

Enhancing 3D Animation Through AI: Leveraging Computer Vision and Neural Networks

Yunpeng Tang^{*ID}, Bunchoo Bunlikhitsiri, Poradee Panthupakorn

Visual Arts and Design, Faculty of Fine and Applied Art, Burapha University, Chonburi 20000, Thailand

Corresponding Author Email: 63810082@go.buu.ac.th

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380235>

ABSTRACT

Received: 15 January 2024

Revised: 28 February 2024

Accepted: 12 March 2024

Available online: 24 April 2024

Keywords:

3D animation, artificial intelligence, computer vision, deep learning, back propagation neural network, fully convolutional network, transformer, graph neural network

As digital technologies rapidly evolve, 3D animation has become increasingly prevalent in fields such as multimedia, gaming, cinema, and advertising, establishing itself as a vital mode of visual communication. Advances in artificial intelligence (AI), particularly in computer vision and deep learning, have opened new avenues for the creation and dissemination of 3D animations. However, existing methods of 3D animation creation still face numerous challenges, particularly in handling 2D view feature points and producing high-quality 3D animations. This study addresses these limitations by proposing an optimized approach that integrates a back propagation (BP) neural network and a fully convolutional network (FCN), aimed at enhancing the accuracy and efficiency of processing 2D view feature points. Furthermore, a novel pyramid graph neural network (GNN) algorithm based on the Transformer model has been developed, designed to generate natural and high-quality 3D animations depicting agrarian scenes in the Jiangnan region. The application of these technologies not only holds promise for improving the efficiency and quality of 3D animation production but also plays a significant role in advancing the application of AI technologies in artistic creation.

1. INTRODUCTION

In the contemporary digital era, 3D animation has emerged as a crucial component of visual arts and communication [1-3]. With the continuous advancements in AI technologies, particularly in computer vision and deep learning algorithms, the methods of creating and disseminating 3D animations are undergoing unprecedented transformations [4, 5]. These changes not only provide artists with innovative tools for expression but also introduce new research directions for scholars. Despite their significant potential to enhance the efficiency and quality of 3D animation production, these technologies still encounter numerous challenges in practical applications, especially in the processing of 2D view feature points and the precision and naturalness of 3D animation generation [6-8].

The exploration of AI-based 3D animation creation and dissemination, particularly the application of computer vision and algorithms in this process, holds substantial theoretical and practical significance [9-11]. Firstly, it facilitates technological innovation in the 3D animation industry, meeting market demands for high-quality 3D animation content by improving production efficiency and quality. Secondly, from an academic perspective, such research deepens the understanding of the application of AI technologies in the field of visual arts and offers insights for technological innovations in other domains.

Although significant progress has been made in the field of 3D animation creation, existing research methods still exhibit numerous flaws and deficiencies [12-15]. For example,

traditional algorithms often struggle to accurately capture details in complex scenes, leading to low precision in 3D model reconstruction. In terms of animation generation, current methods find it challenging to balance generation efficiency with the naturalness and fluidity of the animations [16-19]. These issues limit the widespread application of 3D animation creation technologies and urgently require resolution through more advanced algorithms.

This study aims to explore new methods for AI-based 3D animation depicting agrarian scenes in the Jiangnan region creation and dissemination. By integrating BP and FCN, the process for handling 2D view feature points has been optimized, enhancing the accuracy and efficiency of feature point recognition. Additionally, a novel pyramid GNN algorithm based on the Transformer model has been proposed to generate high-quality 3D animations. This algorithm not only tackles complex animation generation tasks but also significantly enhances the naturalness and fluidity of animations while ensuring efficiency. Through these core studies, the study not only offers new technological pathways for 3D animation creation but also paves the way for future applications of AI in the field of visual arts.

2. OPTIMIZATION ALGORITHM FOR PROCESSING 2D ANIMATION VIEW FEATURE POINTS

Figure 1 outlines the technical route adopted in the study for the creation and dissemination of 3D animation. An optimization algorithm for processing 2D animation view

feature points, based on BP and FCN, is proposed. Initially, the BP neural network computes the visual complexity and contrast information of the input 2D animation view, which determines the required number of feature points. The total number of feature points is adaptively controlled by adjusting the contrast threshold in the SIFT algorithm to meet the visual processing requirements of the animation. If the initial adjustment does not achieve the preset number of feature points, the algorithm iteratively adjusts to ensure the accuracy of the results. Moreover, the FCN is utilized to identify the

pixel range of the main animation elements, concentrating feature point detection on key animation objects and effectively avoiding interference from the background and irrelevant elements, thus enhancing the accuracy of feature point detection and the efficiency of subsequent matching. During the feature point matching phase, the characteristics of movement and deformation of elements within the animation are analyzed to further optimize the matching process, ensuring the coherence and stability of the animation view across different scenes.

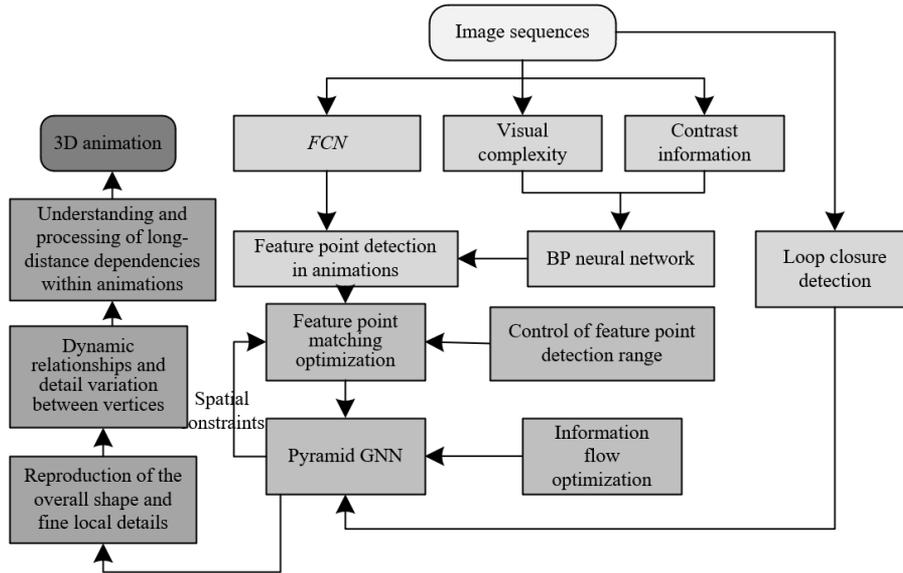


Figure 1. Technical route of the study

2.1 Control of feature point detection quantity

For 2D animation views, the contrast threshold is not only crucial for the stability and quantity of feature points but also directly impacts the visual effect and subsequent processing efficiency of the animation. The root mean square method is typically employed to calculate the contrast in the animation, providing a quantifiable metric that assists the BP neural network in automatically adjusting the contrast threshold in the Scale-Invariant Feature Transform (SIFT) algorithm based on the complexity of the animation content and visual requirements. Using this method, the density and distribution of feature points are effectively controlled. Particularly in 2D animations, this method not only reduces interference from background noise but also ensures that key visual elements are accurately identified and tracked, thereby supporting high-quality animation creation and smooth visual communication.

Assuming the number of pixels is represented by v , the grayscale value of the u -th pixel by a_u , and the average of the image pixel grayscale values by \bar{a} :

$$RMS = \left[\frac{1}{v-1} \sum_{u=1}^v (a_u - \bar{a})^2 \right]^{1/2} \quad (1)$$

In the algorithm, the image's contrast, entropy, correlation, energy, and edge ratio are key parameters that together describe the visual complexity and content characteristics of the 2D animation. Contrast reflects the significance of light and dark differences within the animation, influencing the sensitivity of feature point detection. Entropy quantifies the

randomness of the pixel value distribution in the animation image, with higher entropy indicating richer and more complex content. Correlation reveals the grayscale relationship between pixels, while high correlation typically indicates the consistency of the image's texture structure. Energy describes the consistency of the image texture's repeated patterns, with higher energy implying a more uniform texture. The edge ratio reflects the proportion of edge elements to total pixels in the animation, serving as an important indicator for evaluating the dynamic characteristics and visual complexity of the animation. In 2D animation views, these parameters not only determine the generation and optimization process of feature points but also have a decisive impact on the animation's dynamic characteristics and visual coherence. Assuming the number of image grayscale levels is represented by j , the total quantity of the m -th grayscale level by v_m , the total number of image pixels by V , the number of pixels outlining the main object by O_{ED} , and the value at (u,k) in the grayscale co-occurrence matrix by $o(u,k)$, with $\omega_a = \sum_{a=1}^j \sum_{b=1}^j a \cdot o(a,b)$, $\omega_b = \sum_{a=1}^j \sum_{b=1}^j b \cdot o(a,b)$, $\delta_a^2 = \sum_{a=1}^j \sum_{a=1}^j o(a,b) \cdot (a - \omega_a)^2$, and $\delta_b^2 = \sum_{a=1}^j \sum_{b=1}^j o(a,b) \cdot (b - \omega_b)^2$, the following formula can be derived:

$$\begin{cases} G = -\sum_{m=1}^j \frac{v_m}{V} \log\left(\frac{v_m}{V}\right) \\ ZPN = \left[\sum_{u=1}^V \sum_{k=1}^V uko(u,k) - \omega_a \omega_b \right] / \delta_a \delta_b \\ K = \sum_{u=1}^V \sum_{k=1}^V o(u,k)^2 \\ O = \frac{O_{ED}}{V} \end{cases} \quad (2)$$

compensate for the loss of pixel positional information incurred during downsampling, thereby enhancing the network's segmentation accuracy of the main animation targets.

2.3 Optimization of feature point matching

Due to the frequent presence of similar textures in 2D animations, traditional SIFT algorithms and nearest neighbor-based matching methods may lead to incorrect feature point pairings, such as a feature point being mistakenly matched with another visually similar feature point. To address this issue and to reduce mismatches while enhancing matching accuracy, a strategy for optimizing feature point matching was proposed. This strategy introduces spatial constraints on pixel positions during the selection of matching points. Specifically, matching points must not only be close in feature description but also satisfy a certain proximity in pixel position. This constraint was enforced by setting a threshold, Sg , typically 10% of the image's diagonal length, to ensure that the positional differences between corresponding feature points in two images are within an acceptable range. Assuming the pixel coordinates of a feature point are represented by (a,b) and the length and width of the image are denoted by I_C and I_R , respectively, the following formula is applied:

$$\begin{cases} L = \{((a_1, b_1), (a'_1, b'_1)), \dots, ((a_v, b_v), (a'_v, b'_v))\} \\ (a'_1 - a_1)^2 + (b'_1 - b_1)^2 \leq Sg_e^2 \\ Sg_e = \sqrt{(I_C^2 + I_R^2)} \cdot 0.1 \end{cases} \quad (6)$$

3. GENERATION OF 3D ANIMATION BASED ON PYRAMID GNN

Following the optimization of 2D animation view feature point processing, a pyramid GNN algorithm based on the Transformer was proposed to effectively handle complex view relationships and enhance the representation of object details within animations. This algorithm addresses the issue of insufficient information exchange between vertices in traditional GNNs when processing large-scale images. A pyramid structure was adopted to adapt to different levels of view details, thereby maintaining local information while strengthening the integration of global information. Each layer's SGT unit optimizes the flow of information through an adaptive sampling module that samples key view features, a local GNN module that processes local connections, and a global Transformer module that enhances the remote communication capabilities between vertices. This structural design is particularly suitable for generating 3D animations from 2D animations, as it not only requires precise geometric construction but also needs to portray coherent and smooth dynamic details. The introduction of SGT units effectively captures fine shape variations in animations, such as minute structures on object surfaces, significantly enhancing the quality and realism of the generated animations.

3.1 Adaptive sampling module

To more effectively process point cloud data, an adaptive sampling module was incorporated into the pyramid GNN model for generating 3D animations, thus better reproducing

the overall shape and fine local details of objects within 3D animations. The workflow of this module is shown in Figure 3. Unlike traditional 3D reconstruction, which focuses on static geometric precision, 3D animation generation also needs to capture dynamic subtle changes to enhance the realism and visual effects of the animation. Traditional point cloud processing methods, such as those in Pixel2Mesh which involve upsampling through the insertion of midpoints between neighboring edges, cannot be directly applied to point cloud data due to the lack of inherent edge structures in point clouds. Therefore, the algorithm employs an adaptive sampling mechanism based on attention weight matrices, which dynamically adjusts sampling strategies based on the spatial and semantic relationships between points. Through this method, the algorithm meticulously captures and expresses local features and details of objects while maintaining their macroscopic contours, thereby achieving higher levels of visual coherence and detail richness in 3D animation generation.

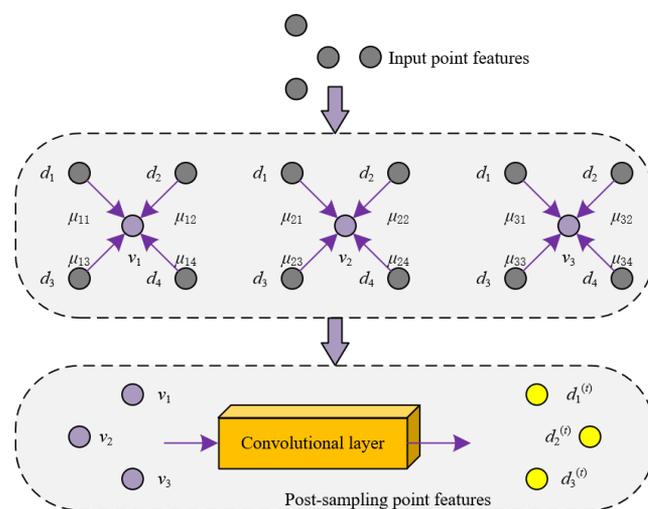


Figure 3. Workflow of the adaptive sampling module

Specifically, the network utilizes the attention weight matrix Q to determine which points should be prioritized for sampling and how the sampling should be conducted. The dimensions of the weight matrix Q are $V2 \times V1$, where $V1$ is the number of input point features, $V2$ is the number of point clouds after sampling, and $F1$ is the dimensionality of point features. Through this weight learning mechanism, the network distinguishes the relationships between different shape parts, assigning higher weights to similar or adjacent points, thereby preserving more local detail information. When the number ($V1$) of input points exceeds the required number ($V2$) of sampling points, the module performs downsampling, reducing the number of points while retaining key features. Conversely, when $V1$ is less than $V2$, upsampling is executed to increase the number of points and enrich details. Assuming the point features of each input view are represented by D , and the point features after sampling are denoted by D_i . The feature vector corresponding to the i -th point feature in D_i is represented by $d^{(i)}_i$, and the feature vector of the k -th point in input point features D by d_k . The value in the u -th row and k -th column of the attention weight matrix Q to be trained is denoted by μ_{uk} , the output dimension of the convolutional layer by F , and the weights and biases of the 1D convolutional layer by Q_1 and y_1 , respectively. Thus, the sampling process can be mathematically defined as:

$$d_u^{(t)} = \left(\sum_{d_k \in D} \mu_{uk} d_k \right) Q_1 + y_1 \quad (7)$$

3.2 Local GNN module

In the process of generating 3D animations, handling complex 2D animation scenes requires the meticulous and accurate portrayal of the dynamic relationships and detail changes between vertices. To overcome the issue of traditional GNNs potentially overlooking the diversity among vertices during vertex feature aggregation, a local GNN module was introduced in the proposed 3D animation generation algorithm. This module, through an attention mechanism, precisely captures and emphasizes the local relationships between different vertices within the animation, especially in complex dynamic changes and fine structural expressions, which are crucial for high-quality 3D animation generation.

The working mechanism of this module is based on establishing a k -nearest neighbor graph between vertices and their neighbors, applying an attention mechanism to dynamically learn and adjust the weight of each edge. Such a design allows the module to adaptively allocate the weight of information transfer based on the spatial positions and possible semantic relationships between vertices, thus more effectively capturing local details and dynamic changes. Each vertex aggregates features not only from neighboring vertices within the same view but also from neighboring vertices across multiple views. Specifically, suppose the set of all neighboring points of the current reference point u is represented by V_u , and the point features of vertex u and k post-sampling are denoted by $d_u^{(t)}$ and $d_k^{(t)}$, respectively. The point feature of vertex u after being updated by the GNN is represented by h_u , the normalized relevance coefficient by β_{uk} , the increase in point feature dimension to enhance point features by matrices Q_{x1} and Q_{x2} , and the activation function by $\Sigma(\cdot)$. The proposed algorithm, during the update in the GNN, transmits both $d_k^{(t)}$ and $d_u^{(t)} - d_k^{(t)}$, and concatenates them in the feature dimension to simultaneously learn the global associations and local differences between vertices within the GNN. The specific update process can be defined as follows:

$$\beta_{uk} = \text{softmax} \left(Q_{x1} \left(d_k^{(t)} \left\| \left(d_u^{(t)} - d_k^{(t)} \right) \right\| \right) \right) \quad (8)$$

$$h_u = \delta \left(\sum_{k \in V_u} \beta_{uk} Q_{x2} d_k^{(t)} \right) \quad (9)$$

3.3 Global Transformer module

In the field of 3D animation generation, there is a high demand for global consistency and coherence in complex animation scenes, necessitating the model's capability to understand and address long-distance dependencies among various parts of the animation. While traditional GNNs excel in local feature processing, they exhibit limitations in modeling long-range dependencies within data. Based on this, a Transformer module was introduced, utilizing its robust global information processing ability to effectively capture and integrate dynamic information from different characters and scenes within the animation. This enhances the model's understanding of dynamic changes throughout the entire animation sequence, ensuring the visual and emotional

coherence and appeal of the generated animations. The structure is shown in Figure 4. Specifically, the application of the global Transformer module within the pyramid GNN primarily addresses the challenges of the unordered nature of point cloud data and global relationship modeling. Unlike sequences in natural language processing, point clouds do not possess a fixed order, making traditional position-encoding methods inapplicable. To overcome this issue, an innovative approach was adopted by projecting the initialized point cloud onto a feature map using the camera intrinsic matrix, thus implicitly embedding positional information. This method not only preserves the spatial information of the point cloud but also allows the Transformer to process this information effectively through its self-attention mechanism, thereby strengthening the global dependencies among the point cloud data.

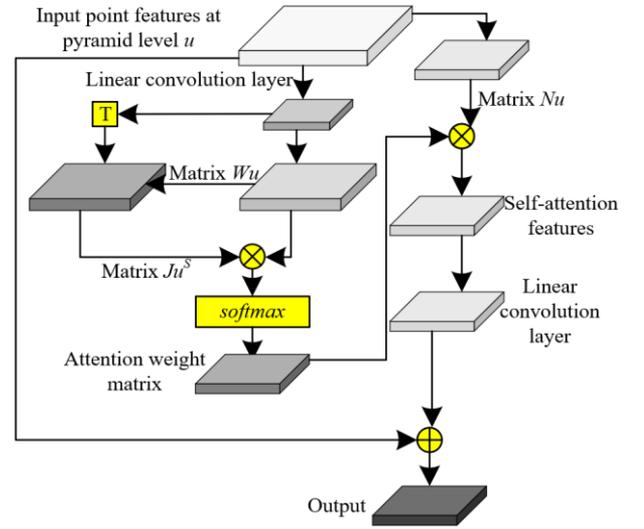


Figure 4. Structure of the global Transformer module

It is assumed that the point features updated by the GNN at pyramid level u are represented by H_u . The number of viewpoints is denoted by VI , the number of points at pyramid level u by V_u , the dimension of point features at pyramid level u by F_u , and the output feature dimension by F_p . The conventional Transformer structure comprises three matrices, W , J , and N , corresponding to *Query*, *Key*, and *Value*, represented by W_u , J_u and V_u , respectively. The linear transformation matrices are denoted by Q_{wu} , Q_{ju} and Q_{nu} , with the feature dimensions of the transformation matrices represented by F_{xu} and F_{ru} . To compute these three matrices, the following transformation was proposed for the algorithm:

$$(W_u, J_u, N_u) = H_u \times (Q_{w_u}, Q_{j_u}, Q_{n_u}) \quad (10)$$

The transpose matrix of J_u is represented by J_u^S . The algorithm first multiplies W_u and J_u^S , and the product is then normalized through the $\text{softmax}(\cdot)$ function. Finally, the self-attention features are added to the input features H_u , resulting in the output point features at pyramid level u , denoted as D_{OU_u} . The feature dimension size is denoted by F_{xu} , with the entire transformation expressed as:

$$D_{OU_u} = \text{softmax} \left(\frac{W_u \times J_u^S}{\sqrt{F_{xu}}} \right) + H_u \quad (11)$$

3.4 Hierarchical loss function

In the algorithm, a hierarchical loss function was established to ensure that animation details generated from each level adhere to high-quality standards. This design primarily addresses the challenges of various complex dynamic scenes in animation, where features at different levels need to accurately express dynamic information of varying granularity. Through the hierarchical loss function, the algorithm optimizes features individually at each level, thus allowing for more precise control of every detail in the animation, whether it involves large-scale motion or subtle facial expressions.

The hierarchical loss function optimizes the quality of reconstruction by calculating the difference between the reconstruction results at each level and the actual animation data. Specifically, the algorithm employs the FPS-CD loss term and CD loss term to respectively ensure the global shape consistency and local detail accuracy of the animation. The FPS-CD loss term ensures that reconstruction results obtained from different viewpoints approximate the overall shape of the same object, which is particularly important for multi-view animation scenarios. The CD loss term focuses on the local details of the point cloud, effectively reducing discontinuities and structural errors in the animation generation process, such as the formation of holes. Through this dual loss mechanism, animation generation at each layer is independently optimized, thereby enhancing the overall quality and visual effects of the animation and ensuring that every scene in the animation accurately reflects the intended design and dynamic changes.

Assuming the actual point cloud result is represented by hs , the reconstruction result at pyramid level u by L'_u , and the point cloud result after merging multi-view reconstruction results by L_u , the CD loss function is denoted by $zf()$, and the CD loss function after farthest point sampling by $dotzf()$. The number of layers in the pyramid is represented by m , and the hyperparameters by variables β and ε . The loss function is defined as follows:

$$LOSS = \sum_{u=1}^m \varepsilon_u (\beta_1 * zf(hs, L) + \beta_2 * dotzf(hs, L')) \quad (12)$$

4. EXPERIMENTAL RESULTS AND ANALYSIS

From the data in the Table 1, a comparison of time consumption in the feature point matching stage between the Kohonen Clustering Network (KCN) algorithm and the algorithm presented in this study is observed. Examining the time data from experiments 1 to 10, it is generally noted that the time taken by the proposed algorithm is higher than that of the KCN algorithm. For instance, in experiment 1, the KCN

algorithm took 2.31 seconds, while the proposed algorithm took 2.41 seconds; in experiment 10, the KCN algorithm took 3.47 seconds, compared to 3.56 seconds for the proposed algorithm. Calculating the average time consumption, the KCN algorithm's average was 2.84 seconds, whereas the average for the proposed algorithm was slightly higher at 3.03 seconds. This indicates that across all experiments, the proposed algorithm has a marginally higher processing time than the KCN algorithm, though the overall difference in time is not significant. Despite the slight increase in time consumption by the current study's algorithm, this difference primarily stems from enhancements in feature point recognition accuracy and efficiency. By integrating BP and FCN, the proposed algorithm aims to improve the quality of feature point processing and the naturalness and smoothness of the final animations. Therefore, although there is a slight increase in processing time, it can be considered an investment in the quality of feature point handling, thereby enhancing the overall quality of the final 3D animations depicting agrarian scenes in the Jiangnan region.

According to the data presented in Figure 5, the model proposed in this study exhibited a significant convergence trend during the training process. With an initial error rate of 0.8000 at the start of training, there was a rapid decline to 0.0060 by the 500th epoch, indicating swift initial improvements in feature point processing optimization. Subsequently, the error rate continued to steadily decrease, reaching 0.0020 by the 2000th epoch and gradually diminishing to 0.0010 by the 5000th epoch. This stable downward trend and the ultimate stabilization of the error demonstrate the efficacy of the optimization algorithm, which consistently improved until nearing the ideal minimal error value. The analysis reveals that after a period of rapid learning and adaptation, the error rate stabilized at 0.0010, consistent with the set target error value and historical best performance. This high level of consistency and the model's sustained low-error performance underscore the efficiency and accuracy of the developed algorithm in processing feature points for 2D animation views.

To evaluate the effectiveness of the algorithm proposed in this study, metrics including ASPD, ASPD1, ASPD2, FD-ASPD, FD-ASPD1, and FD-ASPD2 were employed. The ASPD loss represents the average shortest point distance between the reconstructed and actual point clouds. The loss encompasses two components, ASPD1 and ASPD2, where ASPD1 denotes the distance between the reconstructed point cloud result and the actual point cloud result, representing accuracy. ASPD2 indicates the coverage of the actual point cloud result in the reconstructed point cloud result, representing completeness. FD-ASPD loss refers to the ASPD loss after farthest distance sampling, comprising FD-ASPD1 and FD-ASPD2, each corresponding to the components ASPD1 and ASPD2 of ASPD loss.

Table 1. Time consumption comparison in the feature point matching stage between the KCN algorithm and the proposed algorithm

| Experiment No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average Time (Seconds) |
|--------------------------------------|------|------|------|------|------|------|------|------|------|------|------------------------|
| KCN algorithm | 2.31 | 2.56 | 1.12 | 2.47 | 3.05 | 3.58 | 2.89 | 3.74 | 3.21 | 3.47 | 2.84 |
| The algorithm proposed in this study | 2.41 | 2.74 | 1.32 | 2.56 | 3.15 | 3.78 | 3.24 | 4.02 | 3.52 | 3.56 | 3.03 |

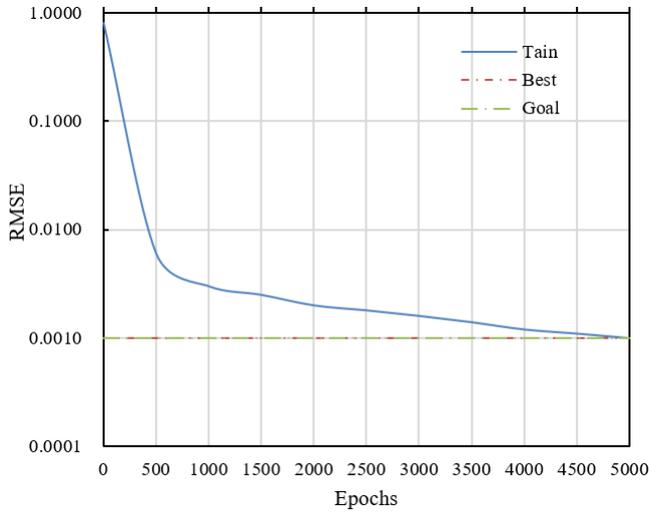


Figure 5. Convergence of the optimization model for feature point processing in 2D animation views

Table 2 provides a comparison of ASPD metric results for various pyramid GNN variants and the proposed algorithm for reconstructing 3D animations of Jiangnan agricultural cultural heritage. The data indicate that the proposed algorithm generally exhibits lower ASPD loss values across all categories compared to the other four algorithms. For example, in the "production tools" category, the proposed algorithm shows an ASPD of 1.985, significantly lower than the others, with the multi-scale pyramid GNN reaching as high as 5.124. Similarly, in the "beliefs" category, although all algorithms record higher losses, the proposed algorithm still demonstrates the lowest ASPD value of 3.562, whereas the multi-scale pyramid GNN records a high of 10.236. This indicates that,

regardless of the simplicity or complexity of the category, the proposed algorithm effectively reduces reconstruction errors and enhances the quality of 3D animations. These results highlight the lower ASPD loss exhibited by the proposed algorithm across different cultural heritage categories, thereby ensuring the accuracy and visual effectiveness of the animations. This is particularly crucial for the complex reconstruction of agricultural cultural heritage animations, effectively enhancing the visual experience for cultural transmission and education.

Table 3 displays a performance comparison for the FD-ASPD metric among the proposed algorithm and four other pyramid GNN variants. The data reveal that the reconstruction errors of the proposed algorithm are generally lower across all cultural categories compared to the other algorithms. For instance, in the "production tools" category, the FD-ASPD loss for the proposed algorithm is 2.231, significantly reduced compared to the highest value of 5.124 from the multi-scale pyramid GNN. Moreover, in the complex category of "beliefs," the proposed algorithm also shows a lower error value of 4.215, while other algorithms range between 5.412 and 10.236. These results demonstrate a significant advantage of the current study's algorithm in accurately reconstructing details and overall structures, especially maintaining a low error rate even after the farthest distance sampling. Analysis indicates that the proposed algorithm exhibits high accuracy and efficiency in generating 3D animations, as clearly evidenced in the FD-ASPD loss comparison. By optimizing the feature point processing in 2D views and integrating an innovative pyramid GNN algorithm based on the Transformer, not only is the accuracy of feature point recognition enhanced, but also the high quality of complex animation generation tasks is ensured.

Table 2. ASPD metric results for 3D animation generation by the proposed algorithm and comparative algorithms

| Category | Multi-scale Pyramid GNN | Spatio-Temporal Pyramid GNN | PGATs | PGCNs | The Proposed Algorithm |
|----------------------|-------------------------|-----------------------------|-------|-------|------------------------|
| Production tools | 5.124 | 2.789 | 2.651 | 2.684 | 1.985 |
| Household utensils | 5.326 | 4.125 | 3.235 | 3.214 | 2.541 |
| Apparel | 4.569 | 5.021 | 4.658 | 3.695 | 2.632 |
| Architecture | 4.458 | 3.652 | 4.235 | 3.102 | 2.754 |
| Agricultural customs | 5.789 | 4.458 | 4.127 | 3.568 | 3.124 |
| Folk art | 5.562 | 5.562 | 6.125 | 4.125 | 3.023 |
| Beliefs | 10.236 | 5.412 | 5.891 | 5.235 | 3.562 |

Table 3. FD-ASPD metric results for 3D animation generation by the proposed algorithm and comparative algorithms

| Category | Multi-scale Pyramid GNN | Spatio-Temporal Pyramid GNN | PGATs | PGCNs | The Proposed Algorithm |
|----------------------|-------------------------|-----------------------------|-------|-------|------------------------|
| Production tools | 5.124 | 2.789 | 2.651 | 3.021 | 2.231 |
| Household utensils | 5.326 | 4.125 | 3.235 | 3.895 | 3.214 |
| Apparel | 4.569 | 5.021 | 4.658 | 4.895 | 3.325 |
| Architecture | 4.458 | 3.652 | 4.235 | 3.885 | 3.357 |
| Agricultural customs | 5.789 | 4.458 | 4.127 | 4.321 | 3.698 |
| Folk art | 5.562 | 5.562 | 6.125 | 4.895 | 3.689 |
| Beliefs | 10.236 | 5.412 | 5.891 | 6.235 | 4.215 |

Table 4. Results of single-category 3D animation generation by the proposed algorithm and comparative algorithms

| Method | ASPD | ASPD1 | ASPD2 | FD-ASPD | FD-ASPD1 | FD-ASPD2 |
|-----------------------------|-------|-------|-------|---------|----------|----------|
| Multi-scale pyramid GNN | 5.685 | 2.485 | 3.215 | 5.784 | 2.358 | 3.254 |
| Spatio-temporal pyramid GNN | 4.321 | 2.023 | 2.315 | 4.325 | 2.014 | 2.314 |
| PGATs | 4.125 | 1.895 | 2.214 | 4.128 | 1.896 | 2.210 |
| PGCNs | 3.326 | 1.326 | 1.897 | 4.056 | 1.689 | 2.356 |
| The proposed algorithm | 2.568 | 1.124 | 1.542 | 3.269 | 1.425 | 1.874 |

vivid and lifelike, and the scenes more realistic. Overall, this study provides robust methodological support and empirical results for the high-quality generation of 3D animations, offering new perspectives and technical pathways for the application of AI in artistic creation, particularly in the field of animation production.



Figure 8. Scene color



Figure 9. Animation scene

5. CONCLUSION

This study successfully explored a novel approach that combines BP and FCN with a Transformer-based pyramid GNN algorithm for generating high-quality 3D animations of Jiangnan agricultural culture. The study primarily focused on optimizing the processing of 2D view feature points, enhancing the accuracy and efficiency of feature point recognition while ensuring the naturalness and fluidity of the generated animations. A series of experimental analyses validated the effectiveness and superiority of the proposed method. The experimental results demonstrated that, although the proposed model consumes slightly more time than the KCN algorithm, it performs better in improving the quality of feature point matching and the details of animations. The model exhibited rapid and effective convergence capabilities, ensuring stability and efficiency in processing. Across multiple categories, the proposed algorithm outperformed comparative algorithms in both ASPD and FD-ASPD metrics, showing its advantages in detail accuracy and completeness.

This study significantly enhanced the quality of 3D animation depicting agrarian scenes in the Jiangnan region generation, especially in handling animations with complex and culturally specific content. The application of advanced neural network technologies not only improved the naturalness and fluidity of the animations but also optimized the efficiency and effectiveness of the animation production process. However, the study also has certain limitations, such as the high computational resource demands of the algorithm, which may restrict its application on low-power devices.

Future research could focus on further reducing the

computational demands of the algorithm to accommodate a broader range of applications, such as mobile devices and real-time animation generation. Additionally, exploring the adaptability and scalability of the algorithm across different types of animation content, such as motion capture and virtual reality environments, will be an important direction for future work. Furthermore, further optimization of the algorithm, especially in terms of efficiency and precision when processing larger datasets, remains a key area for future research.

REFERENCES

- [1] Yusufu, A. (2022). Research on 3D animation production system of industrial internet of things under computer artificial intelligence technology. In 2022 IEEE 2nd International Conference on Data Science and Computer Application, ICDSCA 2022, Dalian, China, pp. 1415-1418. <https://doi.org/10.1109/ICDSCA56264.2022.9988175>
- [2] Lv, Y. (2022). Application of 3D animation cluster system based on artificial intelligence and machine learning. Computational Intelligence and Neuroscience, 2022: 2904607. <https://doi.org/10.1155/2022/2904607>
- [3] Gao, Q. (2022). Design and implementation of 3D animation data processing development platform based on artificial intelligence. Computational Intelligence and Neuroscience, 2022: 1518331. <https://doi.org/10.1155/2022/1518331>
- [4] Hu, Q. (2024). Artistic characteristics and innovative application of ink animation in animation scenes in the age of artificial intelligence. Applied Mathematics and Nonlinear Sciences, 9(1): 1-16. <https://doi.org/10.2478/amns-2024-0591>
- [5] Fu, H., Wang, Z., Gong, K., Wang, K., Chen, T., Li, H., Kang, W. (2024). Mimic: Speaking style disentanglement for speech-driven 3D facial animation. In Proceedings of the AAAI Conference on Artificial Intelligence, 38(2): 1770-1777. <https://doi.org/10.1609/aaai.v38i2.27945>
- [6] Kumar, A., Jilani Saudagar, A.K., Alkhatami, M., Alsamani, B., Abul Hasanat, M.H., Khan, M.B., Singh, K.U. (2022). AIAVRT: 5.0 transformation in medical education with next generation AI-3D animation and VR integrated computer graphics imagery. Traitement du Signal, 39(5): 1823-1832. <https://doi.org/10.18280/ts.390542>
- [7] Bao, Y. (2022). Application of virtual reality technology in film and television animation based on artificial intelligence background. Scientific Programming, 2022: 2604408. <https://doi.org/10.1155/2022/2604408>
- [8] Ma, L., Yu, S., Xu, X., Amadi, S.M., Zhang, J., Wang, Z. (2023). Application of artificial intelligence in 3D printing physical organ models. Materials Today Bio, 23: 100792. <https://doi.org/10.1016/j.mtbio.2023.100792>
- [9] Wang, B., Shi, Y. (2023). Expression dynamic capture and 3D animation generation method based on deep learning. Neural Computing and Applications, 35(12): 8797-8808. <https://doi.org/10.1007/s00521-022-07644-0>
- [10] Tian, X., Li, C. (2024). Augmented reality animation image information extraction and modeling based on generative adversarial network. Computer-Aided Design and Applications, 21(S3): 77-91.

- <https://doi.org/10.14733/cadaps.2024.S3.77-91>
- [11] Bedoya, M.G., Montoya, D.R., Tabilo-Munizaga, G., Pérez-Won, M., Lemus-Mondaca, R. (2022). Promising perspectives on novel protein food sources combining artificial intelligence and 3D food printing for food industry. *Trends in Food Science & Technology*, 128: 38-52. <https://doi.org/10.1016/j.tifs.2022.05.013>
- [12] Peng, T., Wu, W., Liu, J., Li, L., Miao, J., Hu, X., Li, L. (2023). PGN-Cloth: Physics-based graph network model for 3D cloth animation. *Displays*, 80: 102534. <https://doi.org/10.1016/j.displa.2023.102534>
- [13] Bouali, N., Cavalli-Sforza, V. (2023). A review of text-to-animation systems. *IEEE Access*, 11: 86071-86087. <https://doi.org/10.1016/j.displa.2023.102534>
- [14] Lan, C., Wang, Y., Wang, C., Song, S., Gong, Z. (2023). Application of ChatGPT-based digital human in animation creation. *Future Internet*, 15(9): 300. <https://doi.org/10.3390/fi15090300>
- [15] Gao, Y. (2023). The application of digital media art in film and television animation based on three-dimensional interactive technology. *Applied Mathematics and Nonlinear Sciences*, 9(1): 1-17. <https://doi.org/10.2478/amns.2023.2.00313>
- [16] Li, J., Li, Z., Jiang, P., Wang, L., Li, X., Hao, Y. (2024). Guiding 3D digital content generation with pre-trained diffusion models. *International Journal of Advanced Computer Science & Applications*, 15(1): 1220-1230. <https://doi.org/10.14569/ijacsa.2024.01501120>
- [17] Liu, H. (2023). Creative design and technical analysis of animation public service advertisement based on ant colony optimization algorithm. *International Journal of Digital Multimedia Broadcasting*, 2023: 5059665. <https://doi.org/10.1155/2023/5059665>
- [18] Gao, W., Wang, C., Li, Q., Zhang, X., Yuan, J., Li, D., Gu, Z. (2022). Application of medical imaging methods and artificial intelligence in tissue engineering and organ-on-a-chip. *Frontiers in Bioengineering and Biotechnology*, 10: 985692. <https://doi.org/10.3389/fbioe.2022.985692>
- [19] Diao, J., Xiao, J., He, Y., Jiang, H. (2023). Combating spurious correlations in loose-fitting garment animation through joint-specific feature learning. In *Computer Graphics Forum*, 42(7): e14939. <https://doi.org/10.1111/cgf.14939>