Research Article

Swimmer's Posture Recognition and Correction Method Based on Embedded Depth Image Skeleton Tracking

Haiyan Wang, Junhua Shi , and Xiguang Luo

College of Sports and Leisure, Guangdong Ocean University, Zhanjiang 524088, Guangdong, China

Correspondence should be addressed to Junhua Shi; shijunhua@gdou.edu.cn

Received 5 December 2021; Revised 5 January 2022; Accepted 15 January 2022; Published 10 February 2022

Academic Editor: Haibin Lv

Copyright © 2022 Haiyan Wang 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.

With the continuous emergence of depth image recognition technology, human motion recognition technology has gradually come to life. However, because the current technology of image recognition and skeletal tracking is too backward, people cannot use gesture recognition robots to perform gesture recognition and correction with high accuracy for athletes. In this study, the skeletal tracking depth image is obtained through the Kinect sensor. When using the Kinect sensor to acquire images, the requirements for the environment around the measured object are very low, and it will not be affected by conditions such as light, shadow, and object occlusion, and the pose can be segmented in real time in a complex background. In this study, the depth map feature and the bone information feature are selected for fusion. The HOD feature is difficult to deal with the occlusion problem, and it is not easy to detect the excessive range of the human body gesture or the change of the object direction. Based on this, the HOD feature is improved in many places to form a new 3D-HOD feature and DMM-HOG feature. In this study, the technical research on posture recognition and correction of swimmers in the experiment has realized posture collection, posture segmentation, posture analysis, posture modeling, and posture recognition. In this study, random forests are combined with HOD features to classify pixel-by-pixel points in depth images. The accuracy of training pixel classification is as high as 85%, and the average recognition time of the decision tree for each pose is about 9% higher than that of random forests. The use of Meanshift algorithm to cluster the classified pixels to form skeletal joint points is fast and efficient, and can quickly and accurately find joint points. It has also achieved good experimental effects on human motion recognition and continuous motion recognition, is more suitable for real life, and has commercial practical value.

1. Introduction

As a natural interaction method, gesture recognition will be widely used in robot control. With the advent of somatosensory technology, this kind of somatosensory technology is slowly being applied to people's lives, especially the robot gesture recognition and correction algorithm in swimming sports competitions. In swimming, the motion of the robot is controlled by somatosensory, which can avoid the tedious teaching process and programming of the robot program, thereby greatly improving the utilization rate and production efficiency of the robot.

With the advancement of technology, motion sensing has been widely used in various devices, among which the gesture control machine is considered to be the most convenient one and the most commonly used one [1]. In

reference [2], the author introduced the motion component (MC), which is a new method for single gesture recognition. The author uses the ChaLearn Gesture dataset to verify the experimental results of gesture recognition and prove the effectiveness of the method. In reference [3], the author proposed an ultrasonic gesture recognition method based on context-aware information. In reference [4], the author proposes a novel dynamic system for gesture recognition. The gesture tracking algorithm is used to extract the gesture trajectory and predict the gesture coordinates of the next frame. In reference [5], the author introduced a short-range compact 60 GHz millimeter-wave radar sensor that is sensitive to fine dynamic hand movements, and for real-time dynamic gesture recognition [6]. In reference [7], the author discloses a gesture recognition device, including a substrate, a light-emitting device, an image sensor, and a processing

unit, and also provides a complex optical device. In references [8, 9], the authors used high-order singular value decomposition (HOSVD) to factorize the data tensor associated with each order of the tensor, and evaluated the proposed method using three gesture databases. Experimental results show that the method not only performs well on the standard benchmark data set, but also has a good generalization in a learning gesture challenge [10]. In reference [11], the author evaluated the recently proposed method of recognizing static gestures. This work not only considered the accuracy of the identification, but also proposed a suitable solution for each method based on its advantages and limitations. In reference [12], the author proposed a gesture recognition system based on vector quantization (VQ) and Hidden Markov Model (HMM). The method presented allows unsupervised estimation of the parameters of the recognition system given the example of gesture recording, saving computation time and improving performance [13]. In recent years, noncontact interaction has received considerable attention due to the elimination of physical contact barriers. It can be used to collect dynamic gesture information, including finger coordinates, acceleration, and direction [14]. In reference [15], the author proposes a tracker inspired by the cognitive mental memory mechanism, and also proposes an appearance shape learning method to update the 2D appearance model and the 3D shape model appropriately. This method is superior to the latest 2D and 3D trackers in terms of efficiency, accuracy, and robustness. In reference [16], the author proposes an online control programming algorithm for human-computer interaction systems, in which robot movements are controlled by the operator's gesture recognition results based on visual images [17].

Traditional human body recognition and human tracking algorithms need to be initialized frequently and repeatedly, otherwise, the recognition and tracking algorithms will often be disastrously lost. Traditional human body recognition methods can also better recognize human body movements, but are more susceptible to environmental influences. Furthermore, the connection between the front and back frames needs to be made. Once frame loss occurs, catastrophic consequences will occur [18, 19].

Despite the increasing need to use depth data in computer vision applications, the spatial resolution of depth images is still limited compared to typical visible light images. The method proposed by the author qualitatively and quantitatively improves the results of the first-order method based on a simple four-connect MRF graph structure [20]. The author determined the baseline predictor tracking related to the positive deviation of the baseline, and used the depth image information to calculate the target overlap rate and to determine the target overlap rate [21, 22]. In reference [23], the author proposes a new 3D interpolation algorithm for generating digital geometric 3D models of bones from existing image stacks by peripheral quantitative computed tomography (pQCT) or magnetic resonance imaging (MRI). In reference [24], the author proposes a skin parameter optimization method based on depth image sequences. This method has a better visual animation effect, and there is less error. Depth

images play an important role in 3D applications. However, due to the limitation of depth acquisition equipment [25], the acquired depth images usually have a limited resolution [26]. In references [27, 28], this study proposes an effective salient object segmentation method based on depth perception image layering. Experimental results show that this method has better performance than the latest depth perception salient object segmentation method. In reference [29], the author believes that Kinect has the function of tracking gestures and explaining its actions according to the depth data stream, tracking human gestures in the form of clouds and simultaneously synchronizing. In reference [30], the author proposes a novel image-based method that reconstructs a set of continuous 3D lines used to create such objects, where each line consists of an ordered set of 3D curve segments. In reference [31], in order to enhance the prediction of depth and intensity images in the sparse photon state, the author used a custom image restoration strategy based on clustering. Custom algorithms can reconstruct depth images with millimeter-level depth uncertainty at a distance of approximately 2 meters [32, 33]. Layered depth images (LDIs) can compactly represent multiview images and videos, and are widely used in image-based rendering applications [34]. Compared with traditional LDI, multiview LDI (MVLDI) creates fewer layers and eliminates more redundancy. In references [35, 36], the author proposes a novel method that uses a convolutional neural network to enable a mobile robot to estimate its rough position in a 3D map using only a monocular camera. The article uses a pretrained convolutional neural network model to generate a depth image descriptor, and retrieves the location by calculating the similarity score between the current depth image and the depth image projected from the 3D map. The relevant research contents of the above-mentioned scholars all have certain characteristics and advantages in motion recognition, but undoubtedly they need a large number of data samples to support, but most of the scholars' samples are biased and cannot represent the whole.

For the analysis of depth images and the research on human motion recognition, most of the current research status at home and abroad has the problems of low reliability, slow recognition speed, and low accuracy. In this study, the skeletal tracking depth image is obtained through the Kinect sensor, and the pose can be segmented in real time in a complex background. In this study, the depth map feature and the bone information feature are selected for fusion, and the HOD feature is improved in many places to form a new 3D-HOD feature and DMM-HOG feature. In this study, the technical research on posture recognition and correction of swimmers in the experiment has realized posture collection, posture segmentation, posture analysis, posture modeling, and posture recognition. In this study, random forests are combined with HOD features to classify pixel-by-pixel points in depth images. The accuracy of training pixel classification is as high as 85%, and the average recognition time of the decision tree for each pose is about 9% higher than that of random forests. This study has achieved good experimental results on human motion recognition and continuous motion recognition, and is more suitable for robots to determine the swimmer's posture.

2. Method

2.1. Feature Extraction Algorithm Based on Kinect Bone Information

2.1.1. HOD Features (Histogram of Oriented Displacements). The HOD feature is also called the orientation gradient histogram. This method splits the motion trajectory of each joint into three plane motion trajectories, extracts the displacement transformation feature for each trajectory, and finally displaces all the joints on the three planes. The features are transformed for splicing to form HOD features.

The HOD feature is a descriptor that depicts three dimensions in two dimensions. The three-dimensional trajectory needs to be projected on the Cartesian plane first. Suppose the motion trajectory S of a node, its trajectory projected on the XY plane is as follows:

$T = \{P_1, P_2, P_3, \dots, P_n\}$, where P_t is the position of the node on the XY plane at time t . For each pair of P_t and P_{t+1} , the angle between the line segment P_tP_{t+1} and the X axis is calculated, and the formula is as follows:

$$\text{Slope} = \frac{P_{t+1} \cdot y - P_t \cdot y}{P_{t+1} \cdot x - P_t \cdot x} \quad (1)$$

For an angle, the voting information corresponding to it should be placed in a specific column of the histogram. The calculation formula is as follows:

$$\text{Histogram_bin} = \frac{\text{angle} \times \text{histogram_length}}{360} \quad (2)$$

2.1.2. 3D-HOD Features. $T = \{P_1, P_2, P_3, \dots, P_n\}$ is a trajectory in three-dimensional space, where $P_t(x_t, y_t, z_t)$ is the coordinates of the node at time t . It is necessary to calculate the direction of each pair of P_t and P_{t+1} , and perform weighted voting in the corresponding direction range to update the column corresponding to the histogram. First, it is necessary to determine whether P_tP_{t+1} is in the Z -axis direction, and then calculate whether P_tP_{t+1} belongs to the X -axis direction or Y -axis direction to determine the direction range of P_tP_{t+1} in the three-dimensional space, and finally obtain the corresponding histogram column. To determine whether P_tP_{t+1} is calculated in the Z -axis direction, the following equations are to be considered:

$$\begin{aligned} \text{Slope}_{xz} &= \frac{z_{t+1} - z_t}{x_{t+1} - x_t}, \\ \text{Slope}_{yz} &= \frac{z_{t+1} - z_t}{y_{t+1} - y_t} \end{aligned} \quad (3)$$

The voting weight is the length L of P_tP_{t+1} , and the calculation formula is as follows:

$$L = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2 + (z_{t+1} - z_t)^2} \quad (4)$$

Each displacement is calculated in the entire three-dimensional trajectory, and L is added to the column corresponding to the histogram. This histogram is the descriptor of the node's motion trajectory.

2.1.3. DMM-HOG Characteristics. The DMM-HOG feature is to set the DMM into uniform cells to extract HOG features. DMM-Hog feature extraction framework is as follows: first project the original data to three orthogonal projection planes, respectively calculate the 2D-MHI image sequence, and then generate the DMM from the 2D-MHI image sequence, and finally extract the features on the DMM splicing.

For the contribution of image feature information, the role of color information is negligible. For subsequent convenience, the image is converted into a grayscale image. The contrast of the image plays a larger role. At the same time, in order to reduce the influence of illumination changes and local shadows, it is necessary to perform gamma correction on the image. In addition to compensating for the loss of brightness, gamma correction has a very important role in graphics. It makes the data we use when calculating lighting is correct. The formula used is as follows:

$$I(x, y) = I(x, y)^{\text{gamma}} \quad (5)$$

Gamma = 1/2 can be taken. For each pixel in the image, the gradient of the X and Y directions is calculated, and the gradient is derived to obtain the gradient direction value of the point. This operation, once again, reduces the effects of lighting and facilitates the collection of texture and contour information. The specific calculation method uses the following formula:

$$\begin{aligned} G_x(x, y) &= I(x+1, y) - I(x-1, y)n, \\ G_y(x, y) &= I(x, y+1) - I(x, y-1)n, \\ G(x, y) &= \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \\ \alpha(x, y) &= \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right), \end{aligned} \quad (6)$$

where $G_x(x, y)$ and $G_y(x, y)$ represent the gradient of the point in X and Y directions, respectively, $G(x, y)$ represents the gradient increase of the point, and $\alpha(x, y)$ represents the gradient direction of the point.

2.1.4. Multifeature Fusion. The specific method of multifeature fusion is as follows: given depth map DMM-HOG feature $V = \{f, t, s\}$ and 3D skeleton information 3D-HOD feature F . Among them, f, t, s are the characteristics of the three directions of the depth map. SVM probability modeling is performed on DMM-HOG features and 3D-HOD features respectively, comparing the probabilities of the same behavior, selecting the larger one, and outputting the corresponding label. The formula is as follows:

$$\text{Label} = \max(\text{Model}_{\text{svm}}(E)); E \in \{V, F\} \quad (7)$$

Algorithm steps of multifeature fusion are as follows (Figure 1):

- (1) Kinect equipment acquires depth image information and three-dimensional skeleton information.

FIGURE 1: Algorithm flow of multifeature fusion.

- (2) Extract the DMM-HOG features from the depth image, expressed as V .
- (3) Extract 3D-HOD features on the three-dimensional skeleton, denoted as F .
- (4) Respectively recognize the motion of DMM-HOG features and 3D-HOD features.
- (5) Comprehensive recognition results and output corresponding tags.

2.2. Preprocessing of Gesture Recognition Based on Depth Image

2.2.1. 3D Point Cloud Structure. The Kinect sensor not only provides depth information but also angle information, so the ratio of the image area and the real cross-sectional area at this time can be obtained by the angle value, so as to obtain the true x and y axes according to the trigonometric similarity ratio. Therefore, in order to obtain the three-dimensional coordinates of the swimmer's posture in the real scene, a 3D point cloud reconstruction is needed for each pixel. Three-dimensional reconstruction is the reconstruction of three-dimensional objects in the virtual world. In layman's terms, what we do is the inverse operation of the camera (the camera presents the objects in reality in a two-dimensional picture, while the three-dimensional reconstruction is a two-dimensional picture of the information appears in the three-dimensional virtual space).

The specific conversion steps are as follows:

- (1) Obtain the Kinect horizontal and vertical viewing angle values, namely α and β .
- (2) On the horizontal axis, there are x axis, z axis, and α angle relationship, so that the horizontal axis scale factor $\text{RealWorld } Z \text{ to } X = 2 \tan(\alpha/2)$ is obtained according to the actual depth value of the z axis. Similarly, on the horizontal axis, the vertical axis scale factor is obtained according to β angle $\text{RealWorld } Z \text{ to } Y = 2 \tan(\beta/2)$. That is the ratio of the width w , depth z , height h , and depth z of the true cross-section where the depth value at that point is at that moment.
- (3) The symbol normalization of the results is as follows:

$$\begin{aligned} \text{NormalX} &= \frac{x}{\text{output.nX Res} - 0.5}, \\ \text{NormalY} &= \frac{0.5 - y}{\text{output.nY Res}}. \end{aligned} \quad (8)$$

Among them, output.nX Res, output.nY Res is the width and height of the acquired image, then this formula means that the left side of the depth image is in the negative x -axis direction, and the right side is in the positive x -axis direction. Similarly, the upper side is in the positive y -axis direction, and the lower side is in the y -axis negative direction.

- (4) Therefore the actual coordinates are as follows:

$$\begin{aligned} X &= \text{NormalX} \times Z \times \text{RealWorld } Z \text{ to } X, \\ Y &= \text{NormalY} \times Z \times \text{RealWorld } Z \text{ to } Y. \end{aligned} \quad (9)$$

By performing the above transformation steps on all pixels in the image, a 3D point cloud set of the image can be obtained, thereby realizing the mapping from virtual to reality. On this basis, a variety of human-computer interaction research and applications can be carried out, providing rich real information for post-swimmer posture segmentation and feature extraction, and achieving skeletal tracking.

2.2.2. Morphological Treatment. The application of mathematical morphology can compress and simplify image data, remove the unrelated structures in the image, and retain the basic shape features of the image.

Corrosion: After shifting the selected structural element B in Y by a , the element Ba covered by the translation process is obtained. If Ba is included in Y , this point a is kept, all points a in Y that satisfy this condition are found, and the set they form is called Y corroded by B , which is expressed as follows:

$$E(X) = \{a | Ba \in X\} = X \oplus B. \quad (10)$$

Dilation: it can be regarded as the dual operation of corrosion; the selected structural element A will be translated into b in X to get Ab . If Ab and X have an intersection, this b point is kept, all b points in X that satisfy this condition are found, and the set they form is called X inflated by A , which is expressed as follows:

$$D(X) = \{b | Bb \cap X \neq \Phi\} = X \oplus B. \quad (11)$$

2.3. Human Body Part Recognition and Posture Correction Method

2.3.1. Use Decision Tree to Identify Pixels. Decision tree is a simple but widely used classifier. Constructing a decision

tree through training data can efficiently classify unknown data. It is a decision analysis method to determine the feasibility of the probability that the expected value of the net present value is greater than or equal to zero by constructing a decision tree based on the known probability of occurrence of various situations.

Training decision tree is a very important first step in human body part recognition. The following focuses on the steps of decision tree training:

Step 1: initialization, the conditions for the cut-off of the initialization tree: depth and the minimum pixel number of the node.

Step 2: input the sampling information of all the pixels marked as true-value image samples as the recognition set Q , and input all the depth images and the split coefficients to be optimized.

Step 3: identify the pixel set Q to the left and right child nodes, and split the pixel set Q of the node to the left and right child nodes according to the value calculated by the following formula to form the left and right child pixel sets Q_{left} and Q_{right} .

$$\begin{cases} x_i \in Q_{\text{left}}(\phi), & f(x_i|\phi) \geq \tau, \\ x_i \in Q_{\text{right}}(\phi), & f(x_i|\phi) < \tau. \end{cases} \quad (12)$$

Step 4: choose the best splitting coefficient $\phi_p = \{\theta, \tau\}$. We need to select a set of splitting coefficients $\phi = \{\theta, \tau\}$ to maximize the information gain from the set of splitting coefficients composed of many pairs of offset vectors θ and multiple thresholds τ to be selected. This set of splitting coefficients is the best splitting coefficient we get.

2.3.2. Use Random Forest to Optimize Decision Tree. For our body part recognition, for each given pixel x , we will get a probability distribution histogram $p_i(c|x)$ through each decision tree of the random forest. Then finally we need to get the average probability distribution histogram $p(c|x)$ of all decision trees of this pixel. According to the average probability distribution histogram obtained by the following formula, the body part with the highest average probability distribution is taken as the body part to which this pixel belongs.

$$p(c|x) = \frac{1}{T} \sum_{i=1}^T p_i(c|x). \quad (13)$$

2.3.3. Use Meanshift Algorithm to Draw Human Skeleton. The Meanshift algorithm belongs to the kernel density estimation method. It does not require any prior knowledge and completely relies on the calculation of the density function value of the sample points in the feature space. Using the kernel function estimation method, under the condition of sufficient sampling, it can gradually converge to any density function, that is, the density can be estimated for data subject to any distribution.

The initial position of the skeleton point of the three-dimensional coordinate set is initialized. For the calculation of x_i , when the modulus value of $M_h(x_i)$ is greater than a certain threshold, this study believes that this pixel point has no contribution to the skeleton point. For this threshold, this article chooses 500 mm. Therefore, x_i is taken where $M_h(x_i)$ is smaller than the threshold as the initial skeleton point position x , and $M_h(x_i)$ is used as the offset vector for vector offset. By using the following formula, we can get the new skeleton point estimated position:

$$x = M_h(x) - x. \quad (14)$$

In order to improve the accuracy of clustering, the original body parts are divided into 31 parts, and the body parts are merged first, for example, the four parts of the head are first merged into the first part, as shown in Figure 2. There is a correspondence between the clustering parts of the clustering skeleton points and the skeleton points: head-head skeleton points, neck-neck skeleton points, L\R shoulder-L\R shoulder skeleton points, L\R elbow-L\R elbow skeleton point, L\R hand-L\R hand skeleton point, U\W torso-U\W torso skeleton point, LU\RU leg-LU\RU leg skeleton point, L\R knee-L\R knee skeleton point, L\R foot-L\R foot skeleton point.

3. Experiment

3.1. Kinect Skeletal Tracking. The setting of the experimental environment mainly depends on the hardware parameters of the Kinect device. Understanding the hardware parameters helps to obtain more accurate depth images and bone information. The direction of the Kinect coordinate system is different from the ordinary coordinate system, and the 3D space coordinates are used by the Kinect in the camera space. The coordinates are as follows:

- (i) The origin of the coordinates ($x=0, y=0, z=0$) is located at the center of the Kinect infrared camera
- (ii) The x -axis direction is the left direction along the irradiation direction of the Kinect
- (iii) The y -axis direction is the upward direction along the Kinect's irradiation direction
- (iv) The z -axis direction is the irradiation direction along the Kinect

The key parameters of Kinect equipment are listed in Table 1, and the physical picture is shown in Figure 3.

The key hardware and composition of Kinect are as follows:

- (1) Microphone array: the microphone array includes a total of 4 microphones, which will actively filter background noise.
- (2) Infrared camera: receive the reflected infrared spectrum to create a depth image of objects in the visible range.
- (3) Infrared projector: the principle of infrared projector is to read infrared information by actively projecting near infrared spectrum.

FIGURE 2: Schematic diagram of head clustering effect.

TABLE 1: Kinect device hardware parameters.

Project	Parameter
Effective range	1.2–3.5 meters
Visible range	Horizontal: 57 degrees, vertical: 43 degrees
Body rotation range	± 27 degrees
Data flow	Color image: 640 * 480, 30 fps Depth image: 320 * 240, 30 fps

FIGURE 3: Kinect device.

- (4) Elevation angle control motor: the motor is used to adjust the pitch angle of the Kinect body in order to obtain the best angle of the body, and the pitch angle range is $-27^\circ \sim 27^\circ$.
- (5) USB cable: the color data stream and infrared data stream obtained by the Kinect camera are communicated with the PC side through a cable, and the color data stream, depth data stream, and audio information are transmitted to the PC side.
- (6) Color camera: capture color video images of objects in the field of view of the camera.

It can be seen from the table that the viewing angle of the Kinect device has a certain range, and the device needs to be placed at a certain height to facilitate the collection of RGB image and depth image information. The height was tested at 0.75 meters, 1 meter, and 1.25 meters, respectively. The experimental results show that the effect is about 1 meter. Then, the effect of distance on the accuracy value was tested on a 1-meter platform. The actual results are shown in Figure 4.

After testing, it was finally decided to place the Kinect device on a platform with a height of 1 meter. The distance

FIGURE 4: Depth value accuracy diagram.

between the target and the device should be within a range where the measured value and the actual value are small, so that the extracted information is closer to the true value.

3.2. MSR-Action3D Gesture Recognition Data Set. The data set used in the experiments in this article is MSR-Action3D. Before calculating the descriptors, there is no standardization of the original data. The MSR-Action3D data set is a data set containing in-depth action recognition, with 20 action types, and each action is executed 2 or 3 times, for a total of 567 depth map sequences. Based on the difference in action types, the mainstream divides the data set into three sets as follows: AS1, AS2, and AS3. There are 8 action types in each set, and there is partial overlap between different sets. Table 2 lists the classification of actions on the MSR-Action3D dataset.

3.3. Posture Recognition and Correction Process Based on Depth Image Skeleton Tracking. This application uses C++ language and Matlab mixed programming on the VS2010 platform to encapsulate preprocessing, segmentation, skeletal tracking, feature extraction, and gesture recognition in different categories. By specifically implementing the method proposed above, the original image data obtained by Kinect are used as input. Then enter the OpenNI class object, OpenNI is a multi-language, cross-platform framework. Its main purpose is to form a standard API to build a communication bridge between vision and audio sensors and vision and audio perception middleware, and then to proceed through the tracking class object and segment class object operation in turn to get the output of the segmented image. The image is segmented and then the feature class is entered, and finally, the feature value is obtained to Matlab hybrid programming for recognition class objects. The details are shown in Figure 5.

Six major categories were designed during the implementation process. Among them, the OpenNI class implements the interface interaction with the hardware to perform data acquisition and storage functions; the tracking class implements the swimmer's skeletal tracking and obtains hand node functions; the segment class implements the 3D point cloud construction and segmentation of image pixels; then the segmented image data are passed to the

TABLE 2: Classification of actions on the MSR-Action3D dataset.

ActionSet1 (AS1)	ActionSet2 (AS2)	ActionSet3 (AS3)
Horizontal wave	High wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw X	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Hands wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup and throw	Side boxing	Pickup & throw

FIGURE 5: Gesture recognition and correction algorithm implementation process.

feature class to achieve body contour extraction and different biological feature extraction; the feature is given to the recognition class to complete gesture recognition.

4. Results and Discussion

4.1. Posture Recognition and Quantitative Analysis of Correction Parameters

4.1.1. Analysis of Decision Tree Depth. The depth of the decision tree determines the accuracy of pixel recognition and correction. The deeper the number of layers, the more judgment times the pixel recognition and correction generally go through, and the more accurate the result. Table 3 is a histogram of the depth of the decision tree and the average accuracy of recognition. From the histogram, we can see that as the depth of growth increases, the accuracy of its identification is higher, but the growth rate is relatively slow, which means that the effect of depth on accuracy is gradually reduced as the depth increases.

Through experimental analysis, we found that as the number of layers of the decision tree increases, the recognition rate of each image also increases significantly. Considering the effects of training time and recognition time, we chose 25 layers as the depth of the training decision tree. At this time, the average accuracy of the Kinect depth image prediction reached 0.82, the training time was about 5 hours, and the recognition time was 34 ms. In addition, the time for clustering of joint points is about 50 ms, and the time for each frame of depth image to get the skeleton point image is about 85 ms, which can basically meet the real-time requirements we need (Figure 6).

TABLE 3: Relationship between decision tree depth and recognition accuracy.

	Kinect depth image (%)	Dataset image (%)	Training image (%)
10	38	33	19
15	67	58	36
20	72	66	47
25	75	68	48
30	80	73	51

The depth of the decision tree is 25, which can satisfy both the accuracy of the recognition result and the real-time performance of the algorithm, so the depth of the decision tree is 25.

4.1.2. Influence of the Number of Random Forest Decision Trees. Random forest is an algorithm that integrates multiple trees through the idea of ensemble learning. Its basic unit is a decision tree, and its essence belongs to a major branch of machine learning-ensemble learning methods. Although the more decision trees, the more accurate the recognition result, but each decision tree is time-consuming during the recognition, and the greater the number of decision trees, the corresponding time consumption will increase. Figure 7 shows the recognition time under different numbers of decision trees. It can be concluded that when the number of decision trees is 6, the decision speed no longer meets the real-time requirements of our experiments, and it has little effect on accuracy. Therefore, considering the accuracy and time consumption of the identification and the

FIGURE 6: Decision tree depth and time consumption table.

FIGURE 7: Number of decision trees and time consumption.

space occupied by the nodes produced, we prefer to choose 3 decision trees as the number of decision trees included in our random forest.

4.1.3. Influence of Feature Offset Vector. M (vector modulus) is obtained by Gaussian sampling from 0 to $M_{\max}$. $M_{\max}$ represents the maximum length of M (value range of M). The larger the value of $M_{\max}$, the larger the range searched by the offset vector, the higher the accuracy of its distinguishability, and the more body parts searched. It can be drawn from the histogram in Table 4: when $M_{\max} = 50$ pixels, the recognition accuracy has a greater impact on the test image than on the training image. When $M_{\max} = 100$ pixels and $M_{\max} = 200$, the accuracy of training and test depth image recognition is slightly improved. When $M_{\max} = 300$ pixels, the recognition accuracy of the test image appears backward. This phenomenon is due to the fact that the offset vector is too long, which is too sensitive to features, so that some misjudgments have occurred in the recognition of image pixels.

In summary, the selection of 100 for M value range $M_{\max}$ is optimal, which not only reduces the number of feature parameters and saves training time, but also has a better recognition effect on training images and test images.

4.1.4. Influence of Threshold. Table 5 lists the relationship between the number of thresholds and the recognition accuracy. From the relationship between the number of

TABLE 4: Relationship between pixel $M_{\max}$ and recognition accuracy.

	Kinect depth image (%)	Dataset image (%)	Training image (%)
20	31	19	16
30	61	54	22
50	73	67	37
100	82	79	50
200	83	77	51
300	84	75	50

TABLE 5: Relationship between the number of thresholds and recognition accuracy.

	Kinect depth image (%)	Dataset image (%)	Training image (%)
5	45	31	19
10	57	42	27
20	73	59	40
30	87	79	52
50	89	81	51

thresholds τ and the average recognition accuracy rate, it can be concluded that with the increase in the number of thresholds, the average accuracy of lawyers increased. Before the number reaches 20, the recognition accuracy rate rises rapidly, but when the number is 20, it starts to increase slowly until 30, and it basically does not increase. Considering that the number of thresholds increases by 1 training time will be doubled, it is a better choice to consider when the number of thresholds τ is 20.

In summary, considering the comprehensive recognition accuracy and training time, for the selection of the number of thresholds τ , we choose 20. This value can satisfy the accurate recognition of the pixels by the trained random forest, and can save a lot of training time.

4.2. Multi-Feature Extraction and Fusion Result Analysis Based on Swimmer's Posture

4.2.1. DMM-HOG Characteristics. On the basis of gesture recognition, the accuracy of DMM-HOG features and HOD features is evaluated. Using such a test method, for each subset, the subset of Test1.1/3 is used as the training set and the remaining part is tested, the subset of Test2.2/3 is used as the training set and the remaining part is tested, and half of Test3 serves as the training set and the other half serves as the test set.

The DMM-HOG feature is an improvement of the HOD feature, and the two are compared to show the improved results of this article. There are many different configurations of HOD features: 2D-HOD feature, 3D-HOD feature, and DMM-HOG feature. The experimental results of various methods are shown in Figure 8.

First, the different configurations of HOD features are discussed. Simply increasing the number of columns in the histogram does not bring about improvement in recognition accuracy. When the number of histogram columns increases

FIGURE 8: Test results of different HOD features on the MSR-Action3D dataset. (a) HOD characteristics, (b) 2D-HOD characteristics, (c) 3D-HOD features, and (d) DMM-HOG features.

to a certain threshold, it will bring a negative impact. This may be because too many columns lead to excessive dispersion of effective information, reducing the difference in characteristics between different actions. Increasing the number of layers of the time pyramid will make the timing information more prominent in the features and improve the recognition accuracy. Of course, blindly increasing the number of layers of the time pyramid means that each histogram of the newly added level cannot form a meaningful histogram because the number of available frames is too small to increase the redundant information in the features. Experimental results prove that 3D-HOD performs well.

Compared with 2D-HOD of the same level, DMM-HOG has considerable advantages in recognition rate. Compared with the 3D-HOD with more levels, the DMM-HOG feature does not fall in recognition rate. DMM-HOG overcomes some of the disadvantages of HOD features, further improves the recognition rate, and proves the effectiveness of the improvement work.

4.2.2. Multi feature Fusion. Also on the basis of gesture recognition, the following test method is adopted: half of each subset is used as training data, and the other half is used as test data. Since the MSR-Action3D dataset only has depth maps and bone maps, this study selects some well-performing algorithms for comparison. The results of multiple

identification methods tested on the entire data set are shown in Figure 9.

It can be seen from the experimental results that, compared with many methods, the DMM-HOG feature already has a certain lead. Although the 3D-HOD features perform generally, they are further improved in recognition accuracy after being fused with the DMM-HOG features. This is due to the fusion of multiple features, combining the advantages between different features, and the difference in data sources also complements the disadvantages between these features, thereby improving the overall recognition rate. Therefore, the fusion of different features has a very positive effect on human behavior recognition.

4.3. Analysis of Swimmers' Posture Recognition and Correction Results. This article selects 6 common dynamic postures for swimmers, collects samples through Kinect in the early stage of the experiment, and collects 10 experimenters, each experimenter has 5 samples per pose, a total of 300 samples. Half of the sample book is used for training, and the other half is used for testing. Posture segmentation and feature extraction steps are performed on the training samples, HOD, 2D-HOD, 3D-HOD, and DMM-HOG are used as the feature input values of the samples, and the records are stored in the .dat file. The obtained training sample.dat files are sorted and numbered by type, and every 25th constitutes a sample set of swimming styles. That is, nos. 1–25 are all

FIGURE 9: Test results of multiple feature extraction algorithms on the MSR-Action3D data set.

TABLE 6: Identification and correction results of test samples.

		Diving	Spread arm	Paddling	Backstroke	Turn head	Effluent
Decision tree	Correct number	15	15	16	17	17	18
Random forest		18	17	20	19	19	21
Decision tree	Correct rate	60%	60%	64%	68%	68%	72%
Random forest		72%	68%	80%	76%	76%	84%
Decision tree	Average recognition time	0.856	0.842	0.865	0.886	0.901	0.914
Random forest		0.713	0.724	0.749	0.738	0.787	0.796

sample sets of swimming style “diving,” nos. 26–50 are all sample sets of swimming style “spread arm,” and so on, so there are 150 sample.dat files. The success rate obtained by using the test samples for testing is listed in Table 6.

From the above experimental results, it can be seen that the recognition method of swimmers’ posture recognition based on depth image skeleton tracking in this study has significantly improved the recognition rate. Among them, the recognition rates of “diving” and “open arms” gestures are relatively small, accounting for only 72% and 68%. The “turning head” and “flowing water” gestures have the highest recognition rate due to the obvious changes in their characteristics, with accuracy rates of 76% and 84%. The feature changes of the gestures “paddling” and “backstroke” before and after significant improvement are relatively high in recognition rate, and the recognition rate after improvement also increases. At the same time, for the average recognition time, it is found that the average recognition time of the decision tree for each pose is about 9% higher than that of the random forest, indicating that the random forest method is indeed higher than the decision tree in the recognition time of the pose library in this study. Therefore, the random forest improved on the decision tree has higher reliability and real-time performance on the swimmer’s posture recognition rate established in this study.

In summary, the experimental results confirm that the swimmer pose recognition method based on the depth image skeleton tracking in this study improves the recognition rate in the dynamic pose of this study, and the recognition time has reliable real time.

5. Conclusions

This article is based on the Kinect sensor based on depth image skeletal tracking swimmer posture recognition and correction. The image is obtained through the sensor and combined with skeletal tracking technology, which can

segment the posture in real time in a complex background. This article makes several improvements to the HOD feature to form new 3D-HOD and DMM-HOG features. In this study, the depth map feature and the bone information feature are selected for fusion, and the late fusion strategy is adopted to adjust the appropriate parameters for various features separately to fully demonstrate the complementarity of multifeature fusion.

In this study, the technical research on posture recognition and correction of swimmers in the experiment has realized posture collection, posture segmentation, posture analysis, posture modeling, and posture recognition. In this study, random forests are combined with HOD features to classify pixel-by-pixel points in depth images. The accuracy of training pixel classification is as high as 85%, and the average recognition time of the decision tree for each pose is about 9% higher than that of random forests. The use of Meanshift algorithm to cluster the classified pixels to form skeletal joint points is fast and efficient, and can quickly and accurately find joint points. It has also achieved good experimental effects on human motion recognition and continuous motion recognition, is more suitable for real life, and has commercial practical value.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by “Introduction to Sports Management,” the Online Open Course Construction Project of Guangdong Undergraduate Teaching Quality Engineering in 2020.

References

- [1] Y. Wu, S. Yu, and M. Yang, "High-performance gesture recognition system," *Sensors and Materials*, vol. 31, no. 3, p. 923, 2019.
- [2] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, "Principal motion components for one-shot gesture recognition," *Pattern Analysis and Applications*, vol. 20, no. 1, pp. 167–182, 2017.
- [3] X. Zhong, Y. Chen, H. Yu, X. Yang, and Z. Hu, "Context-aware information based ultrasonic gesture recognition method," *Journal of Computer-Aided Design & Computer Graphics*, vol. 30, no. 1, p. 173, 2018.
- [4] M. Yuan, H. Yao, C. Qin, and Y. Tian, "A dynamic hand gesture recognition system incorporating orientation-based linear extrapolation predictor and velocity-assisted longest common subsequence algorithm," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 9, pp. 4491–4509, 2017.
- [5] S. Hazra and A. Santra, "Robust gesture recognition using millimetric-wave radar system," *IEEE Sensors Letters*, vol. 2, no. 4, pp. 1–4, 2018.
- [6] A. S. Soares and A. L. Apolinário Jr., "Real-time 3d gesture recognition using dynamic time warping and simplification methods," *Journal of WSCG*, vol. 25, no. 1, pp. 59–66, 2017.
- [7] E. F. Hsu, M. T. Kao, Y. H. Huang, Y. H. Ko, and N. T. Chen, "Gesture recognition apparatus and complex optical apparatus," *Perspectives in Education*, vol. 22, no. 2, pp. 101–114, 2017.
- [8] Y. M. Lui, "Human gesture recognition on product manifolds," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3297–3321, 2017.
- [9] M. Manasi and S. Chaudhary, "Computer vision technology using gesture recognition," *International Journal of Computer Applications*, vol. 179, no. 19, pp. 1–4, 2018.
- [10] F. negin, J. Bourgeois, P. Robert, and F. Bremond, "A gesture recognition framework for cognitive assessment," *Gerontechnology*, vol. 17, no. s, p. 169, 2018.
- [11] H.-H. Huynh and D.-H. Vo, "Short survey on static hand gesture recognition," *International Journal of Advanced Computer Science & Applications*, vol. 8, no. 7, pp. 447–451, 2017.
- [12] P. Glomb, M. Romaszewski, A. Sochan, and S. Opozda, "Unsupervised parameter selection for gesture recognition with vector quantization and hidden Markov models," *Proceedings of the American Mathematical Society*, vol. 72, no. 2, pp. 422–424, 2017.
- [13] X. Zhang and X. Li, "Dynamic gesture recognition based on MEMP network," *Future Internet*, vol. 11, no. 4, p. 91, 2019.
- [14] W. Zeng, C. Wang, and Q. Wang, "Hand gesture recognition using leap motion via deterministic learning," *Multimedia Tools & Applications*, vol. 77, no. 4, pp. 1–22, 2018.
- [15] N. An, S. Y. Sun, X. G. Zhao, and Z. G. Hou, "Remember like humans: visual tracking with cognitive psychological memory model," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, pp. 1–9, 2017.
- [16] B. Chen, C. Hua, B. Dai, Y. He, and J. Han, "Online control programming algorithm for human–robot interaction system with a novel real-time human gesture recognition method," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, pp. 1–18, 2019.
- [17] K. G. Srinivasa, B. J. Sowmya, A. Shikhar, R. Utkarsha, and A. Singh, "Data analytics assisted internet of things towards building intelligent healthcare monitoring systems," *Journal of Organizational and End User Computing*, vol. 30, no. 4, pp. 83–103, 2018.
- [18] Z. Wan, Y. Dong, Z. Yu, H. Lv, and Z. Lv, "Semi-supervised support vector machine for digital twins based brain image fusion," *Frontiers in Neuroscience*, vol. 15, Article ID 705323, 2021.
- [19] M. Hu, Y. Zhong, S. Xie, H. Lv, and Z. Lv, "Fuzzy system based medical image processing for brain disease prediction," *Frontiers in Neuroscience*, p. 965, 2021.
- [20] E. Shabaninia, A. R. Naghsh-Nilchi, and S. Kasaei, "High-order Markov random field for single depth image super-resolution," *IET Computer Vision*, vol. 11, no. 8, pp. 683–690, 2017.
- [21] L. Xue-Qing, Y. De-Dong, M. Ning, and Y. Fu-Cai, "Depth kernel correlation filtering tracking based on multi-template," *Chinese Journal of Liquid Crystals & Displays*, vol. 32, no. 12, pp. 993–998, 2017.
- [22] J. Cai and C. Jung, "Image-guided depth propagation for 2-d-to-3-d video conversion using superpixel matching and adaptive autoregressive model," *Journal of Electronic Imaging*, vol. 26, no. 5, p. 1, 2017.
- [23] U. Mittag, A. Kriebbaum, and J. Rittweger, "A novel interpolation approach for the generation of 3d-geometric digital bone models from image stacks," *Journal of Musculoskeletal & Neuronal Interactions*, vol. 17, no. 2, pp. 86–96, 2017.
- [24] F. Gao, "Research on 2d animation design based on depth image sequence," *Revista de la Facultad de Ingenieria*, vol. 32, no. 14, pp. 490–495, 2017.
- [25] Y. Zhang, Q. He, Y. Xiang et al., "Low-cost and confidentiality-preserving data acquisition for internet of multimedia things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3442–3451, 2017.
- [26] G. Zhong, S. Xiang, P. Zhou, and L. Yu, "Spatially adaptive tensor total variation-tikhonov model for depth image super resolution," *IEEE Access*, vol. 5, no. 99, pp. 13857–13867, 2017.
- [27] H. Du, Z. Liu, and R. Shi, "Salient object segmentation based on depth-aware image layering," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 1–14, 2018.
- [28] M. S. D. Rasouli and S. Payandeh, "A novel depth image analysis for sleep posture estimation," *Journal of Ambient Intelligence & Humanized Computing*, vol. 10, no. 5, pp. 1999–2014, 2019.
- [29] A. H. Basori, "Real time interactive presentation apparatus based on depth image recognition," *International Journal of Electrical & Computer Engineering*, vol. 7, no. 3, pp. 1308–1315, 2017.
- [30] L. Liu, D. Ceylan, C. Lin, W. Wang, and N. Mitra, "Image-based reconstruction of wire art," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 63, 2017.
- [31] X. Ren, P. W. R. Connolly, A. Halimi, Y. Altmann, and G. S. Buller, "High-resolution depth profiling using a range-gated cmos spad quanta image sensor," *Optics Express*, vol. 26, no. 5, p. 5541, 2018.
- [32] B. Wang, B. F. Zhang, and X. W. Liu, "An image encryption approach on the basis of a time delay chaotic system," *Optik*, vol. 225, Article ID 165737, 2021.
- [33] H. Wei, A. Sehgal, and N. Kehtarnavaz, "A deep learning-based smartphone app for real-time detection of retinal abnormalities in fundus images," in *Proceedings of the Real-Time Image Processing and Deep Learning*, Baltimore, MD, USA, 2019.

- [34] R. D. Anjos, J. M. Pereira, J. A. Gaspar, and C. Fernandes, "Multiview layered depth image," *Journal of WSCG*, vol. 25, no. 2, pp. 115–122, 2017.
- [35] M. Sun, S. Yang, and H. Liu, "Convolutional neural network-based coarse initial position estimation of a monocular camera in large-scale 3D light detection and ranging maps," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, pp. 1–11, 2019.
- [36] S. Mohammadi and O. Gervei, "Using nonlocal filtering and feature extraction approaches in three-dimensional face recognition by Kinect," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, pp. 1–15, 2018.