International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7807-7819

https://doi.org/10.70517/ijhsa463669

The Practice of Motion Capture-based Virtual Character Performance Generation Technology in Film and Television Animation

Na Zhao[1,*]
[1] College of Ministry of Sports, Xi'an Aeronautical University, Xi'an, Shaanxi, 710089, China
Corresponding authors: (e-mail: zhaona89722390@126.com).

Abstract Virtual character performance generation technology is widely used in film and television animation, especially the combination of motion capture and deep learning, which can effectively improve the naturalness and fluency of the performance. In this paper, a virtual character performance generation method is proposed, which adopts motion capture technology to obtain character movement data and combines deep reinforcement learning for training. The study introduces hierarchical policy learning based on the Actor-Critic framework and uses the PPO algorithm to optimize the motion control strategy. The experimental results show that the reward value of the virtual character in completing the one-legged squatting movement tends to stabilize after 6000 training rounds. In terms of muscle-driven control, the ablation experiments verified the importance of the degree of muscle activation in generating movement fluency and variety. In the one-legged squat maneuver, the maximum reward value was 32.08, while the maximum value after removing the muscle reward decreased to 16.39. Through the user research, the smoothness and naturalness of the virtual character's movements were highly evaluated, and the system's usability and visualization received positive feedback. The technique proposed in this paper has important application value in virtual character performance generation.

Index Terms virtual character, deep reinforcement learning, motion capture, reward function, muscle-driven, user research.

I. Introduction

With the development and innovation of science and technology, virtual reality film and television production has become a new trend in the film and television industry [1], [2]. In this trend, the virtual character performance generation technology based on motion capture plays a crucial role [3]. Motion capture-based virtual character performance generation technology is an important link in virtual character modeling, which can help designers accurately capture real-world movements and apply them to virtual character models, thus making virtual characters more realistic and natural when performing actions [4]-[6]. This technology is widely used in many fields such as film and television animation and virtual reality, and its appearance makes virtual characters more realistically mimic human movements, bringing a more immersive experience to the audience [7]-[9].

Generally speaking, the steps of motion capture virtual character performance generation technology in film and television animation usually include selecting motion capture equipment, setting up the capture scene, implementing motion capture and processing the capture data, so as to realize the application in film and television animation [10]-[13]. However, the use of virtual character performance generation techniques in virtual reality film and television production faces some challenges [14]. For example, the technical threshold for operating the equipment is high, requiring a specialized team to operate and process it. At the same time, the captured data need to be interpreted and processed accurately, otherwise it will lead to incoherent or unrealistic movements of virtual characters [15], [16]. In addition, the capture equipment itself has certain cost and technical limitations, and the producer needs to, weigh the pros and cons according to the budget and demand [17].

With the continuous development of virtual character performance generation technology, the film and animation industry has put forward higher requirements on the performance capability of virtual characters. Traditional virtual character generation methods usually rely on fixed action libraries or predefined control methods, which results in the virtual character's action performance often appearing to be single, lacking in personalization and fluency. Therefore, how to improve the performance of virtual characters in complex animation scenes by means of emerging technologies has become an important research topic.

In recent years, the application of deep learning, especially deep reinforcement learning (DRL), in virtual character generation has attracted widespread attention. Unlike traditional methods, deep reinforcement learning enables virtual characters to autonomously learn how to generate actions according to the environment and task requirements without relying on fixed action sequences. This approach not only enhances the adaptability and expressiveness of virtual characters, but also provides greater flexibility in character generation. In particular, the combination with muscle-driven control strategies ensures that the virtual character maintains the coordination and naturalness of its movements while executing complex actions.

In this context, this paper proposes a virtual character performance generation method based on deep reinforcement learning and muscle-driven control strategies. The action data of the character is acquired through motion capture technology and trained using deep reinforcement learning to optimize the action generation process of the virtual character. The research focuses on improving the smoothness and expressiveness of virtual character actions through the combination of deep reinforcement learning and muscle-driven control strategies. To this end, this paper introduces a hierarchical strategy learning framework, which is trained by the PPO algorithm and combined with muscle control strategies to optimize the precision and diversity of character movements.

II. Deep Reinforcement Learning-based Virtual Character Performance Generation Technology

II. A.Motion Capture for Virtual Character Performance

II. A. 1) Marker point labeling and processing

Combining the principle of ergonomics and the characteristics of virtual character performance in film and television animation, 65 marker points are selected to be pasted on the whole body of the virtual character. Viconshgun is used to solve the masked marker points, build the model, and capture the performance movements. Finally, the skeleton is solved and the output data is the complete human body marker points.

II. A. 2) Constructing avatar models

(1) Character role modeling

Based on the modern dance movement dataset, Maya animation production software is used to construct the virtual character model [18], [19]. Specifically using polygonal modeling technology to model the virtual character's head and body; then composed of a complete human body model; and finally adjusted and modified the body parts of the sewing and details, and ultimately realize the construction of a complete character model. On the basis of character modeling, the creation of clothing is carried out. First of all, we create clothing sample curves before and after the body of the character model; then we simulate the material and texture of the real fabric through the 2D texture method; finally, we adjust the editing UV and texture editor to apply the texture to the corresponding objects, thus realizing the creation of the character's clothing model.

(2) Neural Fusion Shape Technology

The framework of neural fusion shape technology mainly includes envelope deformation branch and residual deformation branch. By inputting the joint rotation and character model data, the bone structure can be obtained; then the envelope deformation branch is used to bind the skin weights for the bone construction; finally, it is fused with the hybrid shapes and hybrid coefficients in the residual deformation branch to obtain the fused character model.

a) Envelope deformation branch

This part focuses on learning the parameters of a specific skeleton level, predicting the skeleton, skin and weight bindings. A pose represented by local joint rotations is added in each iteration $R=\{R_i\}\ where\ R_i \in R^{3\times3}$, which guides the deformation of the input data and the predicted bindings and skinning. Therefore, a local mapping transformation $\{R_i, O_i\}$ needs to be performed first for each joint in the character model using the positive kinematic cumulative transformation; the global transformation for each vertex is computed:

$$T_{Rj} = \sum_i W_{ji} T_i \tag{1}$$

After the operation, the vertex-by-vertex mapping transformation $T_R = \{T_{R_i}\}$ can be applied to the input characters:

$$\bar{V}_{Rj} = T_R \Theta V \tag{2}$$

where T_R is the global mapping transform and Θ is the vertex-by-vertex operation.

b) Residual deformation branching

Let the given vertex position be V and the connectivity of each vertex be F, the output skin W is connected to the depth vertex V' along the channel $\left(\{V',W\} \in R^{V\times(K+J)}\right)$, and then use the edge feature representation and mesh convolution width to generate a set of N residual shapes $\{B_i\}_{i=1}^{N}, B_i \in R^{V\times 3}$. A small neural network of J MLP blocks is input simultaneously, and finally the pose-dependent coefficients $\{a_{ij}\}_{i=1}^{N}$ of each joint J are output and added to the residual shape of V:

$$\bar{V} = V + \sum_{j=1}^{J}\sum_{i=1}^{N} \alpha_{ij} M_j B_i \tag{3}$$

M_j is a binary mask that specifies the vertices associated with joint j.

Finally, the loss function L_v is utilized to find the difference between the task role and the corresponding true value. As a result, high-quality 3D character models in *.FBX format are automatically generated by the neural fusion shape technique of envelope deformation branches and residual deformation branches.

II. A. 3) Finger bone actuation

In this paper, a data fusion algorithm is used to complete the driving of finger movement in the character model.

(1) Read the data in the motion capture file.

(2) Read the information in the model file.

(3) Match the skeletal nodes of the two files, and according to the matching result, read the position of the corresponding nodes in the motion capture file, and make the nodes in the model display in the node position of the motion capture file. Finally, the nodes in the model are matched with the nodes in the motion capture file.

(4) Repeat the cycle of step (4) to completely fuse the task model to the captured action sequence, so as to realize the effect of motion capture data driving the movement of the character model.

II. B.Deep reinforcement learning

Deep Reinforcement Learning (DRL) is a combination of reinforcement learning and deep learning that is able to learn the mapping relationship between inputs and outputs and gradually converge the decisions given by the model to the optimum through continuous learning.

II. B. 1) Traditional reinforcement learning

In reinforcement learning Markov Decision Process (MDP) can be represented using the quaternion (S, A, P, R), where S is the state space, which is the set of all states i.e. $S = \{s_1, s_2, \cdots, s_n\}$; A is the action space, which is the set of all possible actions i.e. $A = \{a_1, a_2, \cdots, a_n\}$; P is the state transfer probability, which represents the probability of transferring to the next state after taking an action in the current state i.e., $P^{a}_{s_t \to s_{t+1}}$; R is the reward, which is a scalar function with the combination of the current action, the current state and the next state as $R = R(s_t, a_t, s_{t+1})$.

Set the current state of the environment to be s, define the cumulative reward obtained under the policy π as the state-value function $V^{\pi}(s)$, and the cumulative reward obtained after the execution of the action a using the policy π as the state-action-value function $Q^{\pi}(s,a)$. $V^{\pi}(s)$ and $Q^{\pi}(s,a)$ are defined as:

$$V^{\pi}(s) = \bar{a}_{\pi}\left|\sum_{t=0}^{+\infty} \gamma^t r_t \,|\, s_0 = s\right| \tag{4}$$

$$Q^{\pi}(s,a) = \bar{a}_{\pi}\left|\sum_{t=0}^{+\infty} \gamma^t r_t \,|\, s_0 = s, a_0 = a\right| \tag{5}$$

where γ - discount factor and $\gamma \in (0,1)$; r - reward obtained after the execution of the current action.

Under the Markov model, the future change of each state does not depend on any previous state, but only on its current state, so the above equation can be written in recursive form. I.e:

$$V^{\pi}(s) = \sum_{a\in A} \pi(s,a) \sum_{s_{t+1}\in S} P^{a}_{s\to s_{t+1}}\left(r^{a}_{s\to s_{t+1}} + \gamma V^{\pi}(s_{t+1})\right) \tag{6}$$

$$Q^{\pi}(s,a)=\sum_{s_{t+1}\in S}P^{a}_{s\to s_{t+1}}\left(r^{a}_{s\to s_{t+1}}+\gamma V^{\pi}(s_{t+1})\right) \tag{7}$$

where s_{t+1} - the next state given by the environment after the execution of action a.

It can be seen that the state-value function $V^{\pi}(s)$ and the state-action-value function $Q^{\pi}(s,a)$ are related as:

$$V^{\pi}(s)=\sum_{a\in A}\pi(s,a)Q^{\pi}(s,a) \tag{8}$$

Model-free reinforcement learning can be used to find the optimal solution using Monte Carlo methods, but Monte Carlo methods require a large number of average samples, and this leads to learning inefficiency, and when the number of samples is not up to the optimal solution using Monte Carlo methods is imprecise, which is one of its larger drawbacks.

II. B. 2) Value function based algorithm

The temporal difference algorithm in reinforcement learning solves the problem of imprecision in model-free reinforcement learning, $Q-$ learning algorithm is a typical temporal difference algorithm. The $Q-$ learning algorithm is based on the method of value iteration, and learns in constant interaction with the environment, through which the decision rule is constantly optimized to maximize the cumulative value of the reward.

The $Q-$ learning algorithm takes as its core idea the learning of a state-action value function $Q(s,a)$, which represents the value that can be obtained by choosing an action a in the current state s. The $Q-$ learning algorithm uses the Bellman equation to update the Q value. To wit:

$$Q(s,a)=Q(s,a)+\alpha\left(r+\gamma\max_{a'}Q(s',a')-Q(s,a)\right) \tag{9}$$

where α - learning rate and $\alpha\in(0,1]$; s' - next state obtained by performing action a in current state s; r - - the reward obtained by performing the action a in the current state s; $\max_{a'}Q(s',a')$ -- the next state s' under which all actions in that state s' are executed a' - the maximum Q value that can be obtained.

It is common to take $\alpha=1$ to simplify the above equation:

$$Q(s,a)=r+\gamma\max_{a'}Q(s',a') \tag{10}$$

From the Bellman equation, the reward r obtained after performing action a in state s plus the discounted value of the maximum value function $Q(s',a')$ that can be obtained in the next step is closer to the true value of $Q(s,a)$. The $Q-$ learning algorithm consists of the following five steps:

(1) Initialize the Q values with random values or 0;

(2) Selecting actions using a greedy strategy that balances exploration and exploitation, randomly selecting actions to execute with a probability of ε, and selecting actions to execute with a probability of $1-\varepsilon$ with the largest Q value of all actions in the current state;

(3) Execute the action obtained in step 2 and obtain the reward r for executing the action in the current state and the state s' for the next step;

(4) Update the Q value using the Bellman equation;

(5) Loop over steps 2-4 until the termination state.

After learning and updating the Q function, the optimal policy can be obtained by continuous iterative updating of the Q function and the policy π:

$$\pi(s)=\arg\max_{a\in A}Q(s,a) \tag{11}$$

II. B. 3) Deep learning

Deep Learning (DL) constructs a model by simulating the neural structure in the human body, which processes the data to obtain the laws and representation levels in the data, so that the model has the ability to classify and recognize. This paper is mainly based on convolutional neural networks. The neurons all have weights and the data is subjected to a weighted summation operation and the bias in the neurons is added and the result obtained is processed through an activation function to get the output:

$$O_j=f\left(\sum_{i=1}^{n}w_{ij}x_i+b_j\right) \tag{12}$$

where x_i - the i th input of the j th neuron; O_j - the i th output of the j th neuron; w_{ij} - the - the weight corresponding to the input x_i; b_j - the bias of the j th neuron; f - the activation function.

The output of a neuron is very dependent on the activation function and bias. The commonly used activation functions are Sigmoid, Tanh, ReLU, Softmax.

(1) The Sigmoid function has the property of being continuously conductible, with a maximum derivative at 0, which accelerates the training of the neural network. Namely:

$$f(x)=\frac{1}{1+e^{-x}} \tag{13}$$

(2) The Tanh function alleviates the output bias problem of the Sigmoid function and has an accelerating effect on the convergence of the model:

$$f(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{14}$$

(3) The ReLU function is effective in mitigating and accelerating the convergence of the gradient vanishing problem of Sigmoid and Tanh functions in practice:

$$f(x)=\max(0,x) \tag{15}$$

(4) Is more commonly used in multicategorization problems, where its output is a value representing a probability distribution such that the probabilities of all the categories are between 0 and 1 and add up to 1. i.e:

$$soft\max(x_i)=\frac{e^{x_i}}{\sum_j e^{x_j}} \tag{16}$$

II. B. 4) Actor-Critic Framework

The Actor-Critic algorithm [20] is able to update the strategy while exploring the environment, so the strategy will be able to adapt to the environment, and the exploration time is reduced. Actor makes the intelligent body output action a according to the current state s, and Critic calculates the value of Q according to the current state s and action a and passes it to Actor.

The Bellman equation of action value function $Q(s,a)$ and state value function $V(s)$ is established according to the Markov decision process, and the values of action and state are evaluated respectively to make the decision of the intelligent body better. Namely:

$$Q^{\pi_\theta}(s,a)=r^{a}_{s\to s'}+\gamma\sum_{s'}P_{s\to s'}V^{\pi_\theta}(s')=\bar{a}\left[r+\gamma V^{\pi_\theta}(s')\right] \tag{17}$$

$$\backslash V^{\pi_\theta}(s)=\sum_{a\in A}\pi_\theta(a|\ s)Q^{\pi_\theta}(s,a)=\bar{a}\left[Q^{\pi_\theta}(s,a)\right] \tag{18}$$

where $r^{a}_{s\to s'}$ - the reward obtained by performing the action a in the current state s; $P_{s\to s'}$ - the probability of transferring from the current state s to the next state s'.

The formula for updating the gradient of the strategy can be obtained from the above formula:

$$\nabla_\theta J(\theta)=\bar{a}_{\pi_\theta}\left[(r_t+\gamma V^{\pi_\theta}(s_{t+1})-V^{\pi_\theta}(s_t))\nabla_\theta\log\pi_\theta(s_t,a_t)\right] \tag{19}$$

In general both the action value function and the strategy are unknown, so a neural network is used to replace both functions, so the Actor-Critic algorithm learns both networks and uses gradient descent when updating the parameters, denoted $\theta_{t+1}\to\theta_t-\alpha\nabla_\theta\log\pi_\theta(s_t,a_t)V_t$, and α denotes the learning rate.

II. C.Deep Reinforcement Learning Based Method for Virtual Human Motion Generation

II. C. 1) Layered Strategy Learning Framework

The trajectory tracking [21] layer of hierarchical policy learning takes the joint pose information $q_{ref}=\{\hat{q}_1,\cdots,\hat{q}_n\}$ from the motion database as the reference action, builds a neural network, and uses the proximal policy optimization algorithm to train to obtain the optimal control policy $\pi_\theta\left(q_{tar}|\ s_s\right)$, to obtain the mapping of the virtual human from

the skeletal state s_s to the action q_{tar}. The stabilized proportional differential (SPD) controller receives the target action q_{tar} output from the trajectory tracking strategy π_θ and converts it into the desired moment τ_{tar} and the desired acceleration $\ddot{q}_{tar}$ for each joint to be passed to the muscle control layer. The muscle control layer forms a muscle synergy strategy $\pi_\phi\left(a|\ \ddot{q}_{tar}, s_m\right)$ through supervised learning and outputs muscle activation parameters a. The kinetic simulation system updates and feeds the muscle and bone states $s=(s_s, s_m)$ in real time based on a.

II. C. 2) Trajectory tracking layer

The strategy architecture of PPO algorithm based on Actor-Critic framework is analyzed in the specific case, and two neural networks sharing the parameters of strategy and value function are built to complete the strategy optimization, in which Actor network is used to get the trajectory tracking strategy π_θ, and the strategy loss term updating network is obtained by using the old-new network ratio and the dominance function calculated by GAE; Critic network is used to obtain the value function $V(s)$, using TD-error as the loss term update network.

The state space, i.e., to meet the strategy decision-making needs but not too redundant to affect the training efficiency of the algorithm. Since the control object of the trajectory tracking strategy is the joint posture of the virtual character, the dynamic information of its bones and joints is mainly chosen as the content of the state space, i.e., $s=\left(p, \dot{p}, q, \dot{q}, \phi\right)$, which is 210-dimensional data.

In motion control studies using joint actuation, the action can be simply set as $a=\left(f_1, f_2, \cdots, f_n\right)$. In order to optimize the control effect while achieving the motion of the virtual character generated using the .muscle actuators, a higher level of abstraction control is used to set the action space of the trajectory tracking strategy to the desired pose of each joint, i.e., $a=\left(\hat{x}_{root}, \hat{q}_{root}, \hat{q}_1, \cdots, \hat{q}_n\right)$.

Reward function design:

The reward function is an incentive mechanism, trajectory tracking requires the virtual human to be as similar as possible to the reference motion in each frame, thus designing the reward function r that can respond to several imitation metrics simultaneously, and due to the strong positive correlation between the metrics, the metrics are chosen to be multiplied to improve the accuracy of imitation:

$$r = r_q \cdot r_{\dot{q}} \cdot r_e \cdot r_{com} \tag{20}$$

r_q is the stance bonus, which encourages the virtual character to match the joint rotations of the reference action as the sum of all joint stance quaternion differences:

$$r_q = \exp\left(-\frac{1}{\sigma_q^2}\sum\nolimits_i \left\|\hat{q}_i\ !\ \ q_i\right\|^2\right) \tag{21}$$

where q_i and $\hat{q}_i$ are the joint poses of the virtual character and the reference motion expressed in quaternions, i is the joint index, and σ_q is the pose reward coefficient, respectively.

$r_{\dot{q}}$ is the velocity reward, which encourages the virtual character to match the motion velocity of the reference motion as the sum of all joint angular velocity differences:

$$r_{\dot{q}} = \exp\left(-\frac{1}{\sigma_{\dot{q}}^2}\sum\nolimits_i \left\|\hat{\dot{q}}_i - \dot{q}_i\right\|^2\right) \tag{22}$$

where $\dot{q}_i$ is the joint angular velocity, $\hat{\dot{q}}_i$ is the desired joint angular velocity calculated by differencing the reference motion data, and $\sigma_{\dot{q}}$ is the velocity reward coefficient.

r_e is the end position reward, which encourages the end parts of the virtual character (including the left foot, right foot, left hand and right hand) to match the corresponding values of the reference motion in terms of their 3D positions in the world coordinate system, and is the sum of the differences of all end position coordinates:

$$r_e = \exp\left(-\frac{1}{\sigma_e^2}\sum\nolimits_i \left\|\hat{p}_j^e - p_j^e\right\|^2\right) \tag{23}$$

where p_j^e and $\hat{p}_j^e$ are the end positions of the virtual character and the reference motion, respectively, j is the end position index, and σ_e is the end position reward coefficient.

r_{com} is the center of mass reward, which encourages the center of mass position of the virtual character to match the corresponding value of the reference motion:

where p_{com} and $\hat{p}_{com}$ are the center of mass positions of the virtual character and the reference motion, respectively, and σ_{com} is the center of mass reward coefficient.

SPD Controller:

The SPD controller is used to complete the computation of the joint desired pose $\hat{q}_j$ to the joint desired moment τ_j and provided to the muscle control layer. There are two types of joints with different degrees of freedom in the virtual human model: rotary joints and ball-and-socket joints. The rotary joints have only one degree of freedom, and the scalar value $\hat{q} \in \square$ can be used to directly represent the desired rotation angle, and the calculation of the moments is accomplished by treating the SPD controller as a spring-damped system; The ball and socket joints have three degrees of freedom, and the rotation angle is expressed using quaternions to avoid the gimbal deadlock problem caused by the use of Euler angles, so the moment calculation formula of the SPD controller is rewritten as:

$$\tau^n = -k_p \left[map(q^{n+1} \,!\, \hat{q}^{n+1}) \right] - k_d \dot{q}^{n+1} \tag{25}$$

where $\dot{q} \in \square^3$ is the angular velocity of the ball-and-socket joint, and $q^1 \,!\, q^2$ denotes the quaternion difference, computed as $q^1 \,!\, q^2 = q^1 \overline{q}^2$, with $\overline{q}^2$ being the conjugate of q^2. To compute the target moment using the difference in rotation angles represented by $q^1 \,!\, q^2$, the quaternion q is mapped to the rotational form of the axial angle representation via $map(q)$:

$$\text{map}(q) = \begin{cases} \theta = \|q\|^2 \\ v = \dfrac{q}{\|q\|^2} \end{cases} \tag{26}$$

where θ is the rotation angle expressed in radians and v denotes the rotation axis.

Set the SPD controller of the root joint with proportional gain $k_p = 1000$ and differential gain $k_d = 100$, and the SPD controllers of the other joints with $k_p = 500$ and $k_d = 50$.

II. C. 3) Muscle control layer

In the human skeletal-muscular system, since the number of human muscles is much larger than the sum of the degrees of freedom of the human joints, deriving the degree of muscle activation with known moments or accelerations of the joints results in an infinite solution, so muscle-based motion control is actually a redundant control problem, which requires determining the muscle synergistic control strategy through the optimization concept.

The muscle control strategy training process uses a loss function containing three subterms L_φ^{muscle}:

$$L_\varphi^{muscle} = \mathrm{E}\left(w_{ma} L_\varphi^{ma} + w_{ta} L_\varphi^{ta} + w_{ee} L_\varphi^{ee} \right) \tag{27}$$

where ω is the weight of the corresponding subterm.

L_φ^{ma} is the muscle activity loss term, which reflects the muscle work efficiency and is an important indicator for deciding the muscle driving strategy under many possible coordination schemes, and is the sum of squares of the muscle activation level:

$$L_\varphi^{ma} = \left\| a(\varphi) \right\|^2 \tag{28}$$

L_φ^{ta} is the movement tracking loss, which reflects the overall movement completion and is the sum of the acceleration differences across all joint degrees of freedom:

$$L_\varphi^{ta} = \sum\nolimits_i \left\| \hat{\ddot{q}}_i - \ddot{q}_i \left[a(\varphi) \right] \right\|^2 \tag{29}$$

where $\hat{\ddot{q}}$ is the incoming target acceleration at the end position from the trajectory tracking layer, and $\ddot{q}$ is the actual measurement of the degree of muscle activation that will be current.

L_{φ}^{ee} is the end position loss, which reflects the degree of movement completion at the end position, and is the sum of the acceleration differences at the end position:

$$L_{\varphi}^{ee} = \sum_{j} \left\| \hat{\ddot{p}}_j - \ddot{p}_j \left[a(\varphi) \right] \right\|^2 \tag{30}$$

where $\hat{\ddot{p}}$ is the incoming end-joint acceleration from the trajectory tracking layer, and $\ddot{p}$ is the actual measurement of end-position acceleration at the current level of muscle activation.

III. Practical effects of virtual character performance generation technology in film and television animation

III. A. Deep Reinforcement Learning Based Virtual Human Motion Generation

In fact learning a muscle-driven control strategy is three times faster than learning a moment-driven control strategy. The motion tracking layer learns and operates at the frame rate of the reference data, which is typically 45 frames per second. The muscle-driven layer learns and operates at the rate of the forward dynamics simulation, which is typically 800 to 1600 frames per second. The construction, training and prediction of the neural network was done using the Pytorch deep learning framework under the Ubuntu operating system, and the model was optimized using the Adam optimizer. The design of the neural network hyperparameters is shown in Table 1; Table 2 shows the design of the SPD controller gains, which affect the learning effect of the model and the result output. In addition, in the SPD controller, if a single gain parameter is used it may not be applicable to the whole motion control system. It is usually necessary to select different gain parameters according to the characteristics of the joints to ensure system stability and control accuracy.

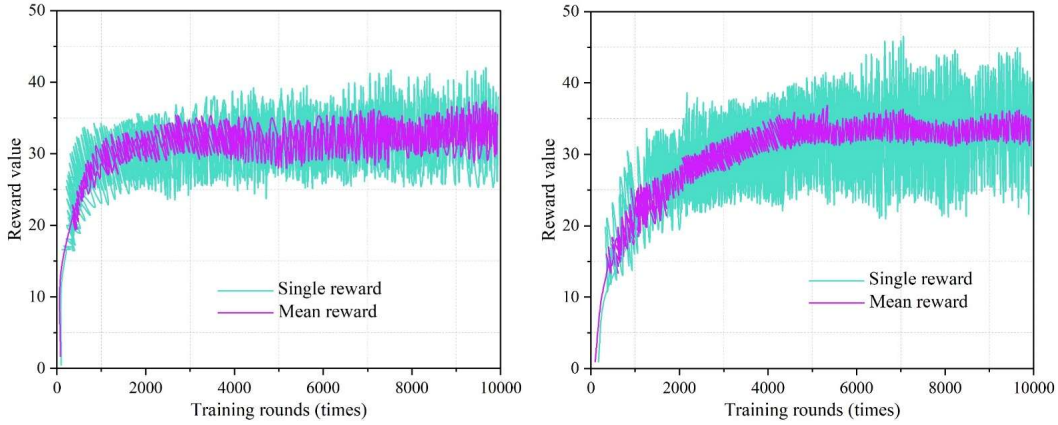
Table 1: Neural network hyperparametric design

Hyperparameter	Data
Learning rate	10^{-3}
Batch size	225
Muscle experience pool size	55000
Discount factor	0.999
Cross entropy coefficient	0.005
GAE parameter	0.98

Table 2: SPD controller gain design

	Proportional gain k_p	Breeze gain k_d
Root joint	2000	200
Other joints	1000	100

Input data of 2 movements extracted from the video, single-leg squat and rotational kick, to complete the training under the hierarchical strategy learning based on deep reinforcement learning. The training results are shown in Fig. 1, where (a) and (b) represent the single-leg squat reward value and the kick reward value, respectively. The blue color indicates a single reward, and the red color is the curve made by taking the average value of every 3 rewards, which can be seen that the reward curve rises smoothly. Since the running and walking movements are relatively simple, the virtual human learns and trains better, and the reward value increases faster, and after about 6000 training rounds, the average reward value starts to stabilize. The rotation and kicking movement is the most complex, the virtual human has to maintain balance and coordinate the muscles of the whole body to complete the rapid rotation and kicking movement, the training effect is relatively the worst, after 10,000 training rounds, the reward value stays at about 30. In addition, the learning effect of the dummy decreased for both the dancing movement of one-legged squatting and the ballet movement of spinning and kicking, with the reward value remaining at around 30.

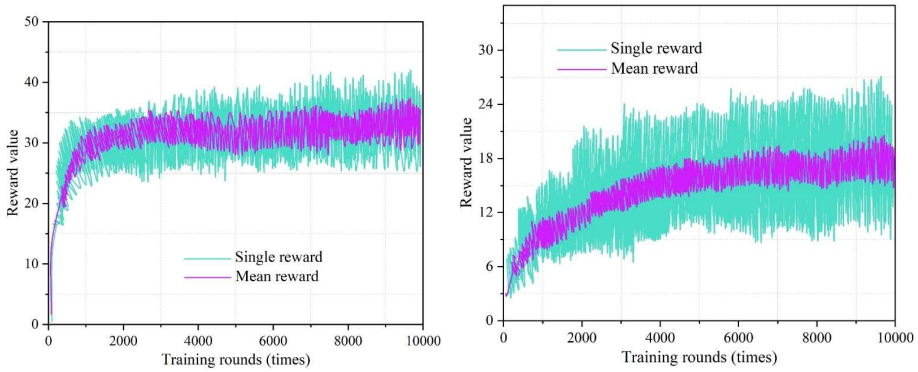


(a) The reward value of a squat at the foot (b) Kicks reward

Figure 1: Training results

In this study, the energy reward of muscles is considered during the design of the reward function to illustrate the effect of muscle activation degree on the realization of diverse motions by the virtual human. In order to verify the effectiveness of the muscle-driven virtual human movement proposed in this paper, ablation experiments are done on the energy reward function of the muscles to compare the change of the reward curve, the optimal value of the reward and the simulation effect.

The results of the ablation experiments are shown in Fig. 2, where (a) and (b) represent the single-leg squat reward value and the reward value after ablating the muscle reward for single-leg squat, respectively. Taking the one-legged squat as an example, the reward curve rises slower and the maximum reward value decreases from 32.08 to 16.39 after the ablation experiment, and fluctuates after reaching the maximum reward value, which indicates the effectiveness of the muscle-driven virtual human movement, and shows the importance of effectively controlling the degree of muscle activation for the virtual human to generate a more natural and fluent diversity.



(a) The reward value of a squat at the foot (b)The reward of the loss of muscle reward at the foot of the single foot

Figure 2: Ablation experiment results

III. B. Research on users of virtual human film and animation system

In order to evaluate the system objectively, this paper organizes a user study using system testing combined with a questionnaire. We recruited 20 volunteers (13 males and 7 females), with participants in the age range of 20-25 years old, a small number of whom claimed to have good human-computer interaction skills and a certain level of understanding of VR/AR technology. The test hardware environment consisted of a PC, a HoloLens headset and a Leap Motion somatosensory controller, and the software environment was a Unity 3D environment based on ML-Agents. Volunteers first need to record the interaction trajectory of human hands and objects in the virtual environment through Leap Motion, and then observe the generated virtual human motion sequence in an all-round way through the HoloLens headset, and score the visualization effect and usability of the system in terms of subjective experience and visual presentation effect of film and animation.

III. B. 1) Evaluation of visualization effects

This paper focuses on the naturalness and stability indexes of the virtual person's grasping of objects, so the evaluation of the visualization effect is the core of this system. In this evaluation system, volunteers mainly play the role of students, according to which they evaluate the animation effect of the virtual teacher's demonstration. We use a Likert scale (from 5 (strongly agree) to 1 (strongly disagree)) to record the user's subjective evaluation. The evaluation results of the visualization effects designed in this paper are shown in Table 3. Through the obtained rating data, after analyzing, we found that the volunteers' average ratings of the performance content are all greater than 4. It is evident that most volunteers think that the system generates the virtual teacher's finger movements more smoothly and naturally, the accuracy of the operation is higher, and they can accept the virtual teacher's film and television effects.

Table 3: Visual effect evaluation results

Evaluation index	Mean	Standard deviation
The teacher's fingers are fluent and natural	4.2751	0.6485
The experimental operation of the demonstration teacher is accurate	4.4392	0.4903
The auxiliary information of the system is clear and easy to understand	4.4507	0.5705

III. B. 2) Evaluation of usability

In addition to visualization results, usability metrics that reflect the operational flow of the system are equally important for interactive systems. Usability mainly includes five aspects: ease of learning, ease of memorization, ease of use, error frequency, and overall satisfaction. In the subjective evaluation system of the system framework, the user mainly plays the role of a teacher, independently designs a specific user interaction trajectory, and then participates in the UI interaction and observes the generated interaction animation sequence through the HoloLens headset. The usability evaluation results are shown in Table 4. Analyzing the data, the following results can be obtained:

(1) The usability data of the system had generally high mean values (3.6212-4.6616) and low standard deviations (all <0.5), indicating that most of the volunteers thought the system was easy to learn, easy to memorize, easy to use, had a low error rate, and had a high level of overall satisfaction.

(2) The mean values of the indicator "The navigation mode of this system has a relatively high operational efficiency" in "Ease of Use" and the standard deviation of the indicator "The misoperation rate of this system is relatively low" in "error frequency" are relatively low. Some of the users who participated in the research suggested that if they operated the system for a long time, their upper limbs would be fatigued due to the suspension of the upper limbs. This is also due to the fact that these users have not been exposed to virtual character interaction systems before, and it is still difficult for them to get started.

Table 4: Usability evaluation results

	Evaluation index	Mean	Standard deviation
Learnability	I can use the system faster on the ground	4.6616	0.4197
	I can use the system quickly and skillfully	4.2402	0.4638
Erotica	I can easily remember how to use the system quickly	4.1617	0.4306
	The operating process of the system is not easy to forget	4.177	0.3834
Ease of use	It's easier to find the information I need	4.5308	0.3374
	The system is efficient in navigation	3.7555	0.2449
Error frequency	The error rate of the system is low	3.6212	0.3187
Overall satisfaction	In general, I think the system is easy to use	4.1263	0.1631
	In general, I think the system is more efficient	4.1716	0.1391
	In general, I think the system is more satisfactory	4.4003	0.4041

III. B. 3) Comparative Evaluation of the Effectiveness of Virtual Character Performance Generation

In order to conduct a horizontal comparison among the performance generation effects of different virtual characters, we also conducted the above-mentioned Liket table investigation on two different performance effects, namely "real-person demonstration" and "video explanation", and added two indicators, "low cost" and "high scalability". The comparison and evaluation results of the virtual character performance generation effect are shown in Table 5. We can learn that:

(1) Although the visualization effect of the virtual character performance demonstration system (4.3873) is slightly lower than that of the real-life demonstration (4.8685) and video explanation (4.6644), its scalability is much higher than the other two. If the prototype system designed in this paper is supplemented with appropriate visual or auditory special effects, it may further increase the experimenter's immersion and enhance the performance quality.

(2) The average value of the “low cost” index of the virtual character performance demonstration system (1.9034) is slightly higher than that of the live demonstration (1.776). It should be noted that the AR system in this paper replaces the more complex motion capture devices or massive data sets, but compared with lightweight media devices such as video, the hardware overhead of this system is still large. With the development of computer hardware and software technology, we believe that the cost problem will eventually be effectively solved.

Table 5: Comparison and evaluation results of virtual role performance

Evaluation of teaching effect	Pattern name	Life demonstration	Video	Virtual role performance demonstration
Visual effect	Mean	4.8685	4.6644	4.3873
	Variance	0.1068	0.5339	0.5694
Good availability	Mean	4.8303	4.0374	4.1252
	Variance	0.0604	0.7597	0.8675
Cost low	Mean	1.776	4.0523	1.9034
	Variance	0.632	0.328	1.0486
High scalability	Mean	1.8512	3.6786	4.7528
	Variance	0.2799	0.4068	0.4375

III. C. Analysis of the effect of virtual character stylized action generation

In this paper, a user study was also conducted to further validate the diversity and rationality of the present method for generating action style attributes. The experiment invited 20 subjects with age distribution between 18 and 45 years old, all of whom had normal or corrected-to-normal vision, were not color-blind, and had long-term experience in animation, games, and other fields. In order to verify the diversity and reasonableness of the action style attributes, the experiment compared the method of this paper with the random sampling method, which is still based on the instruction parsing method proposed in this paper, but replaces the module-generated style labels with randomly sampled style labels, and generates the action results of not shorter than 150 frames for each of the 10 instructions respectively. All subjects were asked to watch 10 groups of videos, each group of videos consisted of the same instruction as input, which were the results generated by the method in this paper and the random sampling method respectively. Subjects filled out a questionnaire after each set of videos. The following questions were included:

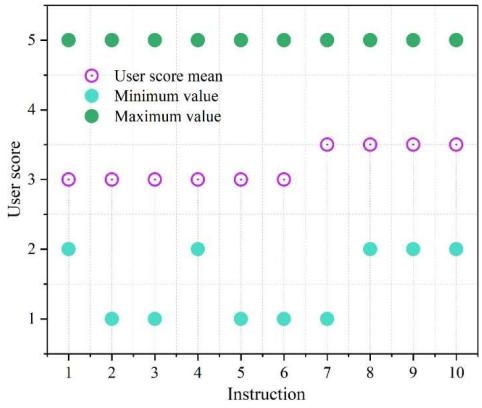
1) Score the difference between the two results with a rating interval of an integer between 1 and 5, where 1 represents the smallest difference and 5 represents the largest difference.

2) Score the reasonableness of the two results separately, where 1 represents the least reasonable and 5 represents the most reasonable.

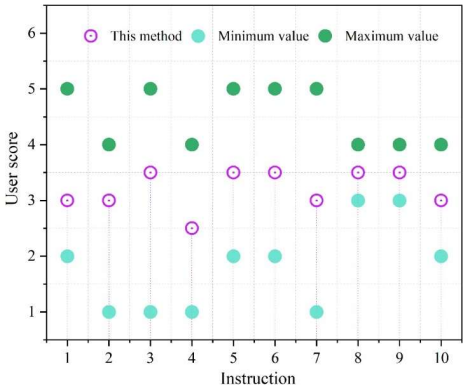
The results of user scoring analysis are shown in Fig. 3, where (a) to (c) represent the user scores on the differences, the user scores on the reasonableness of this paper's method and the user scores on the reasonableness of the random sampling method, respectively. The horizontal axis coordinates “instructions 1-10” in the figure correspond to the corresponding instructions.

In terms of user ratings, this paper's method improves the mean of its action ratings by 1.3 points in terms of reasonableness compared to the random sampling method. The paired-sample t-test was applied to the user scores, and the significance level was $\alpha=0.01$. The results showed that the mean score of the rationality score of this paper's method was 3.2, which was significantly better than that of the random sampling method with an average score of 1.9.

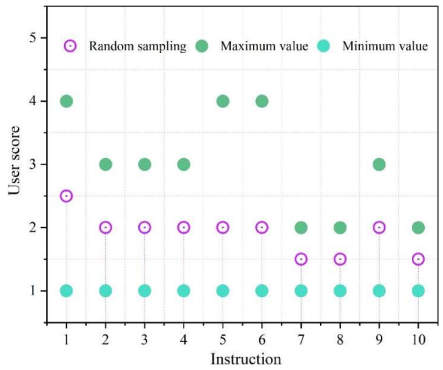
Different characters usually have different attribute features such as tall, short, fat, thin, etc., and the instructions described in the text also generally contain a variety of emotional features, this paper unifies these diverse features into style attribute labels, generates a style that conforms to the character and instructions through instruction parsing, and drives the action generation module to generate a sequence of actions that matches the style, which avoids the problem that the output is too monotonous, and enhances the reasonableness of the experimental results,. It proves the effectiveness of the method framework of this paper. The results show that this paper's method can generate actions with large differences according to different style labels, proving the diversity of action style attributes.



(a)User ratings for diversity



(b)The user's evaluation of the rationality of the method



(c)The user's score on the rationality of random sampling method

Figure 3: User score analysis results

IV. Conclusion

The virtual character performance generation technique combined with deep reinforcement learning and muscle-driven control strategies can effectively enhance the performance of virtual characters in film and television animation. With the method proposed in this study, the virtual character is able to achieve a high degree of naturalness and fluency when performing complex actions. In the experiment, when the virtual character completed the one-legged squatting action, the reward value stabilized after 6000 training rounds, indicating that the learning process was effective. Especially in the muscle drive control, the critical role of the degree of muscle activation on action generation was verified by ablation experiments. In the one-legged squat, the maximum value of the reward for controlling the degree of muscle activation was 32.08, while the maximum value after removing this reward was 16.39, which proved that the effect of muscle drive on action generation was significantly improved.

The results of the user survey also showed that the naturalness and smoothness of the virtual character's movements were highly rated, and the operability and visual effects of the system received positive feedback from users. Although there is a certain gap in hardware cost compared to traditional methods, the system has higher scalability and application potential, and the hardware cost problem is expected to be solved in the future with the advancement of technology. Therefore, this method provides an effective technical path for virtual character performance generation, which has a wide application prospect in the field of film and television animation.

Acknowledgements

1. This work was supported by Analysis of the impact of "Internet +" on the sports industry in the context of the new crown pneumonia epidemic - using Xi'an as an example.

2. This work was supported by Research on the Development Path of Aerobics Dance in Shaanxi Province from the Perspective of Healthy China.

References

[1] Chanpum, P. (2023). Virtual production: Interactive and real-time technology for filmmakers. Humanities, Arts and Social Sciences Studies, 9-17.

[2] Chen, D., & Yang, F. (2020, February). Application of VR virtual reality in film and television post-production. In IOP Conference Series: Materials Science and Engineering (Vol. 750, No. 1, p. 012163). IOP Publishing.

[3] Al-Kamil, S. J., & Szabolcsi, R. (2024). Optimizing path planning in mobile robot systems using motion capture technology. Results in Engineering, 22, 102043.

[4] Niu, Z., Lu, K., Xue, J., Qin, X., Wang, J., & Shao, L. (2024). From Method to Application: A Review of Deep 3D Human Motion Capture. IEEE Transactions on Circuits and Systems for Video Technology.

[5] Papadimitriou, I. (2024). Employing emerging technologies such as motion capture to study the complex interplay between genotype and power-related performance traits. Frontiers in Physiology, 15, 1407753.

[6] Pizzo, A. (2016). L'attore e la recitazione nella motion capture. ACTING ARCHIVES REVIEW, 6(11), 38-69.

[7] Mishra, V., & Kiourti, A. (2021). Wearable sensors for motion capture. Antenna and sensor technologies in modern medical applications, 43-90.

[8] Furtado, J. S., Liu, H. H., Lai, G., Lacheray, H., & Desouza-Coelho, J. (2019). Comparative analysis of optitrack motion capture systems. In Advances in Motion Sensing and Control for Robotic Applications: Selected Papers from the Symposium on Mechatronics, Robotics, and Control (SMRC'18)-CSME International Congress 2018, May 27-30, 2018 Toronto, Canada (pp. 15-31). Springer International Publishing.

[9] Peng, M. (2024). The application of digital media technology in the post-production of film and television animation. Media and Communication Research, 5(2), 129-134.

[10] Yang, J. (2024). Analysis of Motion Capture Technology Research and Typical Applications. Applied and Computational Engineering, 112, 130-138.

[11] Zeng, R. (2021, May). Research on the application of computer digital animation technology in film and television. In Journal of Physics: Conference Series (Vol. 1915, No. 3, p. 032047). IOP Publishing.

[12] An, D. (2022). Technology-driven Virtual Production: The Advantages and New Applications of Game Engines in the Film Industry. Revista Famecos, 29(1), e43370-e43370.

[13] Vugar, A. E. (2015). The impact of the Motion Capture technology on artistic process in film art. European Journal of Arts, (2), 10-12.

[14] Zhang, M. Y. (2013). Application of performance motion capture technology in film and television performance animation. Applied Mechanics and Materials, 347, 2781-2784.

[15] Woodcock, R. (2016). Capture, hold, release: an ontology of motion capture. Studies in Australasian Cinema, 10(1), 20-34.

[16] Haratian, R. (2022). Motion capture sensing technologies and techniques: A sensor agnostic approach to address wearability challenges. Sensing and Imaging, 23(1), 25.

[17] Sharma, S., Verma, S., Kumar, M., & Sharma, L. (2019, February). Use of motion capture in 3D animation: motion capture systems, challenges, and recent trends. In 2019 international conference on machine learning, big data, cloud and parallel computing (comitcon) (pp. 289-294). IEEE.

[18] LiboSun,YongxiangWang & WenhuQin. (2024). A language-directed virtual human motion generation approach based on musculoskeletal models. Computer Animation and Virtual Worlds,35(3),

[19] Zhongkai Zhan. (2024). Application of Skeletal Skinned Mesh Algorithm Based on 3D Virtual Human Model in Computer Animation Design. International Journal of Advanced Computer Science and Applications (IJACSA),15(1),

[20] Zhu Hongjun,Xie Yong & Zheng Suijun. (2024). A double Actor-Critic learning system embedding improved Monte Carlo tree search. Neural Computing and Applications,36(15),8485-8500.

[21] Jiaju Zhu,Zhong Zhang,Runnan Liu & Junyi Liu. (2025). Footwork recognition and trajectory tracking in track and field based on image processing. Scientific Reports,15(1),10813-10813.