2024 2nd International Conference on Mechatronics, IoT and Industrial Informatics (ICMIII) | 979-8-3503-8663-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMIII62623.2024.00052

Research on AI-Driven Virtual Character Modeling Technology through UV Seagull Algorithm

Junyan Deng
Chengdu Neusoft University, Chengdu, China
DengJunxi@nsu.edu.cn

***Abstract*—In the era of AI-driven virtual character modeling technology, UV Seagull Algorithm (UVSA), as an innovative integrated technology solution, creatively integrates UV mapping and reverse UV mapping with Seagull optimization algorithm to effectively solve the problem of fidelity and efficiency of virtual character construction in the meta-universe environment. In the core of 3D modeling, although the UV mapping process is indispensable, it is often accompanied by bad effects such as texture distortion and distortion, which limits the real visual experience of virtual characters. For this reason, UVSA has a unique approach, using the seagull optimization algorithm derived from the natural seagull cluster foraging behavior principle, which has a strong global search ability and flexible processing of complex problems. In the practical application of UVSA, the 3D model is initially expanded to the 2D plane through the standard UV mapping to form the UV coordinate layout, and then the seagull optimization algorithm is used to deeply optimize it, targeted to eliminate texture distortion and optimize texture distribution, in order to pursue the most ideal UV layout scheme. After optimization, UVSA uses reverse UV mapping technology to accurately restore the optimized texture map to the surface of the 3D model, making the virtual character present unprecedented highly realistic texture details and clear and delicate appearance texture. Thanks to this, UVSA not only greatly improves the visual authenticity and viewing quality of virtual characters in the meta-universe environment, but also greatly improves the overall efficiency of modeling and rendering, and opens up a new technical path for many industries, including virtual reality, game development, film and television production, and virtual social communication. It can be said that the emergence of UVSA is not only a major breakthrough in existing modeling technology, but also provides a strong and practical technical support and innovative ideas for the development of virtual character creation in the future meta-universe, creating a solid technical cornerstone for the construction of high-precision virtual characters in the meta-universe era, and effectively promotes the continuous innovation and development of related industries. It also opens the door to more possibilities.**

Keywords*—*AI drive, seagull optimization algorithm, virtual characters

I. Introduction

In the context of the rapid development of artificial intelligence and virtual reality technology, especially in the construction of lifelike virtual characters in the meta-universe environment, technological innovation is particularly important. Among them, UV Seagull Algorithm (UVSA), as a novel and advanced comprehensive model, successfully combines UV mapping, reverse UV mapping technology and Seagull optimization algorithm, and opens up a new path for realizing highly realistic and efficient modeling of virtual characters.

UV mapping is a key technique in 3D computer graphics, which projects the surface of a complex 3D model onto a 2D plane in order to draw and edit the texture information of the model on that plane [1]. However, due to the complexity and nonlinear nature of 3D surface, texture stretching, compression and overlap often occur in the mapping process, which will seriously affect the authenticity and beauty of the appearance of virtual characters. To overcome this common technical bottleneck, UVSA models are designed with the UV layout optimized to minimize such distortions in mind.

On this basis, UVSA introduced the Seagull Optimization Algorithm. Seagull optimization algorithm is a heuristic optimization method developed to simulate the foraging behavior of seagull groups in nature. It has attracted wide attention in the engineering field because of its excellent global search ability, efficient convergence speed and ability to deal with complex optimization problems [2]. In the framework of UVSA, the gull optimization algorithm is creatively applied to the optimization stage of UV mapping. By simulating the flight path of gull, the optimal solution is found, and the UV coordinate layout of the virtual character model is intelligently adjusted, so as to achieve the optimal texture distribution state, so as to avoid texture deformation and ensure the accurate reproduction of texture details.

First, UVSA uses traditional UV mapping technology to expand the surface of the 3D virtual character model onto a two-dimensional plane to form an initial UV coordinate system. Then, the gull optimization algorithm began to work, simulating several "gull" to search for optimization in the UV space, and constantly iteratively updating the distribution of UV coordinates until an optimal UV layout scheme was found that both met the seamless texture stitching and minimized distortion.

After UV Mapping optimization is complete, the UVSA then performs the Inverse UV Mapping process. Reverse UV mapping is to accurately map the texture information on the optimized map back to the surface of the 3D model, ensuring that the texture details perfectly match the model geometry, and finally make the virtual character present the extremely realistic and detailed visual effect.

The successful application of the UVSA model plays a key role in enhancing the realistic and immersive experience of AI-driven virtual characters in the meta-universe environment. On the one hand, it significantly improves the rendering quality of virtual characters' skin, clothing and other textures, enabling users to get a visual enjoyment closer to the real world [3]. On the other hand, by optimizing the modeling process, UVSA also greatly improves work efficiency and reduces repetitive labor, thus shortening the design and iteration cycle of virtual characters [4].

979-8-3503-8663-9/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMIII62623.2024.00052

In addition, the UVSA model has a broad application prospect, which not only serves the creation of virtual characters in the meta-universe scenario, but also shows great potential in virtual reality (VR) content development, video game industry, movie special effects production, virtual social platforms and other fields. This innovative idea of deep integration of natural inspiration and engineering technology has undoubtedly set a new benchmark for the development of future virtual role modeling technology, laid a solid foundation for the construction of a more detailed, vivid and intelligent virtual world, and also brought breakthrough changes and development momentum for related industries.

II. RELATED FIELD

The research of 3D face modeling began in the early 1970s and has a history of more than 50 years. In this field, relevant researches at home and abroad have made many remarkable achievements, and many face modeling methods have appeared. As early as 1999, Blanz et al. [5] took the lead in adopting CyberWare TM laser scanner and established the first 3D face database MPI. In 2009, Paysan et al. [6] adopted ABW-3D structured light scanner to establish a more accurate 3D Face database BFM(Basel Face Model) in less time, and added facial expressions in 2017 and 2019. BFM2017 and BFM2019 databases were released. Domestic research in this regard is relatively late. In 2009, Yin Baocai et al. [7] established the first Chinese 3D face database BJUT-3D by using CyberWare 3030RGB/PS laser scanner. In 2014, Zhou Kun et al. from Zhejiang University used Kinect depth camera to establish FaceWareHouse, a 3D face database with expressions. The model reconstructed by this method has high accuracy and is often used in the construction of 3D human face database. However, it has been difficult to carry out commercial application because of the expensive equipment, strict requirements for application occasions and the need for close cooperation of professionals.

In recent years, 3D face models have been applied in face recognition, animation film and television production, medical plastic surgery and so on. In terms of face recognition, three-dimensional faces have richer feature information and higher recognition accuracy. In 2017, Apple Inc. first launched iPhone X based on 3D structured light technology and Face ID 3D face recognition, opening the era of "unlocking mobile phones with 3D face recognition". In the aspect of animation film and television, 3D model can bring people more real visual experience and achieve immersive effect.

Although great progress has been made in 3D face modeling, and the application of face models is becoming more and more extensive, high-precision and realistic 3D face modeling is still difficult [8]. The main reason is that the human face is complex and expressive, and the high-precision modeling of details such as hair, beard and wrinkles is an unsolved problem. There are two main types of implementation methods for 3D face modeling. The method of 3D face modeling through active physical scanning equipment has high accuracy, but it is difficult to expand the application because of high equipment cost and harsh application conditions. The passive method of face modeling through captured images is easy to popularize due to its low cost and low application requirements, and is the mainstream method of 3D face modeling in recent years [9].

III. UV MAPPING AND REVERSE UV MAPPING

Reverse UV mapping and UV mapping technology are applied in texture domain to record and change color texture information of 3D objects. Inspired, Gu et al. [10] applied these two technologies to the location coordinate domain for the first time in 2002 to realize the recording and alteration of the geometric shape information of three-dimensional objects.

No matter the two techniques are applied in texture domain or position coordinate domain, as shown in Figure 1, it is necessary to establish the mapping relationship between three-dimensional object space (x, y, z), parametric two-dimensional space (s,t) and normalized two-dimensional space (u, v) first. Specifically, it is realized through the calculation of parameterization and normalization of three-dimensional objects, that is, the three-dimensional objects are parameterized to two-dimensional space by the following formula, and then normalized by formula (1), and then the mapping relationship is established.

$$(s,t)=Para(x,y,z) \tag{1}$$

$$(u,v)=Norm(s,t) \tag{2}$$

After building such a reverse UV Map, the attribute transfer process from the vertex of a three-dimensional object to a standardized two-dimensional plane can be achieved, resulting in an image that reflects the two-dimensional properties of the object, often called a UV map. If the spatial coordinates of the vertices of a three-dimensional object (x, y, z) are recorded in a UV graph, a so-called UV Position Map will be formed. This concept is given according to formula (3), wherein the third dimension of the UV graph carries the position coordinate data containing three channels. Similarly, when the vertices' color Texture information is stored (r, g, b), a UV Texture Map is generated. The definition follows formula (4), meaning that the third dimension of the UV map now contains the color texture information consisting of three channels.

$$Pos(u,v)=(x,y,z) \tag{3}$$

$$Tex(u,v)=(r,g,b) \tag{4}$$

For simple three-dimensional objects, such mappings can usually be established using linear functions, cylindrical functions, paraboloid functions, or hemispherical functions. [50] As shown in Figure 1, a cylinder with a base surface radius and height of 1 is taken as an example and a cylindrical function is used to map it, then the parameterization and normalization of the cylinder are calculated as shown in formula (5) and formula (6).

$$(s,t)=Para(x,y,z)=(\arccos x,z) \tag{5}$$

$$(u,v)=Norm(s,t)=\left(\frac{s}{2\pi},t\right) \tag{6}$$

Among them, (x y, z) $\in R^3$said cylinder of at an arbitrary point on the 3 d coordinates, and $x^2 + y^2 = 1$, $z \in [0, 1]$, this is determined by the bottom, radius of the cylinder and high; $(s,t) \in R^2$ represents the two-dimensional parameterized coordinates of the vertex, $s \in [0, 2\pi]$ represents the Angle of rotation of the vertex about the central z-axis,$t \in [0, 1]$ represents the height of the vertex; $(u, v) \in R^2$ represents the two-dimensional normalized coordinates of this vertex.

By applying formula (5) to the coordinates of any vertex on the surface of the cylinder (x, y, z), its corresponding values in two-dimensional parametric space (s,t) can be obtained. Then these parametric coordinates are converted into two-dimensional normalized coordinates (u, v) by means of formula (6). Thus, a transformation mechanism from the surface of the cylinder to the standardized UV coordinates is established. Using this mechanism, the color texture data or position coordinate information attached to the vertex of each surface of the cylinder can be encoded in UV space, and then the corresponding UV map representation can be generated. However, it is worth noting that for face models with complex structural features, such mapping operations cannot be effectively completed by relying only on simple functions.

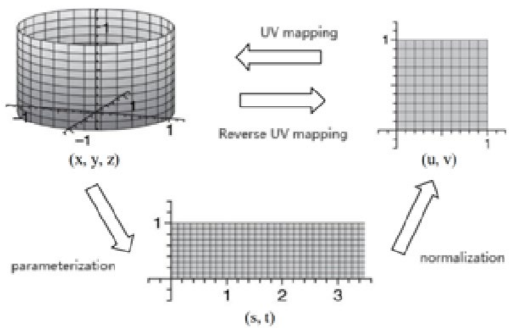

Fig. 1. Mapping and reverse UV mapping

IV. Gull optimization algorithm

SOA simulates the migration behavior of seagulls in nature and the attack behavior in the migration process to conduct mathematical modeling, so as to achieve the optimization of the objective function, as shown in Figure 2.

A. Migration model

Migration behavior refers to the behavior of seagulls moving from a position of low fitness to a position of high fitness in nature, and migration behavior will guide seagulls to find more abundant food sources. During the migration process, each individual seagull is in a different position to avoid collision during migration; At the same time, each individual gull will move to the position of the best individual in the whole gull population. The migration behavior affects the SOA's global search capability.

During their migratory behavior, seagulls need to meet three conditions:

(1) Prevent collisions between seagulls

In order to avoid the collision of seagull individuals when moving, SOA introduces A variable A to calculate the individual in search for a new location in space.

$$C_s = A \times P_s(x) \tag{7}$$

Where C_s does not collide with other search agents; Ps search the current location of the agent; x number of current iterations;

(2) Move in the direction of optimal survival

The search agent will follow the rules of collision avoidance for individual seagulls, flying towards the most fit individual in the vicinity (i.e. the most fit individual in the flock).

$$M_s = B \times \left(P_{bs}(x) - P_s(x)\right) \tag{8}$$

Where, M_s represents the position of individual seagull; B is a random value factor responsible for the proper balance between development and utilization; Optimal location of individual seagulls in P_{bs} (x) populations.

(3) Update the initial position

Finally, the gull population updates the position of the individual and the best agent, as follows:

$$D_s = \left|C_s + M_s\right| \tag{9}$$

B. Attack model

Seagull populations need to prey during migration to maintain resources for survival. During migration, gulls can constantly change their Angle and speed to attack their prey. The attack model aims to make the best use of the history and experience of biome search during migration to further carry out local search for optimization targets. When attacking prey, seagull populations form a spiral formation in the air, which can be described in three-dimensional space by the following formula:

$$x^{"} = r \times \cos(k) \tag{10}$$

$$y^{"} = r \times \sin(k) \tag{11}$$

$$r = u \times e^{kv} \tag{12}$$

Where, r is the radius of each spiral flight; k is a random number with the value range of $[0, 2\pi]$; u and v define the constant of the spiral curve; e The base of the natural logarithm.

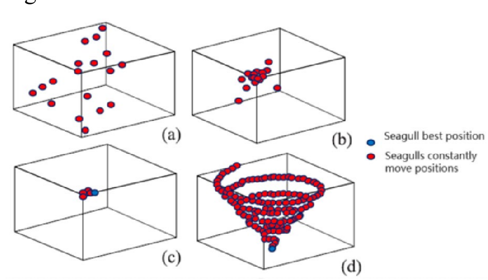

Fig. 2. Seagull optimization algorithm summary diagram

V. UV Seagull Algorithm Model (UVSA)

Suppose there exists a model that combines UV Mapping and reverse UV mapping with Seagull Optimization Algorithm, named UV Seagull Algorithm (UVSA), and its working principle is described in Figure 3. The specific work flow is as follows:

UV mapping and reverse UV mapping:

In 3D computer graphics, UV mapping refers to the process of mapping each vertex coordinate of a 3D model surface to a 2D texture coordinate system in order to facilitate the application of texture mapping to the model surface. The inverse UV mapping is derived from the 2D texture coordinates to the corresponding vertex coordinates on the 3D model surface.

The UVSA model envisages how it works:

1) UV mapping optimization:

UVSA model uses gull optimization algorithm as an efficient global optimization tool to optimize the UV layout of 3D models. The goal of the model may be to maximize the stretch minimization of the texture, seam minimization, or other visual quality metrics.

Each "gull" in the gull optimization algorithm represents a possible UV mapping scheme, and the algorithm searches for the optimal solution in the UV mapping parameter space by simulating the flight behavior of the gull, such as foraging, congregating, and following the pilot.

2) Iterative optimization process:

In each generation of optimization iteration, the "Seagulls" will update their own UV mapping strategy according to the rules of the Seagulls optimization algorithm, such as transforming coordinates, adjusting stretch ratios, etc.

The quality of each UV mapping scheme is evaluated using a pre-defined fitness function, which can be designed according to the UV mapping criteria required for optimization, such as reducing the joint area, evenly distributing UV coordinates, etc.

3) Reverse UV mapping application:

In the optimization process, in addition to looking for a good UV map forward, the UVSA model may also include reverse UV mapping, such as checking the effect of texture mapping on the 3D model during optimization, or applying the optimized UV map to the model surface in reverse to observe and evaluate the optimization effect in real time.

4) Convergence and termination conditions:

When the Gull optimization algorithm finds an excellent enough UV mapping layout (that is, the fitness function value reaches the optimal or meets the preset threshold), or the significant improvement is not achieved after a certain amount of algebra, the optimization process ends and the best UV mapping result is output.

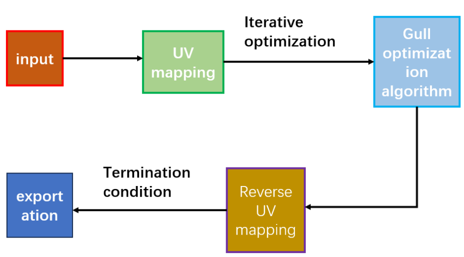

Fig. 3. UVSA job description diagram

VI. EXPERIMENT

A. Experimental environment

In the experimental environment, we used Ubuntu 20.04 operating system and built a virtual environment with Anaconda for the experiment. In this environment, the PyTorch deep learning framework version 1.7.3 is installed, and the CUDA version 11.0 is the matching version. All model training tasks are performed on an NVIDIA RTX3080 GPU.

The backbone network architecture used in the experiment is ViT-B_17, and its image input processing method is to divide the original image into 16×16 non-overlapping blocks. To improve the model performance, we load the ViT-B_16 weights pre-trained on the ImageNet dataset as initialization parameters. In addition, on the regularization strategy, we set L2 regularization parameter β to 0.003.

In the data preprocessing stage, we carried out the following operations on the input image: First, ensure that the short side of the image is adjusted to 500 pixels, and keep the aspect ratio unchanged; Then flip the image horizontally with a 50% probability; Finally, a subgraph of fixed size (224×224) is selected from the randomly cropped area as the final input. The generalization ability of the model is improved by means of data enhancement.

In terms of optimizer selection and training Settings, we used the SGD optimizer, with the initial learning rate set to 0.01 and the momentum parameter set to 0.9, while not using heavy attenuation. The whole training process consists of 30 training cycles, and the cosine annealing learning rate scheduling strategy with warm-up stage is adopted. In the first two training rounds, the learning rate slowly rises to the predetermined value at the initial value of 1e-7, and in the subsequent training process, the learning rate gradually decreases according to the law of cosine annealing until it reaches the minimum value of 1e-6.

B. Evaluation index

In order to deal with the problem that the model may not learn enough difficult samples due to the large number of easily classified samples in the field of target detection, an improved loss function is introduced, which is optimized based on the basic cross-entropy loss function. By introducing a dynamically adjusted weight coefficient, this function can make the model training process more focused on the attention and learning of those samples that are difficult to distinguish. In other words, in general, such a loss function can be expressed as:

$$L_{ce}\left(y,\hat{p}\right)\begin{cases}-\log\left(\hat{p}\right), y=1\\ -\log\left(1-\hat{p}\right), y=0\end{cases} \tag{13}$$

In a general framework, the loss function is often expressed as:

$$L_{fl}=-\left(1-p_t\right)^{\pi}\log\left(p_t\right) \tag{14}$$

This function aims to make the model pay special attention to the learning and recognition of difficult samples when dealing with object detection tasks by dynamically adjusting the weight strategy, so as to avoid the deterioration of model generalization ability caused by the dominant training of easily classified samples.

C. Experimental analysis

Loss function is very important in the training of neural networks, usually a good loss function has a larger design space, can have a more flexible form.

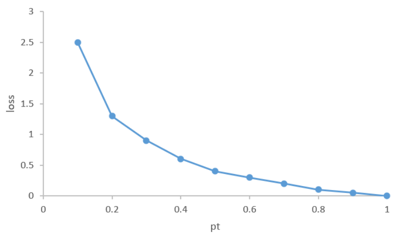

Fig. 4. UVSA loss function

In Figure 4, the relational data of the interaction between "learning rate" and "loss function value" within the UVSA model is revealed. In the machine learning and deep learning training process, the loss function plays a core role in quantifying the difference between the model's predicted output and the actual target, and the learning rate is a key factor determining the intensity of parameter adjustment in the optimization algorithm.

Each set of data corresponds to the loss function value at a certain point (or iteration cycle) in the training process under a specific learning rate setting:

When the learning rate is set to zero, the loss value reaches 2.5, which means that even if the parameters are not updated (because the learning rate is zero), the model still has a significant prediction error.

As the learning rate increases from 0.1 to 1.0, the loss function value shows a gradually decreasing trend: it successively goes through 2.5, 1.3, 0.9, 0.6, 0.4, 0.3, 0.2, 0.1 until 0.05, and finally reaches 0. This shows that as the model parameters are adjusted more greatly, the model performance gradually improves and successfully converges to the optimal state, at which point the loss function value drops to the lowest -- 0, suggesting that the model can match the training data nearly perfectly or has reached the preset termination condition under this configuration.

In the conventional neural network and other model training scenarios, it is very important to choose the learning rate reasonably. The low learning rate may cause the training process to be too slow and easy to fall into the trap of local optimal solution. However, too high learning rate may cause the loss function value to fluctuate greatly, or even miss the global optimal solution. In this example, as the learning rate increases from 0.1 to 1.0, the loss function value continues to decline and converges smoothly, which proves the effectiveness of the learning rate adjustment strategy adopted for the current problem.

TABLE I. COMPARISON OF DIFFERENT MODEL DATA

	MSE	MAE	RMSE
Color Space Mapping	0.512	0.412	0.444
Network Mapping	0.543	0.401	0.451
UVSA	0.471	0.353	0.383

Table I shows the modeling performance of three different machine learning models (Network Mapping, Color Space Mapping, and UV Seagull Algorithm model (UVSA)). Three common image quality evaluation indexes were compared: mean square error (MSE), mean absolute error (ESA) and root mean square error (RMSE).

MSE represents the average of the square of the error per pixel point and is a unitless, purely numerical error measure. The lower the MSE value, the higher the quality of the image recovery. On this measure, UVSA also achieved the lowest MSE value (0.471), showing the smallest average error, better than the two methods.

The MAE representation refers to the average absolute error, that is, the average of the absolute values of the difference between the values of each pixel. Similarly, the lower the value, the smaller the overall error at the pixel level. UVSA also had the lowest MAE value (0.353), indicating that it performed best in terms of absolute error.

RMSE is the square root of the MSE and is the same as the original pixel error scale, also used to measure the difference between the predicted value and the actual value. As with MSE, lower RMSE values represent higher image quality. UVSA also has the lowest RMSE in this table (0.383), once again demonstrating its excellent performance in the test scenario.

Through these objective quality evaluation indicators, it can be seen that among the several color space conversion or mapping methods given, UVSA method has the best overall performance, and its output images have obtained the best results in PSNR, MSE and RMSE indicators. This suggests that UVSA may be a more effective way to maintain image quality and reduce reconstruction errors.

VII. Conclusion

The UVSA model is an innovative model that combines UV mapping, reverse UV mapping technology and Seagull optimization algorithm, designed for efficient modeling and highly realistic modeling of AI-driven virtual characters in a meta-universe environment. In the meta-universe, an immersive digital environment, the authenticity and texture of virtual characters are crucial. As the basic technology of 3D computer graphics, UV mapping is used to project the surface of 3D model to 2D plane for texture mapping, but the traditional method is easy to cause problems such as texture stretching and compression. On this basis, UVSA uses the powerful global search and optimization ability of Gull optimization algorithm to intelligently optimize the UV coordinate layout of the virtual character model. By simulating the dynamic behavior of seagulls foraging, the algorithm effectively solves the problem of texture distortion and realizes the optimal UV layout, thus improving the texture fit degree and the overall visual effect. After optimizing the UV mapping, UVSA also uses reverse UV mapping technology to ensure that the optimized texture information is accurately returned to the surface of the 3D model, ensuring that the texture details of each part of the virtual character are perfectly matched with the geometry, further enhancing its sense of reality. The application of UVSA not only greatly improves the visual quality of virtual characters in the meta-universe environment, but also greatly improves the modeling efficiency, shortens the design cycle, and reduces the cost of manual intervention. This model is expected to be widely used in VR content development, game production, film and television special effects, virtual social and other fields, with its unique natural inspiration and advanced technology integration, leading the innovation direction of virtual role modeling technology, and providing a strong support for the construction of a delicate, vivid and intelligent meta-universe world.

References

[1] Kikuchi, Shinichi, et al. "Dynamic modeling of genetic networks using genetic algorithm and S-system." Bioinformatics, 19.5, (2003): 643-650.

[2] Tahmasebi, Pejman. "HYPPS: A hybrid geostatistical modeling algorithm for subsurface modeling." Water Resources Research, 53.7, (2017): 5980-5997.

[3] Mrak, Marta, Detlev Marpe, and Thomas Wiegand. "A context modeling algorithm and its application in video compression." Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429). Vol. 3. IEEE, 2003.

[4] Runions, Adam, Brendan Lane, and Przemyslaw Prusinkiewicz. "Modeling trees with a space colonization algorithm." Nph, 7.63–70 (2007): 6.

[5] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces[C]//Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. ACM,1999: 187-194.

[6] Paysan P, Knothe R, Amberg B, et al. A 3D face model for pose and illumination invariant face recognition[C]//2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, 2009: 296-301.

[7] Yin Bao-Cai, Sun Yan-Feng, Wang Cheng-Zhang, et al. BJUT-3D face database and its processing technology [J]. Journal of Computer Research and Development, 2009, 46(6): 1009-1018. (in Chinese)

[8] He Jiayu, Huang Hongbo, Zhang Hongyan, et al. A review of 3D face reconstruction in single image based on deep learning[J]. Computer Science, 2022, 49(2): 40-50. (in Chinese)

[9] Zeng X, Wu Z, Peng X, et al. Joint 3D facial shape reconstruction and texture completion from a single image[J]. Computational Visual Media, 2022, 8(2): 239-256.

[10] Gu X, Gortler S J, Hoppe H. Geometry images[J]. ACM Transactions on Graphics, 2002,21(3): 355-361.