf{y}^* \in \mathbb{R}^D$ denotes the new HD data sample, $\mathbf{x}^* \in \mathbb{R}^D$ denotes the LD data of $\mathbf{y}^* \in \mathbb{R}^D$. If b is known, the mapping from \mathbf{y}^* to \mathbf{x}^* can be given as follows:

$$\mathbf{x}^* = \mathbf{g}(\mathbf{y}^*) = (\boldsymbol{\Phi}(\mathbf{y}^*) \widehat{\mathbf{W}}_D)^T = \widehat{\mathbf{W}}_D^T (\boldsymbol{\Phi}(\mathbf{y}^*))^T, \quad (6)$$

where $\boldsymbol{\Phi}(\mathbf{y}^*) = [\phi(\mathbf{y}^*, \mathbf{c}_1), \phi(\mathbf{y}^*, \mathbf{c}_2), \dots, \phi(\mathbf{y}^*, \mathbf{c}_{Nk})]$, then we can get the equation as follows:

$$\mathbf{X} = \boldsymbol{\Phi} \widehat{\mathbf{W}}_D + \mathbf{e}. \quad (7)$$

In Equation (7), $\mathbf{e} \in \mathbb{R}^{N \times Nk}$ is the error matrix, let $\mathbf{e} = [\mathbf{e}_1, \dots, \mathbf{e}_{k_1}, \dots, \mathbf{e}_N]^T = [\mathbf{e}'_1, \dots, \mathbf{e}'_{v_1}, \dots, \mathbf{e}'_q] \in \mathbb{R}^{N \times q}$. Then, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{k_1}, \dots, \mathbf{x}_N]^T = [\mathbf{x}'_1, \dots, \mathbf{x}'_{v_1}, \dots, \mathbf{x}'_q] \in \mathbb{R}^{N \times q}$, $\widehat{\mathbf{W}}_D = [\widehat{\mathbf{w}}_1, \dots, \widehat{\mathbf{w}}_{k_2}, \dots, \widehat{\mathbf{w}}_{Nk}]^T = [\widehat{\mathbf{w}}'_1, \dots, \widehat{\mathbf{w}}'_{v_1}, \dots, \widehat{\mathbf{w}}'_q] \in \mathbb{R}^{Nk \times q}$, and $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{k_2}, \dots, \boldsymbol{\phi}_{Nk}] \in \mathbb{R}^{N \times Nk}$. Let $\boldsymbol{\Phi} = \widetilde{\mathbf{W}} \widetilde{\mathbf{A}}$, and $\widetilde{\mathbf{W}} = [\widetilde{\mathbf{w}}_1, \dots, \widetilde{\mathbf{w}}_{k_2}, \dots, \widetilde{\mathbf{w}}_{Nk}] \in \mathbb{R}^{N \times Nk}$ is a diagonal matrix ($\widetilde{\mathbf{w}}_{k_2}^T \widetilde{\mathbf{w}}_{k_2}' = 0$, $k_2 \neq k_2'$; $\widetilde{\mathbf{w}}_{k_2}^T \widetilde{\mathbf{w}}_{k_2}' \neq 0$, $k_2 = k_2'$). $\widetilde{\mathbf{A}} \in \mathbb{R}^{Nk \times Nk}$ is an invertible matrix ($\widetilde{\mathbf{A}}_{i,j} = a_{i,j}$, $i < j$; $\widetilde{\mathbf{A}}_{i,j} = 1$, $i = j$; $\widetilde{\mathbf{A}}_{i,j} = 0$, $i > j$). We have $\widehat{\mathbf{W}}_D = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{X}$, then let $\widehat{\mathbf{g}} = (\widetilde{\mathbf{W}}^T \widetilde{\mathbf{W}})^{-1} \widetilde{\mathbf{W}}^T \mathbf{X}$ and $\widehat{\mathbf{g}} = [\widehat{\mathbf{g}}_1, \dots, \widehat{\mathbf{g}}_{k_2}, \dots, \widehat{\mathbf{g}}_{Nk}]^T = [\widehat{\mathbf{g}}_1, \dots, \widehat{\mathbf{g}}_{v_1}, \dots, \widehat{\mathbf{g}}_q] \in \mathbb{R}^{Nk \times q}$. The equation can be got as follows:

$$\begin{aligned} \boldsymbol{\Phi} \widehat{\mathbf{W}}_D &= \boldsymbol{\Phi} (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{X} = \widetilde{\mathbf{W}} \widetilde{\mathbf{A}} (\widetilde{\mathbf{A}}^T \widetilde{\mathbf{W}}^T \widetilde{\mathbf{W}} \widetilde{\mathbf{A}})^{-1} \widetilde{\mathbf{A}}^T \widetilde{\mathbf{W}}^T \mathbf{X} \\ &= \widetilde{\mathbf{W}} (\widetilde{\mathbf{W}}^T \widetilde{\mathbf{W}})^{-1} \widetilde{\mathbf{W}}^T \mathbf{X} = \widetilde{\mathbf{W}} \widehat{\mathbf{g}}. \end{aligned} \quad (8)$$

Thus, Equation (8) can be written as:

$$\mathbf{X} = \widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \mathbf{e}. \quad (9)$$

According to the properties of least squares, $\widetilde{\mathbf{W}}^T \mathbf{e} = \mathbf{0}$, $\mathbf{e}^T \widetilde{\mathbf{W}} = \mathbf{0}$, we have:

$$\begin{aligned} \mathbf{X}^T \mathbf{X} &= (\widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \mathbf{e})^T (\widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \mathbf{e}) \\ &= \widehat{\mathbf{g}}^T \widetilde{\mathbf{W}}^T \widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \widehat{\mathbf{g}}^T \widetilde{\mathbf{W}}^T \mathbf{e} + \mathbf{e}^T \widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \mathbf{e}^T \mathbf{e} \\ &= \widehat{\mathbf{g}}^T \widetilde{\mathbf{W}}^T \widetilde{\mathbf{W}} \widehat{\mathbf{g}} + \mathbf{e}^T \mathbf{e}, \end{aligned} \quad (10)$$

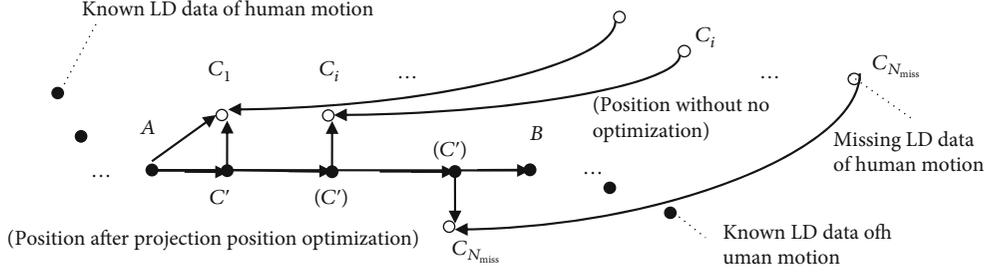


FIGURE 2: The framework of projection position optimization.

where $\mathbf{e}^T \mathbf{e} = \mathbf{X}^T \mathbf{X} - \widehat{\mathbf{g}}^T \widehat{\mathbf{W}}^T \widehat{\mathbf{W}} \widehat{\mathbf{g}}$. When training, the Nk orthogonal vectors can be replaced; the equation can be got as follows:

$$\begin{aligned}
& \min \left(\left\| \mathbf{X} - \Phi \widehat{\mathbf{W}}_D \right\|^2 / (N \times q) \right) \\
& = \min \left(\left\| \mathbf{X} - \widehat{\mathbf{W}} \widehat{\mathbf{g}} \right\|^2 / (N \times q) \right) \\
& = \min \left(\text{tr}(\mathbf{e}^T \mathbf{e}) / (N \times q) \right) \\
& = \min \left(\frac{1}{q} \sum_{v_1=1}^q \left(\frac{1}{N} (\mathbf{e}'_{v_1})^T \mathbf{e}'_{v_1} \right) \right) \\
& = \min \left(\frac{1}{q} \sum_{v_1=1}^q \left(\frac{1}{N} (\mathbf{x}'_{v_1})^T \mathbf{x}'_{v_1} - \frac{1}{N} \sum_{k_2=1}^{Nk} \tilde{\mathbf{w}}_{k_2}^T \tilde{\mathbf{w}}_{k_2} \widehat{\mathbf{g}}_{v_1}^2(k_2) \right) \right). \tag{11}
\end{aligned}$$

Equation (11) is equivalent to the equation as follows:

$$S_w = \arg \max_{S_w} \left(\sum_{v_1=1}^q \left(\sum_{k_2=1}^{Nk} \left(\tilde{\mathbf{w}}_{k_2}^T \tilde{\mathbf{w}}_{k_2} \widehat{\mathbf{g}}_{v_1}^2(k_2) / (\mathbf{x}'_{v_1})^T \mathbf{x}'_{v_1} \right) \right) \right), \quad S_w \in S'_w, Nk \leq N. \tag{12}$$

In Equation (12), $S_w = \{\tilde{\mathbf{w}}_1, \dots, \tilde{\mathbf{w}}_{k_2}, \dots, \tilde{\mathbf{w}}_{Nk}\}$ and $S'_w = \{\tilde{\mathbf{w}}'_1, \dots, \tilde{\mathbf{w}}'_{k_2}, \dots, \tilde{\mathbf{w}}'_{Nk}\}$ both are the sets of orthogonal vectors. S_w is the subset of S'_w , S'_w is the set containing ϕ'_{k_2} which is the vector from $\Phi' = [\phi'_1, \dots, \phi'_{k_2}, \dots, \phi'_N] \in \mathbf{R}^{N \times N}$, then $\phi'_{k_1, k_2} = \phi(\mathbf{y}_{k_1}, \mathbf{y}_{k_2}) = \exp(b \|\mathbf{y}_{k_1} - \mathbf{y}_{k_2}\|^2)$. When the tolerance $\|\mathbf{X} - \Phi \widehat{\mathbf{W}}_D\|^2 / (N \times q) < \varepsilon_1, \varepsilon_1 > 0$ is satisfied, the training can be finished. It means that the vector ϕ'_{k_2} is selected as few as possible to minimize the variable Nk for the satisfaction of the tolerance, so that the mapping training can be finished.

2.2. Projection Position Optimization. The learning of the incomplete gait of human motion needs projection position optimization in the LD space. Let us give some definitions:

$\text{Prj}_{\overrightarrow{AB}}$ denotes the projected operation of vector \overrightarrow{AB} , A is the

first known LD data before the missing human motion sequence, B is the last known LD data after the missing human motion sequence, and $C_i, i = 1, 2, \dots, N_{miss}$ denotes the LD data of the missing frames. According to Figure 2, we have:

$$\text{Prj}_{\overrightarrow{AB}} \overrightarrow{AC} - \frac{i \|\overrightarrow{AB}\|}{N_{miss} + 1} = 0, \tag{13}$$

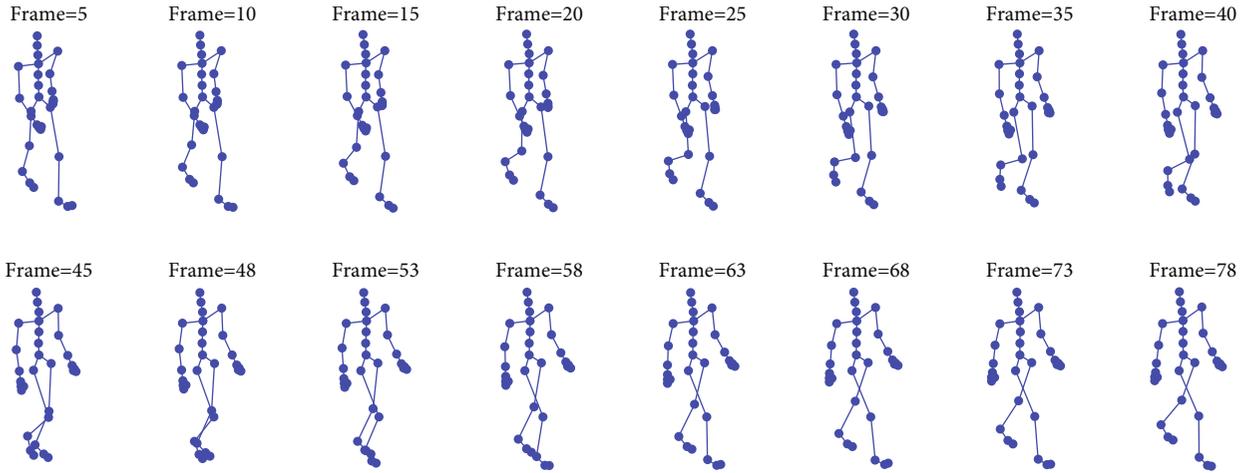
$$\left\| \overrightarrow{C'C} \right\| - \frac{c}{N_{miss} + 1} = 0. \tag{14}$$

After dimension reduction, c in Equation (14) is a preset parameter which denotes the distance between the missing dot and projection dot in Figure 2. The position of missing frames should satisfy Equation (13) and Equation (14); thus, Equation (3) can be trained optimally during the second training. Then, according to Equation (13) and Equation (14), the objective function and gradient function can be got respectively, as follows:

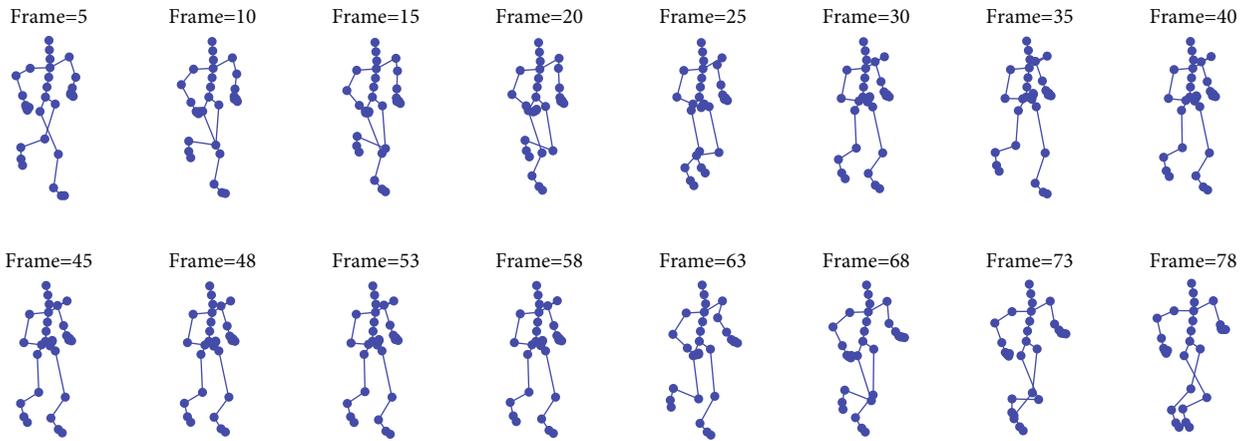
$$\begin{aligned}
\min F(\mathbf{X}_{miss}) & = \sum_{i=1}^{N_{miss}} \left(\text{Prj}_{\overrightarrow{AB}} (\mathbf{x}_i - A)^T - \frac{i \|\overrightarrow{AB}\|}{N_{miss} + 1} \right)^2 + \\
& \sum_{i=1}^{N_{miss}} \left(\left\| \mathbf{x}_i - \left(\text{Prj}_{\overrightarrow{AB}} (\mathbf{x}_i - A)^T \right) \left(\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \right) \right\| - \frac{c}{N_{miss} + 1} \right)^2, \tag{15}
\end{aligned}$$

$$\begin{aligned}
dF(\mathbf{X}_{miss}) / d\mathbf{x}_i^T & = 2 \left(\text{Prj}_{\overrightarrow{AB}} (\mathbf{x}_i - A)^T - i \|\overrightarrow{AB}\| / (N_{miss} + 1) \right) \overrightarrow{AB} + \\
& 2 \left(\left\| \mathbf{x}_i - \left(\text{Prj}_{\overrightarrow{AB}} (\mathbf{x}_i - A)^T \right) \left(\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \right) \right\| - c / (N_{miss} + 1) \right), \\
& 1 / \left\| \mathbf{x}_i - \left(\text{Prj}_{\overrightarrow{AB}} (\mathbf{x}_i - A)^T \right) \left(\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \right) \right\| \left[\left((\mathbf{x}_i - A)^T \overrightarrow{AB} \right) / \|\overrightarrow{AB}\|^2 \right], \\
& (I_{q \times q} - \mathbf{A} \mathbf{B}_R \bullet \mathbf{A} \mathbf{B}_R^T). \tag{16}
\end{aligned}$$

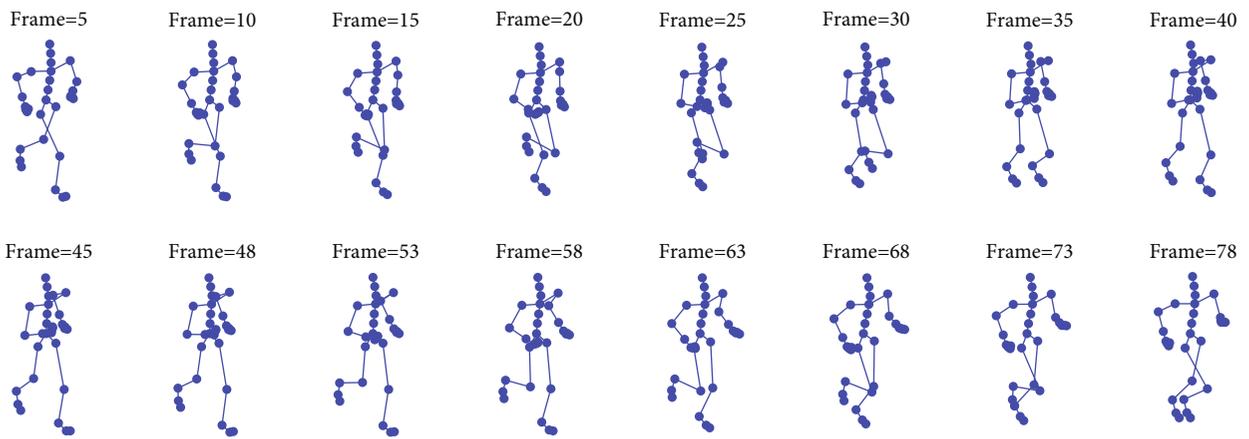
From Equation (16), $\mathbf{A} \mathbf{B}_R = [\overrightarrow{AB} \ \overrightarrow{AB} \ \overrightarrow{AB}]^T$, “ \bullet ” denotes product of the entry of matrix. The solution of Equation (15) will not be a unique solution, but any of the solutions can keep the relative position of each missing frame in the LD space during training. Thus, the second training can



(a) The samples of the input walking motion



(b) The samples of the output human running motion from IDRNPPO



(c) The samples of the output human running motion from IDRPPPO

FIGURE 3: The visual comparison of generating the human running motion between IDRPPPO and IDRNPPO.

obtain the LD data samples of missing frames. The solution of Equation (15) can be got by some traditional gradient optimization methods [23].

2.3. *The Procedure of Generating the Human Motion.* Some definitions are listed as follows: Y_I and Y_{II} are denoted as

HD data sample sequences of type I and II human motions, respectively; Y_I contains the missing frames; X_1 and X_2 are denoted as the LD data sequences of Y_I and Y_{II} , respectively; y_I' and y_{II}' are denoted as the new HD samples of type I and II human motions, respectively. Then, the procedure of generating the human motion is summarized as follows:

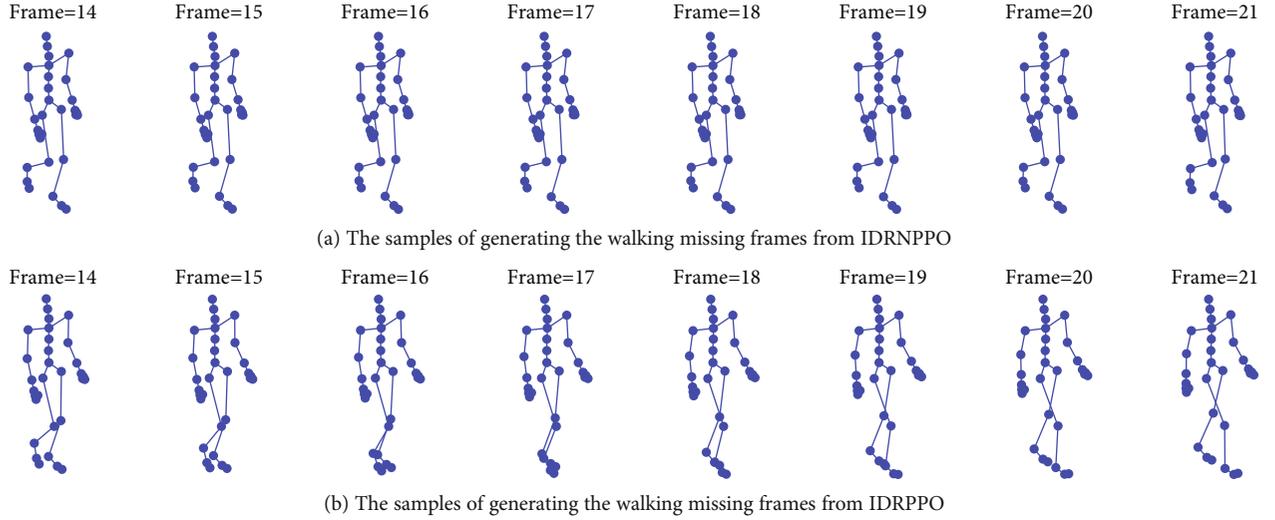


FIGURE 4: The visual comparison of generating the walking missing frames between IDRPPPO and IDRNPPO.

- (1) Equation (3) can be used to process the Y_I which is containing missing frames for dimension reduction; then, X_1 and corresponding training parameters can be obtained (the external and internal iteration numbers of this step are set to S_{11} and S_{12} , respectively)
- (2) Adopt the projection position optimization to process X_1 . It is equivalent to minimize Equation (15) by the help of Equation (16) (the iteration number of this step is set to S_{21})
- (3) The training parameters in step 1 and X_1 processed in step 2 can be took into Equation (3) for the second training, then the training parameters, the updated X_1 and mapping f_1 from X_1 to Y_I can be obtained. The missing frames in the Y_I can be generated from X_1 processed in step 2. Build the mapping g from Y_I to X_1 through Equation (5) next (the external and internal iteration numbers of building f_1 are set to S_{31} and S_{32} , respectively, the iteration numbers of building g is set to $Nk(Nk \leq N)$)
- (4) Build the mapping f_2 from X_1 to Y_{II} through Equation (3), X_1 is obtained from step 3, and X_1 is fixed during this training. After finishing the training of Equation (3), the mapping f_2 can be obtained (the external and internal iteration numbers of building f_2 are set to S_{41} and S_{42} , respectively)
- (5) When there comes y_I' , y_{II}' can be generated by the equation $y_{II}' = f_2(g(y_I'))$.

The computational complexity of the whole algorithm is depending on the iteration number of each step usually. The computational complexity is denoted by $O(\cdot)$, which is mainly described by the time frequency. If the data preprocessing and matrix calculation are without consideration, as the result of which are not the core steps of proposed algorithm, we can get the computational complexity is $O(S_{11}S_{12} + S_{21} + S_{31}S_{32} + Nk + S_{41}S_{42})$. Thus, the computational

complexity is depending on each iteration number which can reach the max iterative magnitude.

3. Experiment and Evaluation

Some heuristic algorithms and dimension reduction models cannot generate one type human motion from the other type mostly. How to optimize the projection position is the key to the generation of human motion. Thus, the algorithm using incremental dimension reduction with no projection position optimization can be called IDRNPPO. IDRNPPO and IDRPPPO will be used to generate the human motion for the experimental tests. In the experiments, the visual effect and error from the missing frames and generated poses will be the evaluation criterion of the performance. The missing frames can adopt the walking motion, and the generated motion can adopt running motion which will be generated by the walking motion. Our test environment is listed as follows:

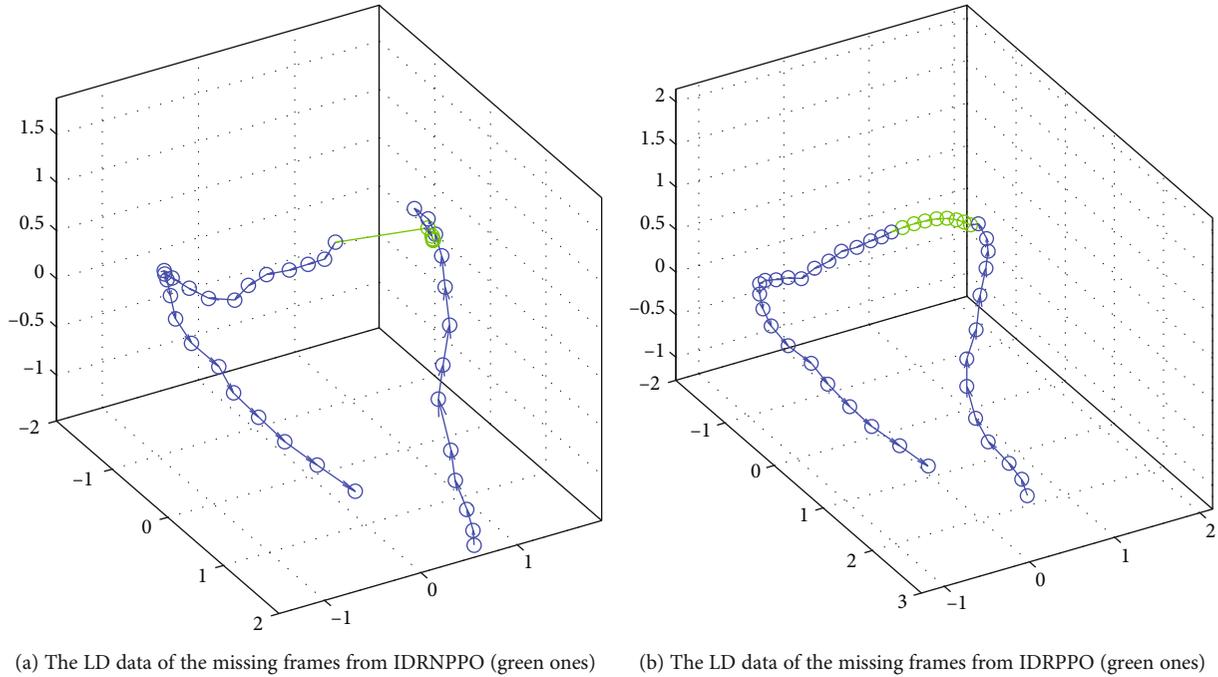
CPU: i7-9750H
RAM: 16GB
GPU: Nvidia GTX 1660Ti 6GB
HD: 1.5TB solid state disk
Software: MATLAB R2009b

3.1. The Visual Comparison. IDRPPPO and IDRNPPO are used to generate the human running motion, respectively, when the input incomplete motion is walking. The results can be seen in Figures 3 and 4.

From Figure 3, the human running poses from IDRPPPO are better than the ones from IDRNPPO in the visual effect. The 30th, 35th, 40th, 45th, 48th, 52nd, and 58th frames from the IDRNPPO are the same, which cannot constitute the smooth motion sequence to show the running process. Furthermore, from Figure 4, the missing frames in the input motion from IDRNPPO are also the same, which cannot display the missing smooth walking sequence. However, the running motion and the missing walking motion from the

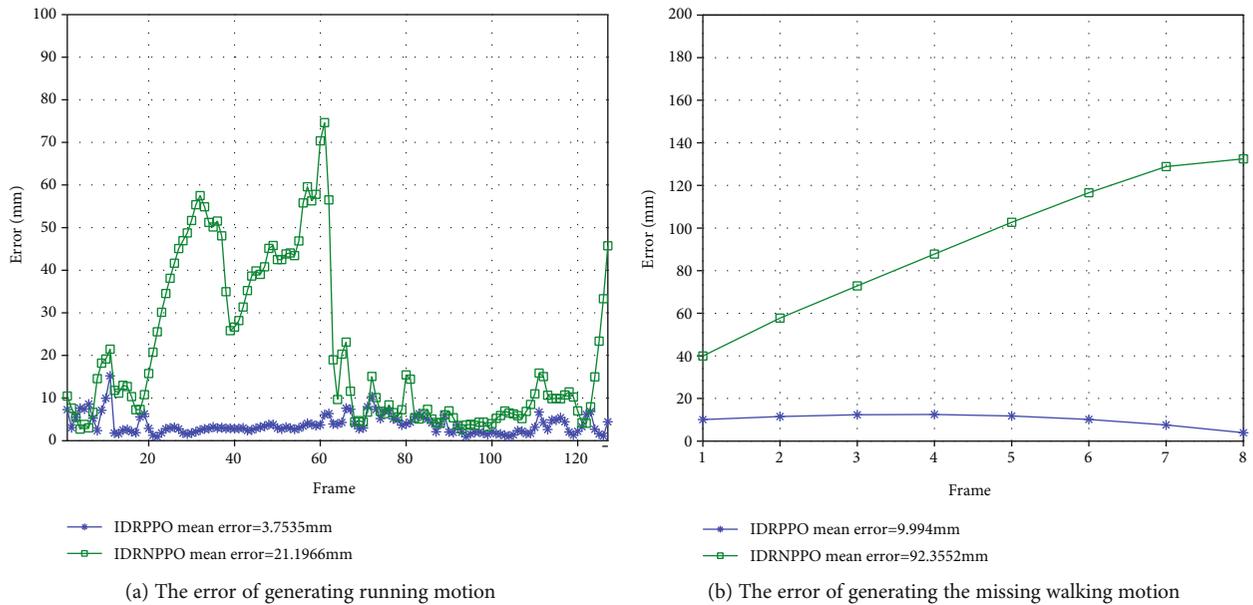
TABLE 1: The running time test (seconds).

Test type	Test content	IDRNPP0	IDRPPO
Visual test	Generating running motion	7.83	7.96
	Generating missing walking motion	2.15	2.13
Error test	Generating running motion and missing walking motion	8.28	9.51



(a) The LD data of the missing frames from IDRNPPO (green ones) (b) The LD data of the missing frames from IDRPP0 (green ones)

FIGURE 5: The LD data of the missing frames in LD space.



(a) The error of generating running motion (b) The error of generating the missing walking motion

FIGURE 6: The error comparison of IDRPP0 and IDRNPPO.

IDRPPO are very smooth, which are constituting the ideal sequences of running motion and the missing walking motion, respectively. The running time testing results are reported in Table 1. From Table 1, when generating the running motion,

the IDRNPPO consumes 7.83 seconds, and the IDRPP0 consumes 7.96 seconds; thus, the running times of both are close. Then, when generating the missing walking motion, the running times of both are also close, the IDRNPPO is 2.15

seconds, and the IDRPPPO is 2.13 seconds. From the running time test, it can be found that the IDRPPPO will not be time-consuming relatively. In Figure 5, the LD data of missing frames from IDRNPPO and IDRPPPO are obviously different, which are denoted by the green ones in Figure 5(a) and Figure 5(b), respectively. The ones of IDRNPPO are without projection position optimization. They are becoming a mess curve, which are difficult to be distinguished. On the contrary, the ones of IDRPPPO are very neat and smooth, which can constitute the missing part from the whole curve. The results of Figure 5 can also explain why the missing frames of IDRPPPO will be the smooth motion sequence in another aspect. On the whole, Figures 3, 4, and 5 can indicate IDRPPPO has better performance than IDRNPPO.

3.2. The Error of the Generation. The IDRPPPO and IDRNPPO can be seen in Figure 6, respectively. How to calculate error can be seen in [24]. From Figure 6, the errors of the human running motion and the missing walking motion from IDRPPPO are lower than IDRNPPO on the whole. It is the normal phenomenon that some frames of both have the close error in Figure 6(a), because some frames of IDRNPPO can display the running motion correctly. However, the tendency of errors can be evaluated by mean error. The mean error from IDRPPPO is lower than IDRNPPO as depicted in Figure 6(b). From Table 1, it can be found that the runtime testing results are 8.28 seconds (IDRNPPO) and 9.51 seconds (IDRPPPO), respectively. The small gap of the required running times for both will also be indicated. Finally, the results of Figure 6 can illustrate the IDRPPPO performance of generating the motion is better than the IDRNPPO again.

4. Conclusion

The IDRPPPO is proposed to obtain the 3D human motion. IDRPPPO with the GIDRM can help to learn the incomplete gait, and generate the other gait, which makes up the defects of some self-supervised or unsupervised algorithms. From the experiments, the projection position is crucial to the performance of IDRPPPO. The experimental results can reveal IDRPPPO is efficacious in making 3D human character animation, which can do great help to generating the motion cycle fast. IDRPPPO can promote the small-scale self-supervised or unsupervised learning undoubtedly. However, IDRPPPO cannot process the complex and irregular human motion samples, which will be improved in the future research. The human motion model can be replaced by a more advantaged model [25], so that the high-level multimedia product can be made by this technique.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work is supported by the Guangzhou Science and Technology Plan Project of (No. 202002030232), the University Young Creative Talent Project of Guangdong Province (No. 2016KQNCX111 and No. 2019KQNCX095), Natural Science Foundation of Guangdong Province (No. 2018A0303130169), Guangdong Province Universities and Colleges Special Innovation Projects (Natural Science) (No. 2018KTSCX163), High Education Teaching Reform Project of Guangdong Province (No. 440, approved in year 2020), Teaching Quality and Teaching Reform Project of Guangdong University of Education (No. 2019jxgg07), Key Disciplines of Network Engineering of Guangdong University of Education (No. ZD2017004), and Computer Practice Teaching Demonstration Center of Guangdong University of Education (No. 2018sfzx01).

References

- [1] E. E. Phelps, R. Wellings, F. Griffiths, C. Hutchinson, and M. Kunar, "Do medical images aid understanding and recall of medical information? An experimental study comparing the experience of viewing no image, a 2D medical image and a 3D medical image alongside a diagnosis," *Patient Education and Counseling*, vol. 100, no. 6, pp. 1120–1127, 2017.
- [2] T. Kühl, S. D. Navratil, and S. Münzer, "Animations and static pictures: the influence of prompting and time of testing," *Learning and Instruction*, vol. 58, pp. 201–209, 2018.
- [3] A. Dowsett and M. Jackson, "The effect of violence and competition within video games on aggression," *Computers in Human Behavior*, vol. 99, pp. 22–27, 2019.
- [4] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1617–1632, 2017.
- [5] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 648–660, 2018.
- [6] J. Gall, B. Rosenhahn, T. Brox, and H. P. Seidel, "Optimization and filtering for human motion capture," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 75–92, 2010.
- [7] W. Y. Li and J. F. Sun, "Human motion estimation based on gaussian incremental dimension reduction and manifold boltzmann optimization," *Acta Electronica Sinica*, vol. 45, no. 12, pp. 3060–3069, 2017.
- [8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [9] J. Xinwei, G. Junbin, W. Tianjiang, and Z. Lihong, "Supervised latent linear Gaussian process latent variable model for dimensionality reduction," *IEEE Transactions on Systems Man & Cybernetics Part B*, vol. 42, no. 6, pp. 1620–1632, 2012.
- [10] J. B. Tenenbaum, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [11] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal Mach. Learn. Research*, vol. 6, pp. 1783–1816, 2005.

- [12] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.
- [13] W. Li, "Manifold latent probabilistic optimization for human motion fitting based on orthogonal subspace searching," *Journal of Information and Computational Science*, vol. 11, no. 15, pp. 5357–5365, 2014.
- [14] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "MonoCap: monocular human motion capture using a CNN coupled with a geometric prior," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 41, no. 4, pp. 901–914, 2019.
- [15] A. Zanchetta and S. Zecchetto, "Wind direction retrieval from Sentinel-1 SAR images using ResNet," *Remote Sensing of Environment*, vol. 253, p. 112178, 2021.
- [16] K. M. Hosny, M. A. Kassem, and M. M. Fouad, "Classification of skin lesions into seven classes using transfer learning with AlexNet," *Journal of Digital Imaging*, vol. 33, no. 5, pp. 1325–1334, 2020.
- [17] X. Xu, M. Xie, P. Miao et al., "Perceptual-aware sketch simplification based on integrated VGG layers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 178–189, 2021.
- [18] M. Hassanpour and H. Malek, "Learning document image features with Squeeze Net convolutional neural network," *International Journal of Engineering*, vol. 33, no. 7, 2020.
- [19] Z. Tang, W. Jiang, Z. Zhang, M. Zhao, L. Zhang, and M. Wang, "DenseNet with up-sampling block for recognizing texts in images," *Neural Computing and Applications*, vol. 32, no. 11, pp. 7553–7561, 2020.
- [20] D. McNeely-White, J. R. Beveridge, and B. A. Draper, "Inception and ResNet features are (almost) equivalent," *Cognitive Systems Research*, vol. 59, pp. 312–318, 2020.
- [21] D. Yuan, X. Chang, P. Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2021.
- [22] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, "Compound rank-k projections for bilinear analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1502–1513, 2016.
- [23] X. Liuqing and Z. Shipeng, *Practical Optimization Method Shanghai*, Shanghai Jiaotong University Press, 2000.
- [24] L. Sigal, A. O. Balan, and M. J. Black, *Humaneva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion*, Brown University, Providence, USA, 2006.
- [25] L. Yebin, G. Juergen, S. Carsten, D. Qionghai, S. Hans-Peter, and T. Christian, "Markerless motion capture of multiple characters using multiview image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 11, pp. 2720–2735, 2013.