Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **1**, February **2024**

Intelligent Generation of Plant Landscaping on Bidirectional Recurrent Network Auto-encoder

Yu-Hang Zhao, Zhen Tang*

School of Art and Design
Shanghai University of Engineering Science
Shanghai 201620, P. R. China
yuh@sues.edu.cn, spritz_sues@126.com

Zhong-Jun He

School of Computer Science
Taylor's University
Selangor 47500, Malaysia
yikaiyue@gmail.com

*Corresponding author: Zhen Tang
Received July 17, 2023, revised September 19, 2023, accepted November 28, 2023.

Abstract. *In the three-dimensional modeling technology, it is very difficult to draw all plants accurately and carefully. The current level of computer software and hardware is far from meeting the requirements. In addition, many application fields require real-time rendering, so we must make the necessary balance between accuracy and complexity. Therefore, in order to solve the above problems, a fast and realistic intelligent generation method of plant landscaping based on deep learning is proposed. Firstly, Bezier parametric curves are used as vertex control curves to generate trunks, branches and leaves, and various plant entities are generated by combining the diversity of geometric models with the diversity of image textures. Then, based on the half-edge folding algorithm, the low-precision three-dimensional geometric model is optimized by using the bidirectional cyclic network auto-encoder, so as to improve the fidelity and reduce the computational overhead. Finally, in the process of detail level drawing, a preprocessing selection stage is added, and the appropriate parallax measurement coefficient is selected for the scene area of each frame through the calculation of projection value. The experimental results show that compared with other modeling methods, the proposed modeling method realizes the rapid rendering of plants, and shows better performance in frame rate change and execution time.*
Keywords: Plant landscaping; Three-dimensional geometric model; Deep neural network; Auto-encoder; LOD

1. **Introduction.** Modern urban landscapes have multiple functions such as ecological, cultural and social services, therefore the simulation and drawing of natural landscapes has always been an important part of modern urban landscape design. Plant landscape simulation, on the other hand, is an important part of natural landscape simulation. In the modern world of economic development and technological advancement, people are more eager to embrace nature and thus improve their quality of life. Botanical landscapes in urban centres help to relieve the stress and negative emotions of city dwellers from their stressful jobs. The simulation and drawing of natural landscapes in modern urban landscape design has always been a hot research topic in related fields [1,2]. With the

rapid development of hardware technology and computer graphics, 3D scene modelling technology has been widely used in animation, virtual reality, landscape design and many other fields. The use of 3D scene modelling techniques to generate a variety of realistic natural scenery has been a challenging topic in computer graphics [3,4].

The goal of nature landscaping is to use various algorithms of computer graphics to construct and draw objects that exist in real nature [5], such as blue skies and white clouds, mountains and rivers, flowers and trees. The quality of the drawing should be highly realistic, so that the observer can feel "immersive". That is to say, when the observer sees the images generated by these computers, it is like being in nature [6,7]. In this way, graphics generation techniques for plant scenes can be used to create realistic virtual environments. However, due to the complexity of plants, many problems arise when a tree model is used in the computer to generate plant landscapes [8,9]. If the model has too few parameters, it is insufficient to describe the morphological characteristics of the plants. If the model has too many parameters, the generated plant scenery is very vague. To date, computer construction and drawing of planted landscapes has mainly included the following aspects [10,11]: 1) terrain construction and grid simplification techniques [12]; 2) plant construction and drawing techniques [13]; 3) cloud construction and drawing techniques [14]; and 4) rain, snow and water simulation and drawing techniques. This paper examines mainly the generation of plant [15].

The study of plants in computer graphics has focused on modelling and drawing. However, due to the complexity of natural plants, constructing and drawing realistic plant landscapes is difficult, especially as the amount of data required to model a 3D scene is a key issue [16,17]. For example, the number of particles required for a single plant is on the order of a million when modelling plant landscapes using particle systems. This requires the algorithm to carefully allocate computer resources such as CPU, memory and storage space, and to take steps to optimise the computation and output. Due to the limitations of computer processing performance, real-time painting of images in 3D scene modelling [18] requires an effective balance between realism and painting speed to provide the best efficiency for the user given the available hardware [19,20]. Plant landscape tasks require the 3D scene to be refreshed at the same rate as the user's viewpoint changes. This requires an effective need to reduce the number of triangular facets without reducing resolution. A refresh rate of at least 40 fps is generally required for the user to have no perception of latency [21,22]. Currently, with the boom in urban planning and design, virtual reality and game development, there is an increasing demand for the rapid generation of large-scale realistic planted landscapes. However, there is relatively little research on complex plant landscape at this stage.

The aim of this study is therefore to combine a number of optimisation techniques for the rapid construction and drawing of complex plant landscapes, including Bezier parametric curves, deep neural network and approximate selection of LOD models. On this basis, modern urban landscape design can provide users with a real-time, realistic experience of the natural ecosystem, so that the observer can feel 'immersive'.

1.1. **Related Work.** At this stage, there are three main optimisation methods for modeling realistic plant landscapes [23,24]: 1) accelerated image-based drawing; 2) LOD techniques; and 3) fractal methods. The fractal approach is a method that uses plant morphological structure (structural self-similarity) to produce plant graphics. L-systems can describe plant ground topology succinctly, but are more difficult to model complex plant forms. The image-based drawing method can draw complex natural scenes with very few polygon constructions. Peterson et al. [25] propose a rapid modelling method based on SketchUp virtual scenes that can meet the needs of high volume scene modelling.

Cubino et al. [26] propose a vectorised modelling method to reduce complexity and enhance realism by realistic drawing. The control of virtual objects is achieved through the objectification of model imports. The final result is a good synthesis in terms of realism, real time and controllability of the scene modelling.

In contrast to traditional geometric model-based methods, image-based drawing methods are independent of the complexity of the scene, which is particularly important in building large-scale, highly complex and realistic natural scenes. Lin et al. [27] proposed a quadratic error calculation method for edge-folded mesh simplification and used it to simplify a 3D scene model. The results show that the algorithm can effectively reduce the complexity of 3D models. Zhou et al. [28] proposed a Levels of Detail (LOD) model simplification algorithm based on half-edge folding to rank the cost of half-edge folding by introducing quadratic error calculation, curvature features and visual feature degrees, thus effectively reducing the cumulative error of the model. Image-based drawing methods essentially operate directly on the image. However, image-based drawing methods also have certain limitations. Firstly, image-based drawing methods require a large computational overhead, resulting in long model rendering times. Secondly, when using image-based drawing methods for plant landscape, it is difficult to select the scene area for each frame during the LOD drawing process, i.e. the computational complexity is high.

The three-dimensional geometric model based on plant parameterization has the problems of low data accuracy and mixed noise. In addition, the three-dimensional set model based on half-edge folding algorithm has a low level of detail, and it needs to be optimized to ensure realistic visual effects. Removing data noise and completing the missing key node information in the data can make the three-dimensional geometric model obtain a higher level of detail. Bidirectional Recurrent Network Auto-encoder (BRNA) is a neural network model, which is used for unsupervised learning and feature extraction. It combines the idea of auto-encoder and cyclic neural network, and performs well in time series data processing and sequence generation tasks. The research results of Li et al. [29] show that BRNA can learn more abundant representation forms and improve the performance of 3D model and data processing effect.

1.2. **Motivation and contribution.** Based on the above research theories, this paper investigates image-based drawing methods for the optimisation of complex plant landscapes. Most of the current algorithms have certain shortcomings in the generation of complex natural landscapes, especially botanical landscapes. Based on the existing research, this paper mainly studies the real-time of plant landscaping. Bezier parametric curves are used as vertex control curves to generate tree trunks, branches and leaves, and by combining the diversity of geometric models with the diversity of image textures, a variety of plant entities are generated. Experimental results show that the proposed optimisation method is able to generate complex plant landscapes at a fast rendering speed while maintaining image quality.

The main innovations and contributions of this paper include.

(1) In order to remove the data noise and complete the missing key node information in the data, the 3D geometric model can obtain a higher level of detail. Based on the half-edge folding algorithm, the parameters of complex 3D geometric model are optimized by using bidirectional cyclic network auto-encoder, so as to improve the fidelity and reduce the computational overhead. The bidirectional cyclic network auto-encoder is used to make the optimized data output by the network have higher position accuracy.

(2) A pre-processing stage for selecting the level of detail is added to the LOD drawing process for plant landscape, and the projection values are calculated to select the appropriate level of detail parallax metric for each frame of the scene area, speeding up the real-time calculation.

2. **The proposed intelligent plant landscape generation method.**

2.1. **Demand analysis.** The aim of 3D scene drawing technology in real time is to use computer technology to give the user a sense of 3D vision when viewing a virtual scene in all directions. In order to provide the user with the best possible sense of realism, the technology needs to maximize the refresh rate of the image. If the refresh rate is below 40 fps, the user will see a frame-dropping phenomenon, which greatly affects the user's experience.

However, in the design of modern urban landscapes it is necessary to include a large number of natural plant features in order to meet ecological requirements. These plant landscapes require the processing of complex image data. Failure to significantly increase the performance of hardware devices such as memory, CPUs and graphics cards can result in significantly lower image refresh rates. However, a high enough image refresh rate must be guaranteed to ensure that the scene display remains somewhat real-time. Otherwise, the user experience in the system is significantly reduced and in severe cases can lead to physiological discomfort such as "vertigo" and "nausea". At this stage, the common method for approximating object modelling is LOD, which can effectively improve the refresh rate of 3D scenes [30,31]. The basic idea of LOD is to reduce the geometric complexity of the scene by simplifying the surface details of the scene one by one without affecting the quality of the image, thus improving the efficiency of the drawing algorithm.

2.2. **Process design.** In order to achieve the optimisation of complex plant landscape in 3D scenes, this paper had to meet the functional requirements described above. Firstly, the terrain is constructed using the HeightMap generation method based on grey-scale maps. Secondly, the terrain is used as the background to construct the types and distribution of plants in the plant scene according to the density distribution range and plant characteristics input by the user. Finally, the process of generating natural plant landscape using the proposed optimisation method is shown in Figure 1.

3. **Plant parameterisation based on stochastic 3D L-systems.**

3.1. **Stochastic 3D L-system.** The distinctive feature of the planted landscape is the distribution of trees on the ground. Other plants are almost identical to trees in their generation methods. Therefore, in this paper, the generation of individual plants is mainly based on trees. Based on the plant structure characteristics of the tree, the tree is divided into two parts for simulation: the trunk and the leaves. In the process of generating trees using the stochastic L system, Bezier curves are used as control curves for axial deformation to construct the tree and the leaf surface. The texture image is then embedded into the local coordinates of the Bezier curve and texture drawing is used to realise the drawing of the tree body, resulting in a naturalistic and realistic tree and foliage.

The branching structure (tree structure) is the most critical step in the simulation of plant morphological structures. The general plant meristematic structure can be generated using the iterative process of L-systems. However, as plant morphology is also subject to many environmental factors, a stochastic parametric stochastic 3D L-system model is used in this paper [32, 33].

$$G = \langle V, w, P, \pi \rangle \tag{1}$$

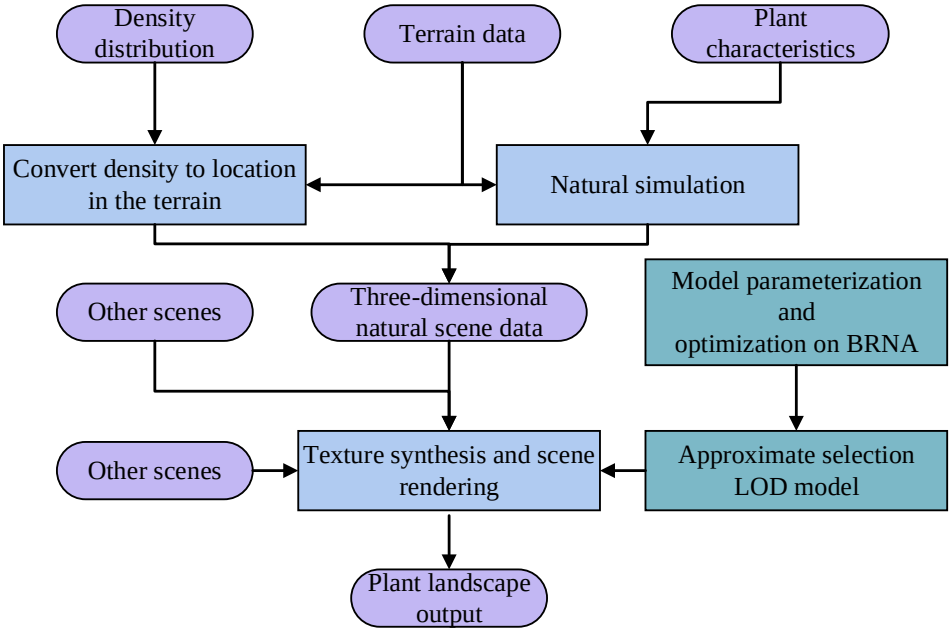


Figure 1. Process of natural plant landscape generation

where V is the basic set of characters with parameters, w is a constant, P is the set of generative rules (the set of branching models) and π is a function.

A generative representation consists of a precursor, a condition and a successor. For example, a generative representation with precursor $A(t)$, condition $t > 5$ and successor $B(t+1)CD(t+2)$ is shown as follow:

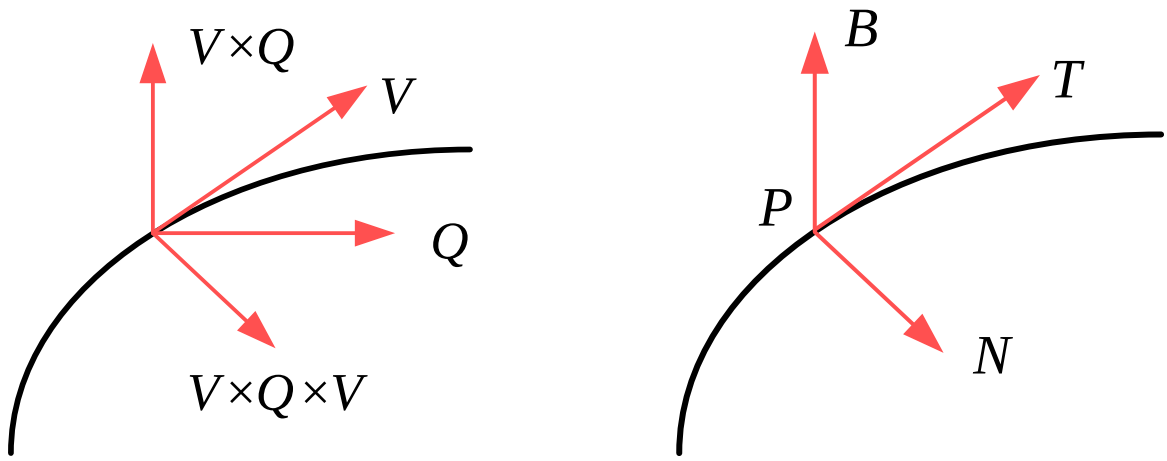
$$A(t) : t > 5 \rightarrow B(t+1)CD(t+2) \tag{2}$$

A generating formula matches the parameterised characters in a branching pattern if it satisfies the following conditions [34]: (1) the characters in the branching pattern and the precursors of the generating formula are the same; (2) the number of actual parameters in the branching pattern is equal to the number of parameters in the precursors of the generating formula. The matching generating formula can be applied to the branching pattern to create a new string using the successors in the generating formula. If there are no matching generators in the set of generating formula P, the branching pattern remains unchanged. The character set of the L-system includes symbols representing different curved surfaces.

3.2. **Construction methods based on Bezier parameter curves.** Axial Deformation (AxDf) is a method for controlling the free deformation of an object by a parametric curve [35]. In the AxDf method, the user first defines a parametric curve. This parametric curve can be located either inside or outside the object. The points on the object are embedded in the local coordinate system of the corresponding points on the parametric curve according to the rule of nearest points. When the user edits the control curve, the object is deformed accordingly. In this paper, the Bezier parametric curve is used as the deformation control curve. The curve is first adaptively dissected according to the change in curve tangency and a minimum rotation Frenet frame is created at the discrete points. The object to be deformed is then embedded in these local coordinate systems.

The curvature of the Bezier parameter curve and the Frenet frame are shown in Figure 2. The position of the point P on the curve can be obtained from its parameter t.

$$p = At^3 + Bt^2 + Ct + D \tag{3}$$



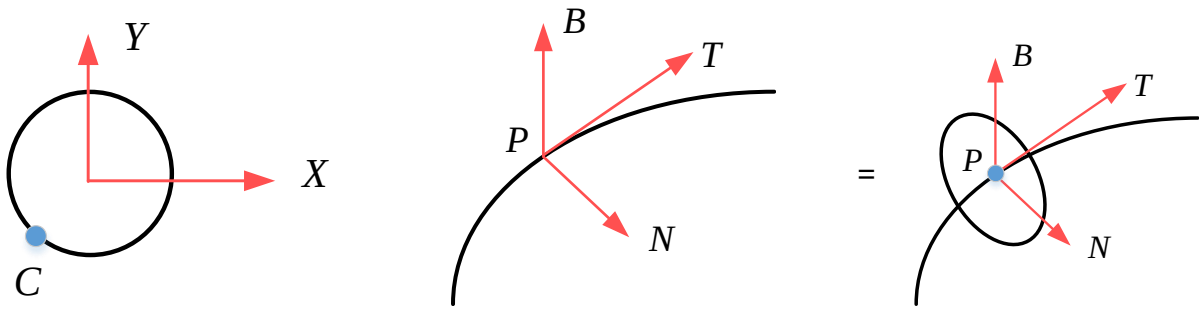
(a) Curvature (b) Frenet frame

Figure 2. Example of a Bezier parameter curve

The Frenet frame consists of a unit tangent vector T, a central axis principal normal vector N and a secondary normal vector B [36]. The unit velocity vector V is calculated as shown as follow:

$$V = 3At^2 + 2Bt + C \tag{4}$$

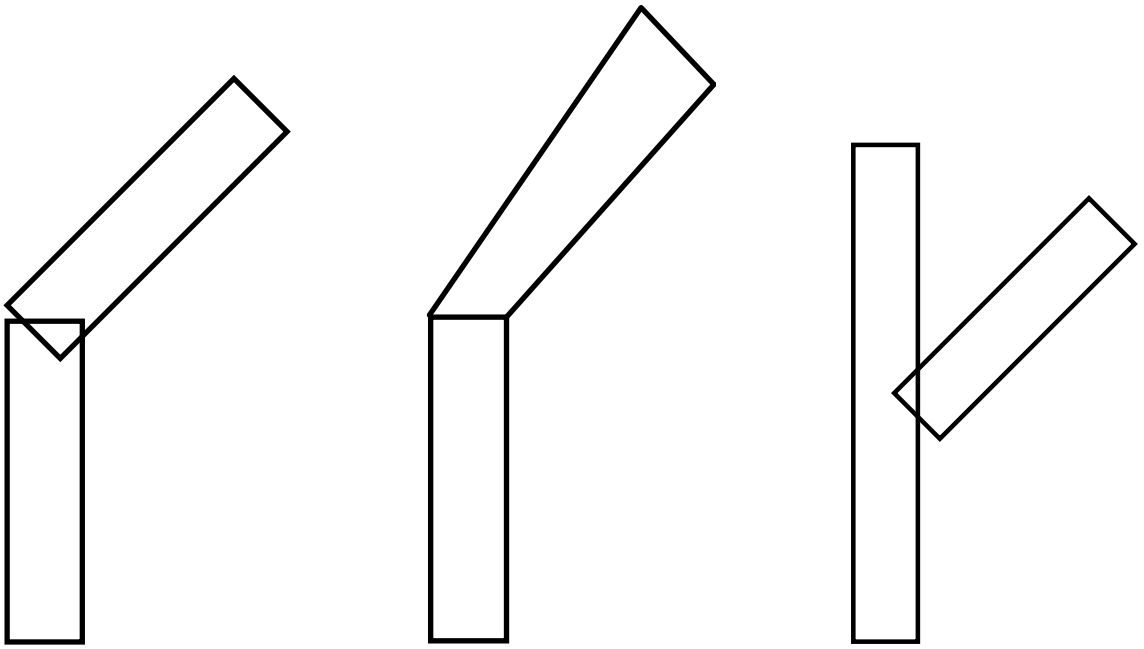
Although the tangent vector T was used in the calculation of the new frame, only three variables were used in converting a cross section to coordinates, as shown in Figure 3.



(a) Variable 1 (b) Variable 2 (c) Variable 3

Figure 3. Transformation of the cross-section

Traditional methods tend to use the following three approaches to generate geometric depictions of tree branches, as in Figure 4: one, simply stacking two column faces together; second, shrinking the child branches and joining them to the parent; and third, embedding the child branches into the parent at the joints.The trees produced by these three methods are often prone to continuity problems at the joint, which is very abrupt visually.In order to achieve natural branch connections, this paper uses Bezier parameter curves to control the branching trend.



(a) Stacked branches (b) Linked branche (c) Dominating branches

Figure 4. Branch geometry

In order to be able to display the generated trees correctly on the computer, a reference plane needs to be selected to realise the projection of the scene. In 3D modelling, the position of each vertex is associated with a reference coordinate system [37-38]. This reference system is called the object coordinates. In 3D rendering, the object coordinates are related to the position and orientation of the camera.

$$f(\bar{P}) = M \cdot \bar{P} \tag{5}$$

where $\bar{P}$ is a vertex and M is the model viewpoint matrix.

3.3. **Texture drawing of foliar textures to parametric curves.** The leaves of plants are difficult to represent as polygons due to their varied morphology. Therefore, in this paper, the midvein of a leaf is used as the axis to fit the flat texture image of the leaf. Firstly, the midvein of the leaf is expressed as a Bezier parametric curve and deformed spatially. The texture image is then embedded in the local frame of the deformed leaf vein curve to create a realistic leaf. The leaves are drawn using OpenGL's alpha checks.

4. Optimization of 3D geometric model based on bidirectional cyclic network auto-encoder.

4.1. **Model representation analysis.** At present, there are two broad types of representation for 3D geometric models: (1) surface representation and (2) body representation, both of which have certain advantages and disadvantages. The body representation shows the full range of surface and internal properties of the object, but takes up more storage space and is therefore more expensive to compute. The surface representation only shows the surface properties of the model, which takes up much less space. This is why surface representation has become a popular method of representing 3D geometric models at this stage.

Among the faceted representations, the triangular mesh in computer graphics is one of the most commonly used representations and can directly represent most complex mesh models. Therefore, in this paper, triangular meshes are chosen as the research content

to achieve model representation and optimisation. Taking the triangular mesh model as an example, the schematic representation of the mesh model is shown in Figure 5. $v_1, v_2, ..., v_6$ denotes each topological vertex which containing spatial coordinate information. $f_1, f_2, ..., f_6$ denotes the number of the triangle.

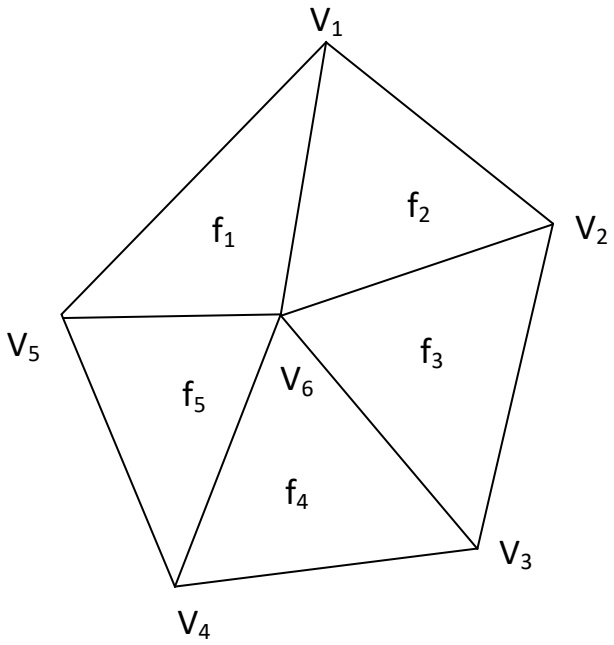


Figure 5. Schematic representation of the grid model

4.2. **Half-folding operation.** A widely used algorithm for simplifying 3D models, the essence of the edge folding algorithm is to delete an edge of a triangle. Performing a single fold operation on an edge achieves the merging of two vertices, thereby eliminating two triangles, as shown in Figure 6.

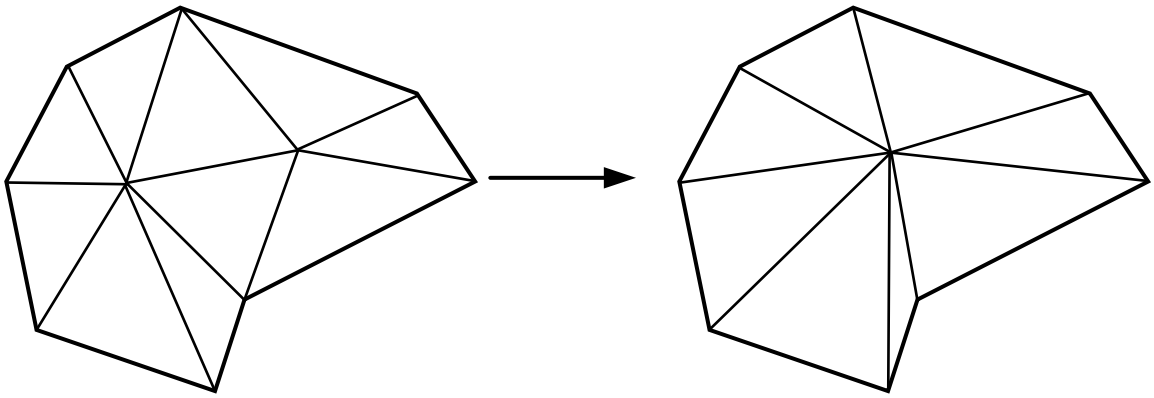


Figure 6. Edge folding operation

The edge folding algorithm based on a quadratic error metric requires less memory and is faster to compute. If any two vertices in the 3D model (p_i, p_j) satisfy any of the following conditions: (1) (p_1, p_2) is an edge; (2) $\|p_1, p_2\| < t$, t represents a threshold value, then (p_1, p_2) is a valid edge. We need to calculate the error metric from a vertex q to a plane in the vicinity of a valid edge (p_i, p_j) on the 3D model.

$$\Delta(q) = \sum_{l \in l(i,j)} (l^T q)^2 \tag{6}$$

where $l(i,j)$ denotes the set of triangular planes connected to the valid side (p_i, p_j). a, b, c, d is the coefficient.

The smaller the value of $\Delta(q)$, the closer the vertex q is to a nearby plane.

$$\Delta(q) = \sum_{l \in l(i,j)} \left(l^T q\right)^2 = q^T \left(\sum_{l \in l(i,j)} K_l \right) q \tag{7}$$

$$M = \sum_{l \in l(i,j)} K_l = \sum_{l \in l(i,j)} \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{12} & m_{21} & m_{23} & m_{24} \\ m_{13} & m_{23} & m_{33} & m_{34} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix} \tag{8}$$

where M is the error matrix for edge folding. The partial derivative of $\Delta(q)$ is calculated according to the least squares method.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{12} & m_{21} & m_{23} & m_{24} \\ m_{13} & m_{23} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{9}$$

The coordinates of the new vertex q are obtained by computing a unique solution to the above equation. If there is no unique solution, otherwise the midpoint of the valid edge (p_i, p_j) is chosen as the new vertex generated by the edge folding algorithm.

The above analysis shows that performing model simplification using the traditional edge folding algorithm may result in multiple LOD models with consecutive transitions, which can affect the speed of model simplification. Therefore, in this paper, half-edge folding is used to reduce the complexity of the simplified model. The edge folding algorithm obtains the new vertex coordinates by a weighted average of the original vertices. However, the half-edge folding algorithm does not require this weighted averaging process and only samples the original vertices, thus reducing the computational overhead and the amount of memory occupied. The half-folding algorithm can be used to improve the rendering speed of the plant landscape model.

The cost function in the half-folding algorithm is calculated as shown as follow.

$$ct(e_{ij}) = \sum_{t \in (T_{v_i} - T_{e_{ij}})} p_t \times IMp_{v_i} \tag{10}$$

where $T_{v_i} - T_{e_{ij}}$ is the set of triangles containing v_i. IMp_{v_i} denotes the visual importance of the vertex v_i.

$$IMp_{v_i} = 1 = \|k_{v_i}\| \tag{11}$$

where k_{v_i} is the visual feature factor.

In addition, this paper introduces a threshold value for the half-folding cost function.

$$ct(e_{ij}) = \sum_{t \in (T_{v_i} - T_{e_{ij}})} p_t \times IMp_{v_i} + \phi_{v_i} \tag{12}$$

Using this threshold adjustment function, the accumulation of errors can be reduced, thus maintaining the important visual features of the model and avoiding a degraded user experience.

4.3. **Model optimization based on BRNA.** The structure of BRNA includes two parts: encoder and decoder, as shown in Figure 7. The encoder transforms the input sequence into a hidden state with a fixed dimension, while the decoder remaps the hidden state into a sequence that matches the original input.

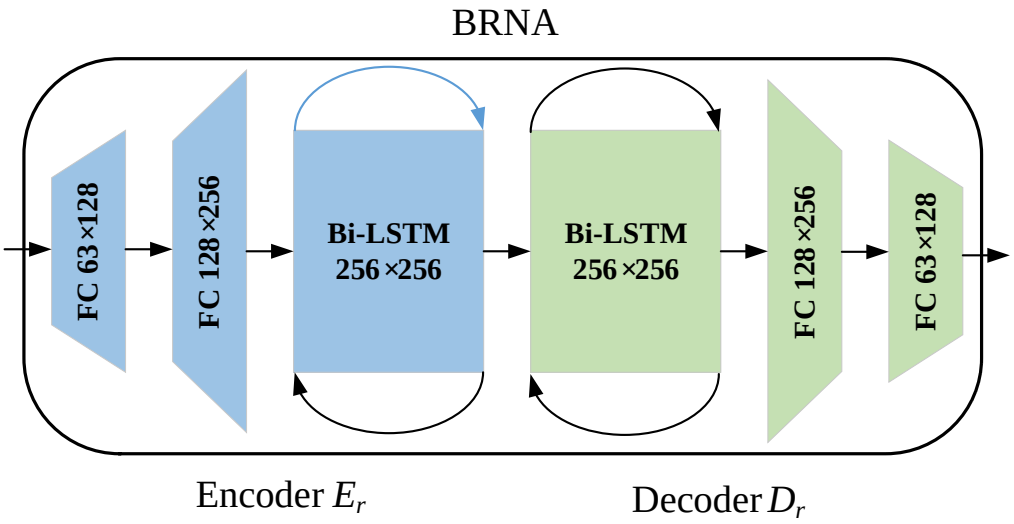


Figure 7. Structure of bidirectional cyclic network auto-encoder

In the training process, the encoder captures the key information of the input sequence by compressing it into a low-dimensional hidden representation. The decoder helps the model to learn the feature representation and generation ability of the input sequence by recovering the reconstruction of the input sequence from the hidden state. Different from the traditional auto-encoder, the bidirectional cyclic network auto-encoder also introduces cyclic connection, which enables the model to model the time series relationship of sequence data.

Encoder E_r consists of two fully connected layers and a bi-directional long-term and short-term memory network Bi-LSTM. The structure of decoder D_r is symmetrical with that of encoder E_r. The function of the encoder is to map the input data to the hidden variable space, and the function of the decoder is to map the hidden variable space data back to the motion sequence. Encoder and decoder can be shown as follow:

$$E_r(X_N) = \mathrm{BiLSTM}_e(\mathrm{FC}_{e2}(\mathrm{FC}_{e1}(X_N))) \tag{13}$$

$$D_r(H_r) = \mathrm{BiLSTM}_d(\mathrm{FC}_{d2}(\mathrm{FC}_{d1}(H_r))) \tag{14}$$

where X_N represents the input noise data, H_r represents the hidden variable, and FC represents the fully connected layer.

The data in the training process is the coordinate data of the middle node of the three-dimensional geometric model. The mean square error is used to calculate the minimum Euclidean distance of data before and after optimization. Therefore, the node position loss function of the three-dimensional geometric model is:

$$L_{\text{position}} = \frac{1}{f \times d}\|Y - X_c\|_2 \tag{15}$$

where f represents the number of frames in the input sequence and d represents the dimension of the data.

The inter-frame smoothness constraint is used to ensure the coordinates of the optimized three-dimensional geometric model are more stable.

$$\mathbf{O} = \begin{pmatrix} -1 & 1 & 0 & & \\ 1 & -2 & 1 & & \\ & & \ddots & & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}_{(j+2)\times(j+2)} \tag{16}$$

5. **Approximate choice of LOD model.** Often, the number of leaves on trees is very large. In order to increase the drawing speed of plant landscape, the LOD technique can be used for foliage that is far from the viewpoint and requires low detail [39-40]. The LOD technique is widely used in dynamic scenes due to its ability to greatly increase the speed of real-time drawing. The LOD technique reduces the geometric complexity of the scene by simplifying the surface details of the scene one by one without affecting the visual effect of the scene, thus increasing the efficiency of the drawing algorithm. the LOD technique can create several geometric models with different degrees of approximation for the original polyhedral model. Each model retains a certain level of detail compared to the original model. When the object is viewed from close up, a fine model can be used. When the object is viewed from a distance, a coarser model is used. The speed of image generation using LOD technology can be increased significantly.

However, when the point of view changes continuously, a noticeable dithering occurs between two different levels of the model. It is therefore necessary to create smooth visual transitions, i.e. geometric transitions, between models at adjacent levels. A suitable LOD model is generally selected according to the distance from the viewpoint to the centre of the scene area d. However, a transition from one LOD to another based solely on the distance from d can easily cause large deformations in the mesh model of the geometric scene, resulting in very violent jitter. This jitter will now significantly reduce the user experience in the system and in severe cases can lead to physical discomfort such as 'dizziness' and 'nausea'.

Instead of using the most compact LOD model, this paper uses an approximate selection of results, thus taking full advantage of the GPU's on-graphics processing power. The projection of the height difference μ_i on the screen ϵ_i is shown in Figure 8.

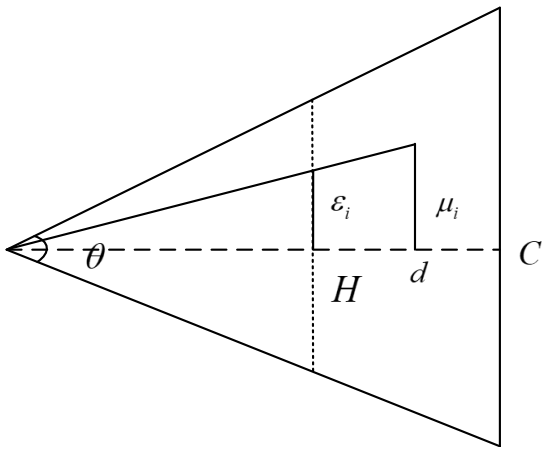


Figure 8. Principle of calculation of projection values

The distance d is calculated as shown as follow:

$$d = \frac{(C - V) \cdot D}{|D|} \tag{17}$$

where V is the viewpoint, D is the viewing direction and the centre point of the scene area block is C.

The projection value ϵ_i on the screen is calculated as shown as follow.

$$\epsilon_i = \frac{H \cdot \mu_i}{2d \cdot \tan(\frac{\theta}{2})} \tag{18}$$

where H is the viewpoint height and μ_i is the height difference.

In the pre-processing phase of LOD selection for complex scenes, the projection values are calculated to select the appropriate LOD parallax metric for each frame of the scene area at δ_i, thus speeding up the calculation. Set λ as a threshold value (generally 4 pixels). If ϵ_i is greater than λ, this will cause significant visual errors. Switching scenes at this point will cause perceptible jitter to the human eye, so ϵ_i must be less than λ.

$$\delta_i = \frac{H \cdot \mu_i}{2\lambda \cdot \tan(\frac{\theta}{2})} \tag{19}$$

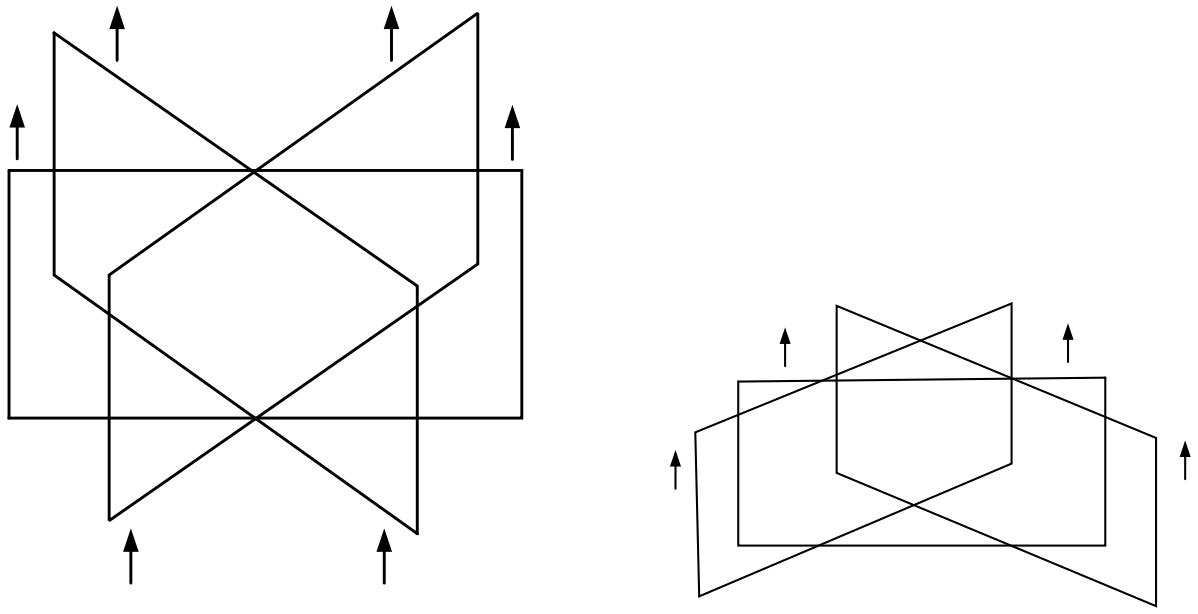
The most important issue when drawing LOD models is how to set a criterion that allows the system to automatically select the appropriate level of detail [41]. Existing criteria mainly consider the maintenance of a stable frame rate. However, the criteria for visual equivalence are not the same in many scenes. Therefore, this paper uses the projected area of sampled points on the screen as a criterion for visual equivalence to ensure smooth transitions between adjacent layers. A sampled point on any layer represents a piece of sampled area. The area of the sampling point $\text{Area}_{\text{orgin}}$ is determined when the LOD is created.

$$Area_{\text{orgin}} <= Area_{\text{orgin}} \cdot \left(\frac{z_{\text{near}} \cdot \max(vw, vh)}{z} \right)^2 \tag{20}$$

where vw and vh are the horizontal and vertical resolution of the screen respectively, z_{near} is the distance between the viewpoint and the plane, and z is the depth of the sampled point in space.

In the approximate choice of LOD model, each point is a circle from the normal direction. Each point can therefore be represented by the position vector, the normal vector and the radius of the point. To speed up the drawing, each point is given a new colour vector as a scattering property of the light. As there is no need to store any topological information between points in the model, the amount of storage is greatly reduced. Scenes can be drawn using points provided that the surface of the object is sampled at a sufficient density. This condition is such that the shape of each polygon is negligible when projecting the points onto the image plane.

In order to ensure that the visual quality of a planted landscape display is not constrained by the line of sight, polygons must be made to produce a crossover effect. The results of several experimental tests have shown that a better effect can be created by using a star structure. Figure 9 gives two possible variants, which consist of three intersecting squares. To obtain a suitable illumination, the normal direction of all the vertices can be made parallel to the perpendiculars of the polygon.



(a) Variant 1 (b) Variant 2

Figure 9. Leaf model

Setting these individuals in a large area using the alpha blending method results in a natural and lush image of the leaves or grass. As the Bezier curve can be determined by the position and tangent vector of the endpoints, for leaf textures, different control points can be created at the two endpoints based on the midvein of each leaf as the axis. This method enables the leaf textures in the initial plane to produce their own surface space, resulting in a three-dimensional effect with different curves.

6. **Experiments and analysis of results.**

6.1. **Performance evaluation of outputs.** In order to verify the effectiveness of the proposed rapid planting method, a set of plant scenes from modern urban landscape design is drawn in this paper. The experimental hardware was a Dell T5820 graphics workstation with a W-2255 CPU (3.7 GHz), 256G RAM, 4T hard drive and RTX4000-8G graphics card. The software platform is MS Visual C++ 2003.NET and Open GL2.0.

The output of the plant landscape drawing is shown in Figure 10. Firstly, before the half-folding algorithm and approximate selection of the LOD model is used, the system output was OUTPUT A. In OUTPUT A, the scene output was often jittery due to changes in viewpoint complexity when the wind blew the leaves or when roaming through the scene.

When the output accuracy is set to the maximum, the jitter is more pronounced and the user is prone to physiological discomfort such as "dizziness" and "nausea". This is because traditional LOD techniques can only set a low loss threshold in order to maintain a high image quality. However, lower loss thresholds result in a significant jitter between two models with different levels when the viewpoint changes continuously, which can seriously affect the speed of image generation. In this case, the plant landscape can only be output in a low precision mode in order to ensure smoothness. The performance of OUTPUT A is shown in Table 1.

Then, after using the proposed method, the system output is OUTPUT B. As can be seen from the drawing results of the OUTPUT B, the single tree models were generated

Figure 10. Output of the plant landscape drawing

Table 1. Performance of OUTPUT A

The quality of the picture	**Average frame rate (fps)**	**Visual Effects**
High precision	15.9682	Occasional flicker
Medium precision	21.4667	More stable
Low precision	44.4851	Good stability

in less than 10 ms. The local close-ups of the tree trunk sections show more detail due to the relatively close view. the performance of OUTPUT B is shown in Table 2. The average frame rate of the system output exceeded 40 fps for all three accuracy settings, resulting in a smooth and intuitive visual experience.

Table 2. Performance of OUTPUT B

The quality of the picture	**Average frame rate (fps)**	**Visual Effects**
High precision	43.3649	Good stability
Medium precision	48.3306	Good stability
Low accuracy	56.2197	Very smooth

6.2. **Performance evaluation of different algorithms.** The average frame rate variation curves for the proposed method and the SR-LOD method [42] are given in Figure 11 when the user is roaming through a modern urban landscape (containing a large number of planted landscapes). A comparison of the number of triangles produced by the two methods is given in Figure 12.

It can be seen that the proposed method is able to maintain a high average frame rate at all times, ensuring real-time performance and user experience when drawing and interacting with the scene. The number of triangles generated by the proposed method increases due to the use of half-folding, i.e. the time complexity increases. However, due to the improved pre-processing mechanism for LOD model selection, the total computation time is less than that of the SR-LOD method. In other words, the proposed method is

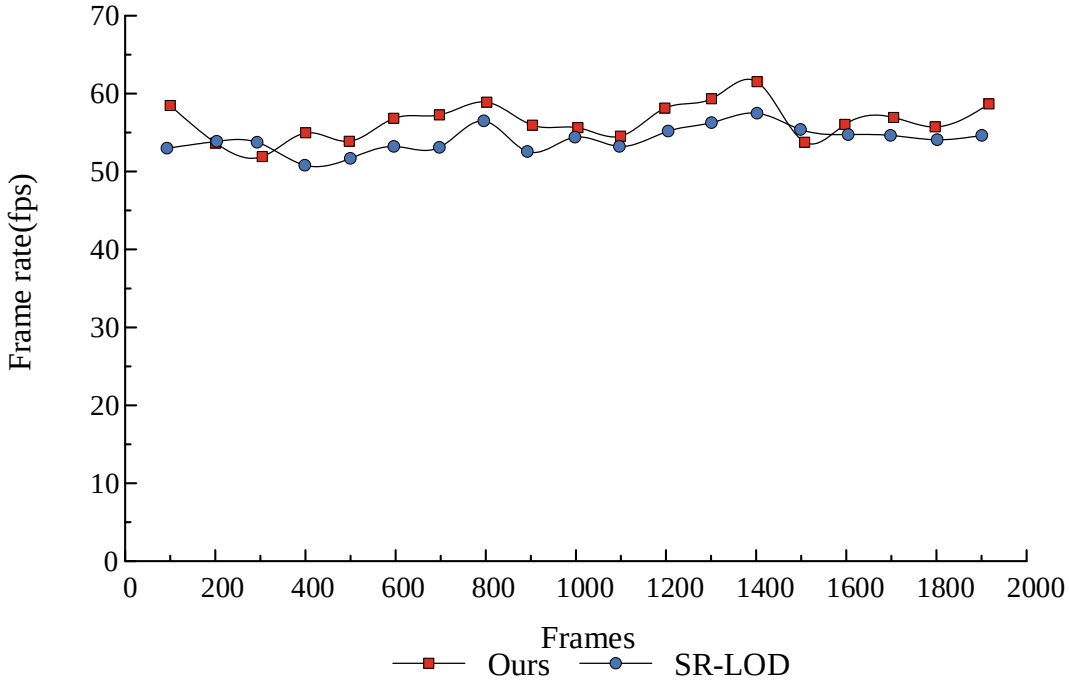


Figure 11. Change curve of frame rate

able to achieve the drawing of complex plant landscapes relatively quickly, as shown in Table 3.

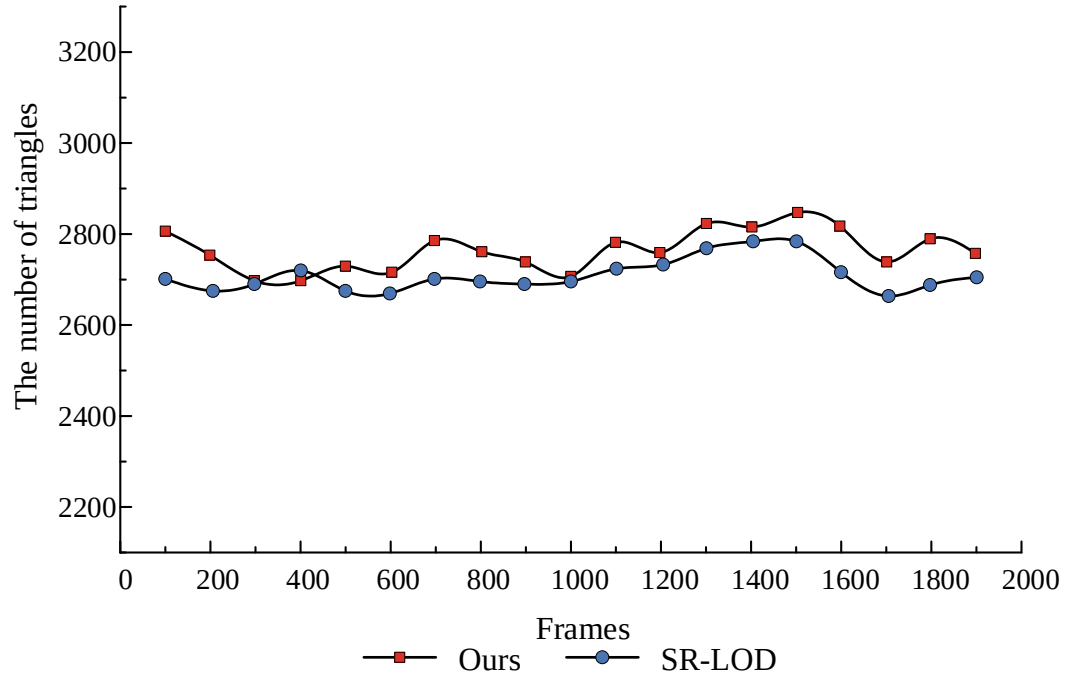


Figure 12. Number of triangles

Table 3. Execution time comparison

	SR-LOD(s)	**Ours(s)**
Initial time	6.046	4.892
Simplify time	25.816	26.253
Total time	31.862	31.145

7. **Conclusion.** In order to realize the rapid construction and drawing of complex plant landscaping while maintaining the picture quality, a rapid plant landscaping generation method is proposed. The optimization is mainly carried out from two aspects, including three-dimensional geometric model optimization and approximate selection of LOD model. In this paper, based on the half-edge folding algorithm, the parameters of the low-precision three-dimensional geometric model are optimized by using the bidirectional cyclic network auto-encoder, so as to improve the fidelity and reduce the computational overhead. By defining the node position loss function of the three-dimensional geometric model, the smooth transition between adjacent layers is ensured. Instead of using the simplest LOD model, approximate selection results are used, thus giving full play to the graphics processing ability of GPU. In order to ensure that the visual quality displayed in plant landscaping is not restricted by the line of sight, a star structure is used to produce a cross effect. The experimental results show that the average frame rate of the output of the proposed method exceeds 40 fps, thus obtaining a smooth visual experience. Compared with other similar methods, the proposed method can realize the drawing of complex plant landscaping relatively quickly.

Funding Statement. This work supported by project of "Youth Project of Shanghai Philosophy and Social Science Planning (No. 2019ECK010)".

REFERENCES

[1] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[2] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[3] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[4] D. Gantar, "Scenario use for fostered adaptation to the future landscape changes," *Acta agriculturae Slovenica*, vol. 93, no. 1, pp. 16-22,2009.

[5] S. K. S. O. Thani, N. H. N. Mohamad, and S. Idilfitri, "Modification of Urban Temperature in Hot-Humid Climate Through Landscape Design Approach: A Review," *Procedia - Social and Behavioral Sciences*, vol. 68, pp. 439-450,2012.

[6] P. Xu, K. Wang, M. M. Hassan, C.-M. Chen, W. Lin, M. R. Hassan, and G. Fortino, "Adversarial Robustness in Graph-Based Neural Architecture Search for Edge AI Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8465-8474, 2023.

[7] K. Wang, Z. Chen, M. Zhu, S.-M. Yiu, C.-M. Chen, M. M. Hassan, S. Izzo, and G. Fortino, "Statistics-Physics-Based Interpretation of the Classification Reliability of Convolutional Neural Networks in Industrial Automation Domain," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2165-2172, 2023.

[8] H. Yoshimura, H. Zhu, Y. Wu, and R. Ma, "Spectral properties of plant leaves pertaining to urban landscape design of broad-spectrum solar ultraviolet radiation reduction," *International Journal of Biometeorology*, vol. 54, no. 2, pp. 179-191, 2009.

[9] W. Aiqing, W. Xitong, and H. Fengwu, "Research on the Digital Multimedia Technology and Urban Landscape Design based on Multimedia Art Perspective," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 10, pp. 371-380, 2016.
[10] B. Alizadeh, and J. Hitchmough, "A review of urban landscape adaptation to the challenge of climate change," *International Journal of Climate Change Strategies and Management*, vol. 11, no. 2, pp. 178-194, 2019.
[11] N. Van Long, and Y. Cheng, "Urban Landscape Design Adaption to Flood Risk: A Case Study in Can Tho City, Vietnam," *Environment and Urbanization ASIA*, vol. 9, no. 2, pp. 138-157, 2018.
[12] S. Koma, Y. Yamabe, and A. Tani, "Research on urban landscape design using the interactive genetic algorithm and 3D images," *Visualization in Engineering*, vol. 5, no. 1, pp. 17-22, 2017.
[13] X. Shi, Y. Zhu, J. Duan, R. Shao, and J. Wang, "Assessment of pedestrian wind environment in urban planning design," *Landscape and Urban Planning*, vol. 140, pp. 17-28, 2015.
[14] F. Xue, Z. Gou, and S. Lau, "The green open space development model and associated use behaviors in dense urban settings: Lessons from Hong Kong and Singapore," *Urban Design International*, vol. 22, pp. 287-302, 2017.
[15] S. Sun, X. Xu, Z. Lao, W. Liu, Z. Li, E. Higueras García, L. He, and J. Zhu, "Evaluating the impact of urban green space and landscape design parameters on thermal comfort in hot summer by numerical simulation," *Building and Environment*, vol. 123, pp. 277-288, 2017.
[16] A. T. Polat, and A. Akay, "Relationships between the visual preferences of urban recreation area users and various landscape design elements," *Urban Forestry & Urban Greening*, vol. 14, no. 3, pp. 573-582, 2015.
[17] H. Yan, S. Fan, C. Guo, F. Wu, N. Zhang, and L. Dong, "Assessing the effects of landscape design parameters on intra-urban air temperature variability: The case of Beijing, China," *Building and Environment*, vol. 76, pp. 44-53, 2014.
[18] V. E. Cooley, and J. Shen, "Factors Influencing Applying for Urban Principalship," *Education and Urban Society*, vol. 32, no. 4, pp. 443-454, 2000.
[19] J. Jiao, H. Liu, and N. Zhang, "Research on the Urban Landscape Design based on Digital Multimedia Technology and Virtual Simulation," *International Journal of Smart Home*, vol. 10, no. 9, pp. 133-144, 2016.
[20] F. M. Shahli, M. R. M. Hussain, I. Tukiman, and N. Zaidin, "The Importance Aspects of Landscape Design on Housing Development in Urban Areas," *APCBEE Procedia*, vol. 10, pp. 311-315, 2014.
[21] X. Fan, B. Zhou, and H. H. Wang, "Urban Landscape Ecological Design and Stereo Vision Based on 3D Mesh Simplification Algorithm and Artificial Intelligence," *Neural Processing Letters*, vol. 53, no. 4, pp. 2421-2437, 2021.
[22] X. Xu, S. Sun, W. Liu, E. H. García, L. He, Q. Cai, S. Xu, J. Wang, and J. Zhu, "The cooling and energy saving effect of landscape design parameters of urban park in summer: A case of Beijing, China," *Energy and Buildings*, vol. 149, pp. 91-100, 2017.
[23] I. Stavi, "On-Site Use of Plant Litter and Yard Waste as Mulch in Gardening and Landscaping Systems," *Sustainability*, vol. 12, no. 18, pp. 7521, 2020.
[24] J. Goodness, "Urban landscaping choices and people's selection of plant traits in Cape Town, South Africa," *Environmental Science & Policy*, vol. 85, pp. 182-192, 2018.
[25] M. N. Peterson, B. Thurmond, M. McHale, S. Rodriguez, H. D. Bondell, and M. Cook, "Predicting native plant landscaping preferences in urban areas," *Sustainable Cities and Society*, vol. 5, pp. 70-76, 2012.
[26] J. Padullés Cubino, M. L. Avolio, M. M. Wheeler, K. L. Larson, S. E. Hobbie, J. Cavender-Bares, S. J. Hall, K. C. Nelson, T. L. E. Trammell, C. Neill, D. E. Pataki, J. M. Grove, and P. M. Groffman, "Linking yard plant diversity to homeowners' landscaping priorities across the U.S," *Landscape and Urban Planning*, vol. 196, 103730, 2020.
[27] J.-Y. Lin, Y.-S. Tsay, and P.-C. Tseng, "Development of Folded Expanded Metal Mesh with Sound Absorption Performance," *Applied Sciences*, vol. 11, no. 15, 7021, 2021.
[28] G. Zhou, S. Yuan, and S. Luo, "Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse," *IEEE Access*, vol. 8, pp. 196341-196350, 2020.
[29] J. Liu, Z. Xie, C. Zhang, and G. Shi, "A novel method for Mandarin speech synthesis by inserting prosodic structure prediction into Tacotron2," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 10, pp. 2809-2823, 2021.
[30] F. Leite, A. Akcamete, B. Akinci, G. Atasoy, and S. Kiziltas, "Analysis of modeling effort and impact of different levels of detail in building information models," *Automation in Construction*, vol. 20, no. 5, pp. 601-609, 2011.

[31] J. Abualdenien, and A. Borrmann, "Levels of detail, development, definition, and information need: a critical literature review," *Journal of Information Technology in Construction*, vol. 27, pp. 363-392, 2022.

[32] K. Ohori, H. Ledoux, F. Biljecki, and J. Stoter, "Modeling a 3D City Model and Its Levels of Detail as a True 4D Model," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1055-1075, 2015.

[33] S. Lafia, W. Kuhn, K. Caylor, and L. Hemphill, "Mapping research topics at multiple levels of detail," *Patterns*, vol. 2, no. 3, pp. 100210, 2021.

[34] J. A. Aguilar, P. Kiraly, R. W. Adams, M. Bonneau, E. J. Grayson, M. Nilsson, A. M. Kenwright, and G. A. Morris, "Ultra-high dispersion NMR reveals new levels of detail," *RSC Advances*, vol. 5, no. 65, pp. 52902-52906, 2015.

[35] E. C. Moreno-Valle, W. Pachla, M. Kulczyk, B. Savoini, M. A. Monge, C. Ballesteros, and I. Sabirov, "Anisotropy of uni-axial and bi-axial deformation behavior of pure Titanium after hydrostatic extrusion," *Materials Science and Engineering: A*, vol. 588, pp. 7-13, 2013.

[36] S. Demouchy, A. Tommasi, T. Boffa Ballaran, and P. Cordier, "Low strength of Earth's uppermost mantle inferred from tri-axial deformation experiments on dry olivine crystals," *Physics of the Earth and Planetary Interiors*, vol. 220, pp. 37-49, 2013.

[37] S. Baydas, and B. Karakas, "Defining a curve as a Bezier curve," *Journal of Taibah University for Science*, vol. 13, no. 1, pp. 522-528, 2019.

[38] A. Tharwat, M. Elhoseny, A. E. Hassanien, T. Gabel, and A. Kumar, "Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm," *Cluster Computing*, vol. 22, no. S2, pp. 4745-4766, 2018.

[39] L. Tang, L. Li, S. Ying, and Y. Lei, "A Full Level-of-Detail Specification for 3D Building Models Combining Indoor and Outdoor Scenes," *ISPRS International Journal of Geo-Information*, vol. 7, no. 11, 419, 2018.

[40] K. Graham, L. Chow, and S. Fai, "From BIM to VR: defining a level of detail to guide virtual reality narratives," *Journal of Information Technology in Construction*, vol. 24, pp. 553-568, 2019.

[41] S. Kwon, B. C. Kim, D. Mun, and S. Han, "User-assisted integrated method for controlling level of detail of large-scale B-rep assembly models," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 9, pp. 881-892, 2018.

[42] F. Biljecki, H. Ledoux, J. Stoter, and J. Zhao, "Formalisation of the level of detail in 3D city modelling," *Computers, Environment and Urban Systems*, vol. 48, pp. 1-15, 2014.