Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 1, 2023

A Spline Based Parametric Modeling Method for Ladder like Facilities

Qun Zhu *, Xiaoxi He, Yinghu Liu, Jiaru Li, Fanlin Meng

School of Software Engineering, Chengdu University of Information Technology, Chengdu, Sichuan, 610225, China

*** Corresponding author:** Qun Zhu (Email: 18783175114@163.com)

Abstract: Existing modeling methods for modeling ladder like facilities are often limited to a single technical means and require a large amount of user input or guidance. To address the above issues, we propose a spline based parametric modeling method for modeling general complex ladder like facilities. Firstly, a facility framework spline generation method was improved to provide an editable initial framework. In addition, the method for calculating the position of contour points is improved, and the surface mesh of the facility is generated by combining contour curve constraint algorithms that meet various dimensions and geometric constraints that control the shape. Finally, a profile mesh generation algorithm was designed to achieve mesh generation of contour surfaces. The experimental results show that this method can be used to model different types of ladders like facilities. Compared with existing methods, facility framework spline generation and component surface mesh generation can significantly improve the efficiency of model generation.

Keywords: Spline; Parametric Modeling; Calculation of Contour Point Position; Contour Curve Constraint.

1. Introduction

Ladder like facilities refer to facilities that have the shape or function of a ladder, including stairs, stairs, climbing ladders, rope ladders, etc., and are composed of multiple parts such as steps, platforms, and railings. In the fields of digital twins and industrial simulation, realistic and detailed scenes are typically characterized, so large-scale scenarios such as ladder like facilities also need to be represented by their appearance. The existing modeling methods are mainly limited to specific types of models or require a large amount of user input or guidance, and the demand for methods that can achieve fast, detailed, and interactive 3D content in these industries is constantly increasing.

Traditional 3D modeling methods typically use modeling software such as 3ds Max and AutoCAD to estimate object contours and heights based on image data, CAD plans, or captured images. This requires the creation and operation of a large number of geometric primitives, as well as the use of various transformations and geometric operations to modify them. The production process of its model requires a large amount of manual participation and professional 3D modeling skills or programming experience, and due to the large number of vertices, the generated model mesh is difficult to re edit. At the same time, the production cycle of the model is long, resulting in low timeliness of the model, which cannot truly meet user needs.

In addition, the 3D modeling method also includes oblique photography technology, which is based on the reverse modeling of real people and objects. It can truly reflect the situation of ground objects, accurately obtain object texture information, and generate a real 3D model. However, oblique photography data is usually stored in OSGB format, which has high requirements for weather and lighting during shooting, low scene precision, and loss of back information; There are numerous and fragmented model files, high complexity of pyramid structure, difficulty in sharing, and difficulties in modifying later models.

This article improves a spline based parametric modeling method for ladder like facilities. This method is based on spline curves that describe the structure of the facility framework, projecting a series of user input 2D contour shapes into 3D space, and ultimately generating a mesh model. The contributions of this article are as follows: 1) Two spline curves, Bezier and B-Spline, were applied to construct facility frameworks. In addition, cylindrical spiral curves and adaptive Bezier curves were also implemented as spiral facility frameworks and handrail frameworks, respectively. 2) Improve the spline modeling method to calculate two types of curve subdivision points based on distance and height, and achieve a geometric array of facility components. 3) Improving the calculation method of contour point position to solve the problem of mesh distortion at inflection points, introducing contour constraint curves to control various dimensions and geometric constraints of input shapes to generate component grids. 4) The algorithm in this article has better generation efficiency than existing mainstream 3D modeling methods, and makes facility frameworks and components easy to edit, enabling the modeling of more types of facilities.

2. Related Work

In the past, modeling ladder like facilities relied on the manual skills of modelers, but later parametric modeling techniques began to dominate, simplifying the modeling phase but requiring an understanding of how object parameters work. In recent years, due to the rapid development of artificial intelligence technology, 3D modeling has also opened up new fields. The existing facility modeling techniques can be divided into four categories: sketch-based methods, parametric modeling, data-driven methods, and spline-based methods.

2.1. Sketch-based Modeling Method

According to Ding's [1] research, using sketches to reconstruct 3D models is intuitive for humans. Nishida et al. [2] proposed a sketch-based model generation method. This method first recognizes and optimizes user sketches based on convolutional neural networks, and then extracts the features and parameters of objects in the sketches to generate a 3D model. Kelly et al. [3] proposed a model generation method that combines sketching and extrusion, but the generated model did not follow the correct structural rules [4] and lacked details. Chen [5] proposed a modeling method based on hand drawn sketches and successfully generated a structurally correct model, but the supported model types are limited. Lopes [6] mentioned multi-story buildings containing stairs and emphasized the importance of setting up stairs. Jonas [7] proposed a program generation algorithm that includes stairwells. This algorithm simply interpolates between the edge lines of stairs to obtain step boundaries, which can achieve several simple types of stairs. Sketch-based methods do not require database support, but for nonprofessional users, drawing accurate facility details is very difficult.

2.2. Parametric Modeling

Huang et al. [8] proposed an HM sketch method to automatically calculate a set of process model parameters, but can only generate 2D/3D shapes similar to the initial input HM sketch. Turrin et al. [9] discussed the benefits of combining parametric modeling with genetic algorithms (GAs) to achieve performance-oriented processes in design. Keshavarzi et al. [10] used hand drawn floor plans and annotations as inputs and outputted a fully parameterized vectorized model in real-time. Unlu et al. [11] proposed the idea of using multiple neural networks to predict different parameter sets that must be correlated. Gumster [12] provides a parameterized modeling tool based on geometric nodes in Blender, providing users with new ways to solve practical problems. However, parameterized entity models correspond to a type of entity, and different modeling systems use incompatible, special, and often internally inconsistent semantics when generating parameterized models.

2.3. A data-driven Modeling Method

Data driven modeling methods are usually based on real data, such as photos or point cloud data from scanning existing objects, to reconstruct surfaces, which is the fastest method for generating 3D models. Guo [13] uses point clouds obtained by scanning real objects as input to the process modeling algorithm to reconstruct a real 3D model. The work of Rutzinger et al. [14] is an example, characterized by detecting objects from point clouds of mobile laser scanners and modeling individual objects. Jeffrey [15] uses mobile robots to locate and model stairs, allowing for multi-level exploration of platforms that can pass through stairs. P é rez [16] provides a new staircase detection and modeling module that can present complete information about stairs. Chan [17] proposed a staircase detection algorithm that can effectively locate and model stairs with different poses. Seungjun [18] proposed a stair mapping method for quadruped robots based on point cloud measurement and stair modeling. The data-driven approach requires a database to provide rich data support, and the generated model does not support further editing.

2.4. Spline based Modeling Method

In computer graphics, splines are commonly used to represent curves. The essence of expressing the coordinates of any position on this line segment is to interpolate between two points, which is the expression of a Bezier curve [19] and a classic method for obtaining a smooth curve. In addition, B-spline [20] is a special representation of Bezier curves, which is flexible, diverse, and computationally efficient. Huang et al. [21] proposed a new interactive procedural modeling method based on splines. This method uses predefined model contours to scan along a spline curve to generate a grid of ancient building models. Huang et al. [22] continued to improve their method. This method allows users to edit splines through two types of control points. Hu et al. [23] adopted a new vertex position calculation method to improve it, allowing it to describe the required component shape, but the prefabricated components in its component library cannot be changed.

3. Research Contents

Frame spline generation and surface mesh generation are the two main modeling processes in this article. In section 2.1, frame splines were generated based on control points, and various curves were also implemented to generate different frame splines. In section 2.2, an improved contour point position calculation method based on spline sweeping was introduced, and contour constraint curves were introduced to constrain the generation direction, size, and combination law of the contour shape, achieving surface modeling. The modeling process is shown in Figure 1.

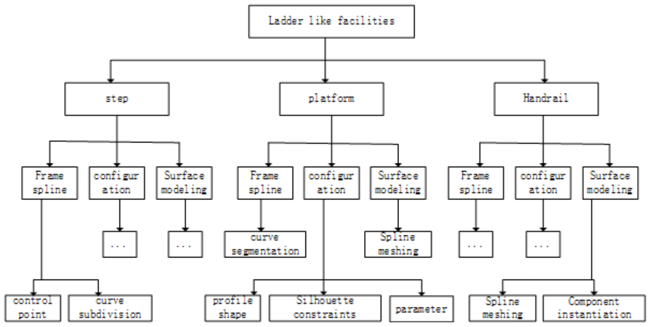


Figure 1. Modeling Process

3.1. Frame Spline Generation

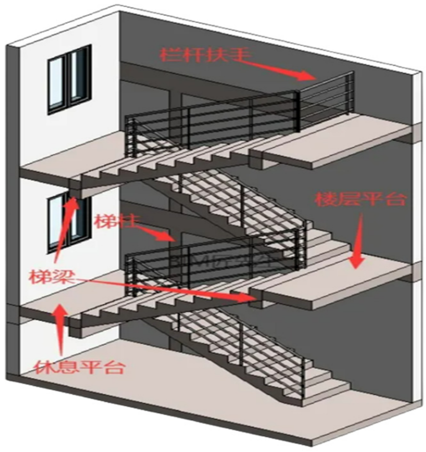


Figure 2. Facility Structure Diagram

The ladder like facilities in this article include stairs, steps, ladders, and rope ladders. Stairs [24] are building components used for vertical transportation between floors, consisting of continuous walking steps, rest platforms, and safety maintenance railings (or boards), handrails, and corresponding supporting structures. Steps [25] are steps set up for people to walk on different levels of outdoor or indoor floors or floors. A ladder is a handrail ladder with a protective cage and can be assembled at will. A rope ladder is a ladder made of ropes on both sides and a ladder made of ropes, wood, or metal. The facility structure consists of three major parts: steps, platforms, and railings. Its structure is shown in Figure 2.

3.1.1. Facility Framework Generation

We have specified a method for generating frame splines through control points, which are used to control the position and curvature of components. This article adopts a segmented cubic Bezier curve, which is intuitively controlled and can effectively describe the shape of the frame.

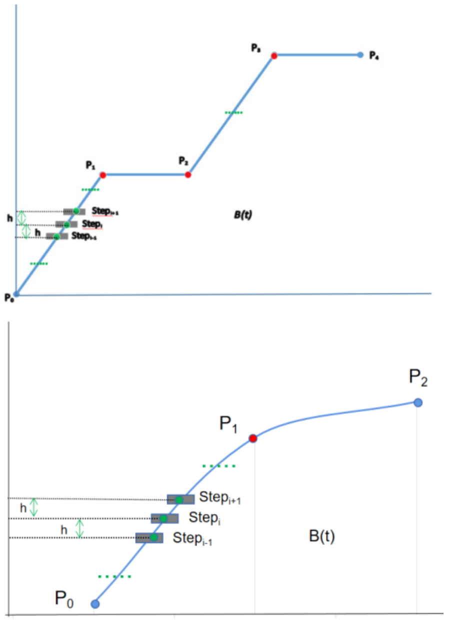


Figure 3. Facility Side Profile

In the left figure of Figure 3, B (t) represents the side profile of the straight double running facility using a single Bezier curve. In the right figure of Figure 3, B (t) represents the side profile of the curved facility using a quadratic Bezier curve, where the blue dot represents the control point and the step+1 represents the (i+1) step.

This article uses the parametric form of a curve, where the parameter Bessel curve at parameter t is defined as:

$$P(\mathrm{t}) = \sum_{i=0}^{n} J_{n,i}(t)B(i) \qquad 0 \le t \le 1 \tag{1}$$

In the equation, n is the degree of curvature,$J_{n,i}(t)$ is the Bernstein polynomial basis function, t is the parameter, and B (i) is the Bessel control point. Usually, the value of t is between 0 and 1. $J_{n,i}(t)$ is the coefficient of the Bessel curve defined as:

$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \tag{2}$$

Due to the fact that curves are generally defined in parametric form rather than explicit form, parameterization is necessary. This process explains how to assign numerical values to each point on the Bezier curve. In this way, each data point will receive a numerical value to describe its position on the Bezier curve.

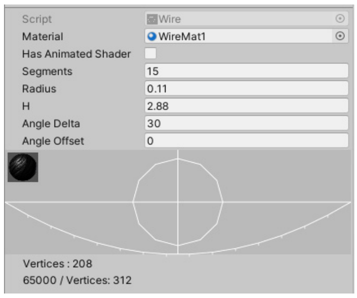
$$t(i) = t_{i-1} + \frac{|x_i - x_{i-1}|}{\sum_{i=1}^{n} |x_i - x_{i-1}|} \qquad i \in (1, \ldots, n) \tag{3}$$

Where t is the parameter value, x is the coordinate of the data points on the x-axis in the three-dimensional space xyz axis, and n is the number of data points. For the y-axis and z-axis, the calculation process is similar.

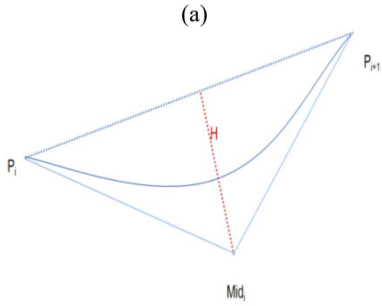
3.1.2. Adaptive Bezier Curve

Based on the control points of the generated frame spline, the handrail frame spline is segmented and fitted using a segmented approximation method, approximating the Bezier curve in each end-to-end connected segment. The adjacent control points are represented by a quadratic Bessel curve to represent the handrail spline, and the feature parameters are adjusted to change the bending effect of the handrail, making the curve modeling more accurate and localized.

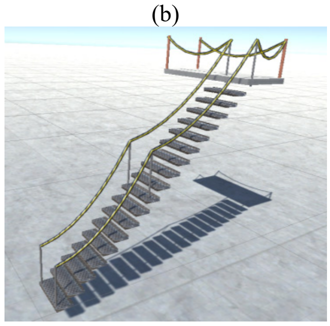
Calculate its frame spline in segments, and the calculation principle is shown in Figure 4 (b). The height H determines the degree of curvature of the Bezier curve.



(a)



(b)



(c)

Figure 4. Shows the generation of rope ladder handrails. Figure (a) shows the configuration file, Figure (b) shows the calculation principle of the Bessel curve, and Figure (c) shows the generated handrails

Find the control point Midi from two adjacent control points, with the height of the control point as the variable H, and points Pi and Pi+1 as the control points on the frame spline. From this, calculate the control point Midi on the handrail frame spline as:

$$Mid_i = (P_i + P_{i+1})/2 + (0, H, 0) \tag{4}$$

The connection of the three control points is $P_i Mid_i$ and $Mid_i P_{i+1}$. A new control point is generated at the position t (value range 0-1) of each connection and continues to be connected with the new control point until the position t of the last connection is the coordinate of the point t of the entire Bezier curve. Adjust the height H to adjust the position of the control point Mid_i, and then adjust the entire handrail frame spline. The spline calculation formula is shown in the above section.

3.1.3. Cylindrical Spiral Curve

A cylindrical spiral is approximated by a polyline, which is a set of multiple straight lines and arcs connected without self-intersection. It is difficult to calculate the convexity value of each arc of the polyline during spline generation. Therefore, in order to effectively parameterize the description, the cylindrical spiral is approximated by the set of straight edges of the polyline. The railing spline obtained by approximating the polysemy line is shown in Figure 5.

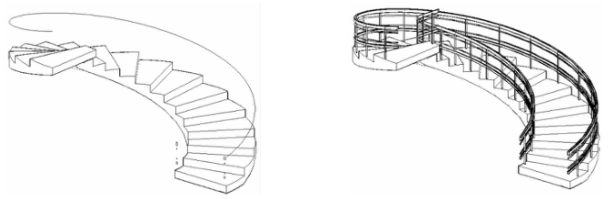


Figure 5. Cylindrical Helix

The parameterized definition of n vertex polylines is as follows:

$$PolyLine = \{E^{(i)}\}, i \in (1,n), E^{(i)} = \{type^{(i)}, \Delta x^{(i)}, \Delta y^{(i)}, \Delta z^{(i)}, \rho^{(i)}\} \quad (5)$$

Among them, $E^{(i)}$ represents the i-th edge of the polyline,$type^{(i)}$represents the type of edge as a straight line or arc,$\Delta x^{(i)}$, $\Delta y^{(i)}$, $\Delta z^{(i)}$represents the coordinate increment between the starting and ending points of the current edge,$\rho^{(i)}$ is the convexity value, when $\rho^{(i)} = 0$ the current edge is a straight line.

Spiral curve winding busbar p_0p_1 generation, starting from p0{x0, y0, z0}, p1{x1, y1, z1}, whose trajectory points are constrained by their projection pos_i on the plane, where $i \in [1, n]$. Derive the projection coordinates of the trajectory point on the xoz plane from the cylindrical helix formula:

$$\text{pos}_i = \begin{cases} x = x_0 + cos(i * \theta) \\ y = y_0 \\ z = z_0 + sin(i * \theta) \end{cases} \quad (6)$$

$spir_i$ represents the point set of spiral curve trajectory points, which are polysemy lines, where r is the radius of the spiral curve, θIs the rotation angle, and d is the step height. $\overrightarrow{U}$ represents the three-dimensional unit vector Vector3 (0,1,0), which can be derived from the above equation:

$$\text{spir}_i = r * pos_i + i * d * \overrightarrow{U} \quad (7)$$

The formula for calculating the straight edge of a polyline is as follows:

$$L_{\text{spir}_i\text{spir}_{i+1}}: \frac{x-x_i}{x_{i+1}-x_i} = \frac{y-y_i}{y_{i+1}-y_i} = \frac{z-z_i}{z_{i+1}-z_i} \quad (8)$$

3.2. Surface Mesh Generation

After generating frame splines, predefined component contour shapes, calculated curve subdivision points, growth direction, and scale factors are used to generate surface models of facility components. When modeling the surface of facilities, this method automatically calculates the mesh vertices of components and performs geometric transformations to construct 3D ladder like facilities. These parameters can be saved, and the constructed 3D object is a parameterized 3D model for further application.

3.2.1. Calculation of Contour Point Position

This algorithm takes a predefined 2D shape as input and generates a mesh model of the component along the frame spline. When the frame is curved, the mesh model generated at the corners is distorted. The main reason for this problem is the uneven grid caused by repeated calculations of contour points at each control point. This article adopts a new vertex position calculation method, which solves this problem by globally iterating and updating contour points through local point surface distance calculation and offset process, enabling them to describe any component shape we need.

This algorithm takes a polygon with m contour points ($Pr\,o^i_j$,…)as input, where i represents the current contour point,$i \in [1, m]$,The predefined polygons are shown in Figure a, and the contour points of the input polygon are calculated on the facility framework. Our algorithm consists of two steps. Step 1: The given polygon is the profile shape of the facility frame component, with the superscript i representing the i-th contour point of the input polygon and the subscript j representing the jth curve subdivision point on the facility frame curve, i.e., the jth iteration. Use equation (9) to calculate the position of each contour point:

$$Pr\,o^i_{j+1} = Pr\,o^i_j + \frac{-\overrightarrow{V_{j+1}}\cdot(P_{j+1} - Pr\,o^i_j)}{-\overrightarrow{V_{j+1}}\cdot d}.d \quad (9)$$

Where $d = \frac{\overrightarrow{P_jP_{j+1}}}{\|P_jP_{j+1}\|}$, $\overrightarrow{V_j}$ represents $\overrightarrow{T_j}$, $\overrightarrow{X}_j$, $\overrightarrow{U}_j$ or $\overrightarrow{F}_j$ unit vector ,and $\overrightarrow{T_j}$, $\overrightarrow{X}_j$, $\overrightarrow{U}_j$ and $\overrightarrow{F}_j$ represents the tangent vector, x-axis, y-axis, and z-axis unit vectors at the jth curve subdivision point, respectively.

The second step is to connect the predefined contour vertices with the calculated contour points in the $\overrightarrow{V_j}$ to obtain the mesh model of the component. The generation process is shown in Figure 6, where k represents the number of triangular meshes, $k \in [1, n]$,and the triangular mesh calculation formula is as follows:

$$\begin{cases} Triangles_k: (Pr\,o^i_j, Pr\,o^{i-1}_{j+1}, Pr\,o^i_{j+1}) \\ Triangles_{k+1}: (Pr\,o^i_j, Pr\,o^{i-1}_j, Pr\,o^{i-1}_{j+1}) \end{cases} \quad (10)$$

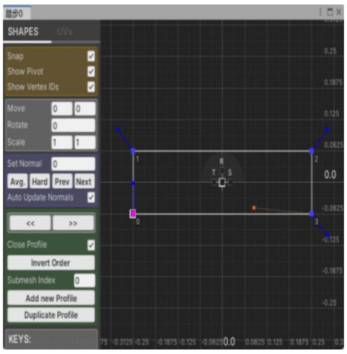
The profile and lofting path of the rope ladder handrail are defined by polysemy lines. Based on the quaternion rotation formula $Rot(\overrightarrow{F}, \theta)$, calculate the quaternion of the target vertex after rotating around the axis, and achieve the rotation of the target vertex around the$\overrightarrow{F}$ vector (0,0,1) θangle, where the angle variable = angleDelta +angleOffset, angle Deltawhere angleDelta is the angle increment and angleOffset is the angle offset. In this method, the multiplication of quaternions and vectors $\overrightarrow{R(1,0,0)}$represents the new vector obtained after rotation. Therefore, the coordinates of each profile point are V_i:

$$V_i = \overrightarrow{V_{center}} + Rot(\overrightarrow{F}, \theta) * \overrightarrow{R} \quad (11)$$

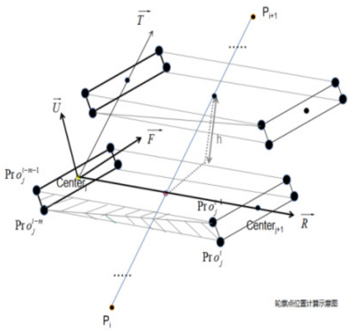
The calculation method for the position of its contour points and grid calculation is the same as above. Divide the spline curve of the rope ladder handrail equally, interpolate and calculate each segment point S_j,of the current curve, connect the segment points in sequence to obtain a polysemy line, j∈[1,segments], $\overrightarrow{s_js_{j+1}}$ generate the jth edge of the polysemy line in the s direction,,$t_j = j/$segments, and the larger the number of segments, the closer the polysemy line is to the rope ladder handrail spline. The calculation of waypoints is as follows:

$$S_j = \frac{(B_i+Mid_i)/t_{j+1} + (Mid_i+B_{i+1})/t_{j+1}}{t_{j+1}} \bullet \frac{\overrightarrow{S_jS_{j+1}}}{\|\overrightarrow{S_jS_{j+1}}\|} \quad (12)$$

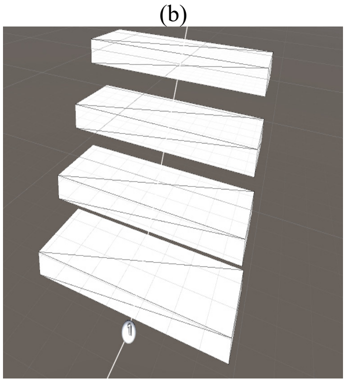
Mid_i calculation is the same as the previous text, where B_i represents the i-th spline control point.



(a)



(b)



(c)

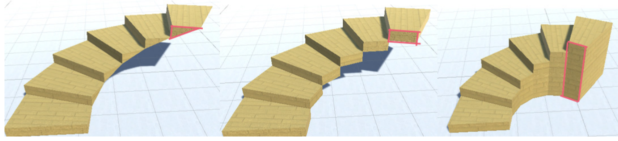
Figure 6. Calculation of contour point position, Figure (a) shows the profile of the step, Figure (b) Calculation principle of contour points, Figure (c) Result model

3.2.2. Contour Curve Constraint

The step height is basically consistent and can be combined and transformed to produce rich forms. The method in this article focuses on predefined constraint curves y(x), x∈[0,1] for predefined contour shapes. For any contour point $Pr\,o_j^i$ of a predefined polygon, calculate the offset along the $\vec{V}$ direction according to y(x) to obtain a new contour point to replace it, and obtain the replaced contour point:

$$\begin{cases} pro_j^i = P_i + offset \\ pro_{j+1}^i = pro_j^i + \phi \bullet y(x) \bullet Length \bullet \vec{V} \end{cases} \quad (13)$$

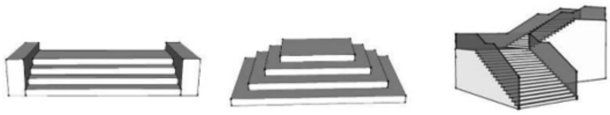
Where $offset = 1/2 \bullet \left[Pr\,o_j^i + \phi \bullet (1 - y(x)) \bullet Length \bullet \vec{V}\right]$. φ is the scaling factor of the predefined polygon to achieve meshing of fan-shaped steps, as shown in is the scaling factor of the predefined polygon to achieve meshing of fan-shaped steps, as shown in Figure7(a) and Figure 7(b). y(x) calculates the ratio coefficient between the x-axis and y-axis of the contour point in the $\vec{V}$ direction to achieve step meshing, as shown in Figure 7(c). Length represents the extrusion length along $\vec{V}$, and $\vec{V}$ is a directional constraint.



(a) (b) (c)

Figure 7. Step Type, where Figure (a) shows a triangular contour step, Figure (b) shows a rectangular contour step, and Figure (c) shows a stepped step

As shown in Figure 8, the extrusion length Length=Length − j ∗ width of each step in a linear ladder is inconsistent, where j represents the step generated at the jth curve subdivision point and width is the step width. The contour points of centralized stepped steps are constrained by φ, and the height of each step is fixed. The calculation method of turning step is the same as that of straight step, so the shape of the step can be easily defined. Figure 8 shows the generated model case created using the method presented in this article.



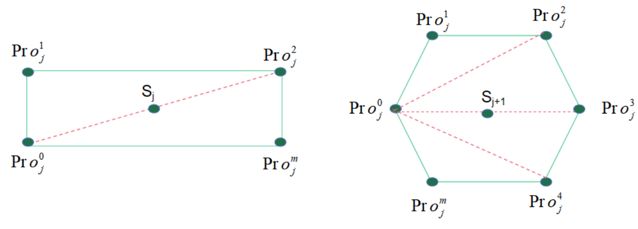
(a) Rectilinear (b) Centralized (c) Inflection type

Figure 8. Several Different Types of Steps

3.2.3. Section Grid Generation Algorithm

In the actual scene construction process, existing modeling methods usually only consider generating surface meshes, but lack treatment for the surfaces at the beginning and end positions, resulting in the problem of mold penetration. To achieve better visual effects, it is necessary to process predefined contour shapes and generate reasonable cross-sectional grids.

The predefined contour shapes of facilities are all closed polygons. The algorithm for generating profile grids is similar to the method for generating facility surface grids. This method is based on the configuration file of the predefined contour shapes, and generates vertex index arrays and other mesh attributes from the contour points in order. The triangulation process is shown in Figure 9:



(a) (b)

Figure 9. Profile Triangulation, Figure (a) shows quadrilateral triangulation, and Figure (b) shows hexagonal triangulation

In the figure, P_j represents the curve subdivision point, which is the center point of the contour shape, $j \in [1, n]$, $Pr\,o_j^i$ represents the i−th contour point at the jth partition point, $i \in [1, m]$. Triangulate the contour points in a clockwise direction, and the formula for triangulation is as follows:

$$Triangles_k: (Pr\,o_j^0, Pr\,o_j^{i+1}, Pr\,o_j^{i+2}) \quad (14)$$

Among them, k represents the k-th triangular mesh, and i=i+2. After each triangulation, the contour point index index index index jumps to the next contour point position that needs triangulation until i=m stops calculating.

4. Experiment

The experimental programming environment used in this study is Unity3D 2021.2.9f1 (64 bit). The computer configuration is as follows: Intel(R) Core (TM) i5-11400HQ 2.70 GHz CPU (8 CPUS), 32.0 GB memory, GeForce RTX 3050 GPU, 8GB graphics memory.

We invited 15 users (9 women, 6 men; ages 21 to 26) to use this tool to complete the modeling tasks of the project. Among these users, all have relevant 3D modeling experience and are familiar with at least one modeling tool. We provided half an hour of training to each testing user to ensure that everyone knew the basic operation of the system and gave them half an hour of practice time.

4.1. Modeling Results

We have divided the tilted photography area for users and modeled the ladder like facilities within that area. Firstly, build your own facility framework based on spline curves and contour shapes, then use our surface modeling operations to generate a complete facility model. Then, adjust the target model by simply adjusting the framework parameters to create a ladder like facility model that looks as close as possible. The model created by the user is shown in Figure 10, which displays the target results, facility framework curves, and user edited results.

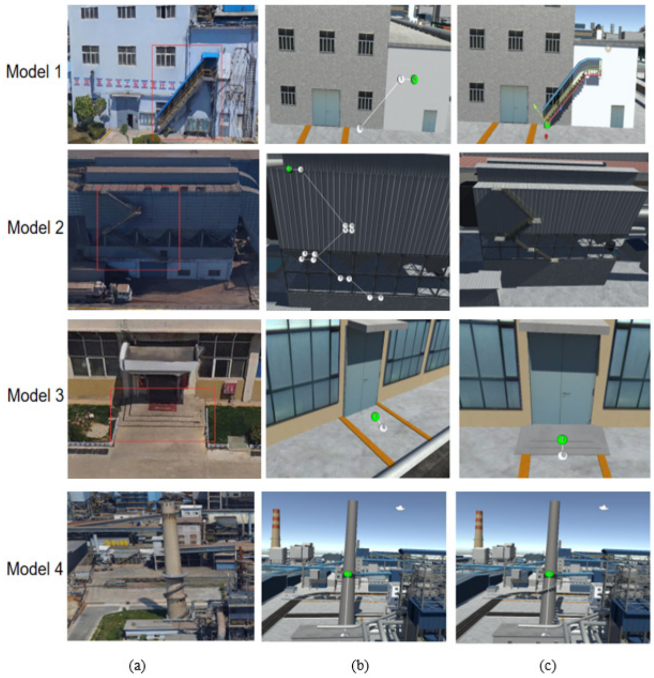


Figure 10. Shows an example model generated based on the framework curve, where (a) is oblique photography, (b) is the facility framework curve, and (c) is the output facility model

4.2. Comparative Experiment

Table 1. Comparison of time and number of grids spent modeling each model

target model	Result model	Grid generation time (in this article)	Number of grids (in this article)	Grid generation time (manual)	Number of grids (manual)
Figure Model 1(a)	Figure Model 1©	2m47s	11.1k	5min24s	11.9k
Figure Model 2(a)	Figure Model 2(c)	5m08s	10.7k	8min40s	13.5k
Figure Model 3(a)	Figure Model 3(c)	21s	4.0k	46s	4.2k
Figure Model 4(a)	Figure Model 4(c)	1m12s	8.6k	3min 41s	8.9k

Firstly, we tested the time required to generate the resulting model using the method presented in this article. Then, we further asked users to generate the target facility using traditional manual modeling methods to test the number of resulting model grids generated in both methods. Some of the results are shown in Table 1, which shows the comparison of editing time and grid quantity for each model.

As shown in Table 1, our method performs faster generation efficiency, better performance, and can be edited twice compared to traditional manual modeling methods in general complex facility modeling.

5. Summarize

This article proposes an extended parametric modeling method for ladder like facilities, which uses two main surface modeling operations: spline mesh generation and component instantiation to complete the modeling. The experimental results show that using this tool to complete the modeling of the target facility in a few minutes can edit the facility components, thereby showcasing the rich details of the facility. Compared with existing methods, this method generates fewer model grids, and the improved framework curve generation method can significantly accelerate the modeling speed. In the future, in addition to solving the problems raised by users, machine learning algorithms can also be used to extract facility frameworks and accelerate the generation speed of framework splines;

References

[1] Ding C, Liu L. A survey of sketch based modeling systems[J]. Frontiers of Computer Science, 2016, 10: 985-999.

[2] Nishida G, Garcia-Dorado I, Aliaga D G, et al. Interactive sketching of urban procedural models[J]. ACM Transactions on Graphics (TOG), 2016, 35(4): 1-11.

[3] Kelly T, Wonka P. Interactive architectural modeling with procedural extrusions[J]. ACM Transactions on Graphics (TOG), 2011, 30(2): 1-15.

[4] Guo Q. Yingzao Fashi: twelfth-century Chinese building manual [J]. Architectural History, 1998, 41: 1-13.

[5] Chen X, Kang S B, Xu Y Q, et al. Sketching reality: Realistic interpretation of architectural designs[J]. ACM Transactions on Graphics (TOG), 2008, 27(2): 1-15.

[6] Lopes R, Tutenel T, Smelik R M, et al. A constrained growth method for procedural floor plan generation[C]//Proc. 11th Int. Conf. Intell. Games Simul. Citeseer, 2010: 13-20.

[7] Freiknecht J, Effelsberg W. Procedural generation of multistory buildings with interior[J]. IEEE Transactions on Games, 2019, 12(3): 323-336.

[8] Huang H, Kalogerakis E, Yumer E, et al. Shape synthesis from sketches via procedural models and convolutional networks[J]. IEEE transactions on visualization and computer graphics, 2016, 23(8): 2003-2013.

[9] Turrin M, Von Buelow P, Stouffs R. Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms[J]. Advanced Engineering Informatics, 2011, 25(4): 656-675.

[10] Keshavarzi M, Hotson C, Cheng C Y, et al. Sketchopt: Sketch-based parametric model retrieval for generative design[C]// Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. 2021: 1-6.

[11] Unlu G, Sayed M, Brostow G. Interactive Sketching of Mannequin Poses[J]. arXiv preprint arXiv:2212.07098, 2022.

[12] van Gumster J, Lampel J. Procedural Modeling with Blender's Geometry Nodes: A workshop on taking advantage of the Geometry Nodes feature in Blender for procedural modeling [M] // ACM SIGGRAPH 2022 Labs. 2022: 1-2.

[13] Guo J, Xu S, Yan D M, et al. Realistic procedural plant modeling from multiple view images[J]. IEEE transactions on visualization and computer graphics, 2018, 26(2): 1372-1384.

[14] Rutzinger M, Pratihast A K, Oude Elberink S, et al. Detection and modelling of 3D trees from mobile laser scanning data[J]. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 2010, 38: 520-525.

[15] Delmerico J A, Baran D, David P, et al. Ascending stairway modeling from dense depth imagery for traversability analysis [C]// 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013: 2283-2290.

[16] Pérez-Yus A, López-Nicolás G, Guerrero J J. Detection and modelling of staircases using a wearable depth sensor [C]// Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13. Springer International Publishing, 2015: 449-463.

[17] Chan D S, Silva R K, Monteiro J C, et al. Efficient stairway detection and modeling for autonomous robot climbing [C]// 2017 IEEE/ RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 5916-5921.

[18] Woo S, Shin J, Lee Y H, et al. Stair-mapping with point-cloud data and stair-modeling for quadruped robot[C]//2019 16th international conference on ubiquitous robots (UR). IEEE, 2019: 81-86.

[19] Rafajłowicz E. Fast algorithm for generating Bernstein-Bezier polynomials [J]. Journal of computational and applied mathematics, 1994, 51(3): 279-292.

[20] Gordon W J, Riesenfeld R F. B-spline curves and surfaces [M] // Computer aided geometric design. Academic Press, 1974: 95-126.

[21] Huang C Y, Tai W K. Ting tools: interactive and procedural modeling of Chinese ting[J]. The Visual Computer, 2013, 29: 1303-1318.

[22] Huang C Y, Sheng Y S, Tai W K. Interactive and procedural modeling of featured Chinese architectures[C]//Smart Graphics: 13th International Symposium, SG 2015, Chengdu, China, August 26-28, 2015, Revised Selected Papers 13. Springer International Publishing, 2017: 16-28.

[23] Hu Z, Qin X. Extended interactive and procedural modeling method for ancient Chinese architecture[J]. Multimedia Tools and Applications, 2021, 80: 5773-5807.

[24] Wang Qian Research on stair form design language [D] Central Academy of Fine Arts, 2011.

[25] Yu yelong Research on architectural step design [d]. Harbin Institute of technology, 2010.