AI Animation Character Behavior Modeling and Action Recognition in Virtual Studio

Yaoyao Xu
Xiangshan Film and Television College, Ningbo University of Finance and Economics, Ningbo, 315000, China

***Abstract*—With the advancement of virtual broadcasting technology, the use of artificial intelligence animated characters in virtual scenes is becoming increasingly widespread. However, there are still a series of challenges and limitations to make the behavior of animated characters more natural, intelligent, and diverse. Therefore, this study proposes a behavior tree based animation character behavior modeling and a short-term memory action recognition method combining human geometric features. The research results indicate that when the behavior modeling model faces different obstacles, the successful avoidance rate is over 80%, and the avoidance reaction time is 0.41s-0.65s. The accuracy and loss function values of the action recognition method gradually converge to 1 and 0 with the quantity of iterations grows. For the recognition of seven types of actions, the accuracy of raising the left hand, raising the right hand, waving the left hand, and waving the right hand reaches 100%, and the recall rate of raising the right hand is 100%. The majority of action types have F-value scores above 0.9. Relative to the recurrent neural network model, the accuracy of the double-layer long-term and short-term memory model is 95.8%, which is significantly better than the former's 86.3%, showing better recognition performance. In summary, modeling and identifying the behavior of artificial intelligence animated characters can make the characters in virtual broadcasting more intelligent, natural, and realistic, thereby improving the viewing experience of virtual broadcasting, which has important practical value and research significance. This has significant practical and research value, providing insightful references for related fields.**

Keywords*—Virtual broadcasting; animated characters; behavioral modeling; action recognition; behavior tree; long and short-term memory

I. INTRODUCTION

Virtual broadcasting technology is a cutting-edge interdisciplinary field in the fields of computer graphics and artificial intelligence, playing an important role in media such as movies, television programs, and games [1]. Through the integration of virtual broadcasting technology and artificial intelligence (AI) animated characters, viewers can experience a fully immersive audio-visual encounter [2]. However, there are still challenges to achieving more natural, intelligent, and diverse behavioral representations of animated characters [3]. Traditional virtual animation character behavior modeling relies on rule design and manual programming, which limits the ability to flexibly respond to complex environments. Action recognition methods are often limited to specific action categories and are difficult in scenarios with high real-time requirements. Therefore, this study proposes a behavior tree based animation character behavior modeling and a Long Short Term Memory (LSTM) action recognition method that combines human geometric features. This study aims to improve the behavior and recognition abilities of AI-animated characters in virtual broadcasting, allowing them to display more natural and intelligent actions in intricate scenarios and interact with users in real-time.

Section I of the study introduces relevant technologies and methods, including existing research results on behavior modeling and action recognition, as well as research on behavior trees and LSTM. Section II provides a detailed explanation of related works. Section III is the establishment of an AI animation character behavior model, experiment and evaluation of animation character behavior model and action recognition is detailed in Section IV. Section V and Section VI delves into comparison and conclusion respectively.

II. RELATED WORK

Behavioral modeling (BM) refers to the process of establishing and describing the behavior of individuals or systems, which has numerous applications in various fields, including AI, computer graphics, virtual reality, game development, robotics technology, etc. As the boost of computer technology and AI, significant progress and development have been made in the field of behavior modeling. Colledanchise et al. presented a method that combines automatic planners and machine learning to automatically generate a behavior tree strategy for the application requirements of industrial robots in unpredictable environments. This approach offers a practical solution for enabling robots to operate autonomously in complex environments and has significant theoretical and practical implications [4]. Kumar proposed a deep learning (DL) classifier method on the ground of children's emotions to predict children's behavior, addressing the problems in establishing a behavioral model for predicting children's current emotional activities. This provides theoretical guidance for predicting children's emotional behavior [5].

Action recognition refers to the recognition and classification of human or objects actions through technologies such as computer vision and machine learning. At present, action recognition technology has made significant progress and has been widely applied in multiple fields. Song et al. addressed the issues of complex and overly parameterized state-of-the-art models, as well as inefficient training and inference, by embedding separable convolutional layers into early fusion multi input branch networks to construct efficient graph convolutional network baselines. It is used for skeleton action recognition, effectively improving accuracy, reducing model parameters and training costs [6]. Gharaee introduced a novel approach to recognizing human actions by employing a self-organizing mapping system. The objective was to address the problem of identifying the start and end times of online unsegmented actions for automated recognition purposes. This method can effectively extract and cluster features of action sequences, thereby achieving accurate recognition and segmentation of actions [7].

Behavior tree is a graphical tool used to model complex behaviors, commonly used in game development and virtual character control. LSTM is a special type of recurrent neural network designed to process sequential data, such as time series or text. Junaidi A et al. presented the utilization of behavior tree algorithm to effectively manage and control the diverse behaviors of NPCs for the type of side scrolling game, to enhance the fun and realism of the game. This accomplishment led to a more precise management and control of behavior, resulting in an enhanced player experience and game quality [8]. Shen et al. proposed a framework that combines bidirectional short-term memory networks and data sorting for real-time prediction of the diameter of shotcrete columns in soft soil, providing more accurate and reliable predictions for shotcrete processing in soft soil, and further improving the design of shotcrete columns [9]. Priyadarshini et al. proposed a combined model of short-term and short-term memory, CNN, and grid search to address the increase in user generated content on the Internet and the challenge of understanding emotions and emotions involved. This model helps to better understand user attitudes, viewpoints, and emotions, providing important basis for decision-making and analysis in various application scenarios [10].

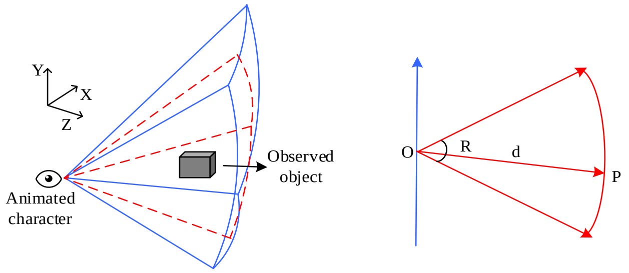
In summary, scholars have conducted in-depth research on behavior modeling and action recognition. However, in many research results, the traditional virtual animation character model is not flexible and lacks real-time performance in action recognition, which needs further research. Therefore, this study proposes an animation character behavior modeling on the ground of behavior trees and an LSTM action recognition method combining human geometric features. This provides an innovative way to improve the behavior performance and action recognition ability of AI-animated characters in virtual broadcasting.

III. Establishment of AI Animation Character Behavior Model and Action Recognition Method

As an execution method for planning models, behavior trees are widely used to construct complex AI character behaviors, enabling virtual actors to exhibit more natural, intelligent, and diverse behavior. The LSTM action recognition method involves combining geometric features of the human body with the LSTM neural network in deep learning to achieve precise recognition of actions.

A. Establishment of AI Animation Character Behavior Model Based on Behavior Tree

AI animated characters require complex behavior and diverse decision-making in virtual broadcasting. Traditional modeling techniques, such as finite state machines, have limitations in representation [11]. To overcome its shortcomings, this study adopts behavior tree modeling. The behavior tree has a parallel mechanism that supports hierarchical structure and node combination, making the behavior logic easier to understand. Its modular design reduces post maintenance costs, dynamically adjusts role behavior, and improves flexibility and scalability [12]. Visual perception possesses an essential influence on modeling the behavior of animated characters, simulating the visual cone of the real human eye, and setting appropriate field of view angles and ranges. The human eye's maximum visual range is 60^{o}, with a comfort range limited to 30^{o}. Only objects located within the visual cone can be perceived [13]. The human eye is more sensitive to dynamic objects, which allows for easier detection of their movements and changes. The visual perception model of animated characters is shown in Fig. 1.



(a)Visual stereogram

(b)Plane decomposition graph

Fig. 1. Visual perception model of animated characters.

In Fig. 1, assuming that a point on the observed object is P. O represents the eye of the animated character, which is the starting point of the line of sight. R represents the angle of view. D represents the field of view distance. If the distance between OPs is less than d and the angle between OP and the Z-axis is less than half of the field of view angle R, it is assumed that the observed object is within the field of view of the animated character. In a virtual environment, animated characters need to further determine whether there is an occlusion relationship. Point occlusion method is used for line of sight detection in this study. Assuming that the world coordinate of point O is $\left(x_0, y_0, z_0\right)$ and point P is $\left(x_p, y_p, z_p\right)$, OP is represented as shown in Eq. (1) [14].

$$\begin{cases} x = x_0 + m\left(x_p - x_0\right) \\ y = y_0 + m\left(y_p - y_0\right), 0 \le m \le 1 \\ z = z_0 + m\left(z_p - z_0\right) \end{cases} \tag{1}$$

In Eq. (1), m is to be solved. Assuming that there is a rectangular object between O and P as an obstruction, it projects it onto the XY plane to obtain a rectangular projection. It sets the coordinates of the four vertices as $\left(x_i, y_i, z_i\right), i = \{1,2,3,4\}$, and the equations for each edge are shown in Eq. (2).

$$\frac{x - x_i}{x_{i+1} + x_i} = \frac{y - y_i}{y_i + y_i}, i = \{1,2,3,4\} \tag{2}$$

It brings Eq. (2) into Eq. (1) and solve for the value of m. If m has no solution, then OP has no intersection with the projection rectangle, indicating that the observed object is not obstructed. If m has a solution, then the obtained value is taken into Eq. (1) to calculate the value of z. If the value of z is greater than the Z coordinate value corresponding to the highest point of the obstruction, it indicates that the observed object is not obstructed. Otherwise, it is obstructed. In addition to the visual model, an auditory model should also be established. The auditory perception model of the animated character is shown in Fig. 2.

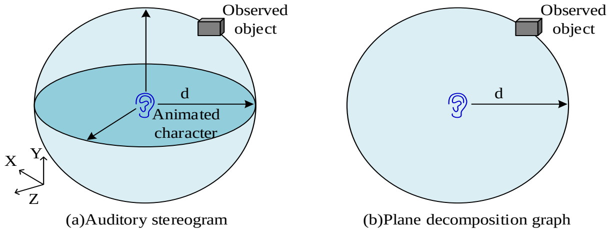
In Fig. 2, in a virtual environment, to simulate real auditory perception, the auditory range of the animated character is limited by a spherical area. Only sound located within the spherical area can be heard by the animated character [15]. It sets the auditory threshold V_0, assuming that the sound intensity is V and the distance between the animated character and the object is d. Only in the case of $V / d > V_0$ an animated character hear sound and make decisions about it. On the contrary, it exceeds the auditory perception range of the animated character, and the animated character will not be able to perceive the sound. In virtual broadcasting, AI-animated characters are modeled through perception to construct their next behavior, which corresponds to the child nodes of the behavior tree, namely the animated character behavior nodes. To achieve complete action recognition and control, this study combines the perception model and behavior model into a whole, forming an animation character control behavior tree, as shown in Fig. 3.

Fig. 3 shows that in virtual broadcasting, the behavior tree of animated characters is composed of a parent node as the root node, connecting the selection node, the order node of different branches, and the standby behavior node. The standby behavior node maintains the initial behavior of the animated character. The decoration node lies beneath the order node and is accountable for continuous visual and auditory perception. Once the environmental information within the perception range of the animated character changes, the sequential logic nodes and decoration nodes come into play, interrupt standby behavior, and execute the corresponding behavior under the sequential nodes. Through this structure, animated characters make intelligent behavioral decisions on the ground of environmental changes, resulting in more realistic performance in virtual broadcasting.

B. LSTM Action Recognition Method Combining Human Geometric Features

AI-animated characters in virtual broadcasting require action recognition ability to make corresponding action responses on the ground of user input or environmental changes. This study adopts the LSTM action recognition method that combines human geometric features [16]. AI characters obtain user action data through sensors, such as geometric features such as posture and joint position, as input. After processing by the LSTM neural network, they learn the temporal evolution patterns of action sequences and extract relevant features. Three different geometric structures of human bone point features, including 3D position difference, 3D angle difference, and bone vector angle feature, are proposed and normalized, and fused as input for the action recognition network model. The framework of the LSTM action recognition method combining human geometric features is shown in Fig. 4 [17].

Fig. 4 shows that the LSTM action recognition method is suitable for processing custom datasets containing sequences of human bone data points, and can effectively process data with temporal features. The method first extracts geometric features and normalizes human bone data points to ensure consistent data scales [18]. Then, the fused features are input into the LSTM network model to capture the dependencies and temporal correlations of action sequences. Next, the custom dataset trains the LSTM network to optimize model parameters and accurately recognize various human movements [19]. Finally, the probability distribution output by the LSTM network is converted into action classification results through the Softmax function, achieving accurate recognition and classification of human actions.



(a)Auditory stereogram (b)Plane decomposition graph

Fig. 2. Auditory perception model of animated characters.

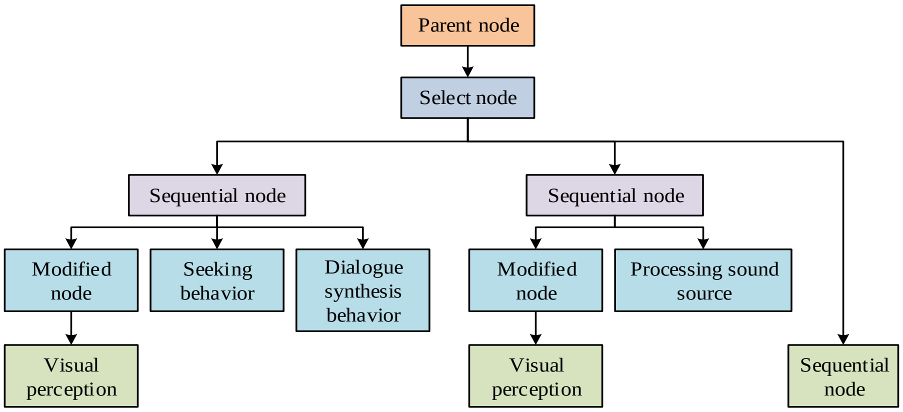


Fig. 3. The animation character controls the behavior tree.

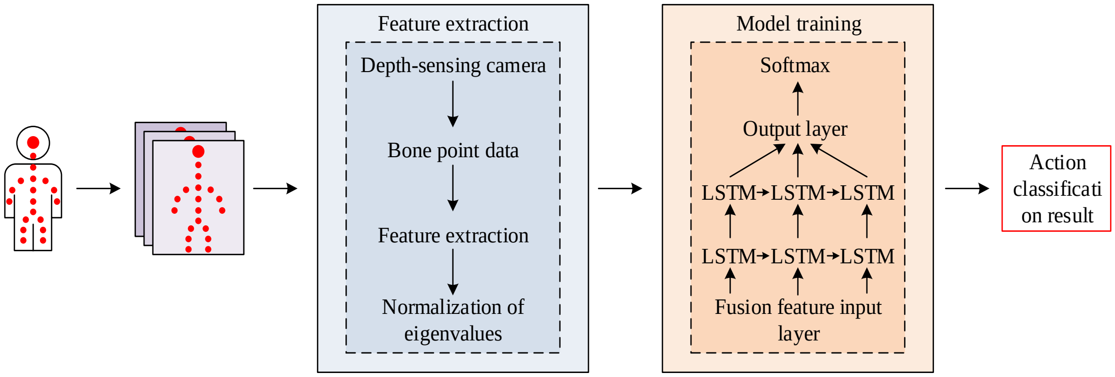


Fig. 4. LSTM action recognition method framework combining human body geometric features.

Recurrent Neural Network (RNN) is commonly used to process serialized data, which exhibit temporal properties and cyclic transitivity. Recurrent neural networks are widely used to handle serialization problems because they can have a "memory" function that links the information of the entire network together. The expression of output layer (OLA) $h^{(t)}$ in the two-layer RNN structure is shown in Eq. (3).

$$h^{(t)} = f\left(h^{(t-1)}, x^{t}\right) \tag{3}$$

In Eq. (3), x^t is the input of the current time (CTI) model. f represents the model function, also known as the activation function. The RNN sequence length is independent of f and is suitable for handling long sequence problems in different time dimensions. During network transmission, the output h_t of RNN nodes at each time point is shown in Eq. (4).

$$h_t = \varphi\left(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b\right) \tag{4}$$

In Eq. (4) φ represents the activation function. W_{xh} and W_{hh} represent the weight matrix (WMA) in the input and output of the network nodes at the CTI, and the WMA of the hidden layer meanwhile. b serves as the bias parameter. RNN transfers the output at the current moment (CMO) to the OLA and the hidden transmission at the next moment. Recursive transmission has defects, and the network gradient gradually decreases with the increase of sequence length, resulting in slower iterative updates and the issue of gradient disappearance or explosion. LSTM solves the above problems and improves the efficiency and stability of the network by adding “memory” units to the hidden layer and selectively “forgetting” long-term sequences. RNN is a standard recursive neural network, while LSTM has added three gate level control units that control the flow of information, including input gate (IGA) i, forgetting gate (FGA) f, and output gate (OGA) o, and has built-in hidden layer memory cell C_t [20]. The FGA f is utilized for controlling the degree to which information from the previous moment is retained in the current memory cell. The formula for calculating FGA f at time t is shown in Eq. (5).

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{5}$$

In Eq. (5), W_f is the WMA of the FGA. h_{t-1} is the hidden state of the previous moment. x_t is the input at the CTI. b_f is the bias parameter of the FGA. σ serves as an S-type activation function, used to map the calculation results to the range [0,1], representing the output value of the FGA. The calculation formula for IGA i is shown in Eq. (6).

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{6}$$

In Eq. (6), W_i is the WMA of the IGA. b_i is the bias parameter of the IGA. The IGA controls the degree to which the input information at the CMO updates the memory cells. The calculation formula for OGA o is shown in Eq. (7).

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{7}$$

In Eq. (7), W_o is the WMA of the OGA. b_o is the bias parameter of the OGA. The OGA controls the degree to which information in memory cells is output at the CMO. The calculation of candidate memory cell $C_t^{'}$ is shown in Eq. (8).

$$C_t^{'} = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{8}$$

In Eq. (8), W_c is the WMA of the input and the previous hidden state. b_c is the bias parameter for candidate cells. $\tanh$ is a hyperbolic function, and the candidate memory cell is a temporary memory cell calculated at each time step to store new input information. The calculation of updated memory cells is shown in Eq. (9).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t^{'} \tag{9}$$

In Eq. (9), C_{t-1} serves as the memory cell of the previous moment. Updating memory cells is achieved through the weighted sum of FGAs, IGAs, and candidate memory cells. The output of memory cells is shown in Eq. (10).

$$h_t = o_t \cdot \tanh\left(C_t\right) \tag{10}$$

In Eq. (10), h_t represents the hidden state of LSTM at the

CTI t, which is also the output of memory cells. To better capture temporal features and improve recognition accuracy, a two-layer LSTM network model is introduced to increase network depth [21]. Fig. 5 showcases the relevant structure.

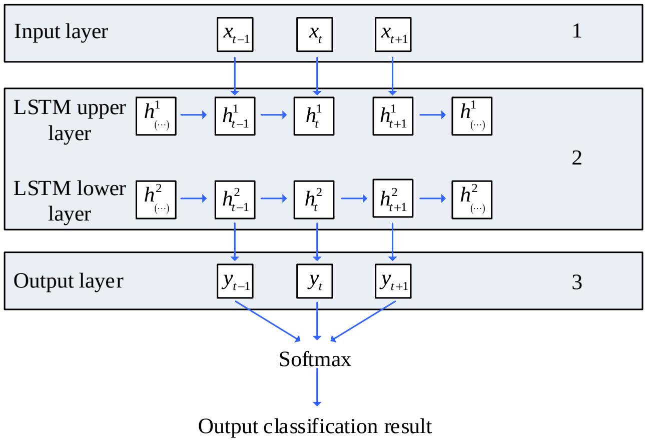


Fig. 5. Two-layer LSTM network model.

Fig. 5 shows that in a two-layer LSTM network model, two LSTM models are set at the same level, with the upper layer (ULA) being the input layer (ILA), and the ULA LSTM model is as the input sequence. The lower layer is the OLA, and the lower layer LSTM model is used as the output model. The forward formula of the double-layer (DLA) LSTM network model is shown in Eq. (11).

$$\begin{cases} i_{l,t} = \delta\left(W_{xi}x_t + U_{hli}h_{l,t-1} + U_{h(l-1)i}h_{l-1,t} + V_{ci}c_{l,t-i} + b_i\right) \\ f_{l,t} = \delta\left(W_{xf}x_t + U_{hlf}h_{l,t-1} + U_{h(l-1)f}h_{l-1,t} + V_{cf}c_{l,t-i} + b_j\right) \\ o_{l,t} = \delta\left(W_{xo}x_t + U_{hlo}h_{l,t-1} + U_{h(l-1)o}h_{l-1,t} + V_{co}c_{l,t-i} + b_o\right) \\ C_{l,t} = f_t \cdot C_{t-1} + i_{l,t} \cdot \tanh\left(W_{xc}x_t + U_{ltc}h_{l,t-1} + U_{h(l-1)c}h_{l-1,t} + b_c\right) \\ h_{l,t} = o_{l,t} \cdot \tanh\left(C_{l,t}\right) \end{cases} \tag{11}$$

In Eq. (11), $h_{l,t-1}$ and $h_{l-1,t}$ respectively represent the hidden states of the ULA LSTM model and the lower layer LSTM model. x_t is the ILA. U_{hl} and $U_{h(l-1)}$ represent the weight matrices of the ULA LSTM model and the lower layer LSTM model, respectively. b represents the offset parameter. All variables are classified as IGA i, FGA f, or OGA o on the ground of i、f、o in the table below. At moment t, the output y_t of the DLA model is shown in Eq. (12).

$$y_t = U_{hlt}h_{l,t} + b_y t \tag{12}$$

In Eq. (12), U_{hlt} represents the WMA. $h_{l,t}$ serves as the hidden state of the lower layer LSTM at time t. b_y represents the offset parameter. The Softmax function is showcased in Eq. (13).

$$\xi_i = \frac{e^{Z^i}}{\sum_{J=1}^{n} e^{Z^j}} \tag{13}$$

In Eq. (13), Z^i represents the i-th action sequence. J serves as the quantity of the action type. n serves as a total of n actions identified. When updating network parameters, the study adopts a cross entropy loss function, as shown in Eq. (14).

$$E\left(y_t, y_t^{'}\right) = -y_t \log y_t^{'} = -\sum_{t=1}^{n} y_t \log y_t^{'} \tag{14}$$

In Eq. (14), y_t represents the label of the actual action. $y_t^{'}$ represents the label that identifies the action. The training process iteratively updates the network, leading to a gradual decrease in the loss function's value. After each update, the gradient values overlay, as demonstrated in Eq. (15).

$$\frac{\partial E}{\partial w} = \sum_{t=1}^{n} \frac{\partial E}{\partial w} \tag{15}$$

In Eq. (15), E represents a loss function. w represents the weight parameter in the network. Through continuous iterative updates and gradient stacking, the DLA LSTM model can gradually optimize during training and adapt to features of the data to improve recognition of temporal features with increased accuracy.

IV. Experimental Analysis of AI Animation Character Behavior Modeling and Action Recognition

To explore the effectiveness and superiority of the behavior tree based animation character behavior modeling and the LSTM action recognition method combined with human geometric features; the study started with simulation experiments and tested the collision avoidance and sound source capture indicators of the behavior modeling. Meanwhile, it conducted testing experiments on the LSTM action recognition method and compared it with the RNN model.

A. *Behavior Modeling Experiment Simulation Based on Behavior Tree*

The objective of the experiment is to test the effectiveness of animation character behavior modeling on the ground of behavior trees in handling obstacles to prevent collisions and capture sound sources. Specifically, this study aims to evaluate the effectiveness of the system in avoiding collision objects and ensuring the safe movement of the modeled object in the environment. It also aims to determine whether the testing system can accurately detect and locate sound sources, thus simulating the modeled object's perception and positioning ability towards sound. The relevant situation of development environment is shown in Table I.

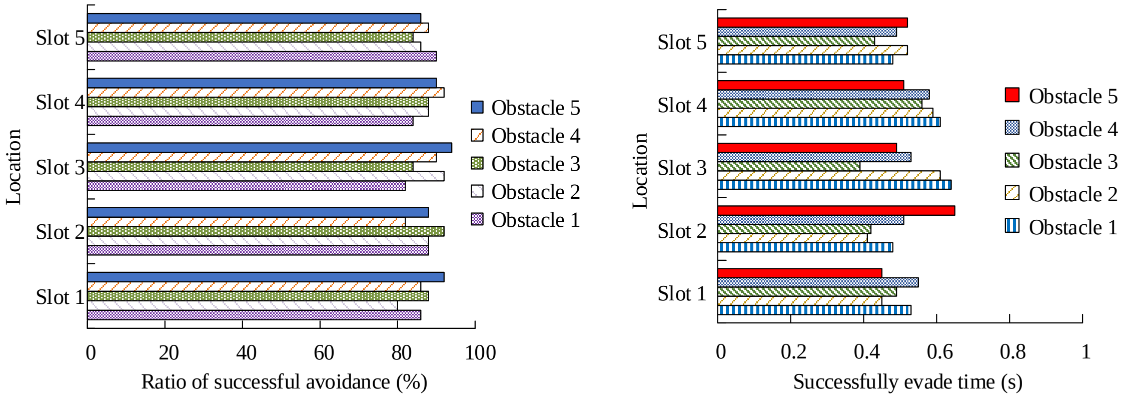
Table I shows that the experimental computer is configured with an AMD Ryzen 5 3600X 6-Core processor, 32GB DDR5 6000MHz memory, and is equipped with an NVIDIA GeForce RTX 3070 high-performance graphics card. This configuration meets the development requirements of the experiment and enables real-time behavior modeling and simulation in complex scenarios. Unity 3D is utilized as the game engine, supporting the development of 2D and 3D games, virtual and augmented reality applications, simulators, and animations. BM adopts the free software DAZ Studio, which provides rich functionality for creating high-quality digital characters, scenes, and animations. To achieve complex AI behavior, it uses the professional behavior tree plugin Behavior Designer. Its evaluation system performance uses collision avoidance and sound source capture metrics. The collision avoidance indicator evaluates the system's contact with collision objects in different scenarios, calculates the successful avoidance ratio and average avoidance time. The experimental results are shown in Fig. 6.

Fig. 6 shows that the successful avoidance rate is above 80% when encountering different obstacles. Among them, when facing obstacle 5, the avoidance rate at position 3 reaches 96%, which means that in most cases, the system can effectively perceive the obstacle and make timely avoidance decisions, thereby successfully avoiding collisions. The reaction time for taking avoidance actions when facing obstacles is within 0.41s-0.65s, indicating that the system has relatively fast response ability in collision avoidance. The system rapidly detects the presence of obstacles and makes appropriate decisions to evade them. The sound source capture indicator uses a virtual sound source to simulate the sound in the environment, and records the system's perception and positioning of the sound source position. It evaluates the sound source capture accuracy of the system by comparing the error between the predicted position and the actual position. The experimental outcomes are showcased in Table II.

Table II shows that the percentage of capture error for different sound sources ranges from 1.40% to 1.88%, with an average error of approximately 1.56% for sound source capture. For the given experimental data, it was observed that the percentage of error between the actual position and the predicted position is not fixed under different obstacles, but fluctuates slightly. As the distance increases, the error also increases. Overall, the predicted results (PRE) captured by the sound source are all near the actual sound source location, with small errors and within an acceptable range. This suggests that the system excels at capturing sound sources with high levels of precision and accuracy.

TABLE I. EXPERIMENTAL DEVELOPMENT ENVIRONMENT

Hardware development platform	/	/
1	Computer processor	AMD Ryzen 5 3600X 6-Core Processor 3.80 Ghz
2	Computer memory	32GB DDR5 6000MHz
3	Computer graphics card	NVIDIA GeForce RTX 3070
Software development platform	/	/
1	Game engine	Unity 3D
2	Modeling software	DAZ Studio
3	Action number plug-in	Behavior Designer



(a) The result of the experiment of the ratio of avoidance times

(b) The result of avoidance time experiment

Fig. 6. Proportion of successful avoidance and average avoidance time of the system.

TABLE II. TEST RESULTS OF SOUND SOURCE CAPTURE EXPERIMENT

Sound source	Actual location of the sound source (m)	Sound source predicted bearing (m)	Error (%)
1	5.62	5.71	1.60
2	8.49	8.33	1.88
3	2.61	2.62	1.14
4	7.36	7.49	1.77
5	4.28	4.34	1.40

B. Experimental Analysis of LSTM Action Recognition Method

The experiment used the UCF101 dataset, created by a team of researchers at the University of Central Florida, and a custom dataset, which included 101 different categories of movements and more than 13,000 video clips covering movements of a variety of daily life and sports. The latter was gathered utilizing a half-body camera and comprised of 12 crucial bone points and seven fundamental movements from a total of 10 participants. The cross-validation numbers are 1-10, the training set contains 634 action sequences, and the test set contains 618 action sequences. The collected 3D position difference feature is an 11-dimensional vector, the 3D Angle difference feature is a 12-dimensional one, and the bone vector Angle feature is an 8-dimensional one. To reduce the risk of overfitting, 0.1 random inactivation is added to the hidden unit of the network. The study first conducted fusion comparative experiments on three different features: 3D position difference feature, 3D angle difference feature, and bone vector angle feature. It records three different features: 3D position difference feature, 3D angle difference feature, and bone vector angle feature, which are A, B, and C. The fusion feature of A and B is D, and the fusion feature of the three features is E. The recognition rate results obtained from the experiments are shown in Table III.

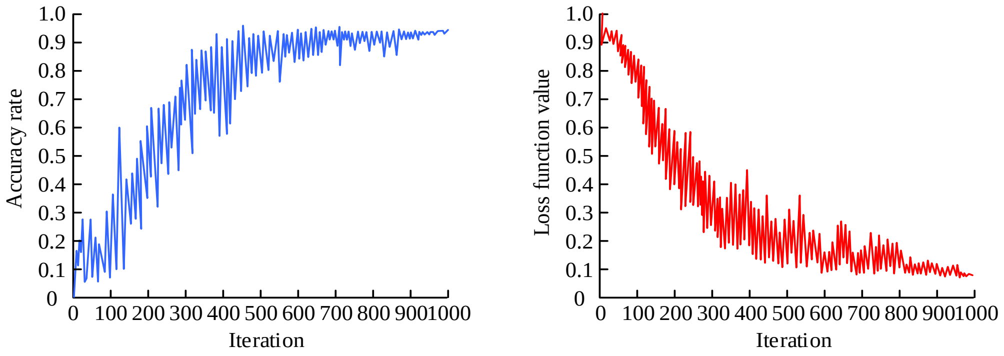
TABLE III. MODEL RECOGNITION RATE WITH FEATURE FUSION

Input feature vector	Maximum recognition rate (%)	Minimum recognition rate (%)	Average recognition rate (%)
A	78.12	72.48	75.23
B	72.54	51.89	65.65
C	86.39	67.73	76.45
D	89.45	76.28	84.32
E	98.23	92.46	95.03

Table III shows that the recognition rate for A is fairly consistent, ranging from a high of 78.12% to a low of 72.48%. The difference between the highest and lowest recognition rates of B is significant, making it the most unstable of the five features. Meanwhile, the average recognition rate is only 65.65%, which is also the lowest. The performance of C and D is average, but D, which combines the two features, has a marked enhancement relative to C. For E, the fusion of three features resulted in a maximum recognition rate of 98.23%, a minimum recognition rate of 92.46%, and an average recognition rate of 95.03%, indicating a significant improvement. Therefore, E was utilized as the input for the double-layer LSTM network model to obtain the accuracy change curve and loss function value (LFV) change curve, as shown in Fig. 7.

In Fig. 7, the accuracy curve shows that the accuracy gradually converges to 1 with an increase in iterations, suggesting perfect classification performance for the training data. The LFV in the LFV change curve gradually converges to 0 with the quantity of iterations grows, indicating that the error between the PRE of the model and the actual labels is effectively reduced during the training. This research uses IDLE for stationary and WALK for walking. RUN stands for running. LHU means raising the left hand. RHU stands for raising the right hand. WLH means waving to the left. WRH means waving to the right. The study evaluated and identified seven actions and obtained their accuracy, recall, and F1 score (F1 Score) results, as shown in Fig. 8.

Fig. 8 shows that the accuracy of LHU, RHU, WLH, and WRH actions is 100%, indicating that the model has no errors in the recognition and prediction of these four actions. The recall rate of action type RHU is also 100%, indicating that the model has excellent capture ability for this action and has not missed any real positive examples. The F1 Score of most action types is above 0.9, indicating that the model performs well on multiple indicators and has good classification performance. The recognition rate of action types WALK and RUN is relatively low because they belong to repetitive dynamic feature sequences. Furthermore, during the capture process, only the upper body mode was employed, thus neglecting bone points such as the knees and ankles, which offer improved distinguishing features between these movements. For demonstrating the superiority of the DLA LSTM model, comparative experiments were conducted with the RNN model. The results are shown in Fig. 9.



(a) The relationship between accuracy and the number of iterations

(b) The relationship between Loss function value and the number of iterations

Fig. 7. Accuracy curve and LFV curve of two-layer LSTM network model.

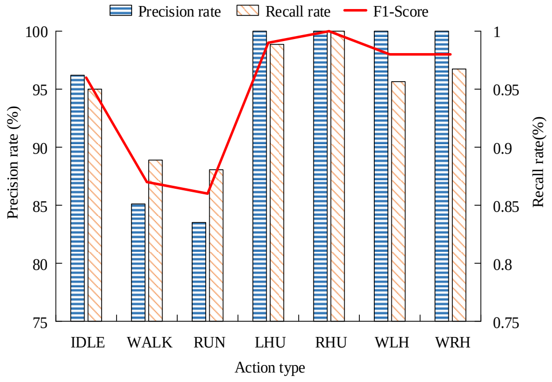
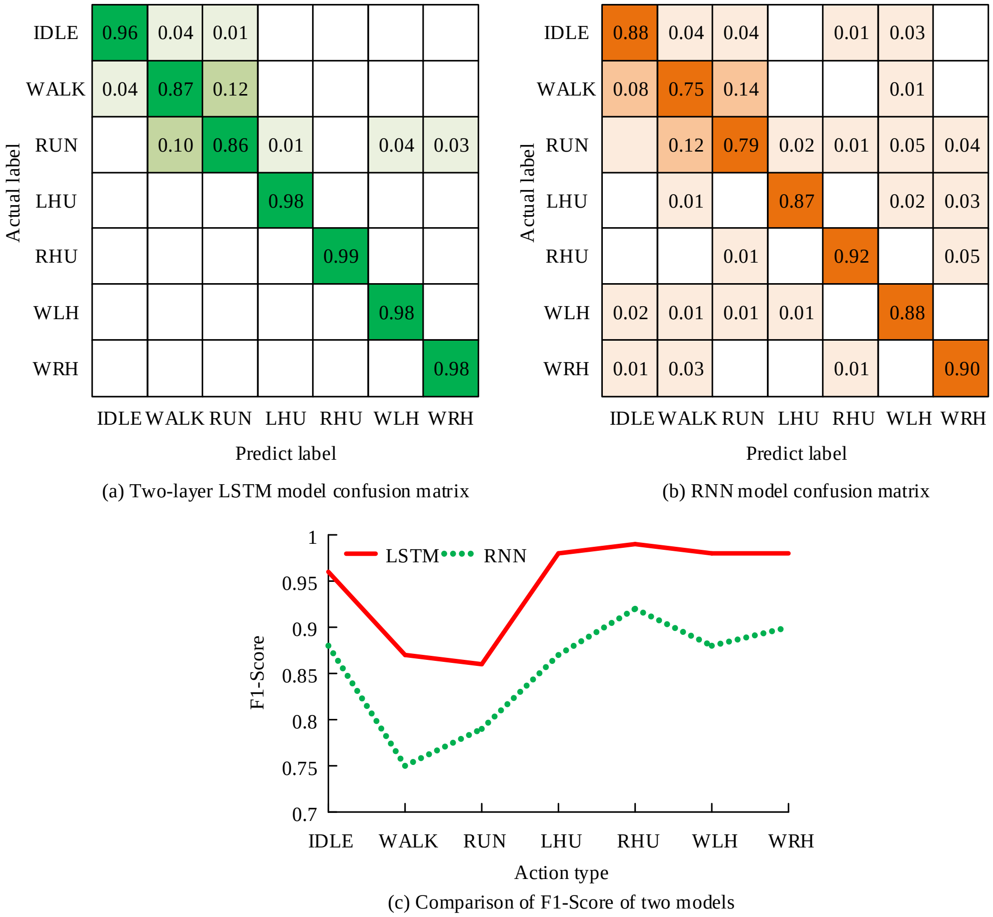


Fig. 8. Model recognition result.



	IDLE	WALK	RUN	LHU	RHU	WLH	WRH
IDLE	0.96	0.04	0.01				
WALK	0.04	0.87	0.12				
RUN		0.10	0.86	0.01		0.04	0.03
LHU				0.98			
RHU					0.99		
WLH						0.98	
WRH							0.98

(a) Two-layer LSTM model confusion matrix

	IDLE	WALK	RUN	LHU	RHU	WLH	WRH
IDLE	0.88	0.04	0.04		0.01	0.03	
WALK	0.08	0.75	0.14			0.01	
RUN		0.12	0.79	0.02	0.01	0.05	0.04
LHU		0.01		0.87		0.02	0.03
RHU			0.01		0.92		0.05
WLH	0.02	0.01	0.01	0.01		0.88	
WRH	0.01	0.03			0.01		0.90

(b) RNN model confusion matrix

(c) Comparison of F1-Score of two models

Fig. 9. Comparison of experimental results between two-layer LSTM model and RNN model.

Fig. 9 shows that the RNN model lacks a “rgetting unit” resulting in a significant increase in misclassification of actions during the recognition process compared to the DLA LSTM model. The diagonal elements of the confusion matrix represent the recall rate of the current model, which is the probability of accurately predicting actions. When comparing the F1 Score values of the DLA LSTM model and the RNN model, it is evident that the LSTM model outperforms the RNN model in all seven classification actions. After calculation, the accuracy of the DLA LSTM model is 95.8%, and the accuracy of the RNN model is 86.3%, indicating that the DLA LSTM model has better recognition performance.

V. Comparison

The conclusions of study [22] were compared with the conclusions of the research, in which study [22] mentioned the application of transformer-based deep neural networks to human action recognition, and the overall success rate reached 94.96% for six kinds of actions. The accuracy rate of the two-layer LSTM model proposed in this research was 95.8%, surpassing that of the aforementioned [22] method, thus demonstrating the superiority of this study.

VI. CONCLUSION

In response to the limitations in traditional virtual animation character behavior modeling and action recognition methods, this study proposes a behavior tree based animation character behavior modeling and an LSTM action recognition method combining human geometric features, and verifies the performance and effectiveness of both methods. The research results indicate that when the behavior modeling model faces different obstacles, the proportion of successful avoidance is over 80%, and the avoidance response time is within 0.41s-0.65 seconds, indicating that the system can effectively perceive obstacles and make avoidance decisions in a timely manner. The accuracy and LFVs of the LSTM action recognition method gradually converge to 1 and 0 as the number of iterations increases. This indicates the model's ability to perform perfect classification on the training data and effectively reduce the error between the PRE and real labels during the training process. Meanwhile, the LSTM action recognition method achieves 100% accuracy in identifying seven types of actions, including LHU, RHU, WLH, and WRH. The RHU achieved a recall rate of 100%, and most action types received an F1 Score higher than 0.9. In the comparative experiment between the DLA LSTM model and the RNN model, the accuracy of the DLA LSTM model was 95.8%, and the accuracy of the RNN model was 86.3%, demonstrating that the DLA LSTM model has better recognition performance. Overall, the behavior modeling and action recognition methods studied and designed have high effectiveness and certain advantages, providing an effective solution for AI animation character behavior modeling and action recognition in virtual broadcasting. However, this study used upper body mode for behavior capture, ignoring bone points such as knees and ankles that are more likely to distinguish between the “walking” and “running” categories, and further discussion is needed.

REFERENCES

[1] Jones D, Lotz N, Holden G. A longitudinal study of virtual design studio (VDS) use in STEM distance design education. International Journal of Technology and Design Education, 2021, 31(4): 839-865.

[2] Kaul V, Enslin S, Gross S A. History of artificial intelligence in medicine. Gastrointestinal endoscopy, 2020, 92(4): 807-812.

[3] Dvorožňák M, Sýkora D, Curtis C, et al. Monster mash: a single-view approach to casual 3D modeling and animation. ACM Transactions on Graphics (ToG), 2020, 39(6): 1-12.

[4] Colledanchise M, Natale L. On the implementation of behavior trees in robotics. IEEE Robotics and Automation Letters, 2021, 6(3): 5929-5936.

[5] Kumar D T S. Construction of hybrid deep learning model for predicting children behavior based on their emotional reaction. Journal of Information Technology and Digital World, 2021, 3(1): 29-43.

[6] Song Y F, Zhang Z, Shan C, Wang L. Constructing stronger and faster baselines for skeleton-based action recognition. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(2): 1474-1488.

[7] Gharaee Z. Online recognition of unsegmented actions with hierarchical SOM architecture. Cognitive Processing, 2021, 22(1): 77-91.

[8] Junaidi A, Yunus A, Wiguna A S. Implementasi Behavior Tree Pada Perilaku Npc Di Game Sidescroller. Kurawal-Jurnal Teknologi, Informasi dan Industri, 2021, 4(2): 92-103.

[9] Shen S L, Atangana Njock P G, Zhou A, Lyu H M. Dynamic prediction of jet grouted column diameter in soft soil using Bi-LSTM deep learning. Acta Geotechnica, 2021, 16(1): 303-315.

[10] Priyadarshini I, Cotton C. A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis. The Journal of Supercomputing, 2021, 77(12): 13911-13932.

[11] Fang Y, Luo B, Zhao T, He D, Jiang B, Liu Q. ST-SIGMA:Spatio-temporal semantics and interaction graph aggregation for multi-agent perception and trajectory forecasting. CAAI Transactions on Intelligence Technology, 2022, 7(4):744-757.

[12] Chen-Kraus C, Raharinoro N A, Lawler R R. Terrestrial Tree Hugging in a Primarily Arboreal Lemur (Propithecus verreauxi): a Cool Way to Deal with Heat? International Journal of Primatology, 2023, 44(1): 178-191.

[13] Wu C, Cha J, Sulek J, Zhou T, Sundaram C P, Wachs J, Yu D. Eye-tracking metrics predict perceived workload in robotic surgical skills training. Human factors, 2020, 62(8): 1365-1386.

[14] Navaneethan S, Sreedhar P S S, Padmakala S, Senthilkumar C. The Human Eye Pupil Detection System Using BAT Optimized Deep Learning Architecture. Comput. Syst. Sci. Eng., 2023, 46(1): 125-135.

[15] Denham S L, Winkler I. Predictive coding in auditory perception: challenges and unresolved questions. European Journal of Neuroscience, 2020, 51(5): 1151-1160.

[16] Yan X, Weihan W, Chang M. Research on financial assets transaction prediction model based on LSTM neural network. Neural Computing and Applications, 2021, 33(2): 257-270.

[17] Xu S, Rao H, Peng H, Jiang X, Guo Y, Hu X, Hu B. Attention-based multilevel co-occurrence graph convolutional LSTM for 3-D action recognition. IEEE Internet of Things Journal, 2020, 8(21): 15990-16001.

[18] Lei Y. Research on microvideo character perception and recognition based on target detection technology. Journal of Computational and Cognitive Engineering, 2022, 1(2): 83-87.

[19] Zheng H, Lin F, Feng X. A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(11): 6910-6920.

[20] Essien A, Giannetti C. A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. IEEE Transactions on Industrial Informatics, 2020, 16(9): 6069-6078.

[21] Singh K, Malhotra J. Two-layer LSTM network-based prediction of epileptic seizures using EEG spectral features. Complex & Intelligent Systems, 2022, 8(3): 2405-2418

[22] Hou Y, Wang L, Sun R, et al. Crack-across-pore enabled high-performance flexible pressure sensors for deep neural network enhanced sensing and human action recognition. ACS nano, 2022, 16(5): 8358-8369.