Article

An Efficient Dynamic Regulated Fuzzy Neural Network for Human Motion Retrieval and Analysis

Xin Huang , Yuanping Zhu and Shuqin Wang

College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China; zhuyuanping@tjnu.edu.cn (Y.Z.); wangshuqin@tjnu.edu.cn (S.W.)

* Correspondence: huxpiero@tjnu.edu.cn

Abstract: Human motion retrieval and analysis is a useful means of activity recognition to 3D human bodies. An efficient method is proposed to estimate human motion by using symmetric joint points and limb features of various limb parts based on regression task. We primarily obtain the 3D coordinates of symmetric joint points based on the located waist and hip points. By introducing three critical feature points on torso and symmetric joint points' matching on motion video sequences, the 3D coordinates of symmetric joint points and its asymmetric limb features will not be affected by shading and interference of limb on different postures. With the asymmetric limb features of various human parts, a dynamic regulated Fuzzy neural network (DRFNN) is proposed to estimate human motion for different asymmetric postures using learning algorithm of network parameters and weights. Finally, human sequential actions corresponding to different asymmetric postures are presented according to the best retrieval results by DRFNN based on 3D human action database. Experiments show that compared with the traditional adaptive self-organizing fuzzy neural network (SOFNN) model, the proposed algorithm has higher estimation accuracy and better presentation results compared with the existing human motion analysis algorithms.

Citation: Huang, X.; Zhu, Y.; Wang, S. An Efficient Dynamic Regulated Fuzzy Neural Network for Human Motion Retrieval and Analysis. *Symmetry* **2021**, *13*, 1317. <https://doi.org/10.3390/sym13081317>

Academic Editor: Basil Papadopoulos

Received: 24 May 2021

Accepted: 14 July 2021

Published: 22 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: human motion retrieval; human motion analysis; limb features; joint points; fuzzy neural network

1. Introduction

With the rapid development of artificial intelligence and computer technology, human beings look forward to obtaining and dealing with more information about themselves, such as ergonomics, human limb motion, virtual reality, etc. Although the high-resolution human point clouds can be obtained by 3D laser scanner, it is difficult to use for the practical application due to processing time and noise data. In order to solve the above problems, video sequences and the reduced 3D human models are applied by many scholars to analyze human motion owing to the advantage of low cost and less limit. Furthermore, human motion retrieval is an effective method to represent the posture of target human body based on video in different areas, such as medical care, sports science and recovery. Although different applications have their special requirements of human motion analysis, accuracy and instantaneity almost become the evaluation criteria on performance of the existing algorithms.

It is difficult to retrieve and analyze human motion data, because different people have various postures on video sequences. Owing to the similar properties with videos and images, human motion data is processed by some multimedia retrieval methods. In some aspects, retrieval and analysis of human motion heavily relies on the feature representation. Huang et al. [1] developed a method based on deep supervised hashing and multi-labels for image retrieval. However, it is difficult to choose precise labels for the given images due to high ambiguity. To overcome the shortcomings, the high-level and low-level morphological and kinematic features using motion capture sequence were proposed [2]. Chen et al. [3]

presented a boosting method for content-based human motion retrieval in order to further improve the performance.

To describe local motion accurately, motion features based on frames are developed. Based on the extracted features on key frames, Bao et al. [4] presented a retrieval method of human motion data. In order to better description of 3D human motion, Ramezani et al. [5] retrieved human action using a low-complexity representation based on the extracted local features. In addition, some statistical learning methods are applied for human motion data. Xiao et al. [6] proposed a human motion retrieval approach based on graph model and statistical learning of motion frame. Similarly, a novel graph-based method was proposed by Li et al. [7] to realize real time 3D human motion retrieval. Xiao et al. [8] proposed an approach for retrieving human motion based on statistical learning and Bayesian fusion. Wang et al. [9] achieved human motion retrieval by thumbnails of key frames on motion capture data based on statistical K-means. However, it is hard to decide which feature and retrieval model is the most suitable for a given application. To cope with the problem, Valcik et al. [10] proposed human motion model evaluator for assessing similarity models with respect to the target application. In order to overcome the limitation of camera view, Wang et al. [11] presented a multi-view feature selection method for human motion retrieval. Moreover, temporal adjacent bag of words and dictionary learning can also be used for human motion retrieval based on capture data [12].

In order to retrieve 3D human motion accurately, Slama et al. [13] presented a 3D human motion analysis framework for 3D shape representation and similarity in video sequences. In recent research, deep learning method is also used to retrieve and analyze human motion using some motion capture data. Ren et al. [14] proposed a video-based human data retrieval approach, and convolutional neural network is applied for extracting motion feature. Furthermore, in order to evaluate accurate information of the actions and effectiveness of models, the complexity of pattern of body motions on different scales was calculated to describe human action [15]. In addition, Tang et al. [16] presented integrated framework of human motion retrieval by sketching several key poses. However, the above methods often required objective frames from video sequences before human motion retrieval.

Recently, a technique called sketch-based searching has been proposed in motion retrieval. Li et al. [17] presented a method for sketch-based 3D model retrieval combining global and local features from 2D views of 3D models. Based on adaptive view clustering and semantic information, Li et al. [18] proposed a novel sketch-based 3D model retrieval framework. However, it requires special experience for using human figures and poses.

A large number of research efforts on 3D human motion retrieval and analysis exist. However, the existing methods need to be perfected in several aspects. For instance, an expensive measurement system is needed, the selection and extraction on 3D human motion features are difficult to deal with, and obtained human postures have a lot of local deviation compared with the accurate poses. The precise extraction on joint points and motion features of symmetric human body are required. However, it is difficult to use an efficient algorithm for obtaining motion features due to various human bodies and poses. Based on the 3D human model database, motion retrieval is a good idea to analyze human motion using video sequences.

In this paper, we introduce limb features and propose a dynamic regulated Fuzzy neural network (DRFNN) to retrieve and analyze human motion for different postures based on regression task. The experimental models are provided by free 3D models database [19] and generated by some algorithms [20,21]. Compared with the existing work, the main contributions include: (1) Three critical feature points on torso and joint points' matching are used for obtaining 3D coordinates. It can overcome the occlusion and interference of limb parts on different poses. (2) Limb features on various limb parts are applied for estimating human motion for different postures by the DRFNN. (3) The regulated strategy of network structures and parameters is introduced for evaluating human motion when the neural network is trained.

Furthermore, in order to prove the advantages of the proposed DRFNN model, a comparative experiment with SOFNN model is executed. In our study, the comparative SOFNN model has been used in many fields in recent years. Sabahi et al. [22] presented novel self-organizing fuzzy neural network based on input-output mapping and validity degrees. Then, a controller is applied for the efficiency of model. Zhang et al. [23] proposed a multi-variable direct self-organizing fuzzy network for modeling wastewater treatment process. Zhou et al. [24] presented self-organizing fuzzy network with hierarchical pruning scheme for modeling nonlinear systems in industrial processes. Therefore, the proposed DRFNN and SOFNN were compared in the experiment part.

This paper is organized as follows. In Section 2, 3D coordinates of joint points are obtained by three critical feature points on torso and joint points matching on various human motion images using a divided skeleton model. In Section 3, human motion models for different postures are retrieved by the extracted limb features and the algorithm of DRFNN using the regulated strategy of network structures and parameters during the network training. In Section 4, the proposed algorithm is applied for predicting motion sequence gestures based on regression task and the experimental results show that the new human motion retrieval and analysis method overcome the influence on different human postures, and the method gives evidence of low cost and effectiveness. Finally, Section 5 summarizes the whole paper.

2. The Extraction of Human Limb Features

The accurate extraction on 3D coordinates of human skeleton feature points is critical for 3D human motion estimation. Our method obtains the coordinates of human joints located on the different human motion frames using the located 2D human joints. Based on this, different human limb features can be obtained accurately.

The estimation on 3D coordinates of human joints is divided into three parts: Firstly, the joint points are defined based on biological structure of symmetric human body and connection mode of different human parts. Secondly, various joint points on human motion image can be obtained using classical method of image processing. Furthermore, photographic focal length is obtained by the three limb parts connection model [25]. Therefore, the 3D coordinates of different joint points are obtained by matching the same joint points located on different human motion images.

2.1. Human Skeleton Model and Divided Limb Part

Tree stick structure [26] is applied for representing symmetric human model. That is, various joint points and rigid limb parts between them [27] are used for indicating the complete skeleton model (see Figure 1a).

All of the models were selected from a free 3D models database when 3D human motion retrieval and analysis are considered. Figure 1b shows several virtual human models with different shapes provided from free human models database. Furthermore, various joint points of skeleton on 3D human models are located combining limb division with the right proportions of symmetric human body. Figure 2 shows another human model selected from database and the located human skeleton on various views by method in [28]. The major work of the paper is to retrieve and analyze the 3D human motion. Therefore, the human torso is considered as a whole and the human limbs are divided into eight parts: left upper and forearms, right upper and forearms, left and right thighs, left and right calves. These limb parts are considered as rigid body when 3D human motion is analyzed.

2.2. Location on Skeleton Feature Points on Symmetric Target Human Body

The location of joint points on target human motion images is critical for 3D human motion estimation. In order to extract the limb features of human model accurately, the positions of different joint points are located by pasting labels when images of human

motion are obtained. In the paper, the method of manual location is applied for location on joint points on symmetric target human body.

The manual location method is introduced to locate various joint points on symmetric target human body in order to extract limb features for 3D human motion estimation. The pasted labels are used for identifying joint points on symmetric target human body before the human motion sequence images are captured. In other words, the candidate pixels with the color values of the labels are used for locating joint points using color histogram.

(a) Human skeleton model

(b) Several models from 3D human models database

Figure 1. Human skeleton and 3D virtual models with different shapes.

Figure 2. 3D human model and its skeleton on various views.

2.3. Estimation on 3D Coordinates of Human Skeleton FEATURE points

Data points in a 3D space are mapped into 2D projected plane when the monocular camera is applied for capturing various human poses, and depth values of data will be lost due to the transformation. That is, restoration of depth information of data points is critical for 3D human motion estimation.

The perspective projection is used to the imaging process, and Figure 3 shows the projection principle. To obtain the 3D coordinates of joint points accurately, the focal length of the camera needs to be obtained. To improve the immediacy of the system, the method proposed in [22] is used for calculating focal length. In the method, the estimated length of limb parts and their corresponding projection coordinates based on human body images are applied for calculating the focal length of the camera.

Figure 3. The perspective projection model.

We calculate the 3D coordinates of joint points by matching on waist and left and right hip points using human sequence images based on obtained positions of joint points and the whole human skeleton. In the first frame of motion sequential image, symmetric target human body located in the standard standing posture, and the 3D coordinates of the above three points in the first image of human motion are calculated using the method of parallel

projection. Subsequently, the 3D coordinates of joint points in the following human motion images need to be determined. In the paper, 3D coordinates of joint points are estimated based on matching on the same joint points on various motion images.

In the human motion sequential images, waist point and left and right hip points are treated as the three critical feature points. 3D coordinates of three critical feature points in the different frame of motion images can be obtained by the iterative solutions based on Newton Method. Subsequently, 3D coordinates of other joint points are calculated by the method of iteration using length sizes of different skeleton segments. Assume that the 3D coordinates of one joint point are known, and the unknown coordinates of another joint point which connect with each other can be calculated by the following equation:

$$\begin{aligned} & \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} = L_{(i,i+1)} \\ \Rightarrow & [s_i \cdot u_i - (s_i + \frac{\Delta z_i}{f}) \cdot u_{i+1}]^2 + [s_i \cdot v_i - (s_i + \frac{\Delta z_i}{f}) \cdot v_{i+1}]^2 + (\Delta z_i)^2 = L_{(i,i+1)}^2 \end{aligned} \quad (1)$$

where (x_i, y_i, z_i) is the 3D coordinates of point g_i . Δz_i is the difference on depth coordinate of g_i and g_{i+1} . In the whole projected plane, u_i and v_i are width and height coordinates for point g_i . The difference of depth information Δz_i between points g_i and g_{i+1} is calculated using Equation (1) because s_i , u_i , u_{i+1} , v_i , v_{i+1} , f , and $L_{(i,i+1)}$ are known. Thus, the 3D coordinates of point g_{i+1} are obtained.

3. Human Motion Model Estimation for Different Postures

A motion model of human limbs is estimated by the trained dynamic regulated fuzzy neural network using different limb features when various frames of human motion images are considered. Based on human physiological structure, human limb feature is defined as the angle between one part with another connected part. Limb features of different limb parts are calculated based on the obtained 3D coordinates of joint points. Therefore, 3D coordinates of joint points are firstly calculated by the correspondence between the known waist and hip points and other unknown joint points. Then, a motion model of human limbs is estimated by the trained parameters and values of DRFNN using the extracted limb features for different postures. Subsequently, human sequential actions corresponding to different postures are presented based on the best evaluation results by DRFNN. The estimation and presentation process on human limb motion is shown in Figure 4.

3.1. The Proposed Dynamic Regulated Fuzzy Neural Network

Due to good applicability and without manual intervention, an improved dynamic regulated fuzzy neural network (DRFNN) is applied for estimating the motion model of human limbs. The learning algorithm of DRFNN is based on structure and parameter learning algorithm which constructs and regulates the FNN automatically and dynamically. In this section, the learning process of the DRFNN, including structure regulation and parameter learning, is presented. In structure regulation, FNN with high accuracy and reasonable structure is constructed. The neurons are generated and regulated dynamically during the learning process. In parameter learning, the corresponding strategy is used to adjust parameters and weights of the DRFNN.

Specifically, we firstly define the system error. In the process of training, the threshold is adjusted dynamically to control the deviation of neurons. At the same time, we redefine the modification, pruning and supplementary rules of fuzzy rules to make the structure of fuzzy neural network more reasonable and efficient. Finally, we supplement and explain the weight adjustment rules of fuzzy neural network, so that compared with traditional networks, network DRFNN can obtain reasonable results using fewer iterations.

Figure 4. The estimation process on human motion model for different postures.

3.1.1. System Errors

With regard to sample data $X^i (i = 1, 2, \dots, n)$, the output error of the whole DRFNN system is an important observation to determine whether the structures and parameters of network should be adjusted or not. Consider the i -th training data (x^i, d^i) where x^i is the input vector and d^i is the desired output. The actual output of DRFNN with the existing structures and parameters of network is denoted by y^i .

Based on the extracted features of human limbs and joint points, the proposed DRFNN deep learning model is used to retrieve and recognize the 3D human posture using the established 3D virtual human models database. That is, our learning task is a regression task. Furthermore, the coefficient of derivative of L1 is larger than that of L2 measure near to $x = 0$, which leads to the dominant role of L1 measure, so we use L1 measure to estimate the system error. Thus, the error is defined as follows:

$$e^i = |y^i - d^i| \quad (2)$$

T_e is a predefined error threshold. The system should adjust the network parameter settings or add new neurons if e^i is greater than T_e . The term T_e changes during the learning process of network as follows:

$$T_{e_{new}} = (1 - \eta) \cdot T_{e_{old}} + \eta \cdot \bar{e} \quad (3)$$

where $T_{e_{old}}$ and $T_{e_{new}}$ are the error thresholds of each sample before and after the learning process, respectively. The term m is the number of samples which have already trained and is defined as follows:

$$\bar{e} = \frac{1}{m} \sum_{i=1}^m e^i.$$

η is the change on error threshold can be described as follows: $\eta = \frac{m}{n}$, where n is the number of all the sample data.

3.1.2. Deviation of Neurons

Every neuron in EBF layer of fuzzy network represents one antecedent of fuzzy rule. Furthermore, the ϵ -completeness of fuzzy rules is for each sample data within the operating

range, there exists at least one fuzzy rule such that the match degree is greater than ε . It indicates that the existing network can cover with the current sample well. Consider the sample X^i , fuzzy rule of j -th neuron is defined as follows:

$$\varphi_j^i(x_1^i, x_2^i, \dots, x_r^i) = \exp[-fd^2(k)] \quad (4)$$

where $fd(k)$ is a M-distance defined as follows: $fd(k) = \sqrt{(X^i - C_j)^T \sum_k^{-1} (X^i - C_j)}$, where $X^i = (x_1^i, x_2^i, \dots, x_r^i)^T \in \mathbb{R}^r$, $C_j = (c_{1j}, c_{2j}, \dots, c_{rj})^T \in \mathbb{R}^r$ and $\sum_k^{-1}$ is calculated as follows:

$$\sum_k^{-1} = \begin{bmatrix} \frac{1}{2\sigma_{1j}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{2\sigma_{2j}^2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{2\sigma_{rj}^2} \end{bmatrix} \quad (5)$$

Therefore, the outputs of each sample X^i and center $C_j (j = 1, 2, \dots, u)$ of the existing EBF neurons are calculated according to the above equations. We define that $J = \operatorname{argmax}_{1 \leq j \leq u} (\varphi_j^i(x_1^i, x_2^i, \dots, x_r^i) - T_d)$.

If the condition $(\varphi_j^i - T_d)_{\max} = \varphi_J^i - T_d < 0$ is satisfied, it implies that the existing network is not satisfied with ε -completeness. Thus, a new neuron needs to be supplemented in EBF layer to cover with the sample X^i .

The updating method of the threshold T_d is calculated as follows during the learning process.

$$T_{d_{\text{new}}} = (1 - \eta) \cdot T_{d_{\text{old}}} + \eta \cdot \overline{\varphi^i} \quad (6)$$

where $T_{d_{\text{old}}}$ and $T_{d_{\text{new}}}$ are the threshold of each sample before and after the learning process, respectively. The term m is the number of samples which have already trained and is defined as follows:

$$\overline{\varphi^i} = \frac{1}{m} \sum_{j=1}^m \varphi_j^i.$$

η is the change on threshold can be described as follows: $\eta = \frac{m}{n}$, where n is the number of all the sample data.

The initial predefined threshold cannot precisely control the system error and antecedent deviation of neurons. Therefore, threshold setting criterion is proposed to overcome the above shortcomings by using threshold record of the learned samples. T_e and T_d will gradually approach the optimal thresholds during the learning process, so that the accuracy of network learning will be enhanced.

3.1.3. Criteria of Fuzzy-Rule Modification

For all the candidate neurons which satisfy with the updated condition, the reduction rate of network error is used for obtaining the neurons which are needed to be modified. The method will make the reasonable modification based on the sensitivity calculation of fuzzy rules on EBF neurons.

Suppose for n observations of network, the network output can be written as a linear regression model in the following form:

$$d(n) = \sum_{i=1}^m \phi_i(n) w_i + \varepsilon(n) \quad (7)$$

where $d(n)$ and $\varepsilon(n)$ are the desired output and error vector. w_i is i -th linear parameter, and $m = u \cdot (r + 1)$ is the dimension of parameter, where u and r are number of neurons and the dimension of input vectors, respectively. Therefore, (7) can be written as the matrix form:

$$D = \psi W + E \quad (8)$$

where $D, E \in \mathbb{R}^n$. By the orthogonal transformation, regression matrix Ψ can be represented as:

$$\psi = TA \quad (9)$$

where T is an $n \times m$ matrix with orthogonal columns, $t_i^T t_j = \delta_{ij} = \begin{cases} 1, i=j \\ 0, i \neq j \end{cases}$, $i, j = 1, 2, \dots, m$. A is a $m \times m$ upper triangular matrix. Substituting (8) into (9), we obtain

$$D = TAW + E = TQ + E \quad (10)$$

According to the orthogonal least squares solution, Q is given by: $Q = (T^T T)^{-1} T^T D$, its component can be described as: $q_i = \frac{t_i^T D}{t_i^T t_i} = t_i^T D, 1 \leq i \leq v$. Moreover, the error vector ER of neuron is given by:

$$er_i = \frac{q_i^2 t_i^T t_i}{D^T D}, 1 \leq i \leq v.$$

Substituting q_i into er_i yields, $er_i = \frac{t_i^T D t_i^T D}{D^T D}, 1 \leq i \leq v$.

Define the matrix $ER = (e_1, e_2, \dots, e_u) \in \mathbb{R}^{(r+1) \times u}$ whose component k of j -th element e_j is obtained from the corresponding value $er_{(r+1) \times (j-1) + k}$:

$$e_{jk} = \frac{er_{(r+1) \times (j-1) + k} - er_{j_{\min}}}{er_{j_{\max}} - er_{j_{\min}}}, j = 1, 2, \dots, u; k = 1, 2, \dots, r + 1$$

where $er_{j_{\min}}$ and $er_{j_{\max}}$ are the minimum and maximum of component $r + 1$ in e_j . Then, the norm $\|e_j\|$ of vector e_j is calculated. Define $j^* = \underset{1 \leq j \leq u}{\operatorname{argmax}} (\|e_j\|)$, and the term j^* is selected as the neuron which is modified.

3.1.4. Criteria of Fuzzy-Rule Pruning

In the learning process of network, the learning velocity will be enhanced greatly if some inefficient neurons are removed. In the paper, output disturbances squared expectation $E[(\Delta y)^2]$ of network is applied for obtaining the neurons which needed to be pruned. Furthermore, the output of whole network can be represented as:

$$y(x) = \sum_{j=1}^u f_j = \frac{\sum_{j=1}^u w_j \exp(-\sum_{i=1}^r (x_i - c_{ij})^2 / 2\sigma_{ij}^2)}{\sum_{k=1}^u \varphi_k}$$

where $(x_1, x_2, \dots, x_r)$ is input vector of sample, w_j represents the weights of hidden and output layer, u is the number of the existed neurons. $(c_{1j}, c_{2j}, \dots, c_{rj})$ and $(\sigma_{1j}, \sigma_{2j}, \dots, \sigma_{rj})$ are the center and width of j -th neuron in EBF layer, φ_k is the output of k -th neuron.

Suppose $(x_1, x_2, \dots, x_r)$ is a random vector, and the relationship of mutual independence among the random variables of several dimensions is established. c_{x_i} and $\sigma_{x_i}^2$ are

the expectation and variance on variable x_i , and its input disturbance Δx_i satisfy with the condition: $\Delta x_i \sim N(0, \sigma_{\Delta x_i}^2)$. Therefore, we can obtain:

$$\begin{aligned} E[(\Delta y)^2] &= E \left(\left[\sum_{j=1}^u w_j \exp\left(\frac{t_j^*}{-2\sigma_{ij}^2}\right) / \sum_{k=1}^u \varphi_k - \sum_{j=1}^u w_j \exp\left(\frac{t_j}{-2\sigma_{ij}^2}\right) / \sum_{k=1}^u \varphi_k \right]^2 \right) \\ &= 1 / \left(\sum_{k=1}^u \varphi_k \right)^2 \left\{ \sum_{j=1}^u (w_j)^2 \exp\left(\frac{\text{var}(t_j)}{2\sigma_{ij}^4} - \frac{E(t_j)}{\sigma_{ij}^2}\right) \right. \\ &\quad - 2 \sum_{j=1}^u (w_j)^2 \exp\left(\frac{\text{var}(t_j^* + t_j)}{8\sigma_{ij}^4} \right. \\ &\quad \left. \left. - \frac{E(t_j^* + t_j)}{2\sigma_{ij}^2}\right) + \sum_{j=1}^u (w_j)^2 \exp\left(\frac{\text{var}(t_j^*)}{2\sigma_{ij}^4} - \frac{E(t_j^*)}{\sigma_{ij}^2}\right) \right\} \end{aligned} \quad (11)$$

where,

$$\begin{aligned} t_j^* &= \sum_{i=1}^r (x_i + \Delta x_i - c_{ij})^2, \quad t_j = \sum_{i=1}^r (x_i - c_{ij})^2 \\ E(t_j) &= \sum_{i=1}^r (\sigma_{x_i}^2 + (c_{x_i} - c_{ij})^2) \\ \text{var}(t_j) &= \sum_{i=1}^r \left(E[(x_i - c_{x_i})^4] - \sigma_{x_i}^4 + 4E[(x_i - c_{x_i})^3] (c_{x_i} - c_{ij}) + 4\sigma_{x_i}^2 (c_{x_i} - c_{ij})^2 \right) \\ E(t_j^*) &= E(t_j) + \sum_{i=1}^r \sigma_{\Delta x_i}^2 \\ \text{var}(t_j^*) &= \text{var}(t_j) + 4 \sum_{i=1}^r (\sigma_{x_i}^2 \sigma_{\Delta x_i}^2 + \sigma_{\Delta x_i}^2 (c_{x_i} - c_{ij})^2 + 0.2\sigma_{\Delta x_i}^4). \end{aligned}$$

According to the candidate column of neurons, the expectation $E[(\Delta y)^2]$ on output disturbance variable can be calculated using (11) when different neurons are removed, respectively. If the condition $E[(\Delta y)^2] < T_y$ is satisfied, it implies that the current neuron causes little impact to the existing network and this neuron should be removed. Here, T_y is a predefined threshold.

3.1.5. Criteria of Fuzzy-Rule Supplementation

Suppose for u neurons in the existing network, the fuzzy-rules of whole network should be adjusted when i -th sample data $X^i (i = 1, 2, \dots, n)$ is considered. Then, X^i is mapped into Gaussian function, the corresponding input variable for which is $x_k^i (k = 1, 2, \dots, r)$.

Here, we define a $n \times u$ matrix $E = \begin{pmatrix} e_{11} & \cdots & e_{1u} \\ \vdots & \ddots & \vdots \\ e_{n1} & \cdots & e_{nu} \end{pmatrix}$, where n and u are the numbers of learned samples and neurons of existing network, and

$$e_{ij} = \exp \left[- \sum_{k=1}^r \frac{(x_k^i - c_{kj})^2}{2\sigma_{kj}^2} \right] - \varepsilon, \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, u),$$

where ε is the antecedent threshold of the corresponding EBF neurons.

(1) Scanning each row of matrix E . If all of the elements in row i satisfy with the following conditions: $e_{ik} < 0 (k = 1, 2, \dots, u)$ when the i -th training sample X^i is considered, this implies that the existing network is not satisfied with ε -completeness and a new EBF neuron should be considered. The criterion proposed by 3.2.2 is used to decide whether or not to supplement the neurons. The center C and width σ of the neuron can be obtained as follows.

Suppose for the supplemented $(u + 1)$ -th neuron, we define the distance between the sample data x_k^i and center set C_j^i as follows when j -th neuron is considered:

$$dis_j(k) = x_k^i - C_j^i(k), k = 1, 2, \dots, r,$$

where r is the vector dimension of input sample, $C_j^i \in \{c_{1j}^i, c_{2j}^i, \dots, c_{rj}^i\}$.

Find $k_{jmin} = \underset{k=1,2,\dots,r}{\operatorname{argmin}} (dis_j(k))$, and then the shortest distance of i -th sample on the existing network can be described as follows:

$$Dis^i = \begin{bmatrix} dis_1(k_{1min}) \\ dis_2(k_{2min}) \\ \vdots \\ dis_u(k_{umin}) \end{bmatrix} = \begin{bmatrix} dis_1 \left[\underset{k=1,2,\dots,r}{\operatorname{argmin}} (dis_1(k)) \right] \\ dis_2 \left[\underset{k=1,2,\dots,r}{\operatorname{argmin}} (dis_2(k)) \right] \\ \vdots \\ dis_u \left[\underset{k=1,2,\dots,r}{\operatorname{argmin}} (dis_u(k)) \right] \end{bmatrix}.$$

The center and width matrix of the supplemented $(u + 1)$ -th neuron are described as follows:

$$C_{u+1} = [c_{1,u+1} \ c_{2,u+1} \ \dots \ c_{r,u+1}]^T, \sigma_{u+1} = [\sigma_{1,u+1} \ \sigma_{2,u+1} \ \dots \ \sigma_{r,u+1}]^T,$$

the neuron's center of member function is:

$$C_{k,u+1} = dis_{J^*}(k_{J^*min}^*), k = 1, 2, \dots, r \quad (12)$$

where $J^* = \underset{j=1,2,\dots,u}{\operatorname{argmin}} (dis_j(k_{jmin}))$.

In order to ensure the ε -completeness of fuzzy rule, the output of $(u + 1)$ -th neuron on i -th sample satisfies with: $\varphi_{u+1}^i > \varepsilon$. Furthermore, the output can be described as follows:

$$\exp\left[-\sum_{t=1}^r \frac{(x_t^i - c_{t,u+1})^2}{2(\sigma_{t,u+1})^2}\right] - \varepsilon > 0, \text{ if } \sigma_{1,u+1} = \sigma_{2,u+1} = \dots = \sigma_{r,u+1},$$

the neuron's width is as follows:

$$\sigma_{t,u+1} \in \left(0, \sqrt{\frac{r \cdot (x_t^i - c_{t,u+1})^2}{2 \cdot \ln \frac{1}{\varepsilon}}}\right), t = 1, 2, \dots, r \quad (13)$$

(2) Scanning each column of matrix E. If all of the elements in column j satisfy with the following conditions: $e_{kj} < 0 (k = 1, 2, \dots, n)$, the column j needs to be recorded. As for all the selected columns, the criterion proposed by 3.2.3 is used to decide to the neurons which need to be modified. The center C and width σ of the neuron can be obtained as follows.

According to the i -th sample, we define the distance between the sample data x_k^i and center set C_j^i as follows when j -th neuron is considered to be modified.

$$er(k) = |x_k^i - C_j^i(k)|, k = 1, 2, \dots, r,$$

where r is the vector dimension of input sample, $C_j^i \in \{c_{1j}^i, c_{2j}^i, \dots, c_{rj}^i\}$.

Find $k^* = \underset{k=1,2,\dots,r}{\operatorname{argmin}}(er(k))$, and then the j -th neuron's center of member function is as follows:

$$c_{kj}^i = c_{k^*j}^i, k = 1, 2, \dots, r \quad (14)$$

In order to ensure the ε -completeness of fuzzy rule, the difference between the actual output φ_j^i and target output ψ_j^i is defined as follows when j -th neuron on i -th sample is considered: $D(j) = \frac{1}{2}(\varphi_j^i - \psi_j^i)^2$.

Therefore, the width σ_{kj}^i should be adjusted as follows owing to the new center $c_{k^*j}^i$ of j -th neuron: $\Delta\sigma_{kj}^i = \sigma_{kj}^{i\text{new}} - \sigma_{kj}^{i\text{old}} = \frac{\partial\varphi_j^i}{\partial\sigma_{kj}^i} = \frac{\varphi_j^i \cdot (x_t^i - c_{k^*j}^i)^2}{(\sigma_{kj}^{i\text{old}})^3}, k = 1, 2, \dots, r$.

Then, the j -th neuron's width is as follows:

$$\sigma_{kj}^{i\text{new}} = \frac{(\sigma_{kj}^{i\text{old}})^4 + \varphi_j^i \cdot (x_t^i - c_{k^*j}^i)^2}{(\sigma_{kj}^{i\text{old}})^3}, k = 1, 2, \dots, r \quad (15)$$

3.1.6. The Adjustment of Network Weights

The adjustment of network weights is essential for the whole learning process of network based on the obtained structure parameters of network. As for the sample data $X^i (i = 1, 2, \dots, n)$, the output error of the network system is an important observation and will directly determine the learning of network. In this paper, an improved system error method is used for weight learning.

We assume that the preceding $t - 1$ sample data have been trained, and the current training sample is (x^t, d^t) , where x^i and d^i are the input vector and desired output, respectively. y^i is the actual output of network. Therefore, the system error is defined as follows: $e^t = \sum_{i=1}^t \frac{1}{2}(y^i - d^i)^2$. Thus, the updating method of the related weight can be described by the following equation.

$$w_{j\text{new}} = w_{j\text{old}} + \eta \cdot \Delta w_j \quad (16)$$

where $w_{j\text{old}}$ and $w_{j\text{new}}$ are the weights of each sample before and after learning process, respectively, and the change on weights can be described as follows:

$$\Delta w_j = - \sum_{i=1}^t \frac{\partial e^t}{\partial w_j} = - \sum_{i=1}^t (y^i - d^i) \frac{\varphi_j}{\sum_{k=1}^u \varphi_k}.$$

η is the change degree of network weight which uses the ratios of error to adjust learning velocity of weight. Where $\eta = \frac{\frac{1}{2}(y^t - d^t)^2}{\sum_{i=1}^{t-1} \frac{1}{2}(y^i - d^i)^2 / (t-1)}$.

Compared with traditional SOFNN model, the changes of the proposed DRFNN model can be summarized into the following three parts. The threshold is adjusted dynamically to control the deviation of neurons for computing the system error. Secondly, the modification, pruning, and supplementary methods of fuzzy rules are redefined to make the structure of fuzzy neural network more reasonable. Thirdly, the weight adjustment rules of fuzzy neural network have been supplemented and explained. The main differences between SOFNN and the proposed DRFNN model are presented in Table 1.

Table 1. The comparison between SOFNN model and the proposed DRFNN model.

Models	System Error	Structure of Neural Network	Weights of Network
SOFNN	L1 measure	Adjustment based on network output	Gradient descent method
DRFNN	L1 measure with error threshold of dynamic adjustment	Fuzzy rule adjustment using sensitivity and disturbance analysis	Gradient descent with weighted adjustment

3.1.7. The running Process of Network

In this section, the cooperative motion model of different limb parts is estimated by the proposed algorithm based on DRFNN. According to the strategies of neurons' supplementation, modification, and deletion, the process of the whole algorithm is described as follows:

(1) Set the related parameters of current initial network using random number, including center matrix C , width matrix σ , and weight matrix w .

(2) Calculate the network error $\varepsilon(i)$ based on current output of network using Equations (2) and (3) when the i -th training sample X^i is considered. Select the target spots i and k in target and template model.

(3) Compute the antecedent error matrix E based on the number of whole neurons u and trained samples n of the current network.

(4) Scanning each row of matrix E . If all of the elements in row i satisfy with the following conditions: $e_{ik} < 0 (k = 1, 2, \dots, u)$ when the i -th training sample X^i is considered, the criteria proposed by 3.2.2 is used to decide whether or not to supplement the neurons. The center and width of the supplemented neuron can be obtained by Equations (12) and (13), if necessary.

(5) Scanning each column of matrix E . If all of the elements in column j satisfy with the following conditions: $e_{kj} < 0 (k = 1, 2, \dots, n)$, the column j needs to be recorded. As for all the selected columns, the criterion proposed by 3.1.3 is used to decide to the neurons which need to be modified. The new center and width of the selected neurons can be obtained by Equations (14) and (15).

(6) Estimate the neurons which need to be pruned using Equation (11) and threshold T_y based on the existed neurons of network.

(7) Optimize the coefficients of weight layer by Equation (16) using the obtained structures and parameters of the network.

(8) Output the structures and values of network which indicate the motion model of current human posture.

Furthermore, the whole process of the proposed DRFNN model is shown in Figure 5.

Figure 5. The whole process of the proposed DRFNN model.

4. Experimental Result and Analysis

In order to demonstrate the effectiveness of the proposed human motion analysis method based on dynamic regulated fuzzy neural network, we compared our method to the method using fuzzy set approaches proposed in [29]. In addition, the proposed DRFNN algorithm is compared to the SOFNN algorithm in [30] when different human postures are considered. In our experiment, all of the human models were provided from free 3D models database [19] and various postures are generated by algorithms in [20,21].

Furthermore, the 3D coordinates of spots locating on the model surface can be obtained. Compared with 3D human scanner, there are no spots in the interior of models. The proposed method is developed by Visual Studio 2005 and executed on a Pentium IV 2.0 GHz personal computer.

Furthermore, our dataset is generated by ourselves based on free 3D models database [19] and Poser software. For the dataset, we have tested SOFNN and DRFNN algorithms. Other models for the dataset will be further studied in the future. Recently, many scholars have studied human posture retrieval and analysis. Hong et al. [31] recovered 3D human pose by retrieving relevant poses with image features. It needed multi-view silhouettes of human body on various images. That is, the cost and processing time of the system will be greatly increased, and the overall efficiency of the model needs to be further improved. Yu et al. [32] estimated 3D human pose from a single image by retrieving pose candidates with 2D features. However, the annotation of 2D human body image at joint points in the image dataset needs a lot of manual operation. The efficiency of the whole system will reduce when a large number of images are processed. Yasin et al. [33] retrieved and reconstructed 3D human pose using 2D landmarks extracted from an annotated 2D image. Thus, 3D human limb features are mapped by the joint information of 2D images, which will also introduce new errors. It will reduce the accuracy of pose retrieval and recognition. In the paper, SOFNN model and the proposed DRFNN model calculate and extract 3D human limb features using 2D joint points based on monocular camera. The whole system has low cost. In addition, direct calculation of 3D human limb features based on single human body image will avoid introducing new errors. Thus, dynamic adjustment of network parameters and structures will also improve the overall efficiency of the whole model. In conclusion, the system combining SOFNN or DRFNN with 3D human limb features can be considered as state-of-the-art models in the area of human pose retrieve and analysis. Therefore, SOFNN and DRFNN models are compared in the experiment.

4.1. Datasets

All of the 3D virtual human models are provided by the free 3D models database [19]. Furthermore, the initial 3D symmetric human body model is composed of point cloud data, which contains several 3D space points, which are represented by various 3D coordinates of X, Y, and Z. The variation range of coordinates is not the same, because various models are selected from different scenes. In order to ensure that the human body model has the same size, all of the models are normalized by ourselves. That is, the variation ranges of the 3D coordinates of X, Y, and Z of the point cloud are within [0, 1].

In order to accurately describe the different postures of symmetric human body, it is very important to segment the limbs and various parts of the whole model. However, it is difficult to use a unified body proportion to divide parts for all of the human models, due to the diversity of shape and height on various human models. In order to ensure the accuracy of human model segmentation, we use the partial segmentation method of Poser to extract various parts of the human model, which is a human modeling software. This method is conducive to the accurate matching between various parts of the symmetric human body model. Figure 6a shows human body segmentation method of Poser. Different colors represent different basic parts, such as head, neck, left and right chests, abdomen, waist, buttocks, left and right upper arms, left and right forearms, left and right thighs, left and right calves, left and right hands and feet (see Figure 6a). All of the divided parts of the human body have clear boundaries, and any two parts do not intersect each other. Therefore, accurate segmentation relationship can be established for the same part of human body belonging to different models, and then different poses of symmetric human body model can be generated. As shown in Figure 6a, the horizontal and vertical directions of the screen are defined as X direction and Y direction, respectively, and the Z direction is perpendicular to the screen.

Figure 6. Divided method and definition of human parts on 3D model.

In order to describe the human parts more clearly, some critical definitions of human parts are described in Figure 6b: (1) direction l of various parts. It determines the variation trend of the skeleton segment corresponding to the current human part, and it is also the common centerline of the sections in different directions. (2) vector (l_x, l_y, l_z) of human part. 3D coordinates are represented for the direction of the part, which is mainly used in the location of the target point and the calculation of normal vector of the direction section. (3) Direction angle θ . If the normal vector of the section is parallel to the Z-axis, θ equals to 0. That is, the angle between the normal vector of the current section and the Z-axis is the direction angle θ of the section. (4) Target point P_0 . It is mainly used to locate the direction angle and obtain the normal vector of the direction section, and it is obtained by the center point Cur of section I paralleling to the ZOY plane, and specific distance λ . (5) Direction section N. It mainly obtains the direction section by intersecting with the current human part. The normal vector of the cross section is obtained by the corresponding direction angle. (6) Direction curve. It is mainly generated by the intersection of the section in different directions and the current human part. That is, shape features of different directions are represented based on corresponding direction curves. The data points of different directions are shown in Figure 6b.

That is, the initial virtual 3D human model is firstly imported into Poser software. Then, various parts of the human model are extracted based on the partial segmentation method of Poser, and sizes and parameters of whole 3D model are calculated based on the above method. Thirdly, different poses of 3D human model are obtained by adjustment of various human parts using rigid body transformation based on the software of Poser, especially for the human limb parts. Therefore, the whole dataset can be generated by obtained different human poses based on divided various human parts' adjustment.

4.2. Behavior Comparison for Different Human Postures

The behavior for different human postures by using the DRFNN and the SOFNN in [30] was first tested. Table 2 shows the number of fuzzy rules, training time, and root mean squared error (RMSE) when the two algorithms are performed. The results using the DRFNN are better than those obtained by the SOFNN in [30] in all of three postures. That is, the DRFNN gets the better solution owing to less RMSE on different human postures, although its computation time is greater than that of the SOFNN. The poor performance of algorithm in [30] in different three postures can be attributed to its computation on fuzzy rules of neurons, which are unstable with respect to network structure adjustment.

Table 2. The results of two network algorithms for different postures.

Different Model Postures	Number of Fuzzy Rules		The Training Time of Algorithms(ms)		Root Mean Squared Error	
	SOFNN	DRFNN	SOFNN	DRFNN	SOFNN	DRFNN
Walking	9	8	309.65	373.46	0.0365	0.0289
Running	9	9	323.53	391.78	0.0378	0.0254
Sitting	10	9	315.24	394.36	0.0421	0.0276

Then, we tested the stability of the SOFNN in [30] and the DRFNN with several limb parts of different postures. In our experiment, different human sequence postures are selected by captured human motion images. Figure 7a1–a3 show the selected original human motion postures, Figure 7b1–b3, c1–c3 are the front and side views of evaluation results of corresponding postures, respectively. As shown in the figure, evaluated results of selected three postures are satisfactory based on the proposed DRFNN model. Furthermore, Table 3 shows the comparison result of the proposed DRFNN and SOFNN models on eight limb parts when the selected three postures are considered. Pos1, Pos2, and Pos3 are the three poses in Figure 7a1–a3. α_1 – α_8 are joint angles of corresponding limb parts (Figure 1). The target examples are the observed joint angles of limb parts on the three poses in Figure 7a1–a3. The error of α_1 – α_8 is the average value of difference between the evaluation results and the observed results based on the selected three poses when SOFNN and the proposed DRFNN models are applied. The obtained errors of the selected three postures by using the DRFNN are slightly less than the corresponding result obtained by the SOFNN in [30]. It is an important improvement since the accuracy of limb parts on symmetric target human body is critical for motion analysis on different human postures.

Table 3. The comparison results of two algorithms with different limb parts.

Rotation Angles of Limb Parts	Target Example (Degree) (Pos1, Pos2, Pos3)	Average Error of Network in [30] (Degree)	Average Error of the Proposed DRFNN (Degree)
α_1	(123.11, 159.46, 129.46)	2.75	2.16
α_2	(111.35, 35.56, 177.20)	3.67	2.78
α_3	(118.72, 149.84, 134.60)	3.41	2.13
α_4	(165.25, 35.31, 74.51)	3.54	2.21
α_5	(144.04, 153.45, 149.79)	2.51	1.67
α_6	(151.39, 170.94, 133.89)	2.31	1.54
α_7	(158.50, 149.65, 144.54)	3.71	2.32
α_8	(163.06, 137.88, 145.28)	3.44	2.15

4.3. Comparison of the Results between DRFNN and SOFNN

In this part, the proposed DRFNN and SOFNN in [30] are applied for analyzing the human motion in order to demonstrate the efficiency of the two algorithms. The angles of various limb parts corresponding to different postures are located for input data of existing neural network. That is, the obtained eight angles are used for the training data. The parameters of the DRFNN are chosen as follows: $T_e = 0.5$, $T_d = 0.8$, $T_y = 0.5$, and $\varepsilon = 0.9$.

In order to examine the noise immunity of the algorithms, training data are mixed with Gaussian noise sequences which have zero mean and different variances. In addition, for the purpose of training and testing, eight angles of various limb parts corresponding to different postures are located. We chose 100 data for preparing the input and output sample data. In order to demonstrate the accuracy of the two algorithms, another 100 data are tested. The overall results are demonstrated in Table 4.

Figure 7. Human postures and corresponding prediction results using the proposed DRFNN model.

Table 4. Results of two network algorithms with noise.

Variances (σ^2)	Training RMSE		Testing RMSE	
	SOFNN	DRFNN	SOFNN	DRFNN
$\sigma = 0$	0.0349	0.0285	0.0418	0.0328
$\sigma = 0.05$	0.0427	0.0347	0.0545	0.0416
$\sigma = 0.1$	0.0485	0.0379	0.0596	0.0437

As shown in Table 4, RMSE of the two algorithms become greater when the noise variance increases rapidly. Furthermore, it is clear that the DRFNN can get better perfor-

mance with less RMSE for training and testing in all of the three Gaussian noise sequences. That is, the DRFNN has better performance than the SOFNN in terms of network structure and RMSE.

Figure 8 shows the obtained angles of four limb parts for human sequence postures which includes left upper arm, right upper arm, left thigh, and right thigh. The unit of obtained angles is degree. Furthermore, the angles of left and right upper arm for sequence postures are obviously greater than that of left and right thigh. In addition, the changes in angle of thigh are relatively smaller than the corresponding result of upper arm for human postures. The phenomenon can be attributed to the greater activity range from upper arms when the human body locates on different postures, such as walking and running.

Figure 8. The angles of different limb parts.

Furthermore, the proposed DRFNN and SOFNN are applied for analyzing human motion based on sample data. Figure 9 shows the result of root mean squared error for human sequence postures when the two algorithms are applied. The fluctuation range of RMSE in DRFNN is smaller than that in algorithm of SOFNN. This is attributed to dynamic adjustment of neural network structure when sample data are trained. In addition, RMSE of most sample data in DRFNN are obviously smaller than that of SOFNN. That is, the proposed DRFNN can be applied for various human postures and has better performance on human motion analysis.

4.4. Comparison of the Motion Retrieval and Analysis on Different Human Postures

Another set of experiments concerning the immediacy and the accuracy of our proposed algorithms is presented. The proposed DRFNN is compared to the algorithm in [30] using SOFNN by the prediction result of human motion analysis when models locate on different postures.

Figures 10 and 11 show the best prediction result of walking posture using the algorithm in [30] and the proposed DRFNN, respectively, when the motion of walking posture is considered. Obvious shortages occur on the positions of thigh and calf when algorithm in [30] is applied for estimating walking posture of human models when algorithm of SOFNN is considered. The angle between left leg and right leg is bigger than that of the proposed DRFNN when Algorithm in [30] is considered. Furthermore, compared with the real posture of human body, the posture of legs is not very reasonable. However, the proposed algorithm using DRFNN successfully solves the above problems and obtains the satisfactory result of walking posture. It demonstrates that DRFNN has advantage

on posture analysis of human body, although its running time is greater than that of the algorithm in [30] shown in Table 5.

Figure 9. Root Mean Squared Error (RMSE).

Figure 10. Prediction result of walking model using the algorithm in [30].

Figure 11. Prediction result of walking model using the proposed DRFNN.

Table 5. The evaluation results of two algorithms of various inputs with different model postures.

Different Model Postures	The Average Output on Poor Inputs		The Average Output on Good Inputs		Average Running Time of Prediction (ms)	
	SOFNN in [30]	The Proposed DRFNN	SOFNN in [30]	The Proposed DRFNN	SOFNN in [30]	The Proposed DRFNN
Walking	(0.05, 0.11, 0.16)	(0.04, 0.11, 0.14)	(0.93, 0.86, 0.8)	(0.95, 0.89, 0.84)	316.79	482.54
Running	(0.05, 0.12, 0.17)	(0.05, 0.11, 0.15)	(0.92, 0.86, 0.81)	(0.95, 0.88, 0.83)	331.64	523.49
Sitting	(0.07, 0.14, 0.2)	(0.06, 0.12, 0.18)	(0.92, 0.87, 0.82)	(0.96, 0.89, 0.85)	304.35	468.96

Table 5 shows the evaluation results of various inputs and computation time of human posture prediction performed with the two above compared algorithms for different postures. Every posture has the three evaluation results when different algorithms are applied. The three average output values on poor inputs are corresponding to the poor levels of 0.1, 0.2, and 0.3, respectively, and the good input levels are selected as 0.9, 0.8, and 0.7. The evaluation results using the above two algorithms are within the constraints of scope. That is, the prediction of human posture is accurate when different levels are selected. Furthermore, the evaluation results of different postures are obviously better than that of SOFNN in [30] when the proposed DRFNN is applied. The good performance of DRFNN can be attributed to human motion computation based on the adjustment of numbers and structures of neural network. Compared with the SOFNN in [30], the evaluation results of various inputs have obvious superiority when the same measured level is considered. It is an important improvement since the evaluation result of different postures is critical for motion retrieval and analysis of human body.

To solve the problem of input-output partitioning, the algorithm in [30] based on SOFNN depends on the structure and parameter identification of neural network. Firstly, a kind of clustering function is introduced to determine the structure and values of network. Then, parameters of network are optimized by supervised learning algorithm. In the proposed algorithm of DRFNN, the error matrix and automatic regulation method are applied for predicting the human sequence posture, and the supplement and pruning of network neurons improve evaluation result of various inputs and reduce the computation time. Then compared with the SOFNN, the DRFNN increases the differentiation degrees among various human postures when different levels are selected. Therefore, the proposed DRFNN can retrieve and analyze human motions locating on various postures and has a wide range of application.

In conclusion, the proposed DRFNN is suitable for retrieving and analyzing human motion owing to its more precise evaluation result of different postures than the algorithm of SOFNN. Although the computation time of DRFNN is greater than that of the SOFNN, it is in an acceptable extent.

Additionally, a set of experiments concerning the applicability of the proposed DRFNN is presented. The proposed algorithm is applied for predicting the sequence motion gestures of human models for three different postures. The models used in the experiments are all provided by free 3D human models database [19]. Figures 12–14 show the evaluation results of human motion sequence corresponding to the posture of walking, running, and sitting, respectively. All of the postures on models shown in Figures 12a–d, 13a–d, 14a–d correspond to the best evaluated results of intervening sequence motion gestures. As shown in the figures, almost all of the evaluated results of sequence gestures are accurate for all of three different postures. Furthermore, the positions and angles of different human limb parts are reasonable when different human postures are considered. All of the evaluated sequence gestures can represent the complete human motion. That is, the evaluation results of different postures are satisfactory when the proposed DRFNN is applied. Therefore, the proposed DRFNN improves evaluation result of various inputs and overcomes the interference arising from different human limbs.

Figure 12. Human sequence motion results of walking posture.

Figure 13. Human sequence motion results of running posture.

Figure 14. Human sequence motion results of sitting posture.

In order to evaluate the performance of DRFNN model in processing arbitrary human postures, motion sequence video of different human postures is captured for recognizing human postures. Figure 15 shows the evaluation results of real human motion sequences of various postures. Figure 15a1–a8 are the captured pose frames of real human motion video, and Figure 15b1–b8 are the corresponding estimated poses using virtual 3D human model based on the proposed DRFNN model. As shown in the figure, almost all of the estimated results of various human poses are accurate, except for slight deviation of right

arm and right leg in pose 3, left leg in pose 6. This phenomenon can be attributed to the adjustment of human joint points on the Z axis, which leads to the slight deviation of 3D coordinates of joint points estimated by our method. However, as a whole, the estimation results of various human postures are satisfactory based on the proposed DRFNN model.

Figure 15. Human sequence motion results of various postures.

5. Conclusions

In this paper, efficient algorithms for retrieving and analyzing limb motion pattern of human models for different postures are proposed. The 3D coordinates of feature points on human limb are firstly calculated by limb features using the images of human motion. Then, a fuzzy neural network is designed, and it is applied for evaluating the similarity with different postures when eight angles of limb parts are considered as input vectors.

Subsequently, the network converges to the optimal solution by measuring the system errors using the dynamic regulation method with structure parameters and network values, which is called DRFNN. Then, the evaluation results of various input vectors are calculated using DRFNN based on regulated network structures and values. Therefore, motion sequence gestures of human models can be obtained by the best evaluation results using the dynamic regulated FNN. The experimental results show that the proposed algorithm can meet the requirements of human motion prediction for different postures in various research fields.

Future work will concentrate on further enhancing the accuracy of human motion analysis and decreasing the computation time of the proposed algorithm DRFNN. The initial estimation of limb parts is beneficial to human motion prediction for different postures. Thus, joining initial estimation of limb parts is considered as a good idea to enhance the accuracy of human motion analysis and decrease the computation time of the proposed algorithm.

Author Contributions: Conceptualization, X.H.; methodology, X.H. and Y.Z.; software, X.H.; validation, X.H. and S.W.; formal analysis, X.H. and S.W.; investigation, Y.Z. and S.W.; resources, X.H.; data curation, X.H.; writing—original draft preparation, X.H.; writing—review and editing, X.H., Y.Z. and S.W.; visualization, X.H.; supervision, Y.Z.; project administration, X.H. and Y.Z.; funding acquisition, X.H. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Science Foundation of China (No. 61703306), the Science and Technology Commissioner Foundation of Tianjin (No. 20YDTPJC02000), the Natural Science Foundation of Tianjin (No. 18JCYBJC85000), the Science and Technology Foundation of Tianjin (19JCTPJC43300), and the Doctoral Foundation of Tianjin Normal University (No. 52XB1302).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in the study are available on request from corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, C.; Yang, S.; Pan, Y.; Lai, H. Object-location-aware hashing for multi-label image retrieval via automatic mask learning. *IEEE Trans. Image Process.* **2018**, *27*, 4490–4502. [[CrossRef](#)] [[PubMed](#)]
- Lv, N.; Jiang, Z.; Huang, Y.; Meng, X.; Meenakshisundaram, G.; Peng, J. Generic content-based retrieval of marker-based motion capture data. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 1969–1982. [[CrossRef](#)]
- Chen, S.; Sun, Z.; Zhang, Y.; Li, Q. Relevance feedback for human motion retrieval using a boosting approach. *Multimed. Tools Appl.* **2016**, *75*, 787–817. [[CrossRef](#)]
- Bao, H.; Yao, X. Human motion data retrieval based on staged dynamic time deformation optimization algorithm. *Complexity* **2020**, *2020*, 6650924. [[CrossRef](#)]
- Ramezani, M.; Yaghmaee, F. Retrieving human action by fusing the motion information of interest points. *Int. J. Artif. Intell. Tools* **2018**, *27*, 1850008. [[CrossRef](#)]
- Xiao, Q.; Liu, S. Motion retrieval based on dynamic bayesian network and canonical time warping. *Soft Comput.* **2017**, *21*, 267–280. [[CrossRef](#)]
- Li, M.; Leung, H.; Liu, Z.; Zhou, L. 3D human motion retrieval using graph kernels based on adaptive graph construction. *Comput. Graph.* **2016**, *54*, 104–112. [[CrossRef](#)]
- Xiao, Q.; Song, R. Human motion retrieval based on statistical learning and Bayesian fusion. *PLoS ONE* **2016**, *11*, e0164610. [[CrossRef](#)]
- Wang, X.; Chen, L.; Jing, J.; Zheng, H. Human motion capture data retrieval based on semantic thumbnail. *Multimed. Tools Appl.* **2016**, *75*, 11723–11740. [[CrossRef](#)]
- Valcik, J.; Sedmidubsky, J.; Zezula, P. Assessing similarity models for human-motion retrieval applications. *Comput. Animat. Virtual Worlds* **2016**, *27*, 484–500. [[CrossRef](#)]
- Wang, Z.; Feng, Y.; Qi, T.; Yang, X.; Zhang, J. Adaptive multi-view feature selection for human motion retrieval. *Signal Process.* **2016**, *120*, 691–701. [[CrossRef](#)]
- Liu, X.; He, G.; Peng, S.; Cheung, Y.; Tang, Y. Efficient human motion retrieval via temporal adjacent bag of words and discriminative neighborhood preserving dictionary learning. *IEEE Trans. Hum. Mach. Syst.* **2017**, *47*, 763–776. [[CrossRef](#)]

13. Slama, R.; Wannous, H.; Daoudi, M. 3D human motion analysis framework for shape similarity and retrieval. *Image Vis. Comput.* **2014**, *32*, 131–154. [[CrossRef](#)]
14. Ren, T.; Li, W.; Jiang, Z.; Li, X.; Huang, Y.; Peng, J. Video-based human motion capture data retrieval via motionset network. *IEEE Access* **2020**, *8*, 186212–186221. [[CrossRef](#)]
15. Ramezani, M.; Yaghmaee, F. Motion pattern based representation for improving human action retrieval. *Multimed. Tools Appl.* **2018**, *77*, 26009–26032. [[CrossRef](#)]
16. Tang, Z.; Xiao, J.; Feng, Y.; Yang, X.; Zhang, J. Human motion retrieval based on freehand sketch. *Comput. Animat. Virtual Worlds* **2014**, *25*, 273–281. [[CrossRef](#)]
17. Li, Y.; Lei, H.; Lin, S.; Luo, G. A new sketch-based 3D model retrieval method by using composite features. *Multimed. Tools Appl.* **2018**, *77*, 2921–2944. [[CrossRef](#)]
18. Li, B.; Lu, Y.; Johan, H.; Fares, R. Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information. *Multimed. Tools Appl.* **2017**, *76*, 26603–26631. [[CrossRef](#)]
19. Free 3D Models Database. Available online: <http://artist-3d.com> (accessed on 15 November 2020).
20. Huang, X.; Gao, L. Reconstructing Three-Dimensional Human Poses: A Combined Approach of Iterative Calculation on Skeleton Model and Conformal Geometric Algebra. *Symmetry* **2019**, *11*, 301. [[CrossRef](#)]
21. Huang, X.; Zhu, Y. An entity based multi-direction cooperative deformation algorithm for generating personalized human shape. *Multimed. Tools Appl.* **2018**, *77*, 24865–24889. [[CrossRef](#)]
22. Sabahi, F. Introducing validity into self-organizing fuzzy neural network applied to impedance force control. *Fuzzy Sets Syst.* **2018**, *337*, 113–127. [[CrossRef](#)]
23. Zhang, W.; Qiao, J. Multi-variable direct self-organizing fuzzy neural network control for wastewater treatment process. *Asian J. Control* **2020**, *22*, 716–728. [[CrossRef](#)]
24. Zhou, H.; Zhang, Y.; Duan, W.; Zhao, H. Nonlinear systems modelling based on self-organizing fuzzy neural network with hierarchical pruning scheme. *Appl. Soft Comput.* **2020**, *95*, 106516. [[CrossRef](#)]
25. Zou, B.; Chen, S.; Shi, C.; Providence, U.M. Automatic reconstruction of 3D human motion pose from uncalibrated monocular video sequences based on markerless human motion tracking. *Pattern Recognit.* **2009**, *42*, 1559–1571. [[CrossRef](#)]
26. Chan, C.K.; Loh, W.P.; Rahim, A. Human motion classification using 2D stick-model matching regression coefficients. *Appl. Math. Comput.* **2016**, *283*, 70–89. [[CrossRef](#)]
27. Fu, Y.B.; Liu, S.; Li, H.H.; Yang, D.S. Automatic and hierarchical segmentation of the human skeleton in CT images. *Phys. Med. Biol.* **2017**, *62*, 2812–2833. [[CrossRef](#)] [[PubMed](#)]
28. Huang, X.; Hao, K.; Ding, Y. Human fringe skeleton extraction by an improved Hopfield neural network with direction features. *Neurocomputing* **2012**, *87*, 99–110. [[CrossRef](#)]
29. Ren, Y.; Li, Q.; Liu, W.; Li, L. Semantic facial descriptor extraction via axiomatic fuzzy set. *Neurocomputing* **2016**, *171*, 1462–1474. [[CrossRef](#)]
30. Liu, S.; Liu, Y.; Wang, N. Robust adaptive self-organizing neuro-fuzzy tracking control of UAV with system uncertainties and unknown dead-zone nonlinearity. *Nonlinear Dyn.* **2017**, *89*, 1397–1414. [[CrossRef](#)]
31. Hong, C.; Yu, J.; Tao, D.; Wang, M. Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3742–3751.
32. Yu, J.; Sun, J. Multispectral embedding-based deep neural network for three-dimensional human pose recovery. *Opt. Eng.* **2018**, *57*, 013107.
33. Yasin, H.; Kruger, B. An efficient 3D human pose retrieval and reconstruction from 2D image-based landmarks. *Sensors* **2021**, *21*, 2415. [[CrossRef](#)] [[PubMed](#)]