*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**

Volume **9**, Number **4**, November **2024**

# 3D Dance Movement Recognition Based on Somatic Interaction Devices and Neural Networks

Qi Zhou

Hunan University of Arts and Science, Changde 415000, P. R. China
quqibinggan77zq@163.com

Dai-Li Jiang*

Yulin Normal University, Yulin 537000, P. R. China
17377071827@163.com

Garrett Wang

Graduate School of Christian Studies
Hongik University, Mapo-gu, Seoul 06695, South Korea
ns5108@163.com

*Corresponding author: Dai-Li Jiang
Received October 25, 2023, revised February 17, 2024, accepted May 4, 2024.

Abstract. *While traditional cameras may not be able to accurately recognise human movements in 3D space, somatosensory interactive devices can achieve more accurate movement recognition through more advanced sensing technologies and algorithms. In addition, the somatosensory interaction device can capture and feedback the information of dance movements in real time, which enables the dancers to adjust and improve their movements in time, thus improving the performance quality of dance. Therefore, this paper proposes a 3D dance movement recognition method based on somatosensory interaction devices and neural networks. Firstly, the principle of measuring depth information based on Kinect, a somatosensory interactive device, is investigated, and the conversion between the world coordinates and the coordinates of the depth image is analysed. Secondly, the depth image obtained by Kinect is used to detect the human body, including edge information and depth change information, to achieve preprocessing and human body localisation. A human joint feature extraction method based on random decision forest is proposed. Then, a Hierarchical Extreme Learning Machine (H-ELM) network containing three hidden layers is constructed using the depth map of human joint features as input features. The number of nodes in each layer is 1024, 512, and 128 respectively. The number of nodes in the input layer is the dimension of the human body's keypoints, and the number of nodes in the output layer is the number of gesture categories. Finally, a 3D movement dataset containing 10 dance movement categories was constructed and tested. The results show that although the training time is increased, the H-ELM algorithm has the highest recognition accuracy in terms of accuracy.*
**Keywords:** Human posture; dance movement recognition; Kinect; single-hiddle layer feedforward neural network; H-ELM

1. **Introduction.** 3D dance movement recognition can be applied to dance teaching and training by using technologies such as sensors to accurately capture and analyse dancers' movements [1, 2], provide real-time feedback and guidance, and help learners better understand and master dance skills. By recognising dance movements, more accurate feedback

and guidance can be provided for dance teaching to help students improve their movement skills. Meanwhile, for dance performers, they can improve the quality of their dance performances by conducting self-assessment and training through the 3D dance movement recognition system. Meanwhile, in dance performances, the dancers' movements can be captured by sensors and combined with virtual stage or virtual reality technology to create more immersive stage effects.

There are several problems with traditional techniques that employ sensors for dance movement recognition [3, 4]. For example, traditional sensors usually need to be worn or mounted on different parts of the dancer's body, such as the wrist, waist, or ankle [5], which limits the dancer's comfort and freedom, and increases the complexity of deploying and using the system. The data collected by the sensors requires complex processing and parsing to extract information about the dancer's movements from them. However, sensor data may be subject to a number of technical limitations in pose recognition and motion tracking, resulting in reduced data accuracy and reliability.

In contrast, somatosensory interaction devices have certain advantages in dance movement recognition. Somatosensory interaction devices are usually in the form of cameras or depth sensors [6, 7], eliminating the need for dancers to wear or install any additional sensor devices. This increases the comfort and freedom of the dancers and facilitates the rapid deployment and use of the system. Somatosensory interaction devices can capture dancers' movements in real time and process them instantly, which can provide more accurate results than traditional sensors. For example, depth sensors can provide information about the depth of an image, thus better capturing the spatial position and posture of a dancer [8, 9]. Somatosensory interaction devices are often flexible enough to adapt to different dance movements and environments. With appropriate algorithms and models, they can recognise and analyse various types of dance movements, including jumps, turns, spins, etc., for different dance styles and genres. All in all, somatosensory interaction devices offer better comfort, real-time, accuracy and system flexibility in dance movement recognition, and can provide better user experience and performance compared to traditional sensors.

However, the accuracy of 3D human movement recognition based on somatosensory interaction devices still falls short of the desired requirements. Therefore, a 3D dance movement recognition method based on somatosensory interaction devices and neural networks is proposed. The aim of this study is to reduce the limitations of traditional sensors by using somatosensory interaction devices to adapt to different dance movements and environments. Meanwhile, the neural network model allows for efficient feature extraction and classification of dance movements, thus achieving higher accuracy in movement recognition.

1.1. **Related Work.** Microsoft's somatosensory interaction device Kinect was originally designed for gaming, but the role of Kinect has gone far beyond gaming. Kinect-based research has become a hot research direction in the fields of pattern recognition, computer vision, virtual reality and human-computer interaction [10, 11].

Schwarz et al. [12] proposed a whole body pose estimation of the human body using anatomical markers, Kinect to obtain depth maps and a human skeleton model, and Geodesic Distances to measure distances between body parts using Geodesic Distances. Wang et al. [13] uses SVM classifier to classify the depth image information of human body into multiple parts of the body. Gahlot et al. [14] used the human body image obtained by Kinect's depth sensor to recognise the 3D human body pose. In view of the cheapness of Kinect, many medical experts also bring this advantage into medical rehabilitation, and track human limbs to determine their positions [15], so as to identify

the movements that enter the body. Vieira et al. [16] proposed the use of games for rehabilitation therapy, which can increase human motivation. The use of Kinect in virtual scenarios can effectively enhance the quality of rehabilitation, improve the psychological quality of patients and reduce their negative emotions.

Research on neural network based human pose recognition has now made great strides in processing video streams in real time and achieving state-of-the-art results on multiple datasets. Kamel et al. [17] proposed a method for real time human 2D pose estimation using Convolutional Neural Networks (CNN). The method uses a kind of part affinity fields to represent the spatial relationships between captured body parts. Rohan et al. [18] proposed a novel method for human pose estimation using CNN. The method uses a novel network architecture called convolutional pose machine which captures the long-term dependencies between body parts. Tian et al. [19] proposed a method for human pose estimation using deep residual networks. The deep residual network uses a stacked module that captures local and global relationships between body parts.

Compared to CNNs and deep residual networks, Extreme Learning Machine (ELM) based on Single-hidden Layer Feedforward Neural Network (SLFN) [20] requires only one iteration to complete the training, whereas the former two require multiple iterations to converge. ELM is insensitive to noise and outliers in the training data and has a strong generalisation ability [21]. ELM has a small computational effort and is suitable for resource constrained devices such as Kinect.Therefore, ELM has a greater advantage in human posture recognition.

1.2. **Motivation and contribution.** In the depth information obtained by Kinect, offset normalisation ensures that features are depth invariant. However, any single such feature provides only a weak signal.

In addition, ELM is widely used in practical applications for tasks such as regression, classification and feature learning. However, since the training samples in classical ELM are always the original dataset, which makes the robustness and generalisation performance of 3D dance movements limited, the Hierarchical Extreme Learning Machine (H-ELM) is introduced in this work [22, 23]. The main innovations and contributions of this work include:

(1) A randomised decision forest-based feature extraction method for human joints is proposed to address the problem that any single feature captured using Kinect can only provide a weak signal about which body component the pixel belongs to.

(2) An H-ELM-based dance movement recognition method is proposed. An H-ELM network containing three hidden layers is constructed. The number of nodes in the input layer is the dimension of the human body's keypoints, and the number of nodes in the output layer is the number of dance movement categories.

## 2. Depth information based on Kinect, a somatosensory interaction device.

2.1. **Introduction to Kinect.** Kinect is a somatosensory interactive device publicly released by Microsoft in 2010 [24]. Kinect has three lenses. The middle lens is an RGB colour camera that acquires 30 frames of colour images per second as shown in Figure 1. In addition, the Kinect has an infrared 3D depth sensor that can be used to detect the relative position of the player. There is a set of microphone arrays (4) on either side of the Kinect for sound source localisation and speech recognition. At the bottom is a base with built-in motors for adjusting the tilt angle, the specific parameters of which are shown in Table 1.

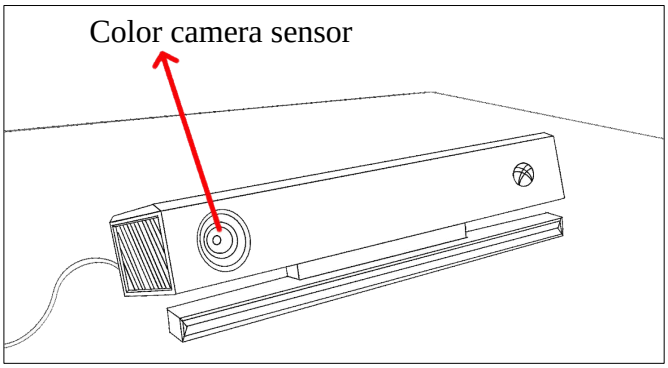


Figure 1. Colour camera sensor

Table 1. Structure and function of Kinect

| Software | Functionality |
|---|---|
| RGB Camera | Resolution 1280×960, Viewing angle: 43° vertically, 57° horizontally |
| Infrared emitter | Emits infrared light to create a diffuse image |
| Infrared receiver | Resolution 640×480, receive scattering image information, build depth image |
| Motor | Adjusting Kinect Elevation Angle |
| Array Microphone | Receive sound signals |

The Kinect sensor provides three major data sources [25], including depth data streams, colour data streams, and raw audio data, which correspond to the three processing processes of skeleton tracking, identity recognition, and speech recognition, respectively. The Kinect was initially designed for gaming, but as gaming enthusiasts as well as academics around the globe have explored the capabilities of the Kinect, it has become much more useful than just the gaming category.

Kinect offers a range of novel and powerful technologies such as depth sensing, skeleton tracking, speech and human recognition that are being experimented with as a new form of human-computer interaction in industries such as education, healthcare, video surveillance and more. In addition to capturing normal video, Kinect is unique in its ability to capture depth data, which allows it to obtain the distance between the target object and the sensor within the viewing angle. The Kinect has high application value in 3D dance movement recognition. Specifically, Kinect can convert dance movements into corresponding digital signals, which can also play an important role in virtual reality and game development.

2.2. **Principle of depth information measurement.** Kinect uses a technique called structured light to measure depth information. Structured light works by projecting a special pattern of light onto an object, and then using the Kinect's infrared camera to capture the pattern. Based on changes in the shape and position of the captured light patterns, Kinect can calculate how close the object is to the camera, and thus obtain depth information. This technique enables real-time depth measurement and is suitable for a wide range of scenarios and environments.

According to the calibration relationship between the selected reference image and the light source, the distance from the object to the light source is calculated, a 3D image is constructed, the distance is normalised and converted into a grey value image, and finally a depth image is output. The field of view of the Kinect sensor is limited, as shown in Figure 2. With a pyramidal reach, the infrared ray covers a bigger cross-section of the economic field of view the farther it is from the camera. Accordingly, the depth value of each pixel represents the distance between the item in the field of vision and the camera,

rather than the height and breadth of the picture and the actual position inside it being a one-to-one connection.

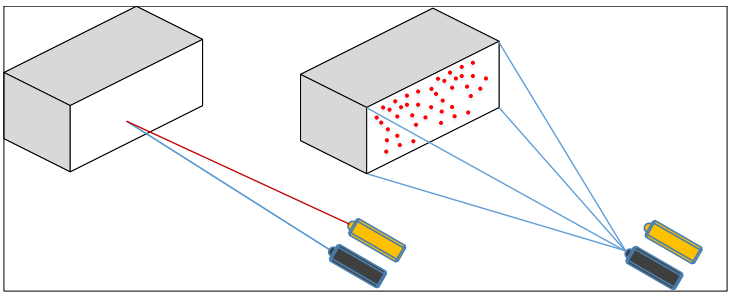

Figure 2. Kinect field of view

In depth frames, each pixel occupies 16 bits, i.e., two bytes, while the depth value occupies 13 bits, as shown in Figure 3, where the depth value is stored in bits 3 to 15.

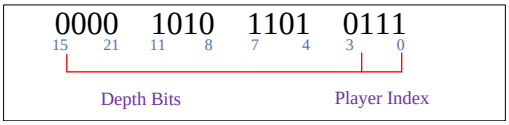


Figure 3. Layout of the depth bits

In order to use Kinect for depth measurement, firstly, we need to calibrate Kinect, including the calibration of RGB color camera and IR camera, and the corresponding coordinate relationship between depth map and RGB map. The original value $d_r$ of 3D space point $P$ is defined as shown below:

$$d = K \tan(H \cdot d_r + L) - O \tag{1}$$

After obtaining the depth of the image, it is possible to recover the world coordinates of the point $P$, as shown below:

$$\begin{cases} x_w = \left(x_d - \frac{w}{2}\right) \cdot (z_w + D') \cdot F \cdot \left(\frac{w}{h}\right) \\ y_w = \left(y_d - \frac{h}{2}\right) \cdot (z_w + D') \cdot F \\ z_w = d \end{cases} \tag{2}$$

where $(x_d, y_d, z_d)$ is the depth coordinate, $(x_w, y_w, z_w)$ is the actual coordinate, $D' = -10$, $F = 0.0021$, and the resolution of the Kinect is $w \times h = 640 \times 480$.

Figure 4 shows a schematic diagram of the world coordinates and the coordinates of the depth image. In the world coordinate system, it is the Kinect sensor as the origin, while the depth map coordinate system uses the depth image origin as the origin of the coordinate system.

## 3. Kinect-based human body detection and feature extraction.

### 3.1. Preprocessing and human body localisation.

In this work, the depth information obtained from Kinect is used to detect the human body, including 2D edge detection and 3D shape detection, respectively. The depth image is first processed with noise reduction and smoothing, and then the human body is localized using the second-order head detection process.

Assuming a continuous space, some points in the depth map obtained using Kinect are subject to pixel offsets, which are considered as a kind of noise. In order to eliminate the

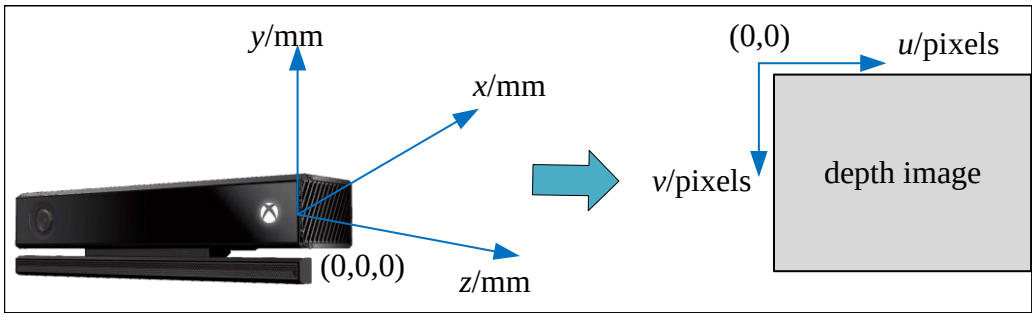


Figure 4. The world coordinates and depth coordinates

interference caused by this noise, it is necessary to recover their depth values. For missing points, assuming that their neighboring pixels are very similar, all 0-pixel points are first assumed to be empty and filled using the nearest neighbor interpolation algorithm.

In order to determine the possible regions of the human body, in this paper, edge detection is used to find the edge of the depth map, and then 2D chamfer distance matching algorithm [26] is used. The specific steps are as follows:

(1) Select a specific 2D template (Sobel operator edge detection template) as a reference template.

(2) Calculate the distance between the depth image obtained by Kinect and the reference template to generate a distance image. The distance formula for each pixel point in the distance image can be calculated using Euclidean distance.

Assuming that $ref(x, y)$ denotes the value of point $(x, y)$ on the reference template and $img(x, y)$ denotes the value of point $(x, y)$ on the depth image, Equation (3) can be used to compute the distance value from each pixel point of the image.

$$D(x,y) = \sqrt{\sum_{i,j} (ref(i,j) - img(x+i, y+j))^2} \tag{3}$$

(3) Using 2D chamfer distance matching algorithm, the distance image is matched to find out the edge location in the depth image that is most similar to the reference template.

The regression function for the depth value and height of the human head is shown below:

$$-y = p_1 \cdot x^3 + p_2 \cdot x^2 + p_3 \cdot x + p_4 \tag{4}$$

where $p_1 = -1.3835 \times 10^{-9}$, $p_2 = 1.8435 \times 10^{-5}$, $p_3 = -0.091403$, $p_4 = 189.38$

The standard height of the head in the depth map is calculated and then the head is searched within a certain range.

$$R = 1.33h/2 \tag{5}$$

where $h$ is calculated using Equation (4).

The hemisphere was used as a 3D head model, which was then fitted. The depth was normalized by the CR to obtain a circular region with a radius of $R_t$.

$$d_n(i,j) = d(i,j) - \min(d(i,j)), \quad i,j \in CR \tag{6}$$

where $d(i, j)$ is the depth value of pixel $(i, j)$ and $d_n(i, j)$ is the normalized depth value.

The mean square error between the 3D model and the circular region $T(i, j)$ is shown below.

$$Er = \sum_{i,j \in CR} |d_n(i,j) - T(i,j)|^2 \tag{7}$$

3.2. **Human Joint Feature Extraction Based on Randomised Decision Forest.** In the RGB image, a person standing on the ground can use gradient features to detect the demarcation line between the human foot and the ground. Whereas in depth images the depth values of human foot and ground plane are same, so the overall contour of human body cannot be detected using common boundary detectors. At the same time, based on the actual situation it is considered that the human foot is usually upright, for any object touched by the human body, the parts of the object and the human body may also have the same depth value.

Using a region expanding method, the depth map's total human body contour is extracted. It is presumed that the human target's surface depth values are continuous within a given area. The location of the seed, or the center of the map found by fitting the 3D model, is where the algorithm's rules begin. The expanding region is determined by how similar the region is to its surrounding pixels. The following definition applies to the similarity between the pixels $x$ and $y$ in the depth map.

$$S(x,y) = |d(x) - d(y)| \tag{8}$$

where $d(\cdot)$ denotes the depth value, the depth of the region is the average depth of all pixels in the region.

$$d(R) = \frac{1}{N} \sum_{i \in R} d(i) \tag{9}$$

It is assumed that the coordinates and velocity of the target change smoothly between neighboring frames. Firstly, the center of the detection block is found. Then, the 3D coordinates and velocity of the human body are calculated for each frame. The coordinates are taken straight from the depth map, and the neighboring frames' coordinate locations may be used to determine the velocity. The energy values for spatial and velocity changes are shown as follows:

$$E = (c - c_0)^2 + (v - v_0)^2 \tag{10}$$

The segmentation of body parts is considered as a pixel-by-pixel classification issue when human contour tracking is implemented. While the appearance of individual body parts still changes greatly depending on the scenario, evaluating each pixel independently eliminates combinatorial search between distinct body joints. A motion capture database is used to sample a variety of stances from people of all sizes and shapes. From this, training data in the form of realistic synthetic depth maps is produced. This work uses eighteen body part components. There are basic features for depth comparison. The features are calculated in the following way for a given pixel $x$.

$$f_\theta(I,x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right) \tag{11}$$

where $d_I(x)$ is the depth of the image $I$ at pixel $x$ and the parameter $\theta = (u,v)$ describes the offsets $u$ and $v$.

Normalising the offsets guarantees that the features are depth invariant. For a given point on the body, whether it is near or far from the camera, (feature computation) gives a fixed offset in world space. However, any single such feature can only provide a weak signal about which component of the body the pixel belongs to. To address this problem,

this work proposes the use of randomised decision forests [27, 28] to combine individual features in order to accurately distinguish all training components.

A randomised decision forest is a totality of $T$ decision trees, each with branch nodes and leaf nodes. Each branch node consists of a feature $f_\theta$ and a threshold $\tau$. To classify the pixels $x$ of an image $I$, the feature values are obtained by continuously calculating Equation (11) starting from the root node. Then, the tree branches to the left or right based on the comparison of the feature value with the threshold value $\tau$. The leaf nodes of the tree store the trained distribution $P_t(c|I,x)$ of body component labels $c$. The distributions of all trees are averaged and used as the final classification.

$$P(c|I,x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,x) \tag{12}$$

Given a random pair of branching candidates $\phi = (\theta, \tau)$ (feature parameter $\theta$ and threshold $\tau$), use each $\phi$ to partition the sample set $Q = \{(I,x)\}$ into left and right subsets.

$$Q_l(\phi) = \{(I,x)|f_\theta(I,x) < \tau\} \tag{13}$$

$$Q_r(\phi) = Q/Q_l(\phi) \tag{14}$$

Determine $\phi$ by solving the maximum information gain problem.

$$\phi^* = \arg\max G(\phi) \tag{15}$$

$$G(\phi) = H(Q) - \sum_{s\in\{l,r\}} \frac{|Q_s(\phi)|}{Q} H(Q_s(\phi)) \tag{16}$$

where $H(Q)$ is Shannon entropy, calculated on the human label $I_r(x)$ normalized histogram.

If the maximum gain $G(\phi^*)$ is still large (large enough) and the depth of the tree has not reached its maximum, the recursion continues in the left and right subsets $Q_l(\phi^*)$ and $Q_r(\phi^*)$.

Human joint point recognition is where a single input depth image is segmented into dense probabilistic body component labels, with components defined as body parts that are spatially similar to the skeletal joint of interest. The inferred components are reprojected into world space, localising the spatial pattern of each component distribution to form a prediction with plausible weights for the 3D position of each skeletal joint. Pixel-by-pixel information can be inferred using body component recognition. This information for all pixels is aggregated to form reliable predictions of 3D skeletal joint positions. These predictions are the final output of the algorithm and can be self-initialised and recovered from failure in the tracking algorithm.

It is possible to estimate the global 3D centre using a per-component cumulative probability (distribution). However, irrelevant pixels can severely degrade the quality of such a global estimate. Therefore, a weighted mean shift-based approach is used in this paper. The density estimates of the body components are defined as follows:

$$f_c(\hat{x}) \propto \sum_{i=1}^{N} w_{ic} \exp\left(-\left\|\frac{\hat{x}-\hat{x}_i}{b_c}\right\|^2\right) \tag{17}$$

where $\hat{x}$ is the coordinates in the 3D world space, $N$ is the number of image pixels, $w_{ic}$ is the pixel weights, $\hat{x}_i$ is the reprojection of the image pixel $x_i$ into the world space for a given depth $d_I(x_i)$, and $b_c$ is the width of each of the trained component's width.

The pixel weights $w_{ic}$ take into account both the probability reasoned over the pixel (which body component it belongs to) and the surface area of the pixel in world space.

$$w_{ic} = P(c|I, x_i) \cdot d_I(x_i)^2 \tag{18}$$

This improves the accuracy of joint forecasts significantly and guarantees that the density estimates are depth invariant. Pre-accumulating across a limited number of components can provide a posteriori probability $P(c|I, x)$, depending on how the body components are defined.

## 4. Neural network-based dance movement recognition.

### 4.1. Single hidden layer feedforward neural network.

The components of SLFN, one of the most basic neural network designs, are an input layer, a hidden layer, and an output layer. Every layer is made up of many neurons connected exclusively forwardly—there are no feedback connections among the neurons. Accordingly, the neurons in the input layer and the hidden layer are only linked to the corresponding neurons in the output layer and the hidden layer, respectively. Due to their high capacity for learning, SLFNs are extensively employed in a variety of sectors.

Suppose a SLFN has a total of $N$ sample data. For each sample, $x = [x_1, x_2, \ldots, x_n]^T$ is the input vector, $b$ is the threshold of the hidden layer node, and $g(x)$ is the hidden layer excitation function; $b_o$ is the threshold of the node in the output layer, and $f(x)$ is the the output layer excitation function. Each sample output is shown as follow:

$$\begin{aligned} y &= f\left(g\left(w_i x + b\right)\beta_i + b_o\right) = g\left(w_i x + b\right)\beta_i + b_o \\ &= \left[g(w_1 x + b_1)g(w_2 x + b_2)\ldots g(w_i x + b)\right]\beta_i + b_o \end{aligned} \tag{19}$$

For the usual SLFN learning algorithm, the weights $w$, the weights $\beta$, the activation threshold $b$ of the implicit layer, and the activation threshold $b_o$ of the output layer, all need to be optimised by continuous iteration.

### 4.2. H-ELM-based dance movement recognition.

Although SLFN has good learning ability, it cannot achieve complex dance movement recognition, which is because the human joint depth feature maps extracted by Kinect above are non-linear data. Therefore, deeper or more complex neural network structures are often required in practical applications. In contrast, ELM can be regarded as a special case of a single hidden-layer feedforward neural network, which uses randomly initialised hidden-layer weights and where the weights of the output layer are obtained by solving a linear equation during training.

ELMs are widely used in practical applications for tasks such as regression, classification and feature learning. However, since the training samples in classical ELM are always the original dataset, which makes the robustness and generalisation performance of 3D dance movements limited, H-ELM is introduced in this work.

The H-ELM algorithm is divided into two main stages: unsupervised hierarchical feature extraction and supervised classification. The first stage extracts the multilevel sparse matrix of the input data, and the latter stage makes the final decision based on the original ELM-based regression. In the first stage, the unsupervised multilayer feature encoding employs ELM sparse auto-coding. Auto-coding is applied as some kind of feature extractor in the multilayer learning framework so that the coded output approximates the

original input and minimises the reconstruction error. Mathematically, the input data $x$ in autocoding can be represented at a higher level by a deterministic mapping.

$$y = h_\theta(x) = g(A \cdot x + b), \theta = \{A, b\} \tag{20}$$

where $g(\cdot)$ denotes the activation function, $A$ is the implied layer weights $d' \times d$, and $b$ denotes the bias.

The purpose of autocoding is to make the reconstructed output data approximate the original input data. The input weights for ELM sparse autocoding are randomly generated, and the randomly generated input weights are able to satisfy any approximation of the input data during training. Once the autocoding is initialised, no further updates and adjustments are required throughout the training process. Unsupervised feature learning in H-ELM should be preceded by transforming the input raw data into the random feature space of the ELM, which allows the use of implicit information in the training samples.

After $N$ layers of unsupervised learning finally high-level sparse features are obtained. The output of each hidden layer can be expressed as follow:

$$H_i = g\left(H_{i-1} \cdot \beta\right) \tag{21}$$

where $H_i$ is the output of the $i$-th layer, $H_{i-1}$ is the output of the $i-1$-th layer, and $g(\cdot)$ is the activation function of the hidden layer.

H-ELM algorithm introduces the pruning mechanism of hidden layer neurons and the adding mechanism of output layer neurons, which can automatically optimize the network structure by adding specific data sets and improve the generalization ability and prediction accuracy of the model. The implementation of H-ELM is shown in Algorithm 1.

**Algorithm 1** H-ELM Implementation Process

**Input:** $X$: $N \times D$ input data, $N$ is the number of samples, $D$ is the feature dimension.
        $Y$: $N \times C$ labels, $C$ is the number of categories.
**Output:** trained $W$, $b$, $W_{\text{out}}$
# Parameter initialisation.
# Network parameters.
$L$: number of implicit layers.
$M$: number of nodes per implicit layer.
$S$: 'ReLU' activation function.
**for** $i = 1$ to $L$ **do**
    $W[i]$ = Initialise weight matrix $(M[i], M[i-1])$;
    $b[i]$ = Initialise bias vector $(M[i])$;
**end for**
# Forward propagation.
$H[1] = S(W[1] \cdot X + b[1])$;
**for** $i = 2$ to $L$ **do**
    $H[i] = S(W[i] \cdot H[i-1] + b[i])$;
**end for**
# Output weights.
$W_{\text{out}}$ = find pseudo-inverse $(H[L] - Y)$;
# Loss function.
$L$ = mean square error $(Y, W_{\text{out}} \cdot H[L])$.
**return**

The steps of H-ELM based dance movement recognition are as follows:

(1) Use the human movement image sequences captured by Kinect sensor and extract the depth map of human joint point features as input features. Set the dance movement category labels, such as kicking, squatting, and waving.

(2) Construct an H-ELM network with three hidden layers. The number of nodes in each layer is 1024, 512, and 128, respectively. the number of nodes in the input layer is the dimension of the human body's key points, and the number of nodes in the output layer is the number of dance movement categories.

(3) Initialise the weight matrix from the input layer to each hidden layer and the hidden layer bias vector. For the training data, forward propagation iteratively calculates the output of each hidden layer. Finally, the matrix pseudo-inverse of the last hidden layer is calculated from the output layer labels to obtain the output layer weight matrix.

(4) After obtaining the trained H-ELM network, the test set input is passed forward to compute the predicted output. Finally, the classification performance on the test set is evaluated.

## 5. **Experimental results and analyses.**

### 5.1. **Dataset.**

In order to evaluate the proposed 3D dance movement recognition method based on Kinect and H-ELM, we constructed a 3D movement dataset containing 10 dance movement categories. The dance movement categories include basic movements such as kicking, spinning and waving.

The dataset was acquired using a Microsoft Kinect v2 sensor, which contains both an RGB camera and a depth camera. The tester faces the Kinect sensor and completes a given action at 30 frames per second. We recorded both RGB video sequences and depth video sequences. To remove the background interference, we only utilise the depth information and extract the 3D coordinates of 18 major keypoints, i.e., heads, necks, shoulders, elbows, wrists, hips, knees, and ankles, in each video frame in real-time based on the OpenPose toolkit as input features.

Table 2. Model training parameters

| **Parametric** | **Instructions** |
|---|---|
| Acquisition equipment | Microsoft Kinect v2 |
| Modal | RGB Video Depth Video |
| Deflection rate | 1920×1080 |
| Frame rate | 30 fps |
| Action category | 10 (legs, spins, etc.) |
| Number of key points | 18 (OpenPose detection) |

The final dataset contains 100 video samples, 10 samples for each action category. All video samples are segmented, and each video contains one complete action cycle. This dataset will be used to train and test the proposed H-ELM-based 3D movement recognition method. The main parameters of the constructed 3D dance movement dataset are shown in Table 2.

The results of Kinect-based 3D dance movement recognition are shown in Figure 5. It can be seen that after using Kinect to obtain the depth information of the dancer, the proposed human joint feature extraction method based on random decision forest can display the 3D skeleton joint position information of the dancer in real time.

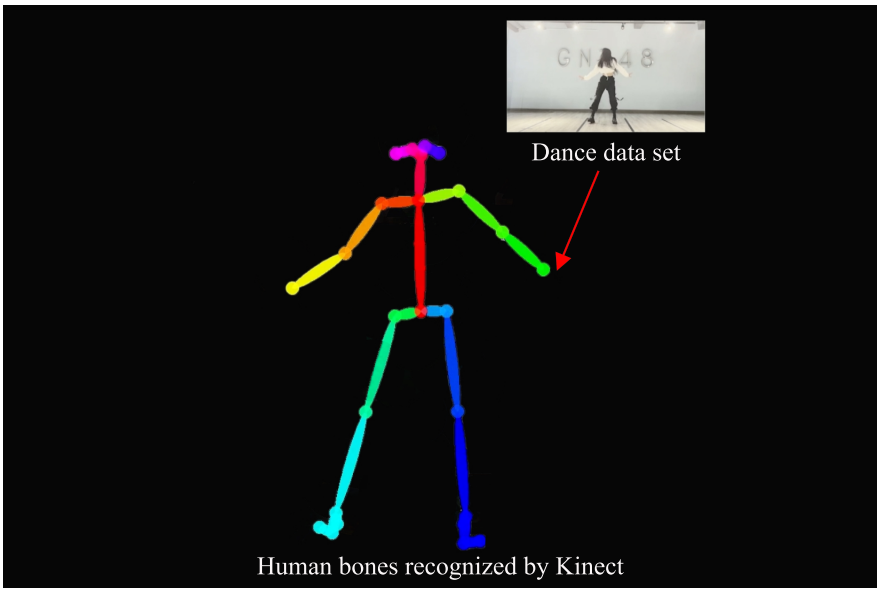


Figure 5. Kinect-based 3D Skeleton Joint Position Information

5.2. **Analysis of recognition results.** The identified 3D skeleton joint position information (feature depth map) was input into the H-ELM model for the 3D dance movement recognition method. The aliasing rate and training time of the three models of BP neural network, ELM and H-ELM were compared, as shown in Figure 6. It can be seen that in terms of training time, the BP algorithm has the shortest training time and the longest is the H-ELM algorithm. However, in terms of accuracy, the H-ELM algorithm has the highest accuracy. This is because H-ELM sparse and multilayer classifies the original data with high recognition accuracy, but it also increases the computational complexity, so it is improving the accuracy at the expense of time.

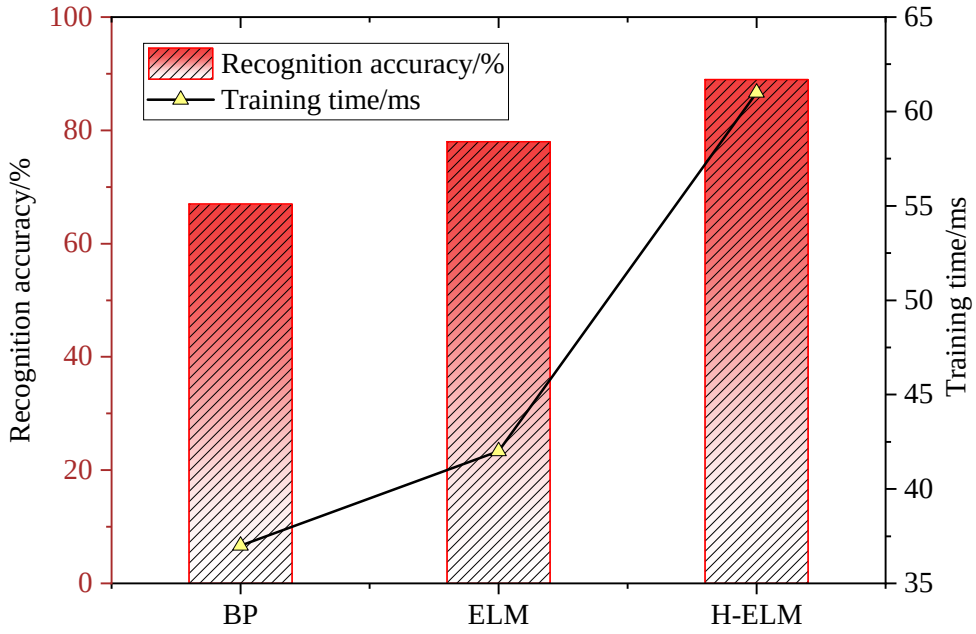


Figure 6. Performance comparison of the three recognition models

6. **Conclusions.** In this work, a 3D dance movement recognition method based on Kinect and H-ELM is proposed. A randomised decision forest based human joint feature extraction method is proposed to address the problem that any single feature captured using Kinect can only provide a weak signal about which component of the body the pixel belongs to. A depth map of human joint features is extracted using the Kinect sensor as input features, and an H-ELM network containing three hidden layers is constructed. The number of nodes in the input layer is the dimension of the human body's keypoints, and the number of nodes in the output layer is the number of dance movement categories. The results show that the proposed human joint feature extraction method based on random decision forest can display the 3D skeleton joint position information of dancers in real time. However, since H-ELM improves the accuracy at the expense of time, it is only suitable for cases with fewer data samples and higher accuracy requirements. Subsequent attempts will be made to solve this problem using On Line Sequential ELM.

**Acknowledgement.** This work is supported by scientific research project of Hunan Provincial Department of Education (No. 22C0391), and the key project of Hunan University of Arts and Science University (No. JGZD2341).

## REFERENCES

[1] W. Ren, O. Ma, H. Ji, and X. Liu, "Human posture recognition using a hybrid of fuzzy logic and machine learning approaches," IEEE Access, vol. 8, pp. 135628-135639, 2020.

[2] S. Zhang, and V. Callaghan, "Real-time human posture recognition using an adaptive hybrid classifier," International Journal of Machine Learning and Cybernetics, vol. 12, pp. 489-499, 2021.

[3] L. Li, G. Yang, Y. Li, D. Zhu, and L. He, "Abnormal sitting posture recognition based on multi-scale spatiotemporal features of skeleton graph," Engineering Applications of Artificial Intelligence, vol. 123, 106374, 2023.

[4] W. Ding, B. Hu, H. Liu, X. Wang, and X. Huang, "Human posture recognition based on multiple features and rule learning," International Journal of Machine Learning and Cybernetics, vol. 11, pp. 2529-2540, 2020.

[5] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," Human-centric Computing and Information Sciences, vol. 9, 40, 2019.

[6] J. Gao, H. Zou, F. Zhang, and T. Y. Wu, "An intelligent stage light-based actor identification and positioning system," International Journal of Information and Computer Security, vol. 18, no. 1/2, 204-218, 2022.

[7] S. P. Xu, K. Wang, M. R. Hassan, M. M. Hassan, and C.-M. Chen, "An Interpretive Perspective: Adversarial Trojaning Attack on Neural-Architecture-Search Enabled Edge AI Systems," IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 503-510, 2023.

[8] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," Communications of the ACM, vol. 56, no. 1, pp. 116-124, 2013.

[9] K. Han, Q. Yang, and Z. Huang, "A two-stage fall recognition algorithm based on human posture features," Sensors, vol. 20, no. 23, 6966, 2020.

[10] D. Brulin, Y. Benezeth, and E. Courtial, "Posture recognition based on fuzzy logic for home monitoring of the elderly," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 5, pp. 974-982, 2012.

[11] J. P. Wachs, M. Kölsch, and D. Goshorn, "Human posture recognition for intelligent vehicles," Journal of Real-time Image Processing, vol. 5, pp. 231-244, 2010.

[12] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, pp. 217-226, 2012.

[13] W.-J. Wang, J.-W. Chang, S.-F. Haung, and R.-J. Wang, "Human posture recognition based on images captured by the kinect sensor," International Journal of Advanced Robotic Systems, vol. 13, no. 2, 54, 2016.

[14] A. Gahlot, P. Agarwal, A. Agarwal, V. Singh, and A. K. Gautam, "Skeleton based human action recognition using Kinect," International Journal of Computer Applications, vol. 975, 8887, 2016.
[15] M. Eltoukhy, C. Kuenze, J. Oh, S. Wooten, and J. Signorile, "Kinect-based assessment of lower limb kinematics and dynamic postural control during the star excursion balance test," Gait & Posture, vol. 58, pp. 421-427, 2017.
[16] A. Vieira, J. Gabriel, C. Melo, and J. Machado, "Kinect system in home-based cardiovascular rehabilitation," Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 231, no. 1, pp. 40-47, 2017.
[17] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 9, pp. 1806-1819, 2018.
[18] A. Rohan, M. Rabah, T. Hosny, and S.-H. Kim, "Human pose estimation-based real-time gait analysis using convolutional neural network," IEEE Access, vol. 8, pp. 191542-191550, 2020.
[19] Y. Tian, W. Hu, H. Jiang, and J. Wu, "Densely connected attentional pyramid residual network for human pose estimation," Neurocomputing, vol. 347, pp. 13-23, 2019.
[20] T. Matias, F. Souza, R. Araújo, and C. H. Antunes, "Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine," Neurocomputing, vol. 129, pp. 428-436, 2014.
[21] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," Neural Computing and Applications, vol. 25, pp. 549-556, 2014.
[22] Y. Lu, F. Weng, and H. Sun, "Numerical solution for high-order ordinary differential equations using H-ELM algorithm," Engineering Computations, vol. 39, no. 7, pp. 2781-2801, 2022.
[23] C. Chen, K. Li, A. Ouyang, Z. Tang, and K. Li, "Gpu-accelerated parallel hierarchical extreme learning machine on flink for big data," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 10, pp. 2740-2753, 2017.
[24] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE Multimedia, vol. 19, no. 2, pp. 4-10, 2012.
[25] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," IEEE Transactions on Cybernetics, vol. 43, no. 5, pp. 1318-1334, 2013.
[26] P. Kaliamoorthi, and R. Kakarala, "Directional chamfer matching in 2.5 dimensions," IEEE Signal Processing Letters, vol. 20, no. 12, pp. 1151-1154, 2013.
[27] L. Rokach, "Decision Forest: Twenty years of research," Information Fusion, vol. 27, pp. 111-125, 2016.
[28] O. Sagi, and L. Rokach, "Explainable decision forest: Transforming a decision forest into an interpretable tree," Information Fusion, vol. 61, pp. 124-138, 2020.