International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-14

https://doi.org/10.70917/ijcisim-2026-0086
Article

Synchronized Generation of Appearance and Behavior of Cinematic Animated Characters Using Deep Generative Models

Wei Peng [1,2,*]

[1] Cheongju University, Chungcheongbuk-do, Cheongju-si, 28497, Korea
[2] Commerce Boustead College, Tianjin University, Tianjin, 300384, China; pengwei6666660205@163.com

Abstract: The continuous development of depth generation technology has improved the efficiency and quality of animated character generation. In this paper, we construct a cascade classifier based on depth generation technology, use the nearest neighbor difference to adjust the size of the expression region and complete the feature screening to determine the face expression feature set. By designing the loss function, the style consistency of the generated expression images is maintained. Introduce foreground mask mechanism and add expression magnitude discriminator in the expression editing model to improve the quality of expression generation. Using behavioral probabilistic finite automata (BPFA) to constrain the uncertain behavior generation of animated characters, and improving the fitness of generated behaviors and expressions through probabilistic calculation. The study shows that the animated character generation frame rate of this paper's method is >90f/s, the number of textures is >60MB, and the accuracy is high in 4 angles. During expression editing, this paper's method achieves stable convergence with only 59 iterations, and has the best effect on extracting the feature points of the face in the frontal angle. The generated movie-level animated character expressions and behavioral synchronization effects are all greater than 90%.

Keywords: deep generation technique; cascade classifier; loss function; BPFA; animated character generation

1. Introduction

Currently, deep learning technology, with its promising prospects, has been successfully applied to multiple fields over the past decade and has achieved notable application outcomes [1-2]. Analysis of previous research findings indicates that before being applied to a new field, deep learning technology requires continuous improvement of its functionalities to better adapt to new application contexts and enhance its practical applicability [3-4]. In image processing, the generation of animated characters represents an emerging and highly engaging area of research [5]. Given the diverse expressive capabilities of animated characters, which have garnered widespread popularity, numerous scholars have conducted research on animated character avatars [6-7]. Furthermore, an increasing number of animated works have emerged in society. This underscores the crucial role of animated character design in the creation of animated works.

Currently, most research on animation character generation based on deep generative models focuses on generative adversarial networks (GANs). For example, Jin et al. [8] trained a GAN model on an animation character facial dataset, and through quantitative analysis and case studies, successfully designed a deep generative model capable of stably generating high-quality animation character expressions. Lee et al. [9] constructed a hybrid GAN that integrates a reversible neural flow generator, while introducing two cartoonization losses, effectively addressing the inability to accurately capture the unique features of cartoon styles, and demonstrated the model's superiority through experiments. Khalid et al. [10] assigned values between 0 and 1 to the training data using a sigmoid function and improved the generator and discriminator of the GAN to enhance the model's performance in generating and classifying game animation character images. Tan [11] compared different GAN architectures, such as

DCGAN, CycleGAN, and SNGAN, in terms of the quality and diversity of generated animated facial images, and summarized the applicability of different GAN architectures for animated facial image generation based on the comparison results. Yi et al. [12] proposed an optimized model based on DRAGAN, employing a Conv-BN-Relu-Pooling structure and combining it with an alternating gradient update procedure for training. The optimized DRAGAN model demonstrated significant improvements in visual quality, FID. Lungu-Stan et al. [13] utilized AnimGPT and DenoiseAnimGPT decoder transducers to enhance the performance of animated character image generation. The results showed that under a 50-frame image context, the errors for AnimGPT and DenoiseAnimGPT were 0.345 and 0.2513, respectively, with satisfactory generation outcomes.

Research on the generation of animated character behavior is relatively scarce. Zheng et al. [14] optimized the generative adversarial network using a particle swarm algorithm, improving the SSIM score by 0.13 and the PSNR by 4.2 dB. The combination of the generative adversarial network with particle swarm optimization enhanced the performance of synchronous generation of animated characters and interactive behavior. Nishimura et al. [15] embedded three factors—interaction intensity, temporal evolution, and temporal resolution—into a deep generative model, to model character behavior during interactions. Subjective evaluation results indicated that this method improved the generation quality based on character behavior. Yang et al. [16] validated the impact of reinforcement learning-based intelligent agent generation models on animated character creation. The results showed that reinforcement learning promoted the creativity and development of character attributes in animated character design, demonstrating potential for assisting in character behavior generation. Previous studies have advanced the application of deep generative models in animation generation, but limitations in character motion generation significantly impact the expressiveness and generalizability of generated animations [17].

In this paper, after normalizing the image attributes using grayscale conversion, we construct a cascade classifier based on improved deep learning techniques to achieve multi-scale, multi-node, and omni-directional face feature detection. The L1 loss function is designed to constrain the expression image style, reduce the loss of details, and improve the expression realism of animated characters. Introducing foreground mask mechanism and adding expression amplitude discriminator, combined with coding and decoding process, to systematically optimize the expression generation effect of expression editing model. Incorporate behavioral probabilistic finite automata (BPFA) to synchronously generate animated character behaviors that are consistent with the expression state by calculating the behavioral transfer probabilities of various types of action functions.

2. Technical Implementation of Synchronized Generation of Appearance and Behavior of Movie-Quality Animated Characters

2.1. Design of Animated Character Appearance Generation Method Based on Improved Deep Learning

2.1.1. Face Expression Feature Extraction Based on Improved Deep Learning

Before generating movie-level animated character facial expressions, real human face facial expression images are used as the basis for generating animated expressions, and before extracting the facial expression features of the real images, images with different attributes are processed using grayscale conversion to unify the image attributes, and then the improved deep learning is used to extract the facial expression features. The improved deep learning technique has the ability to differentiate between facial expression features during facial expression feature extraction, and real facial expression images are processed using the improved deep learning technique, from which facial features are extracted and used as the basis for generating movie-level animated character expressions. An image consists of multiple units, and before processing the image is divided into multiple blocks on the basis of units, each block corresponds to a sub-region denoted as $A_1A_2A_3\cdots A_n$. The corresponding histogram is extracted from each sub-region, and the calculation formula is as follows:

$$g_{i,j} = \sum_{x,y} P((x,y) \in A_i) \tag{1}$$

where $i \in [1,n]$, $j \in [1,n]$, x and y denote the image pixels, and $g_{i,j}$ denotes the histogram within the i th subregion. The total histogram $G_A = (g_{i,0}, g_{i,1}, \cdots, g_{i,n})$ is formed by connecting the histograms of each sub-region, which is used as a sequence of real face expression features in the article.

In detecting real face facial expression features, a cascade classifier is constructed using improved

deep learning techniques, and the input real face features are detected using the cascade classifier multiscale. The multi-scale detection is mainly for images with more pixel points, before detecting the features, the cascade classifier is trained to initialize the processing search window based on the size of the input image, the search window is continuously trained according to the changes in the input image, and the face features are searched, and the same face feature regions are merged together. After the search is completed, a large number of sub-windows are output, and the images are screened by the cascade classifier, and the judgment of whether to throw away the region is carried out once at each node, so as to finally obtain a reasonable set of face expression features.

The cascade classifier detects face facial expression images when the image size is not fixed. In order to improve the efficiency, the image size is adjusted using the nearest neighbor difference, when the operation of image processing is zoomed in, it may lead to increased jaggedness of the image and stiffness of the image edges, and the image edges are softened using the ANTIALIAS method to deal with the edges of the image and increase the realism of the edges of the image. Then the image pixels are normalized to get reliable and complete face expression features.

2.1.2. Designing the Loss Function

In order to ensure that the details of the generated animated facial expressions are consistent with the details of real human face expressions, the style and content of the generated expression images are constrained. Assuming that the set of real human face expression images is I and the set of movie-level animated character expression images is C, and the input images $x \in I$ and $y \in C$, the generated facial expression image is $J(K_t(x), K_s(y))$, denoted as ff. Where J denotes the decoder, K_t denotes the content encoder, and K_s denotes the style encoder. Assuming that the input feature image has two scales, let the outputs of the two scales be Z_1 and Z_2, and the output of the discriminative network with improved deep learning is $Z_1 \times Z_2$, the loss of the discriminative network is denoted as:

$$\varphi = 1 - Z_1(ff) * Z_2(ff)^2 \tag{2}$$

$$\varphi_s = \frac{1}{2}[(1 - Z_1(y)^2) + (Z_1(ff)^2)] + \frac{1}{2}[(1 - Z_2(y)^2) + (Z_2(ff)^2)] \tag{3}$$

In the case of unpaired data, it is difficult to realize the constraints on the content by real expression features. Therefore, the form of reconstruction is used to ensure the content invariance of the expression image. Figure 1 shows the expression image reconstruction process.

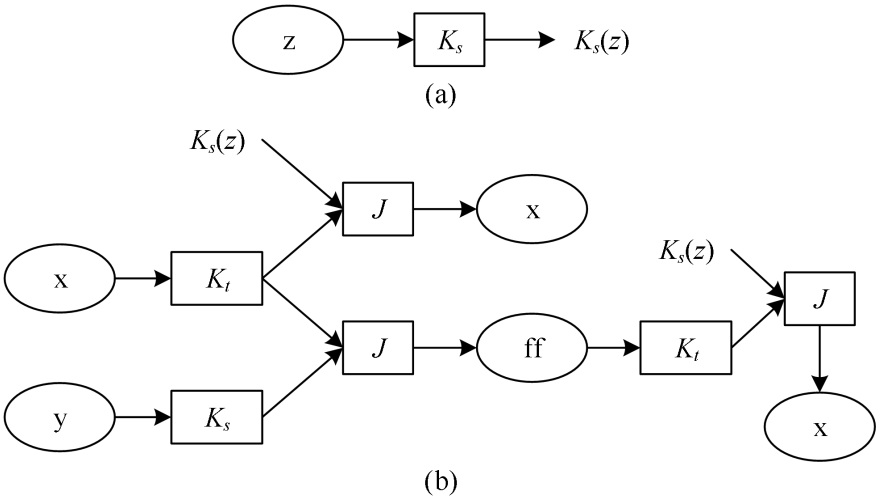


Figure 1. The process of expression image reconstruction.

Input image x, its own content features are fused together with the style features of the real face expression to get the reconstructed image, which is labeled as rec_x1. In order to ensure that the

content remains unchanged, it is necessary to ensure as much as possible that the content of the generated facial expression image is consistent with that of the original image during the reconstruction process by fusing it together with the stylistic features of the real human face expression image, and marking the resulting reconstructed image as rec_x2. The formula for the two images is:

$$rec\_x1 = J(K_t(x), K_s(z)) \tag{4}$$

$$rec\_x2 = J(K_t(ff), K_s(z)) \tag{5}$$

In the above formula, z denotes an image similar in style to the input image, in order to ensure the content and style consistency between the reconstructed image and the input image, the expression image is constrained using the L1 loss function in deep learning as a basis for judging the content loss of the network, the formula is as follows:

$$\begin{cases} L_1 = \left\| x - rec\_x1 \right\|_1 \\ L_2 = \left\| x - rec\_x2 \right\|_2 \end{cases} \tag{6}$$

In order to keep the content of the expression image unchanged and retain the detailed features, the content loss of the expression image is constrained on the basis of Equation (5), and the weights of the content loss of the real expression image and the generated expression image are represented by α_1 and α_2, and the weight size is adjusted with the support of the deep learning judgment network. The specific content is:

$$L' = \alpha_1 L_1 + \alpha_2 L_2 \tag{7}$$

The obtained loss function is used for the optimization of the generative network and discriminative network for generating facial expressions of movie-quality animated characters, and after the optimization is completed, the designed loss function is used as the basis for generating facial expressions of movie-quality animated characters through improved deep learning.

2.2. Expression Editing Model and Important Module Analysis

2.2.1. Expression Editing Model

The expression editing model (AMEE-GAN) refers to the network structure of implicit conditional generative adversarial network (IcGAN) and generative adversarial network for face attribute editing (AttGAN), and makes two improvements based on them, one of which is to introduce the foreground masking mechanism, so that the model focuses on the region where the expression of cinematic animated characters changes, and reduces the correlation with other face attributes; and the second is to add the The second is to add a movie-animated character expression amplitude discriminator, so that the expression of the movie-animated character in the generated image is as close as possible to the expression of the target image.The network of AMEE-GAN can be divided into three parts according to the functional modules: the image generation module, the foreground mask module, and the image discriminator module. The image generation module is responsible for generating the intermediate image after the expression change; the foreground mask module generates the foreground mask, which is multiplied with the intermediate image and the source image to get the final generated image; the image discrimination module consists of an image discriminator and an expression magnitude discriminator, which share most of the weights of the network layer, where the image discriminator is used to determine whether the input image is from the real or generated image, and the expression magnitude discriminator is used to determine whether the input image is from the real or generated image, and the expression magnitude discriminator is used to determine the expression magnitude of the input image. The image discriminator is used to determine whether the input image is derived from a real image or a generated image, and the expression magnitude discriminator is used to recognize the magnitude of the expression in the image.

2.2.2. Image Generation Module

The image generation module accepts the expression parameter a and the input image I to generate the intermediate generated image I_a^m after the expression change. The network structure of this module belongs to the type of conditional generative adversarial network, i.e., some additional conditions are

attached to guide the generator to generate images that meet the conditions, here the additional conditions are the expression parameters. The network in the image generation module adopts the structure of a codec decoder, and Figure 2 shows the specific network architecture.

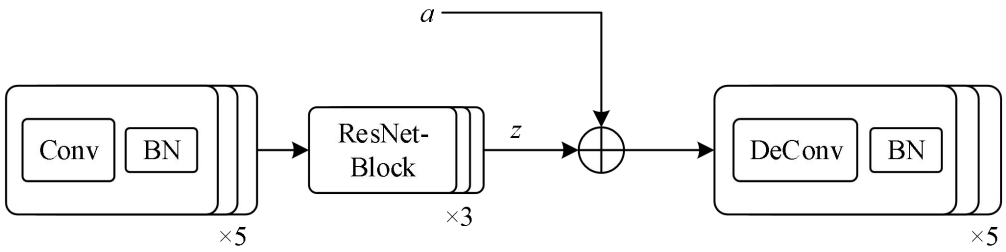


Figure 2. Network structure of image generation module.

The encoder is a 5.0-layer convolutional layer and each CNN layer is followed by Batch Normalization (BN) and then Leaky-ReLU is added as an activation function. The encoder part is mainly used to extract the features of the image. The face image I is input into the encoder G_{enc} to get the potential vector z, and this process is expressed by the available equation as

$$z = G_{enc}(I) \tag{8}$$

Cascading between the encoder and decoder through 3.0 residual network blocks allows interaction between shallow and deep features on the one hand, and on the other hand reduces the problem of gradient vanishing due to the deep network by transferring the original features through a constant mapping. Figure 3 shows the residual network.

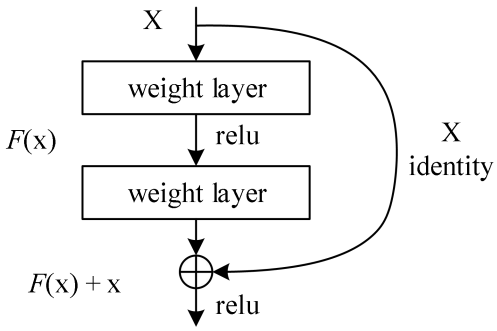


Figure 3. Residual network.

After that, the latent vector z is spliced with the input expression vector a and fed into the decoder. The decoder part is a 5.0-layer inverse convolutional layer, which is similarly back-joined to BN and Leaky-ReLU except for the last layer which is a Tanh activation function.The decoder decodes to get the expression-edited image, but this image is only a generative state in the middle of the model and is not the final generative image, which is notated as I_a^m, and is represented by the formula

$$I_a^m = G_{dec}(z,a) = G_{dec}(G_{enc}(I),a) \tag{9}$$

2.3. Behavioral Probabilistic Automata

Representing and modeling the uncertain behavior of movie-level animated characters can then be reduced to the problem of representing uncertain decision-making, planning, and action. Uncertainty action effects can be described by probability distributions, realized in action functions. Decision making is the process of choosing a goal, and planning is the process of choosing a way to realize the goal. Therefore, the representation can be harmonized as an uncertainty selection problem. For this purpose, probabilities can be introduced in FSMs to form probabilistic finite automata (PFA). The basic PFA is not fully suitable for describing uncertainty decision making and planning and needs to be subjected to certain constraints. This constrained PFA, suitable for describing the behavior of uncertainty, is called a

behavioral probabilistic finite automaton (BPFA).

2.3.1. Behavioral States

The states of the BPFA are the behaviors of the film-level animated characters, and the behaviors are divided into composite behaviors and basic behaviors; basic behaviors are also known as actions, and composite behaviors are combinations of actions or other composite behaviors. The uncertainty of action effects including imprecision, ambiguity and success or failure is not only related to the perceptron and effector, but also related to the intrinsic state of movie-level animated characters. Uncertainty decision-making and planning are reflected in composite behaviors depending on the intrinsic state of the virtual movie-level animated character.

The uncertainty action is denoted as $action_p = \langle n,\ R,\ p \rangle$. Where n is the name of the action, R is the space of possible outcomes of the action, and $p:\ R \to (0,1]$ is the outcome probability assignment function or affiliation function. When it is a probability assignment function, it should satisfy $\sum_{r \in R} p(r) = 1$. When $action$ is a deterministic action, the action can be replaced by n. The set of actions constitutes the set of states Q_A of the BPFA. In the graphical representation of the BPFA, behaviors correspond to circles.

2.3.2. Behavioral Transfer Events

Events are triggers for the transfer of behavior of virtual cinematic animated characters, and events include internal and external events. Internal events are issued by the movie-level animated characters themselves, such as the end-of-behavior event. External events are emitted by the environment (including other movie-level animated characters), such as enemy encounters.

An event can be defined as a six-tuple $event = \langle id,\ s,\ r,\ c,\ t,\ d \rangle$. Where id is the event identifier, s and r are the sender and receiver respectively, c is the event content, and t and d are the timestamp and expiration date respectively. The set consisting of all events forms the alphabet Σ in BPFA. In BPFA, events correspond to letters on the arc.

2.3.3. Transfer of Behavior

A transfer indicates the termination of the previous behavior and the beginning of the next behavior, and a behavioral transfer is usually triggered by an event. Behavioral transfers are defined below. In behavioral probabilistic automata behavioral transfers are represented as arcs with events and probabilities.

2.3.4. Representation of the BPFA

The basic PFA is $PFA = \langle Q_A, \Sigma, \delta_A, I_A, F_A, P_A \rangle$, and Figure 4 shows the basic probabilistic finite automaton structure. Where Q_A is a finite set of states; Σ is an alphabet; $\delta_A \subseteq Q_A \times \Sigma \times Q_A$ is a set of state transfer functions; $I_A : Q_A \to \mathbb{R}^+$ is the initial state probability; $P_A : \delta_A \to \mathbb{R}^+$ is the transfer probability; $F_A : Q_A \to \mathbb{R}^+$ is the end-state probability; I_A, P_A, and F_A satisfy: $\sum_{q \in Q_A} I_A(q) = 1.0$, $\forall q \in Q_A$, $F_A(q) + \sum_{a \in \Sigma, q' \in Q_A} P_A(q,\ a,\ q') = 1.0$.

Virtual movie-quality animated characters usually have a deterministic set of initial and ending behaviors, so I_A and F_A are not needed. In basic probabilistic finite automata, the probability of occurrence of letters needs to be considered, whereas in behavioral transfers there is no need to consider event probabilities, and the difference between the two can be seen from the comparison of Figs. 4 and 5, where the sum of the probabilities of all outgoing chains (including hold-alive and all input letters) is 1.0, while in Fig. 5 the sum of the probabilities of outgoing chains corresponding to each behavioral transfer event is 1.0.

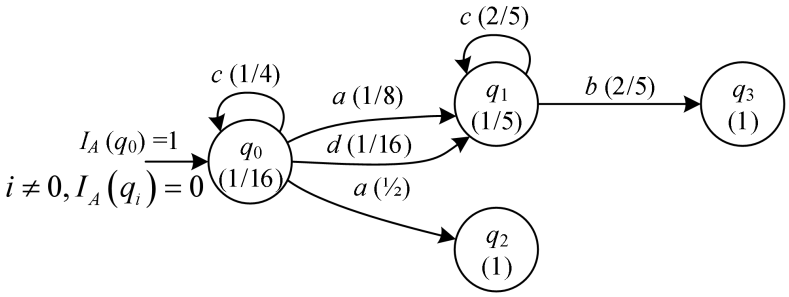


Figure 4. basic probabilis tic fin ite au tom a tic m ach ine.

To accommodate the uncertain behavioral representation, Figure 5 shows a PFA-based definition of a behavioral probabilistic finite automaton.

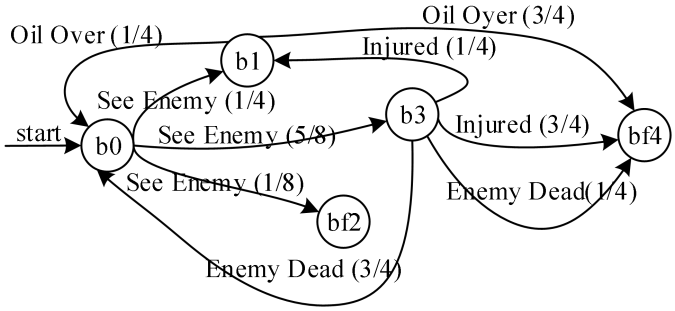


Figure 5. behavior probabilistic finite autom atic m achine.

$$BFPA = \langle n,\ Q_b, \Sigma, \delta_b, p_b, b_0, F_b \rangle \tag{10}$$

where: n is the name of the behavior; Q_n is the set of behaviors, including the set of basic behaviors A and the set of composite behaviors B . The composite behavior $b(b \in B)$ is defined by actions and other composite behaviors according to the rules of behavioral probabilistic finite automata. Thus the BFPA is itself a composite behavior. Σ is the set of behavioral transfer events, $\delta_b \subseteq Q_b \times \Sigma \times Q_b$ is a set of behavioral transfer functions; $p_b : \delta_A \to \mathbb{R}^+$ is the behavioral transfer probability satisfying $\forall b \in Q_b,\ \forall e \in \Sigma,\ \sum_{q \in Q_b} P_b(q,\ e,\ q') = 1.0$.

3. Practice and Effect Analysis of Animation Character Generation Based on Depth Generation Technology

3.1. Comparison of Modeling Effects

3.1.1. Model Frame Rate vs. Texture Number

In order to verify the effect of the animated character model constructed based on the depth generation technique in this paper, the method of this paper is compared with the 3D character modeling method based on semantic tree and the 3D character modeling method based on 3DS MAX, and the frame rate and texture number of each animated character model in the experiments created by the different methods are compared, and four models are randomly selected from the animated character models constructed by each of the three methods for comparison. Table 1 shows the results of comparing the frame rate and texture number of models from different methods. The frame rate of the four animated character models in this paper ranges from 95.23f/s to 98.74f/s, and all of them are over 90f/s. The texture number ranges from [60.15,67.21] MB, and all of them are larger than 60 MB. Compared with the other two 3D modeling methods, the animated character models created in this paper have higher frame rate and texture number, which indicates that the models created in this paper are smoother and more clear

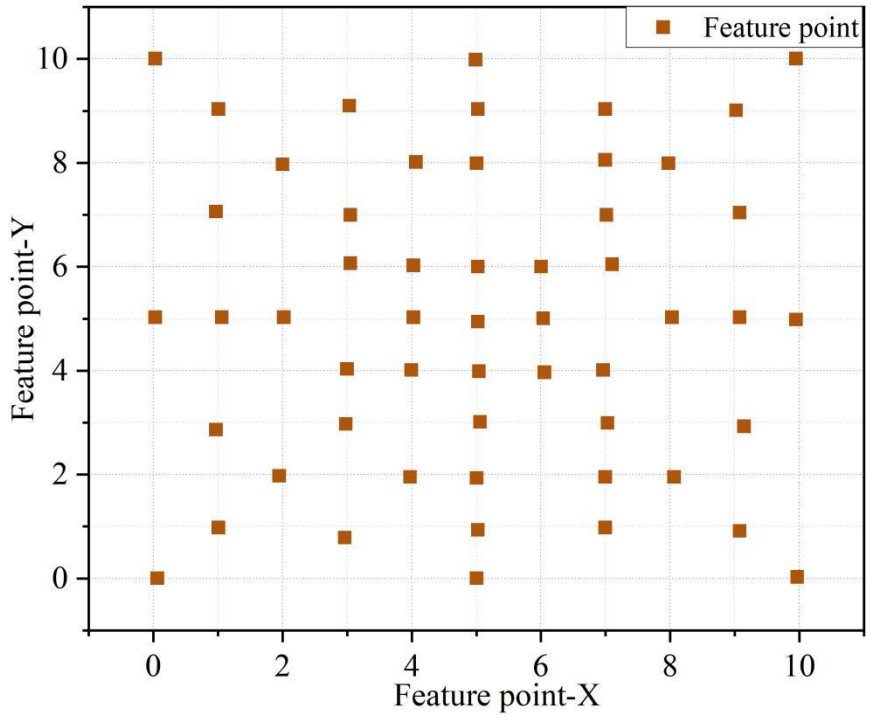
and lifelike during the operation. This indicates that the model created by this method runs more smoothly and the model is more clear and realistic.

Table 1. Comparison of model frame rate and texture count.

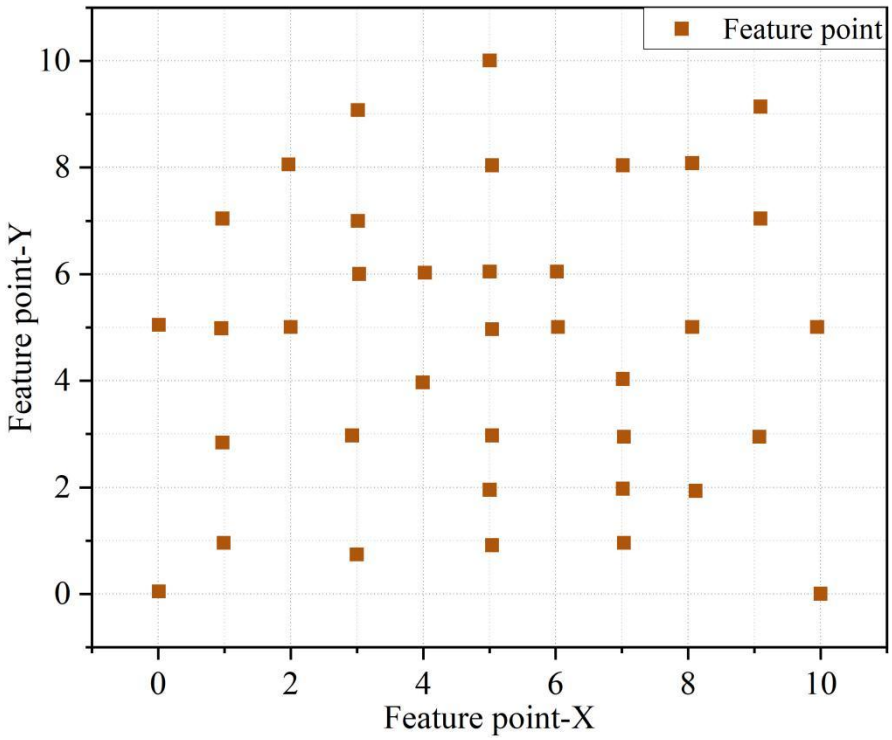
Method	Model serial number	Frame rate (f/s)	Number of textures (MB)
Article method	1	98.17	67.21
	2	96.43	66.47
	3	98.74	60.15
	4	95.23	63.82
Semantic tree	1	71.34	35.97
	2	68.83	34.13
	3	69.02	40.04
	4	73.62	41.78
3DS MAX	1	83.17	54.12
	2	82.46	56.65
	3	84.25	55.03
	4	88.54	57.86

3.1.2. Comparison of the Accuracy of 3D Construction of Animated Character Images

In the construction of animated characters, it is important to have a strong visual impact. In order to be able to construct excellent animated characters, it is necessary to improve the speed of extracting expression features and behavioral features, etc. of animated characters. So the speed and accuracy of 3D construction need to be improved. In this section, after completing the comparison of the extraction speed, a special detection method is used to record the accuracy of the 3D construction of the animated character image by applying the traditional method (semantic tree 3D modeling method) and the method of this paper. Figure 6 shows the comparison of the accuracy of 3D construction of the animated character image. The symmetry of each point in Fig. 6, up and down, left and right, represents the accuracy of 3D construction, and the stronger the symmetry, the higher the accuracy. From Fig. 6, it can be seen that: applying the traditional method to complete the 3D construction of the animated character image, only part of the points can satisfy the symmetry of the upper and lower, left and right directions; applying the method of this paper to complete the 3D construction of the animated character image, all the points can satisfy the symmetry of the upper and lower, left and right directions. It can be proved that compared with the traditional method, the method in this paper can really improve the accuracy of 3D construction of animated character image.



(a) Article method



(b) Traditional method

Figure 6. Accuracy of three-dimensional construction of animated character images.

3.2. Comparative Analysis of Animated Character Expression Generation Effects

3.2.1. Comparison of the Convergence of the Editing Process of Different Methods

In order to verify the editing effect of this paper's method on the expression of 3D animated characters, the semantic tree-based 3D character modeling method and the 3DS MAX-based 3D character modeling

method are used as the experimental control group to compare the convergence of the three methods for the automatic editing of expression under different rotation angles. Figure 7 shows the results of the convergence comparison of the expression editing process of different methods. As can be seen from Fig. 7, the method in this paper reaches a stable convergence value in 59 iterations, while the semantic tree-based 3D character modeling method needs 78 iterations and the 3DS MAX-based 3D character modeling method needs 145 iterations. The method in this paper has better convergence for the 3D animated character expression editing process, which can be converged quickly and has better convergence performance than the two compared methods. The reason for this effect is that the method in this paper extracts a large number of accurate facial expression features using cascade classifiers and constraint functions before expression editing, which enables it to complete the convergence more quickly.

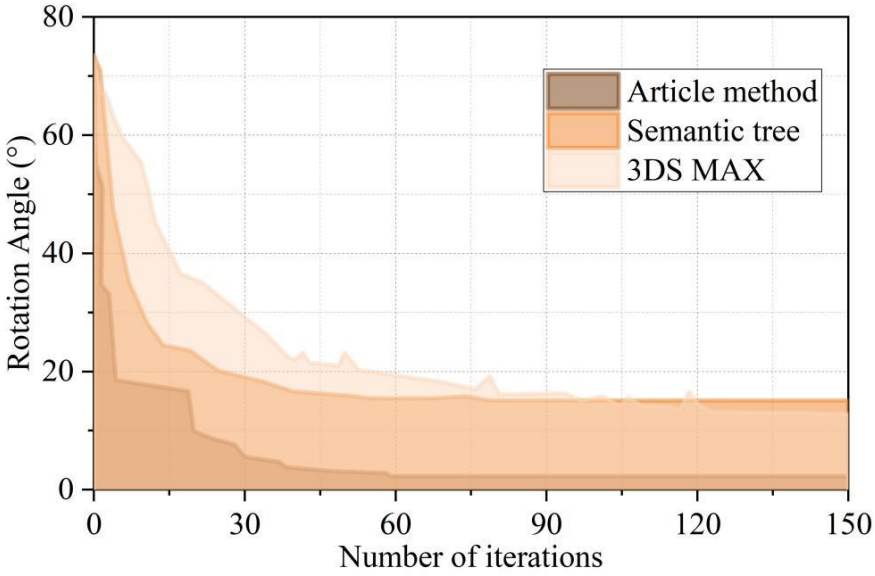


Figure 7. Comparison of convergence of editing process by different methods.

3.2.2. Comparison of Normalization Errors for Facial Feature Points

The feature extraction part of this experiment is mainly to make a comparison of the extraction accuracy. In order to fully illustrate the performance of the feature extraction method in this paper, this experiment uses the normalized mean error (NME) as an evaluation criterion to analyze and compare the accuracy. NME is used to evaluate the change in accuracy of the depth generation model in extracting facial features of faces from multiple angles.

In this experiment, three angles of faces are selected for 3D feature extraction, including frontal face, medium angle deflection face, and large angle deflection face, and Fig. 8 shows the comparison results of normalized mean error of facial feature points. The x-coordinate represents the normalized error value, and the y-coordinate represents the percentage of feature points for which the calculated normalized error value is the current value. In this experiment, three colors of lines are used to represent various types of faces, light orange lines refer to frontal faces, positive orange lines refer to medium-angle deflected faces, and dark orange lines refer to large-angle deflected faces. The accuracy of frontal faces is overall higher than that of all faces with the presence of a deflection angle, and the lines are steeper upfront, with 75.5% of the feature points having an NME value of 0.4 or less, and 90.35% of the feature points having an NME value between 0 and 0.68. The accuracy of medium-angle deflection faces is comparable to that of large-angle deflection faces, with 52.2% of medium-angle deflection faces having NME values between 0-0.4. It can be seen that the feature extraction of frontal faces is the best, and the feature extraction of medium-angle and large-angle deflected faces is relatively low.

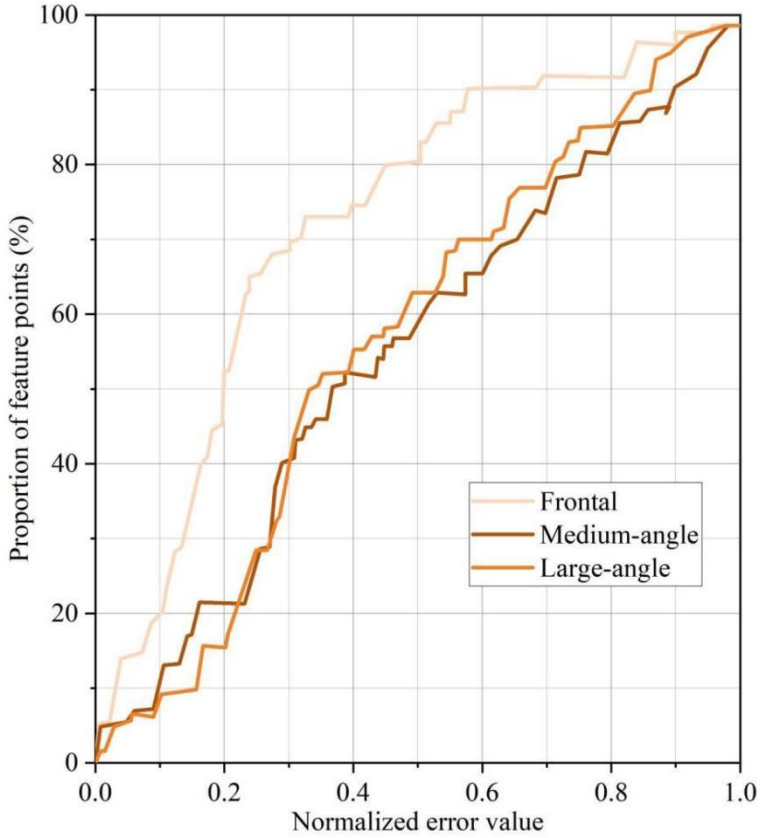


Figure 8. Comparison of normalized mean errors of facial feature points.

3.2.3. Facial Expression Feature Point Accuracy Analysis

Specific 0-degree face, 40-degree off-angle face and 80-degree off-angle face are selected for further accuracy test comparison. Fig. 9 shows the NME values of facial expression feature point accuracy for specific angles. Where the x-coordinate represents the normalized mean error value, and the y-coordinate represents the proportion of features whose normalized mean error value is less than the current value, in addition, this experiment uses four colors of lines to represent the four angles of the face, where the light orange line refers to the frontal 0-degree face, the positive orange line refers to the 40-degree moderately deflected face, and the dark orange line refers to the 80-degree large-angle deflected face. The overall prediction accuracy of the 0-degree frontal face remains at the highest level, the lines are steep and then flat with slopes ranging from high to low, 58.7% of the feature points have NME values below 0.4, and nearly 76.7% have NME values below 0.8. The 40-degree deflected face can be approximated as an exponential function before the NME value of 0.4, and after 0.4, the horizontal and vertical coordinates are linearly correlated, as shown in the data. 61.9% of the feature points have NME values below 0.4, and 78.4% of the points have NME values below 0.8. The feature point accuracy of the 80-degree large-angle deflected face is slightly higher than that of the 40-degree face before the NME value of 0.4, and is surpassed by that of the 40-degree face after 0.4, and the two are at a similar level from 0.4 onwards. It can be seen that the feature points extracted by the depth generation model have the highest accuracy for frontal faces, and the error value is basically kept at a low level, and the smaller the angle in the deflected face, the better the effect is, but the difference is not big, and basically can accurately locate the side face.

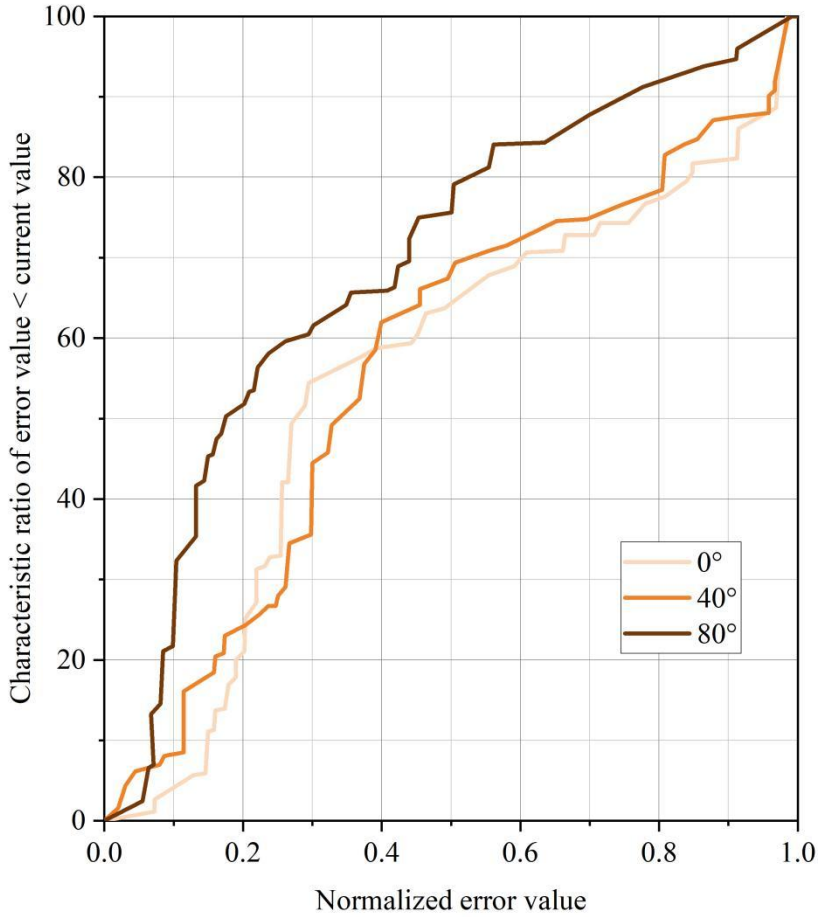


Figure 9. Accuracy NME value of feature points at a specific Angle.

3.3. Behavioral Simulation and Case Studies

3.3.1. Starting and Stopping Angles of Joint Rotation Based on Expression Rotation Angle

Table 2 shows the starting and ending angles of animated character behavior joint rotations based on expression rotation angles. Within the range of expression rotation angles from 0 degree frontal face to 80 degree large angle deflection face, the joint rotation range of the generated animated character behaviors is also between 0-π. Analyzing the specific joint rotation angles with 10 degrees as the expression rotation unit, it is found that the joint rotation angles with different expression rotation angles have diversity in different generation cases. It shows that the depth-based generation technique can have detailed and rich non-specific behaviors to meet the diverse needs of simultaneous generation of animated character appearance and behavior.

Table 2. Start and end angles of joint rotation of the animated character's behavior.

Corresponding Angle (°)	Starting and ending angles (°)		
	Case 1	Case 2	Case 3
0	0~π/18	0~π/18	0~π/18
10	π/9~-π/9	π/10~-π/10	π/6~-π/6
20	π/9~π/6	π/7~π/8	π/5~π/4
30	π/6~-2π/9	π/5~-1π/6	π/4~-1π/3
40	2π/9~5π/18	6π/9~4π/15	5π/7~5π/6
50	5π/18~π/3	4π/13~π/4	3π/4~2π/3

60	π/3~7π/18	π/4~6π/15	π/2~7π/14
70	7π/18~4π/9	5π/13~5π/12	6π/11~5π/9
80	4π/9~0	4π/9~0	4π/9~0

3.3.2. Analysis of the Effect of Synchronization of Expression and Behavior of Animated Characters

Table 3 shows the comparison of the expression and behavior synchronization effects of the six animated characters generated by the three methods. The synchronization effects of expressions and behaviors of the 6 animated characters generated by the methods in this paper are all over 90%, which are 90.78%, 90.26%, 91.53%, 92.15%, 92.67%, and 91.09%, respectively. While the synchronization effect of the semantic tree-based 3D character modeling methods are not more than 85%, and the synchronization effect of the 3DS MAX-based 3D character modeling methods are all lower than 80%. From the comparison results, it can be analyzed that the animated character expressions and behaviors generated by this paper's method are better synchronized and have higher consistency.

Table 3. The synchronization effect of expressions and behaviors.

Animated character	Method synchronization effect (%)		
	Article method	Semantic tree	3DS MAX
1	90.78	80.15	78.29
2	90.26	81.32	79.61
3	91.53	82.13	76.95
4	92.15	80.47	74.68
5	92.67	82.31	79.32
6	91.09	81.45	79.61

4. Conclusion

In this paper, based on the improved deep learning technique, we extract and generate animated characters with synchronized expressions and behaviors, and set up comparative experiments to verify the generation effect. In the overall modeling effect, the modeling frame rate of this paper's method reaches up to 98.74f/s, the texture number is up to 67.21MB, and the accuracy of feature point extraction in 4 directions is high. In the expression editing effect test, the method achieves stable convergence with 59 iterations, which is better than the 78 and 145 iterations of the comparison methods. Meanwhile, the accuracy of facial feature point extraction for different angles is high. The synchronization effect of expression and behavior of the generated animated characters reaches 90.78%, 90.26%, 91.53%, 92.15%, 92.67%, 91.09%, which is a good synchronization. In the future, the synchronization of appearance and behavior generation of multiple animated characters can be studied to explore the possibility of reducing the generation cost and time.

References

1. Mu, R., & Zeng, X. (2019). A review of deep learning research. KSII Transactions on Internet and Information Systems (TIIS), 13(4), 1738-1764.
2. Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. World Information Technology and Engineering Journal, 10(07), 3897-3904.
3. Grewal, P. S., Oloumi, F., Rubin, U., & Tennant, M. T. (2018). Deep learning in ophthalmology: a review. Canadian Journal of Ophthalmology, 53(4), 309-313.
4. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM computing surveys (CSUR), 51(5), 1-36.
5. Yoshino, Y., Nakada, K., Kobayashi, M., & Tatsumi, H. (2019, July). A study on machine learning-based image identification towards assitive automation of commentary on animation characters. In 2019 International Conference on Machine Learning and Cybernetics (ICMLC) (pp. 1-4). IEEE.
6. Lin, K. W., & Wang, Y. J. (2012). The influence of animated spokes-characters in customer orientation. The International Journal of Organizational Innovation, 4(4), 142-154.

7. Shuja, K., Ali, M., Anjum, M. M., & Rahim, A. (2018). Effectiveness of animated spokes-character in advertising targeted to kids. Journal of Marketing Management and Consumer Behavior, 2(2), 31-47.
8. Jin, Y., Zhang, J., Li, M., Tian, Y., Zhu, H., & Fang, Z. (2017). Towards the automatic anime characters creation with generative adversarial networks. arXiv preprint arXiv:1708.05509.
9. Lee, J., Kim, H., Shim, J., & Hwang, E. (2022, October). Cartoon-flow: a flow-based generative adversarial network for arbitrary-style photo cartoonization. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 1241-1251).
10. Khalid, S. B., & Hazela, B. (2021). A Framework for Estimation of Generative Models Through an Adversarial Process for Production of Animated Gaming Characters. In Deep Learning in Gaming and Animations (pp. 123-136). CRC Press.
11. Tan, J. L. (2024). Performance comparison between generative adversarial networks (GAN) variants in generating comic character images (Doctoral dissertation, UTAR).
12. Yi, Z., Wu, G., Pan, X., & Tao, J. (2021, May). The Research of Anime Character Portrait Generation Based on Optimized Generative Adversarial Networks. In 2021 33rd Chinese Control and Decision Conference (CCDC) (pp. 7361-7366). IEEE.
13. Lungu-Stan, V. C., & Mocanu, I. G. (2024). 3D Character Animation and Asset Generation Using Deep Learning. Applied Sciences, 14(16), 7234.
14. Zheng, F., & Zhu, Y. (2024). Exploring the Fusion of Animation and Computer Vision for Enhanced Realism in Virtual Character Interaction. IEEE Access.
15. Nishimura, Y., Nakamura, Y., & Ishiguro, H. (2020). Human interaction behavior modeling using generative adversarial networks. Neural Networks, 132, 521-531.
16. Yang, H., & Hong, S. W. (2025). Creating an Anthropomorphic Folktale Animal: A Pilot Study on Character Design Creativity Derived From Autonomous Behavior Generation Powered by Reinforcement Learning. Computer Animation and Virtual Worlds, 36(1), e70013.
17. Wang, Y., Che, W., & Xu, B. (2017). Encoder–decoder recurrent network model for interactive character animation generation. The Visual Computer, 33, 971-980.