Retraction

Retracted: Multifeature Fusion Human Pose Tracking Algorithm Based on Motion Image Analysis

Wireless Communications and Mobile Computing

Received 15 November 2022; Accepted 15 November 2022; Published 27 November 2022

Copyright © 2022 Wireless Communications and Mobile Computing. 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.

Wireless Communications and Mobile Computing has retracted the article titled “Multifeature Fusion Human Pose Tracking Algorithm Based on Motion Image Analysis” [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process and the article is being retracted with the agreement of the Chief Editor.

References

[1] L. Si and B. Liu, “Multifeature Fusion Human Pose Tracking Algorithm Based on Motion Image Analysis,” Wireless Communications and Mobile Computing, vol. 2022, Article ID 8513093, 12 pages, 2022.

[2] L. Ferguson, “Advancing Research Integrity Collaboratively and with Vigour,” 2022, https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/.
Research Article

Multifeature Fusion Human Pose Tracking Algorithm Based on Motion Image Analysis

Lipeng Si and Baolong Liu

School of Computer Science and Engineering, Xi’an Technological University, Xi’an, 710021 Shaanxi, China

Correspondence should be addressed to Lipeng Si; silipeng@xatu.edu.cn

Received 10 March 2022; Revised 14 April 2022; Accepted 23 April 2022; Published 9 May 2022

Copyright © 2022 Lipeng Si and Baolong Liu. 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.

Human pose and motion detection is an important area of computer vision research, covering the interplay of different fields such as image processing, pattern recognition, and artificial intelligence. Due to the complexity of human motion, existing 3D recognition and pose detection methods based on low-quality depth images are not very accurate and reliable. Due to the time-sensitive nature of features, a single feature cannot adapt to the dynamic changes of the scene, so it is difficult for the target tracking algorithm based on a single feature to achieve robust tracking results. If multiple features are fused and applied in the tracking algorithm, the complementarity between different features can be used to better adapt to the scene changes and achieve robust tracking results. In order to solve the problem of human pose and human motion recognition in low-quality depth images, this paper uses the Kinect somatosensory camera to obtain 20 human skeleton joint points through the Kinect skeleton tracking technology. This paper studies the typical human posture and interactive action recognition technology in daily life. On the basis of understanding the characteristics of skeletal data, this paper proposes a distance feature and angle feature model combined with human body structure. Through the experimental results, it is found that the distance feature and the angle feature value are basically not affected by the distance change. When the subjects turned 45° to the left and 45° to the right, the distance characteristics changed, which was different from the characteristic data when the subjects turned around.

1. Introduction

The so-called moving object tracking refers to the detection, extraction, identification, and tracking of moving objects in images and video sequences. And it acquires relevant parameters of moving objects, such as position, velocity, and trajectory, for postprocessing to achieve higher-level tasks such as position estimation and understanding the behavior of moving objects. Human pose detection and estimation in video images is a hot topic in the field of computer vision with many applications. There are many kinds of human features, and feature selection and feature extraction are the key issues in human motion analysis [1].

As a key technology in the field of computer vision, target tracking aims to accurately identify the region of interest of a target in a continuous image sequence and derive motion data such as target velocity and trajectory, and also provides the basis for solving various subsequent advanced vision problems such as target behavior analysis and target recognition. Human tracking and pose analysis play a very important role in human motion analysis, and content-based indexing and fast retrieval are also required in databases containing a large number of videos, such as sports videos. Although with the development of computer technology, computer hardware is also developing rapidly. But storing and processing large amounts of video data is still a heavy burden on today’s general-purpose computers, especially when the video data is uncompressed. In fact, for many applications, it is only a very small part of the video. Features such as human motion may be of interest. This is
to obtain relevant motion data by analyzing human motion in the video and then index and encode according to the motion data for fast retrieval.

Moving object detection can identify moving objects in a series of images. Currently, object tracking can track not only the entire human body but also a specific body part such as the head, face, arms, and legs. In this work, this paper analyzes the characteristics of Kinect. Kinect is used to store depth information, color information, and bone information of human poses, which are used to create a database of human poses. The images in the database need to be preprocessed for later feature extraction and classification without background and noise interference.

2. Related Work

Bhat et al. proposed a particle filter-based tracking algorithm for tracking objects in videos with vivid and complex environments. Particle filter (PF) algorithm is a typical representative of region matching based target tracking algorithm, which can achieve robust target tracking in complex backgrounds, narrow the target search range, and make the algorithm applicability greatly improved. Objects are represented in feature space by color distributions and KAZE features. The fusion of these two features will lead to efficient tracking as they have better performance under challenging conditions compared to other features. For the color distribution model, the Bhattacharya coefficient is used as the similarity index, while in the KAZE algorithm, the nearest neighbor distance ratio is used to match the corresponding feature points. The update model of the particle filter is based on kinematic formulas, and the weights of particles are controlled by an formula that combines color and KAZE features [2].

Gong F et al. proposed an innovative detection method based on flame multi-feature fusion. First, the motion detection and color detection of the flame are combined as a fire preprocessing stage. Second, although the flame is irregular, it has a certain similarity in the sequence of images. They then extracted features including the flame’s spatial variability, shape variability, and area variability to improve identification accuracy. They finally used the support vector machine for training, completed the analysis of the candidate fire images, and realized the automatic monitoring of the fire. Experimental results show that the proposed method can improve the accuracy and reduce the false positive rate, but it is not widely used [3]. Multiple object tracking understudies many smart video applications. Hassan proposed a new tracking solution. He exploits the power of recurrent neural networks to efficiently model complex temporal dynamics between objects regardless of appearance, pose, occlusion, and lighting. For online tracking, his real-time and accurate association of objects with activity trajectories is the main algorithmic challenge. Also, the reentry of objects should also be properly addressed. He adopted a hierarchical long short-term memory (LSTM) network structure to model the motion dynamics between objects by learning the fusion of appearance and motion cues, using a detection-tracking approach. Long short-term memory neural network (LSTM) is a special kind of recurrent neural network (RNN). Existing work captures the viewpoint of objects within the detected bounding box for tracking. His proposed motion representation and deep features representing object appearance are fused together [4]. Existing convolutional neural network- (CNN-) based trackers have limited tracking performance because the features extracted from single-layer or multilayer linear combinations are insufficient to describe the object appearance. To overcome this problem, Tang et al. proposed a novel tracking algorithm based on the interactive multimodal (IMM) framework to better explore deep features from different layers (IMM_DFT). In this method, they first build a measurement model from convolutional layers by applying correlation filters to hierarchical features. They then designed a hybrid system in order to efficiently estimate the target state of each layer. They use likelihood functions and transition probabilities to dynamically adjust the weights of each system. Extensive experiments show that the proposed algorithm achieves more favorable performance than several state-of-the-art methods [5]. Traditionally, most tracking algorithms use a single feature to describe an object. Considering the weaknesses of traditional localization algorithms in complex environments, Tong et al. proposed an object localization algorithm using multi-feature fusion and visual visibility. First, they utilize the visual visibility mechanism to process color histogram data to obtain visibility features and then extract foreground object models and background object models. Finally, they estimate the scale, rotation, and center point of the object based on the previous frame matching and localization data and obtain the new object position in the current frame. Experiments show that the proposed algorithm is able to track complex backgrounds and adapt to strong illumination changes, local events, fast motion, etc. [6]. Liu et al. proposed an efficient human mocap (motion capture device) data annotation method by using multiview spatiotemporal feature fusion. First, they utilize an improved hierarchical alignment cluster analysis algorithm. They divide the unknown human mocap sequence into several submotion segments; each submotion segment contains specific semantics. Then, two multiview features are discriminatively extracted and temporally modeled by a Fourier temporal pyramid to complement the features of each motion segment. Finally, they use discriminant correlation analysis to fuse these two types of motion features and further employ extreme learning machines to annotate each submotion segment. Extensive experiments in public databases show that the proposed method is effective compared to existing similar methods [7]. The above studies provide a detailed analysis of the application of target tracking techniques and multi-feature fusion. It is undeniable that these studies have greatly contributed to the development of the corresponding fields. We can learn a lot of lessons from the methods and data analysis. However, there are relatively few studies on human pose recognition in the field of target tracking techniques, and there is a need to fully apply these algorithms to the research in this field.
3. Multifeature Fusion Human Pose Tracking Algorithm Based on Moving Image Analysis

Human pose detection and motion recognition are important topics in the field of computer vision, and these topics include the position and motion of objects. These are recognized and analyzed through various image processing methods and techniques, machine learning, and pattern recognition. These include identifying, classifying, and monitoring objects associated with low- and mid-level vision processing and location detection, behavior analysis, and behavior understanding associated with higher-level processing, as shown in Figure 1 [8].

For the four video recognition system functions, the detection and tracking of human objects are very important to the system. These include detecting, recognizing, and monitoring human objects associated with lower-level processing, as shown in Figure 1 [8]. For the four video recognition system functions, the detection and tracking of human objects are very important to the system, involving the main visual problems, and many mature methods have been proposed. Human target detection methods include background removal method, frame difference method, and optical flow method [9]. The basic idea of the background removal method is to compare the input image with the background model and segment the moving target by determining changes in features such as grayscale or by using changes in statistical information such as histograms. This method is simpler to implement and has good real-time performance and can capture objects better when the complete background image is available. Optical flow is defined based on pixel points, reflecting the image changes due to motion during the time interval dt. The optical flow method is used to be able to detect objects independently, without the need for other information from the camera.

In the process of detecting human body posture, the technical difficulties are mainly reflected in the following aspects:

1. Difficulty in isolating areas of the human body in a rapidly changing environment: under ideal lighting and simple background conditions, existing image segmentation methods can extract human regions well. However, when the light is too bright or too dark and the background is complex, it is difficult for the image segmentation method based on monocular or binocular camera to accurately extract the human body area [10, 11]

2. The problem of self-recognition in the process of human gesture recognition

3. The problem of user independence: since different users are different, for example, they are tall or short, fat or thin, etc., these differences will bring great uncertainty to human gesture recognition

4. Independence of perspective: the perspective of the robot body will change according to the positional relationship between the robot and the user, and different perspectives will lead to large differences in the human posture images recorded by the camera. Therefore, it is technically difficult to obtain consistent features from different viewpoints in human pose recognition [12, 13]

Capture human motion video and extract features, then use mean shift to track human motion, and finally integrate the extracted human image into mixed reality.

In combining human body images with mixed reality, first separate the images from the mixed background library, and match the virtual background coordinates with the original human body video and the extracted binary image one by one [14]. Then the separated human body image is identified as the foreground background, so as to obtain the human body image in mixed reality; finally, the foreground image will be combined with the previous results frame by frame. Figure 2 is a schematic diagram of the synthesis of a human body image and a virtual background. Motion detection is an initial part of visual analysis and is the basis for processing procedures such as object classification, tracking, perception, and behavior description [15]. The ultimate goal of motion detection is to extract distant foregrounds from the background to form a subset of motion regions. Segmentation of motion regions is important for postprocessing, which also focuses on pixels within motion regions. However, the segmentation of moving regions is often affected by dynamic changes such as complex backgrounds, light waves, and shadows, which greatly increases the complexity of detection [16].

1. Face foreground segmentation: face foreground segmentation can be achieved through simple image processing techniques. It analyzes the position of each pixel in the human body based on the distance ratio and then uses edge detection to separate the human body from the full depth of the image.

Skeletal tracking is a technology that allows Microsoft to use the depth images captured by the Kinect to locate and locate key parts of the human body. This is the key technology for Kinect’s success. Regions and backgrounds are segmented and then the Kinect camera is calibrated. It uses
this image segmentation method to perform transformation
matrix segmentation on RGB images of human body
regions. This method is less affected by the environment
and has good real-time performance, which meets the
requirements of segmenting human body regions in a diffuse
environment [17].

(2) Human foreground images can be separated using
simple image processing techniques. Each pixel is
checked for proximity to a human body and then
separated from the full depth image by perimeter
detection [18, 19].

Binary segmentation of images can be viewed as a binary
labeling problem. Given an image \( S \) of size \( P \times Q \), the image
matrix can be regarded as a one-dimensional vector in row-
major order, and a one-dimensional label vector \( B = (B_1, \ldots, B_p, \ldots, B_{p\times q}) \) can be defined. The value range of the
element \( B_i \) in \( B \) is \( \{0, 1\} \), and when \( B_i = 0 \) is used, it means that
the \( i \)-th element of the image \( S \) is marked as the background.
On the contrary, it is expressed as the foreground [20]. To
segment the image \( S \) is to define a suitable label vector.
The image segmentation effect can be evaluated by calculating
the cost function of label \( A \).

\[
E(B) = \lambda \cdot R(B) + C, \tag{1}
\]

where

\[
R(B) = \sum_{i=1}^{p \times q} R_i(B_i), \tag{2}
\]

\[
C(B) = \sum_{(i,j)} C_{ij} \zeta(B_i, B_j),
\]

where

\[
\zeta(B_i, B_j) = \begin{cases} 
1, & \text{if } B_i \neq B_j, \\
0, & \text{else}, 
\end{cases} \tag{3}
\]

\[
C_{ij} = \alpha \exp \left( -\frac{(I_i - I_j)^2}{2\sigma^2} \right) / \text{dist}(i,j), \tag{4}
\]

The cost function includes two parts, the regional cost
\( R(B) \) and the relational cost \( R(C) \), and the coefficient \( \lambda \)
reflects the proportion of the regional cost and the relational
cost. The regional cost reflects the penalty for each pixel
being missegmented, such as \( R_i(1) \) represents the penalty
for marking the \( i \)-th pixel as foreground, and \( R_i(0) \) repres-
ts the penalty for marking the \( i \)-th pixel as background
[21]. \( R(.) \) is defined by defining the grayscale histograms
of the foreground and background: \( \Pr(I|O) \) and \( \Pr(I|C) \).

\[
\begin{align*}
R_i(0) &= -\ln \Pr(I|O), \\
R_i(1) &= -\ln \Pr(I|C), \tag{5}
\end{align*}
\]

The relational cost \( C(B) \) reflects the relation between
pixels. If the \( i \)-th pixel has a neighborhood relationship with
the \( j \)-th pixel, when marking the vector element \( B_i \neq B_j \), then
the boundary between \( i \) and \( j \) is the segmentation boundary
[22]. \( C_{ij} \) is used to judge the possibility of whether there is a
boundary between \( i \) and \( j \). It can be seen from the definition
that the \( C_{ij} \) and the adjacent pixel grayscale have an inverse
relationship; that is, the more similar the two pixel grayscales
are, the greater the \( C_{ij} \), and the greater the boundary cost of
considering the \( i \)-th and \( j \)-th pixels as segmentation [23].

In order to segment an image \( S \), the graph cut algorithm
constructs an undirected graph \( G \leq V \) according to the char-
acteristics of the image. Each pixel \( p \) in the image corre-
sponds to a node \( v \) in the graph. In addition, two
additional sink nodes, the target node (S) and the back-
ground node (T), are added to the graph; that is, there are

\[
V = P \cup \{S, T\}. \tag{6}
\]

There are two types of undirected edges \( E \) in the graph:
node edges and neighborhood edges. Each pair of adjacent
pixels \( p \) and \( q \) in the image \( S \) contains a neighborhood edge
\( \{v, u\} \) between the corresponding nodes \( u \) and \( v \) in the
graph \( G \), and all the neighborhood edges \( \{v, u\} \) form a
neighborhood edge set \( N \). For each pixel \( S \) in the image \( S \), the
corresponding node \( v \) in the graph \( G \) has two edges,
which are connected to the target node \( S \) and the back-
ground node \( T \), respectively. That is, the edge set of the

---

**Figure 2:** Schematic diagram of the composition of mixed reality images.
graph $G$ can be expressed as

$$E = N \cup \{\{v, S\}, \{v, T\}\}. \quad (7)$$

The rules for determining the weights of the edges in the graph $G$ are as follows, which belong to the edges in the neighborhood edge set $N$. Its weight is calculated by formulas (3) and (4); for the pixel $p$ in the image, it corresponds to the node $v$ in the graph. The weight of the edge $\{v, S\}$ connected to the target node $S$ can be expressed as

$$Q_{\text{ts}} = \begin{cases} 
\lambda R_p(\text{obj}), & \text{if } p \in P \text{ and } p \notin O \cup C, \\
0, & \text{if } p \in O, \\
F, & \text{if } p \in C.
\end{cases} \quad (8)$$

Among them, $O$ and $C$ represent the target pixel set and the background pixel set, respectively, and $F$ can be calculated by the following formula:

$$F = 1 + \sum_{q \{p, q\} \in N} C\{p, q\}. \quad (9)$$

For any feasible cut $D$ of graph $G$, a unique binary division $B(D)$ of image $P$ can be defined.

$$B_p(D) = \begin{cases} 
\text{obj}, & \text{if } \{p, T\} \in D, \\
\text{bkg}, & \text{if } \{p, S\} \in D.
\end{cases} \quad (10)$$

After segmentation, each pixel can only have one edge connected to the sink node; that is, the pixel is either segmented as the background or as the target. The image corresponding to the minimum cut $\tilde{D}$ is divided into $B(\tilde{D})$, and the segmentation cost defined by formula (1), the segmentation cost of $B(\tilde{D})$ is the smallest among all segmentations. For any partition $D$ in a feasible cut set, according to formula (7) and the weights of the edges in the graph, the cost of the graph cut $D$ can be calculated by the following formula:

$$|D| = \sum_{p \in O \cup C} \lambda \cdot R_p(\text{obj}) + \sum_{\{p, q\} \in N} C\{p, q\} \cdot \delta(B_p(D), B_q(D)), \quad (11)$$

Therefore, the segmentation cost of the minimum cut $\tilde{D}$ is

$$E(\tilde{B}) = |\tilde{D}| + \text{const},$$

$$E(\tilde{B}) = \min_{D \in F} |D| + \text{const},$$

$$E(\tilde{B}) = \min_{D \in F} E(B(D)) = \min_{D \in F} E(\tilde{B}).$$

(3) Body part recognition: this part of the work is to identify body parts based on contour body surface images. Kinect uses terabit data to train a random forest model to classify body parts and interprets the depth image of each random forest pixel to
obtain images with body part labels. Each pixel is labeled with a class label (e.g., headband and shoulders) and the probability that the pixel belongs to a certain class [24]. Currently, the human body can be divided into 32 different parts.

(4) After the previous step, the pixels of each body part are combined into a three-dimensional point of the human body, and the joint position is determined by combining the Gaussian kernel-based mean shift local model detection method; check each pixel to determine the coordinates. This allows the skeletal joints of the human body to be identified from depth images [25]. Kinect is currently able to detect and detect 24 joints, but only 20 are public point coordinate values.

It uses several underlying features such as the trajectory of a set of points on the human body, the speed of the point, the acceleration of the point, the directional gradient histogram of the image, and the optical flow to calculate the self-similar matrix. It uses the trajectory, velocity, and acceleration of points to calculate the self-similar matrix, which needs to identify and track a set of points in the human body during the movement process. And it needs to obtain a set of points on the human body from the actual video and track this set of points to obtain its motion trajectory [26].

Using the three-dimensional position information of human bones, the representation of local features can be achieved. Since the position of the skeleton points will change with the pose and time point, the angle between the skeleton points is used as the feature of pose recognition in this paper. It can easily obtain the skeleton point information of the human body by using the handles and events in the Kinect SDK. The obtained skeleton point image contains...
20 skeleton points of the human body. They are head coordinates, shoulder center coordinates, spine coordinates, hip center coordinates, left shoulder coordinates, left elbow coordinates, left wrist coordinates, left hand coordinates, right shoulder coordinates, right elbow coordinates, right wrist coordinates, right hand coordinates, left hip coordinates, left knee coordinates, left ankle coordinates, left foot coordinates, right hip coordinates, right knee coordinates, right ankle coordinates, and right foot coordinates [27], as shown in Figure 3.

The bone point information obtained by the Kinect camera can easily obtain the x and y coordinates of each bone point, and the z coordinate can be obtained from the depth value at the corresponding point \((x, y)\) in the depth map. There will be a horizontal swing angle and a pitch angle between every two bone points, and 20 bone points will have an \(19 \times 2 = 38\) angle. Since the quadrant of the angle is also taken into account when directly calculating the angle, the trigonometric function value of the angle is taken to construct the final eigenvector. The range of the horizontal swing angle is \([0, 2\pi]\), and the range of the pitch angle is \([-\pi/2, \pi/2]\), so the horizontal swing angle needs the sine and cosine values to be determined together, and the pitch angle only needs the tangent value to be determined. So, the resulting eigenvector will be a \(3 \times 19\) vector of \(R_i = (R_{i_{x}}, R_{i_{y}}, R_{i_{z}} | i = 1, 2, \ldots, 19)\), where

\[
\begin{align*}
R_{ix} &= \sin \alpha_i = \frac{y_{i+1} - y_i}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}}, \\
R_{ic} &= \cos \alpha_i = \frac{x_{i+1} - x_i}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}}, \\
R_{iz} &= \tan \alpha_i = \frac{z_{i+1} - z_i}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}},
\end{align*}
\]

where \(\alpha_i\) is the horizontal swing angle at bone point \(i\), \(\beta_i\) is the pitch angle at bone point \(i\), \((x_i, y_i, z_i)\) is the spatial coordinate at bone point \(i\), and \((x_{i+1}, y_{i+1}, z_{i+1})\) is the spatial coordinate at bone point \(i + 1\). The coordinate system \(i + 1\). The coordinate system is used to identify the unit. During recognition and classification, information from the training unit is used to match data to trained patterns and determine the type of signal received. The method is the one adopted by most motion tracking systems based on the sensor principle. But there are other test methods based on other principles, and the principles are the most typical implementation methods.

Typically, a pattern recognition system has a training module that trains a reference pattern or reference model for recognition. Its data from the training unit is then used to identify the unit. During recognition and classification, information from the training unit is used to match data to trained patterns and determine the type of signal received. The method is the one adopted by most motion tracking systems based on the sensor principle. But there are other test methods based on other principles, and the principles are the most typical implementation methods.

Human motion sensors follow the design guidelines of low cost, low energy consumption, easy portability, and adaptability to the environment. The human motion sensor consists of a microprocessor, a three-axis accelerometer, a three-axis angular rate sensor, and a power control unit, which are responsible for collecting and storing motion data and information. The microprocessor is responsible for collecting the information from the sensors and storing it in its memory buffer. The power management unit is responsible for powering the microprocessor, sensors, and all electronic components. Kinect uses an infrared transmitter and an infrared camera to acquire depth data and a color camera to acquire color images. Four audio microphones are responsible for

![Diagram of identification process](image)

**Figure 6: Modeling identification process.**

| Table 2: Measurement results. |
|-----------------------------|
| Kinect placement height (cm)| 35  55  75  95  115  135  155 |
| Nearest distance (cm)       | 152 161 169 173 178 172 165   |
| Farthest distance (cm)      | 370 369 362 372 368 373 378    |
collecting audio data, and the core component of the central processing is the PS1080 chip. This chip is responsible for controlling other sensors, such as controlling infrared emitters to project external light from the structure. At the same time, the CMOS image sensor receives the light spot formed by infrared light on the object, transmits it to PS1080 for processing to obtain the depth data, and transmits it to the computer through USB. Kinect somatosensory sensor is a device that combines various data. From Table 1, it can understand the basic specifications and parameters provided by Kinect.

4. Experiment of Human Posture Tracking Algorithm

At present, the most widely used method is the method of defining rule sets. This method directly uses the skeleton
data to define the posture and action and can achieve the purpose of real-time online posture recognition. This method is suitable for simple recognition of human body gestures and actions. For example, in somatosensory interaction, a small number of gesture interaction commands are defined to complete interaction tasks. This method generally only selects the relationship of a few joint points to analyze the posture and action of the human body and cannot describe the overall posture of the human body. If it wants to use Kinect to analyze the more subtle gestures and movements of the human body, the recognition method based on feature extraction is a good choice. In this method, a certain amount of posture samples is first collected to form a posture sample library, and a posture and action are defined by the sample library. This process is the process of building a model. Using the sample library to mark the category to which a new pose and action instance belong, this process is the process of recognition. Figure 6 shows the process of building a model and identifying a classification algorithm based on feature extraction.

In experimental program development environment, the experimental program runs under the Windows 8 operating system, the experimental program development environment is Visual Studio 2010, and the C# language, the WPF development framework, and the Kinect SDK v1.7 development kit are used. It uses the Kinect SDK bone tracking-related API to extract the 3D point data and combines the file input and output stream technology to write the 3D point data of the bones into a .csv file. The file can be opened directly in Excel, and data processing and analysis can be performed in Excel.

Test the closest and farthest distance that Kinect can track the whole body of the human body at different heights. Kinect is placed at a height of 40 cm, 60 cm, 80 cm, 100 cm, 120 cm, 140 cm, and 160 cm from the ground. It adjusts the elevation angle of the Kinect camera so that the Kinect can track the whole body exactly (the head and feet are at the boundary of the vertical viewing range) and use a meter ruler to measure the distance from the front toe to the Kinect camera, which is the closest distance to the whole body tracking, then move the body backwards. During this process, all 20 points of the body are in the tracked state until unstable joints appear, stop moving backwards, and measure the distance from the toes to the Kinect, which is the farthest distance for the whole body to track. The experiment takes the standing posture as an example. The height of the experimenter is 178 cm. The measurement results are shown in Table 2.

Test the closest and farthest distance that Kinect can track the whole body of the human body at different heights. Kinect is placed at an increased height, the minimum distance that can track the whole body of the human body first increases and then decreases, but the range of the farthest tracking human body does not change much. In addition, when the experimenter is facing the Kinect, all 20 joints in the body are in a tracked state. When the experimenter’s relative to the Kinect’s head angle changes, some joints will be in a state of speculation or untracked. Here, we test the range of angles within which the joint points of the whole body can be tracked. Standing at a distance of 2 m, slowly rotate the body to the left (right) until the relevant nodes are in a presumed or untracked state and stop rotating. At this time, the angle between the human facing direction and the Kinect optical axis direction is about 45°, which means that the Kinect can track the bones of the human body within this angle range.

Based on the performance analysis of the Kinect’s skeleton tracking, this paper determines the following experimental conditions based on the laboratory environment: (1) the distance between the human body and the Kinect is selected between 1.7 m and 3.5 m. Based on two considerations, first, the experiments verify that the range of Kinect full-body tracking is within this distance range. For taller experimenters, or with a gesture of reaching up, the closest distance for full body tracking will be greater than this value. (2) The relative angle between the human body and the Kinect should not exceed 45°. This paper uses a single Kinect to study human gesture recognition. In the experiment, it is found that if the relative angle rotation is too large, the detection of some joint points will be distorted. (3) During the experiment, keep a person within the visual range of the Kinect, and the experiment environment should be as open as possible without strong lighting. And where there are few backgrounds and obstacles, Kinect sometimes misidentifies objects with backgrounds similar to people as human bodies.

### Table 3: Average angular characteristics of Kinect placement at different heights.

| Height (cm) | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|
| 50         | 0.49| 0.65| 0.52| 0.60| 1.28| 2.30| 1.26| 2.32|
| 100        | 0.50| 0.70| 0.51| 0.76| 1.30| 2.32| 1.40| 2.35|
| 150        | 0.50| 0.70| 0.52| 0.80| 1.32| 2.41| 1.45| 2.45|
| 200        | 0.51| 0.71| 0.51| 0.76| 1.31| 2.45| 1.46| 2.48|

5. Results of Multifeature Fusion Human Body Pose Tracking Algorithm

5.1. Verify whether the Different Distances between the Human Body and the Kinect Have an Effect on the Features. This experiment verifies the influence of the placement height of the Kinect on the features and excludes the influence of other factors. The Kinect is placed at a fixed height of 1.2 m, the viewing angle is adjusted, and the experimenter is standing in front of the Kinect camera. The range of the human body distance from Kinect in the experiment is 1.7 m–3.5 m. This distance range is divided into 4 equal parts, and 4 different distances from Kinect (1.7 m, 2.3 m, 2.9 m, and 3.5 m) are obtained. The experimenter stood facing the Kinect at these 4 distances and collected data 20 times at each distance, each time collecting 60 frames of data, a total of 4800 frames of data. The 4800 frames of bone data are processed and converted into distance feature and angle feature values, and their average values are calculated.

Figures 7 and 8 are the eigenvalues of the distance features and angle features extracted in this paper at different distances from the experimenter to the Kinect. It can be seen from the histogram that these distance features and angle feature values are basically not affected by distance changes,
can maintain a certain invariance, and have a certain degree of aggregation in the feature space.

5.2. Kinect Placed at Different Heights Will Affect the Features. This experiment verifies the effect of Kinect placement height on features. Excluding the influence of other factors, the experimenter stood at a distance of 3.5 m from the Kinect, facing directly in front of the Kinect camera. Place the Kinect at 40 cm, 80 cm, 120 cm, and 160 cm from the ground. At these four heights, 10 times of data are collected at each height, and 60 frames of data are collected each time, for a total of 2400 frames of data. Convert these
2400 frames of skeleton data into distance features and angle feature values to obtain 2400 frames of feature data, and calculate the average value of the feature data.

Table 3 shows the eigenvalues of the angle features extracted in this paper at different distances from the experimenter to the Kinect. From the experimental results, the extracted angle features can be placed under different heights in Kinect, and the feature data can maintain good invariance.

5.3. Experimenter’s Angle Relative to Kinect Affects Characteristics. This experiment verifies the effect of changes in the angle of the human relative to the Kinect on the features. Excluding the influence of other factors, the experimenter stood at a distance of 2.1 m from the Kinect and placed the Kinect at a height of 120 cm from the ground. This article is facing the Kinect, respectively, turning the body to the left by approximately 45° and turning the body to the right by approximately 45°, collecting 10 times in each direction, and collecting 60 frames of data each time, a total of 1800 frames of data. The 540 frames of bone data are converted into distance features and angle features. The experimental results are shown in Figures 9 and 10.

From the results of the experiments in Figures 9 and 10, it is found that when the subjects turned 45° left and 45° right, the distance characteristics changed, which was different from the characteristic data when the subjects turned around. Therefore, if the distance feature is considered to be an available feature set representing the human pose, the human body should try to maintain the positive pose of the Kinect. Generally speaking, the distance and angle features obtained after normalizing the related node data can satisfy the changes of the Kinect space coordinate system and maintain good invariance. Therefore, using this feature vector to describe the human pose can ensure good consistency between samples of the same type. It can obtain a stable pose description model, which is helpful for item reading.

6. Conclusion

The human body is a very complex assemblage, and it can express so many gestures that it is impossible for us to recognize them all. Human gesture and motion recognition has many applications in areas such as human-computer interaction, motion analysis, intelligent video surveillance, form and language understanding, health monitoring, and virtual reality. However, there are still many fundamental problems to be solved in the field of human pose recognition using computer vision. Needless to say, only recording a person’s pose with a traditional camera results in a loss of information. As image capture technology improves, the Kinect body camera, capable of collecting information in depth, has created new opportunities for researchers. Human motion tracking must meet the requirements of high-fidelity real-time algorithms to ensure that the tracked motion trajectories are recorded continuously.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the Science and Technology Program of Shaanxi Province (2017XT-02).

References

[1] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, “Real-time moving object detection in high-resolution video sensing,” Sensors, vol. 20, no. 12, p. 3591, 2020.
[2] P. G. Bhat, B. N. Subudhi, T. Veerakumar, V. Laxmi, and M. S. Gaur, “Multi-feature fusion in particle filter framework for visual tracking,” IEEE Sensors Journal, vol. 20, no. 5, pp. 2405–2415, 2020.
[3] F. Gong, C. Li, W. Gong et al., “A real-time fire detection method from video with multifeature fusion,” Computational Intelligence and Neuroscience, vol. 2019, no. 1, Article ID 1939171, 2019.
[4] E. Hassan, “Multiple object tracking using feature fusion in hierarchical LSTMs,” The Journal of Engineering, vol. 2020, no. 10, pp. 893–899, 2020.
[5] F. Tang, X. Lu, X. Zhang, S. Hu, and H. Zhang, “Deep feature tracking based on interactive multiple model,” Neurocomputing, vol. 333, no. MAR.14, pp. 29–40, 2019.
[6] W. Tong, H. E. Xiao, and L. U. Ying, “Visual saliency and multi-feature fusion for object tracking,” Journal of Frontiers of Computer Science & Technology, vol. 11, no. 3, pp. 438–449, 2017.
[7] X. Liu, M. Xu, S. J. Peng, W. Fan, and J. X. du, “Efficient human motion capture data annotation via multi-view spatiotemporal feature fusion,” IET Signal Processing, vol. 12, no. 3, pp. 269–276, 2018.
[8] L. I. Hongyang, J. Chen, and H. U. Ruimin, “Multiple feature fusion in convolutional neural networks for action recognition,” Wuhan University Journal of Natural Sciences, vol. 22, no. 1, pp. 73–78, 2017.
[9] S. Y. Cho, S. Y. Lee, J. H. Lim, and S. J. Park, “Simultaneous motion-tracking and localisation of a person based on the integration of multiple IMUs and depth camera,” IET Radar, Sonar and Navigation, vol. 11, no. 11, pp. 1679–1688, 2017.
[10] P. F. Shan, “Image segmentation method based on K-mean algorithm,” EURASIP Journal on Image and Video Processing, vol. 2018, no. 1, Article ID 81, 2018.
[11] H. Wang, N. S. Kong, and J. Wen, “Robust video tracking algorithm: a multi-feature fusion approach,” IET Computer Vision, vol. 12, no. 5, pp. 640–650, 2018.
[12] B. Du, L. Ru, C. Wu, and L. Zhang, “Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 12, pp. 9976–9992, 2019.
[13] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, “Deep learning models for real-time human activity recognition with
smartphones,” Mobile Networks and Applications, vol. 25, no. 2, pp. 743–755, 2020.

[14] M. Tabesh, T. Hoffmann, S. Vollmer, H. Schüttrumpf, and R. M. Frings, “In-situ measurement of river-bed sediment porosity using structure-from-motion image analysis,” Geomorphology, vol. 338, no. AUG.1, pp. 61–67, 2019.

[15] Y. Masatoshi, H. Chihiro, and U. Hideharu, “3P-146 study of temperature effect on amoeboid locomotion by motion image analysis (Cell biology, The 47th Annual Meeting of the Biophysical Society of Japan),” Biophysics, vol. 49, supplement, pp. S175–S176, 2009.

[16] T. Matsuda, K. Maeda, A. Yamaguchi et al., “Microscopic image analysis for soil particles motion due to horizontal flow by using Piv,” Journal of Japan Society of Civil Engineers Ser B2, vol. 73, no. 2, pp. I_529–I_534, 2017.

[17] X. Wang, C. Chen, J. Yuan, and G. Chen, “Color reproduction accuracy promotion of 3D-printed surfaces based on microscopic image analysis,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 34, no. 1, p. 2054004, 2020.

[18] H. Song and M. Brandt-Pearce, “A 2-D discrete-time model of physical impairments in wavelength-division multiplexing systems,” Journal of Lightwave Technology, vol. 30, no. 5, pp. 713–726, 2012.

[19] S. Milenkovic and A. N. Bondar, “Motions of the SecA protein motor bound to signal peptide: insights from molecular dynamics simulations,” Biophysica et Biophysica Acta, vol. 1860, no. 2, pp. 416–427, 2018.

[20] S. Zamiran and A. Osouli, "Seismic motion response and fragility analyses of cantilever retaining walls with cohesive backfill," Soils and Foundations, vol. 58, no. 2, pp. 412–426, 2018.

[21] R. Li, Z. Liu, and J. Tan, “Human motion segmentation using collaborative representations of 3D skeletal sequences,” IET Computer Vision, vol. 12, no. 4, pp. 434–442, 2018.

[22] J. Peng, A. Song, C. Wu, H. Wei, and Z. Song, "Human-robot-robot-cooperation based uncalibrated visual servoing control for mobile robotic manipulators without joint-state feedback," Robot, vol. 39, no. 2, pp. 197–204, 2017.

[23] W. P. Hung, C. H. Chang, and T. H. Lee, "Real-time and non-contact impulse radio radar system for µm movement accuracy and vital-sign monitoring applications," IEEE Sensors Journal, vol. 17, no. 8, pp. 2349–2358, 2017.

[24] A. Jevtić, A. F. Valle, G. Alenyà et al., "Personalized robot assistant for support in dressing," IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 3, pp. 363–374, 2019.

[25] X. U. Da and Q. L. Jiao, "Research on the construction method of human body posture examples based on improved flower pollination algorithm and K-means," Binggong Xuebao/Acta Armamentarii, vol. 39, no. 12, pp. 2459–2469, 2018.

[26] D. Liu, X. Wang, and J. X. Pian, "Inertial motion tracking algorithm for human arm based on Lie algebra," Shenyang Gongye Daxue Xuebao/Journal of Shenyang University of Technology, vol. 39, no. 1, pp. 73–77, 2017.

[27] D. Groos, L. Adde, and E. Ihlen, “Approaching human precision on automatic markerless tracking of human movements,” Gait & Posture, vol. 81, no. 11, pp. 117–118, 2020.