J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 4229--4240

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Virtual Landscape Generation Technique Based on Fractal Geometry and Its Optimization Strategy in Game Development

Gaofeng Su[1,✉]

[1] *College of Arts, Beijing Union University, Beijing, 100101, China*

ABSTRACT

Since the introduction of fractal geometry, it has set off a wave of research in the scientific community, and it has been widely used in many fields. This paper firstly introduces the landscape modeling and generating technology based on fractal geometry, and proposes the virtual landscape generating method based on fractal geometry through the study of the regular characteristics of fractal geometry. Combined with the game development of virtual landscape generation diversity, complexity needs, in the fractal Brownian motion model on the basis of the proposed optimization of the generation process for game development. In the simulation experiments of virtual landscape generation, the NME value of virtual landscape generation under the method of this paper is the smallest, which is distributed between 3 and 6, and the generation time is reduced by 31ms and 38ms compared with the average time of the traditional generation method and the SEM method, which shows that the designed virtual landscape generation is able to generate the virtual landscape more realistically. The study concludes with strategies and recommendations for the application of fractal geometry to virtual landscape generation in game development, with a view to contributing to the promotion of virtual generation technology.

Keywords: fractal geometry, Brownian motion, virtual landscape generation, game development

1. Introduction

Virtual reality technology is a computer technology that simulates the real environment, through the use of head-mounted displays and sensing devices, the user is completely immersed in a virtual three-dimensional interactive environment [1-3]. The realization of virtual landscape generation technology cannot be separated from the comprehensive application of computer graphics, human-computer interaction, and perception technology and other fields [4-7]. When designing a virtual reality scene, it is first necessary to establish a mathematical model of the virtual environment, including terrain,

✉ Corresponding author.
E-mail address: guaguakanword@163.com (G. Su).
Received 20 February 2024; Revised 15 April 2024; Accepted 20 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-239

buildings, objects and so on. Then, the model is transformed into image animation using computer graphics rendering techniques, and the image is projected to the user through virtual reality devices, such as helmets and glasses [8-10]. The visual and auditory information obtained by the user through these devices can produce an immersive feeling and enhance the cognition and participation in the virtual environment.

With the continuous progress of technology, the innovative application of virtual landscape generation technology in the field of game development has attracted more and more attention. Game developers can present an immersive game environment for players through the skillful use of virtual world building technology. In the process of game development, the construction of virtual worlds is a crucial link. Virtual reality technology creates a realistic virtual world for players by simulating real-world environments and objects, enhancing the sense of reality experience, and some can even do real-time simulation [11-13]. Players can interact and play with the virtual animation world through specially designed devices, such as head-mounted displays and joysticks, allowing players to feel immersive and immersive, increasing the fun and challenge of the game [14-15].

In this virtual scene, developers can adjust various effects at will to better adjust the design program. Virtual reality technology is also able to simulate different weather such as sunshine, heavy rain, snow, tornado, etc. in specific situations to enhance the haptic-like reality experience [16-19]. This can help developers to better predict the effect of the design, and modify and optimize the design solution.

The proposed fractal geometry model provides an effective means for terrain simulation, and under this model, this paper proposes a virtual landscape generation optimization method based on fractal geometry for game development. In order to comprehensively explore the effective optimization of virtual landscape generation, this paper firstly researches the principle structure and characteristics of fractal geometry. It also deeply analyzes the application method of fractal geometry in virtual landscape generation. On this basis, optimization is carried out based on the fractal Brownian motion model for the complexity demand of landscape generation for game development. Then, the effectiveness of this paper's virtual landscape generation method is verified through simulation experiments, and optimization strategies and suggestions are proposed for game development.

2. Research on virtual landscape generation based on fractal geometry

2.1. Fractal Geometry Theory

2.1.1. Characterization of fractal geometry. While traditional geometry studies integer dimensions, such as zero-dimensional points, one-dimensional lines, two-dimensional surfaces, and three-dimensional cubes, fractal geometry studies fractional dimensions, such as 0.65, 1.33, 2.89, and so on. Fractal geometry is a study of geometry with irregular geometric forms, and with the release of books dedicated to popularizing it to non-mathematical readers, such as Fractals and Chaos, fractals are becoming more and more known to the world, and have gradually moved towards common sense knowledge.

Fractal geometry can be divided into regular and random fractals. Regular fractals are shapes generated by mathematical equations or logic, and are the product of mathematicians rather than nature. Koch curve and Shebinski triangle is a more classical regular fractal, Koch curve construction shown in Figure 1, the construction rules are, starting from a straight line (this line is also known as the initiator), the line is divided into three segments, with a bottom of the equilateral triangle (the length of the side is equal to the length of each equidistant segment) instead of the middle of the line segment, this structure consists of four segments of the line segment known as the generator, and then repeated! Repeat the above steps, replacing each of the four segments with a generator, and the final curve formed is the Koch curve.

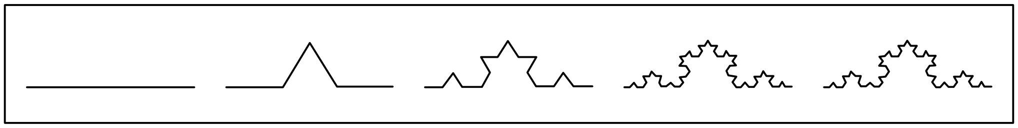

Fig. 1. Koch curve structure

The construction rule for the Shebinski triangle is to start from a triangle, find the midpoints of its three sides, connect the midpoints, divide the triangle into four smaller triangles, remove the middle triangle, and further subdivide the remaining three triangles, repeat the above steps, and the resulting pattern is the Shebinski triangle.

Random fractals, on the other hand, are natural products, characterized by presenting approximate self-similarity and a finite number of scale-free intervals, e.g., cauliflower, branches of trees, etc.

Fractals are characterized by self-similarity, self-affine and scale-free nature. By self-similarity, the essence is that a structure is (strictly) self-similar if it can be decomposed into arbitrarily small parts, each of which is a small replica of the whole. Each small part can be obtained by a self-similar transformation of the whole. This self-similarity property may be exact self-similarity, approximate self-similarity, or self-similarity in the statistical sense. Those that present exact similarity are regular fractal shapes generated by mathematical equations, and those that present approximate similarity are mostly natural products, such as trees, mountains and rivers in nature. Those that present statistical similarity, on the other hand, refer to some similarities that do not appear to be visually obvious, and present self-similarity in a statistical sense, such as through graphs and charts. The difference between self-simulation and self-similarity is that self-similarity zooms in or out at the same scale in each direction, while self-simulation zooms in and out at different rates in different directions.

2.1.2. Fractal dimension. For fractals, calculating the scales of their length, area, etc. seems meaningless, and thus a new metric is needed to articulate the image characteristics of a fractal. Mathematicians therefore proposed a way to measure the complexity of a scale as it increases by analyzing how quickly the measured length, area, etc. increase with it. Mathematicians introduced the concept of dimensionality to refine fractal geometry.

Most natural surfaces have fractal features in both spatial and gray-scale images, and the fractal dimension is a good way to characterize the human eye's perception of surface roughness. The larger the fractal dimension, the rougher the corresponding image surface is, and the smaller the fractal dimension, the smoother the corresponding image surface is. Therefore, the fractal dimension can be used as an index to measure the roughness of image texture, and maintains the consistency with human visual perception.

The mathematical definition of the fractal dimension needs to be illustrated in the context of the notion of self-similarity. Let F be a bounded set of a Euclidean space R^n. If the result F can be obtained from its own N_r mutually non-intersecting subsets by a concatenation operation, then F is considered to be self-similar. Although it is difficult to find strict self-similarity in nature, the similarity dimension can be utilized instead of the fractal dimension. Therefore, the fractal dimension is usually computed using a similar fractal dimension with the fractal dimension D given by:

$$D = \frac{\log\left(N_r\right)}{\log(1/r)} \tag{1}$$

where N_r is the number of mutually non-intersecting subsets and r is the scale factor.

The main methods that have been used to calculate the fractal dimension include the box dimension method, the FBM model and the blanket model. Since the box dimension method is simple to compute and can accurately describe the fractal dimension of an image, it is generally recognized in the field of image research. Therefore, in this paper, the difference box dimension method is used to calculate the fractal dimension. The method utilizes the pixel coordinates to form the xy-plane and the pixel values corresponding to each pixel coordinate to form the z-axis, so that the 2D image is transformed into a 3D model whose z-axis is a surface. The surface model is then stacked with small square boxes of a certain scale size, and then the corresponding number of differential boxes is accumulated by varying the scale factor r, and finally the fractal dimension D of the image is obtained by using the least squares method to fit the scale factor and the number of boxes.

The main steps in calculating the difference box dimension method are as follows:

Set the scale factor $r = s/L$ and partition the xy plane into $s \times s$ grids (where s is an integer, L is a constant, and $L/2 \geq s > 1$). On each $s \times s$ grid, a series $s \times s \times s$ of small square boxes are stacked onto the surface corresponding to the z-axis. Assuming that the minimum and maximum values of the pixel values at the (i,j) grid fall in the kth and lth boxes, respectively, let the box difference at the (i,j) grid be $n_r(i,j)$ when the box scale is r:

$$n_r(i,j) = l - k + 1 \tag{2}$$

The sum of box differences N_r required to cover the entire image is obtained by summing the box differences $n_r(i,j)$ for all meshes in this 3D model:

$$N_r = \sum_{i,j} n_r(i,j) \tag{3}$$

A set of corresponding N_r is obtained by taking different values of scale factor r, and finally $\lg N$ and $\lg(1/r)$ are fitted by the least squares method to obtain the slope of the fitted straight line, which is the box dimension D of the image.

Natural images respond to geometric objects in a natural scene with complex and irregular texture properties. The computer generated image is composed of a limited geometric model, and the texture of the image surface is simpler and smoother than that of the natural image, and the fractal dimension, as a measure of the coarse precision of the image texture, can respond to the difference between the two in terms of texture. Therefore, the fractal dimension can be used as one of the distinguishing features between the two.

2.2. Application of Fractal Geometry in Virtual Landscape Generation

Virtual reality technology is currently one of the popular research topics, with the technical progress and gradual maturity, its application areas have been developed from the past entertainment and simulation training to include aviation, aerospace, railway, construction, civil engineering, scientific computing visualization, medical, military, education, entertainment, communication, art, sports and other broad areas, and the status is also increasingly important. In recent years, in the field of computer graphics with the development of realism graphics generation technology and the introduction of fractal geometry, wavelet analysis and other theories, people have been able to use computer simulation to generate a lot of very realistic virtual natural scenery. In this paper, on the basis of fractal geometry, we further study the virtual landscape generation technology.

Three-dimensional terrain as an important type of natural scenery, in the simulation of three-dimensional terrain, the first thing that comes to mind is still the use of fractal geometry model to model it. The mathematical model of fractal Brownian motion (f Bm) is proposed to further promote the stochastic Brownian motion. f Bm is a complete mathematical model as a self-radiating stochastic process, and its most important feature is that it is statistically self-phase and smooth.

The mathematical model of fractal Brownian motion is proposed in the form of a stochastic process, and gives the construction method of Brownian motion: superimpose a number of stochastic processes obeying Gaussian distribution. Brownian motion is a stochastic process that satisfies the following conditions:

Let the stochastic processes be $X(t)$, $t \in I$, and $I = [0, +\infty)$, satisfying the following three terms, then $X(t)$ is a Brownian motion.

1) $X(t)=0, X(t)$ is a continuous function of t.
2) For any independent variables $0 \le t_1 \le t_2 \le \ldots \le t_n$, $t_i \in I$: The random variables are independent of each other:

$$\Delta X(t_k) = X(t_k) - X(t_{k-1}) \tag{4}$$

3) For any $0 \le t$ and $0 < h$, the random variable $X(t+h)-X(t)$ always obeys the normal distribution $N(0,h)$, i.e:

$$P((X(t+h)-X(t)) \le x) = \frac{1}{\sqrt{2\pi h}} \int_{-\infty}^{x} \exp\left(-\frac{u^2}{2h}\right) du \tag{5}$$

The above three rules define Brownian motion, and in 1968, fractal Brownian motion was proposed on the basis of Brownian motion and described as a stochastic process, and the fBm two-dimensional surface was defined as a stochastic process on a certain probability space and satisfied:

(1) $X(0,0)=0$, $X(x,y)$ are continuous functions of (x,y).

(2) For any variable $(x,y),(h,k) \in R^2$, its two-dimensional increment $X(x+h,y+k)-X(x,y)$ obeys a normal distribution with expectation 0 and variance $\left(h^2+k^2\right)^{2H}$ whose probability is satisfied:

$$X(x+h, y+k) - X(x,y)) \le z) = \frac{1}{\sqrt{2\pi}\left(h^2+k^2\right)^H} \int_{-\infty}^{z} \exp\left\{-\frac{r^2}{2\left(h^2+k^2\right)^{2H}}\right\} dr \tag{6}$$

Nowadays, fractal terrain modeling has been more mature, with more methods, among which five methods are more prominent, i.e., Poisson step method, mid-point displacement method, Fourier filtering method, step-by-step random increase method, and band-limited noise accumulation method.

2.3. Optimization of Fractal Virtual Landscape Generation for Game Development

Natural features, such as trees and rocks, are rough and irregular, and traditional methods cannot accurately model these objects. Natural scenery can be described realistically using fractal geometry methods that use processes rather than equations to model objects.

1) Fractal generation process. A fractal object is generated by repeating a given transformation function for points in a region of space. If $P_0 = (x_0, y_0, z_0)$ is the selected initial point, the computation of the transformation function $F(\cdot)$ is repeated each time to generate a successor layer:

$$P_1 = F(P_0), P_2 = F(P_1), P_3 = F(P_2) \tag{7}$$

For each repetition, we use both fixed and randomized generative processes. Although fractal objects by definition have an infinite number of details, we apply the transformation function a finite number of times. As the number of transformations is increased to produce more detail, the procedural representation approaches the "true" fractal. The amount of detail included in the final object display depends on the number of iterations and the resolution of the display system. It is not possible to display a change in detail smaller than the pixel size.

2) Random midpoint displacement method. Fractal Brownian motion can be utilized to obtain topographical features and other natural landscapes with a high degree of realism, but it is time-consuming to compute the fractal Brownian motion because the height coordinates on the ground are computed using a Fourier series, which is the sum of the cosine and sine terms. The Fast Fourier Transform (FFT) method is usually used, but it is still a slow process for generating fractal landscapes. Therefore, a fast randomized midpoint displacement method similar to the randomized midpoint displacement method used in geometric constructions is used here to represent the fractal Brownian

motion that approximates the ground and other natural phenomena.

Although the random midpoint displacement method is faster than fractal Brownian motion calculations, it reduces the realism of the ground features. A midpoint displacement method for generating random walk paths in the xy plane is illustrated. Starting with a straight line segment, line segment midpoint displacement y values are obtained by averaging the endpoint y values plus a random offset:

$$y_{\text{mid}} = \frac{1}{2}[y(a) + y(b)] + r \tag{8}$$

where is the approximating fractal Brownian motion. The value of r is a Gaussian distribution from 0 to a mean square deviation proportional to $|b-a|^{2H}$, where $H=2-D$ and $D>1$ are the fractal dimensions. Another way to obtain a random offset is to make $r = sr_g|b-a|$, where s is the selected "surface roughness" factor, and r_s is a Gaussian random value from 0 to a mean square deviation of 1. The Gaussian values can be obtained by looking up the table. The process is then repeated by calculating the midpoint displacement y value for each half of the subdivided segment, successively subdividing the segment until it is less than a specified value. At each step, the value of the random variable r decreases because it is proportional to the length $|b-a|$ of the subdivided line segment.

A random midpoint displacement program is used to generate ground features for a rectangular ground plane. To begin, a height is assigned to each of the four corners of the ground plane (a,b,c,d), z . The ground plane is then split at the midpoint of each side to obtain five new grid point locations: e, f, g, h, m . The heights at points e, f, g, and h on the sides of the ground plane can be computed by averaging the heights of the two nearest vertices plus a random offset. The height z_e at point e is calculated using vertices a and b over, and the height at midpoint f is calculated using vertices b and c:

$$z_e = (z_a + z_b)/2 + r_e, z_f = (z_b + z_c)/2 + r_f \tag{9}$$

Random values r_e and r_f are obtained from a Gaussian distribution with mean square deviation from 0 to the power of $2H$ proportional to the grid spacing, where $H=3-D$, $D>2$ are the fractal dimensions of the surface. We can also calculate the random offset as a lookup table value of the surface roughness factor multiplied by the grid spacing multiplied by a Gaussian value from 0 to the mean square deviation 1. The height z_m of the midpoint m of the surface can be calculated using e and g or f and h, or z_m using the specified heights of the four surface corners:

$$z_m = (z_a + z_b + z_c + z_d)/4 + r_m \tag{10}$$

This process is repeated for four new mesh sections at a time until the mesh spacing is less than a specified value.

The triangular facets are formed after generating the height. It represents the eight plane slices formed during the first segmentation. At each level of recursion, the triangle is successively divided into smaller plane slices. When the segmentation process is complete, the facets are drawn according to the light source position, other lighting parameter values, selected color and surface texture.

In addition to the ground, the random midpoint displacement method can be applied to other parts of the landscape. For example, the same method can be used to obtain surface features of water waves or cloud models on the ground plane.

3) Ground Map Control. In the simulation of a fractal ground scene with the midpoint displacement method, mountain and canyon displacements are controlled by constraining the computed heights in a certain interval on different regions of the ground. Then, at each midpoint grid location on the ground, a random height is computed for that point that depends on the difference between the control height and the average height at that location. This process constrains the height within a given interval of the control ground height.

This chapter uses a random midpoint displacement method to calculate the grid heights, coming from a Gaussian distribution to pick random values, where the mean μ and standard deviation σ are functions of the control heights. One way to take the values of μ and σ is that they are both proportional to some difference between the calculated mean height and the predetermined control height at each grid location. For example, for grid position e, let the mean μ_e and standard deviation $\sigma_ò$ be:

$$\mu_c = zc_c - \left(z_a + z_b\right)/2, \sigma_c = s\left|\mu_c\right| \tag{11}$$

where zc_e is the control height at ground position e and $0 < s < 1$ is a predetermined scaling factor. Smaller s values of $(s < 0.1)$ generate a smoother surface, while larger s values give a greater undulation in the ground height.

To determine the value of the control height on a planar control surface, the planar parameters A, B, C and D are first calculated. For any ground plane location (x, y), the height on the plane containing the control polygon can be calculated as follows:

$$zc = (-Ax - By - D)/C \tag{12}$$

An incremental approach can then be used to calculate control heights at ground plane grid locations. To accomplish these calculations efficiently, the ground plane is first partitioned into a xy-position network. Each polygon control surface is then projected onto the ground plane. A process similar to scanline region filling can be used to determine which grid locations fall within the projection area of the control polygon. That is, a scanning area fill is performed for each y on the ground plane mesh that passes through the polygon's edges, counting the scanning line intersections and deciding which grid location falls inside the control polygon's projection area. The height of the control at these grid positions can then be calculated in increments:

$$zc_{i+1.j} = zc_{i,j} - \Delta x(A/C), zc_{i,j+1} = zc_{i,j} - \Delta y(B/C) \tag{13}$$

where Δx and Δy are the mesh distances in the x, y directions. This procedure is particularly fast when dealing with control surface mesh positions using parallel vector methods.

3. Analysis of the effect of virtual landscape generation for game development

In order to verify the optimization effect of this paper's fractal geometry-based virtual landscape generation in game development, on the virtual simulation platform with Visual Studio 2015 as the development environment and UG NX10.0 as the research and development platform, a virtual landscape generation system was built using this paper's fractal virtual landscape generation method to carry out simulation experiments of virtual landscape generation.

3.1. Comparison of the quality of virtual landscape generation

The fractal geometry virtual landscape generation method proposed in this paper, the traditional virtual landscape generation method and the common 3D reconstruction technique SFM modeling are compared. The experiments were collected 20 small field landscape scenes (denoted as LcS1~LcS20) and 15 larger field scenes (denoted as LcB1~LcB15), and these landscape scenes were used as the validation objects to generate virtual landscapes using this paper's method and the traditional method, respectively, and this paper adopts the Normalized Mean Error (NME) as the assessment index. In order to comprehensively study the effect of virtual landscape model generation, the comparison results of virtual landscape generation for small scenes are shown in Figure 2. As can be seen from the figure, the NME value of virtual landscape generation under the method of this paper is the smallest, distributed between 3 and 6, and the NME of SFM and traditional method is distributed between 3 to 9 and 4 to 10. The virtual landscape generation based on fractal geometry in this paper has achieved advantages in the generation of small landscapes for game development.

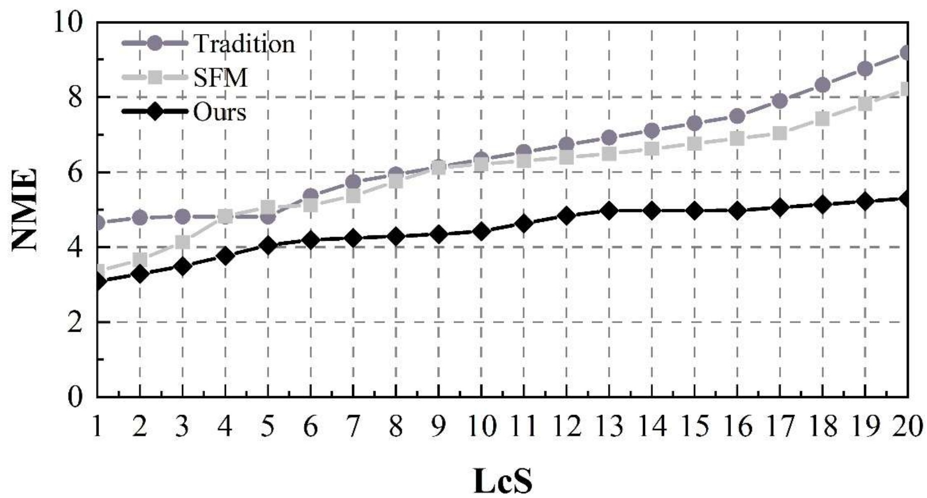

Fig. 2. Small scene virtual landscape generation comparison results

The results of the comparison of virtual landscape generation for large scenes are shown in Fig. 3, where it is observed that the NME values of all three methods have increased in the scene generation of larger virtual landscapes. This is due to the fact that the generation of virtual landscapes for large scenes is more complex and the generation difficulty is high. It is noteworthy that the NME of the SFM method rises to 6-11 and the NME performance is lower than that of the traditional methods. In this paper, we propose a virtual landscape generation technique based on fractal geometry, which is optimal in both large and small scenes.

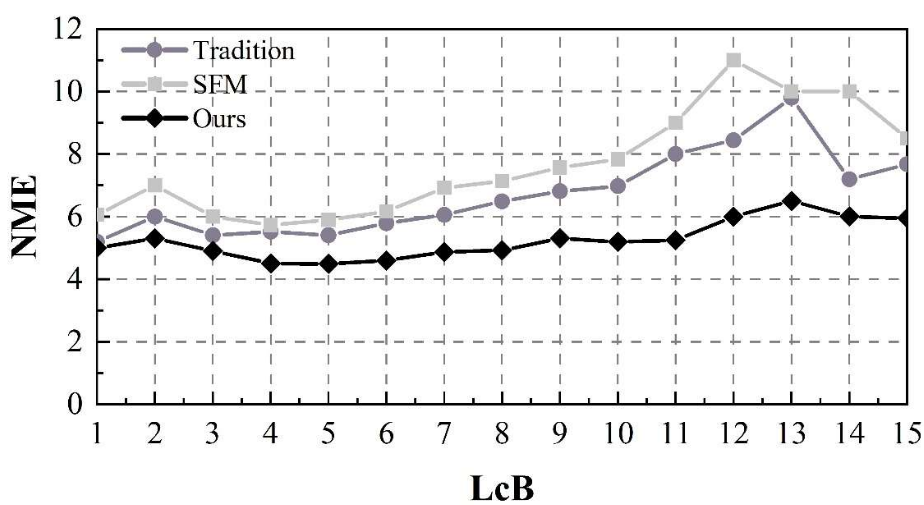

Fig. 3. Large scene virtual landscape generation comparison results

In order to further verify the effectiveness of the virtual landscape generation method in this paper, this paper compares the error and RMSE of the virtual landscape generation as the indexes, and the comparison results of the generation accuracy of different methods are shown in Table 1. From the table, it can also be seen that the average error and RMSE of this paper's method are 2.4738 and 1.7281, respectively, which are higher compared with the model accuracy generated by other methods, and further proves that this paper's fractal geometry-based virtual landscape generation method has a higher application value in game development.

Table 1. Comparison results of different methods of production

Method	Index	LcS	LcB	Mean
Traditional method	Model error(mm)	2.8654	3.1254	2.9954
	RMSE	1.9567	2.0152	1.9860
SEM	Model error(mm)	2.4582	3.5152	2.9867
	RMSE	1.8751	2.9542	2.4147
Ours	Model error(mm)	2.4354	2.5122	2.4738
	RMSE	1.5841	1.8721	1.7281

3.2. Comparison of the efficiency of virtual landscape generation

In this paper, the statistics of different virtual landscape generation methods are used in the generation of larger virtual scenes, and the comparison of the virtual landscape generation time of different methods is shown in Table 2. The higher the grid number of the generated landscape model, the more complex the generated landscape model is. From the comparison of the experimental results, it can be seen that the average time of virtual landscape generation of traditional generation method and SEM method is 1536ms and 1543ms, respectively, and the average time of virtual landscape generation based on fractal geometry in this paper is 1505ms, which is 31ms and 38ms lower than the average time of traditional generation method and SEM method, respectively, and the time-consuming comparison of landscape model generation with different methods proves that this paper's method can be more effective in virtual landscape generation than the traditional method and the SEM method. By comparing the time consumption of landscape model generation with different methods, it proves that the method in this paper has higher efficiency in virtual landscape generation.

Table 2. The virtual landscape generation of different methods is compared

Generated landscape	Model grid number	Time(ms)		
		Traditional method	SEM	Ours
LcB 1	275423	1565	1578	1506
LcB 2	265485	1585	1596	1555
LcB 3	234425	1456	1468	1421
LcB 4	235424	1421	1438	1400
LcB 5	223542	1435	1449	1398
LcB 6	235418	1452	1435	1405
LcB 7	235874	1454	1462	1435
LcB 8	233845	1461	1468	1454
LcB 9	264821	1502	1515	1495
LcB 10	256781	1495	1413	1413
LcB 11	235481	1462	1475	1426
LcB 12	523548	1871	1901	1862
LcB 13	435424	1679	1721	1646
LcB 14	354888	1603	1622	1597
LcB 15	305112	1598	1601	1568
Mean	287699	1536	1543	1505

3.3. Evaluation of virtual landscape generation

With the continuous development of three-dimensional model generation technology, more and more fields through three-dimensional simulation to solve the corresponding scientific research problems, but so far, there is no exact standard to evaluate the model generation simulation effect and build quality to test and constraints. In order to more scientifically and objectively evaluate the effect of this paper's method in the virtual landscape generation, this paper is used for three-dimensional landscape construction of data information, digital elevation model (DEM) into a coordinate set of elevation values to test the actual generation of the virtual landscape model coordinate values, according to the generation of the landscape example, the extraction of the Y-axis coordinate value of Y1 and the original coordinate value of Y2 for comparison, the results of the comparison of elevation data as shown in Figure 4. According to the elevation data comparison results, it can be seen that this study uses the fractal geometry virtual landscape generation method, the depth estimation of the generated landscape derived from the model mesh construction of the vertex height coordinate value Y1 in numerical value and the sampled landscape area of the DEM elevation coordinates of the set of elevation values Y2 has a high degree of similarity, which to a certain extent explains that this paper's landscape modeling has a good generation results.

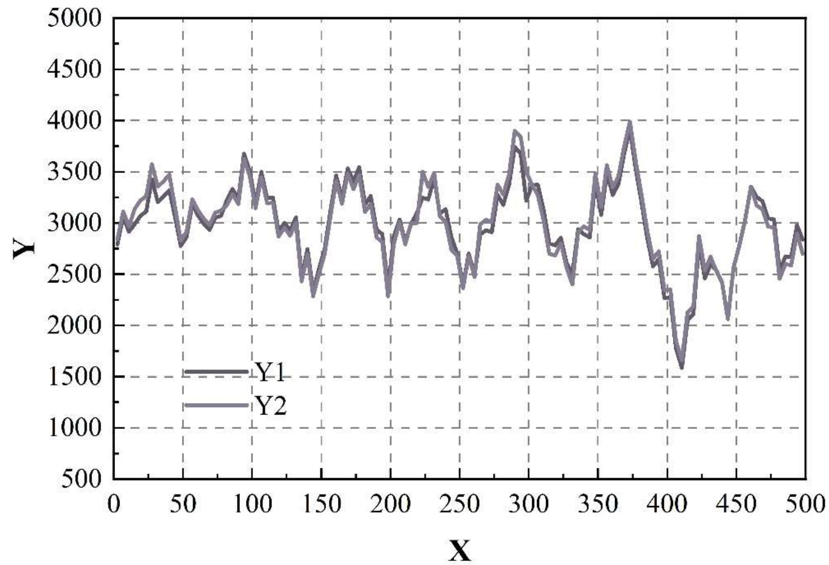

Fig. 4. High range data comparison results

4. Conclusion

The core content of the research in this paper is the virtual landscape generation technology based on fractal geometry, through the study of fractal geometry, this paper realizes the optimization of virtual landscape generation based on fractal geometry, through the simulation experiments, it can be seen that the NME value of virtual landscape generation under the method of this paper is distributed between 3 and 6, and the average time of landscape generation is 1505ms. From the NME, the method in this paper reduces 0~3 and 1~4 compared to the conventional generation method and SEM method, respectively. From the point of view of average time, compared with the traditional generation method and SEM method, it reduces 31ms and 38ms, respectively.In addition, the average error and RMSE of this paper's method are 2.4738 and 1.7281, respectively, and the accuracy of the model generated by this paper is higher compared with other methods. Taken together, the virtual landscape generation method based on fractal geometry in this paper has higher application value in game development.

For the application of fractal geometry virtual landscape generation in game development this paper proposes the following optimization strategies and recommendations:

1) Fractal algorithms for virtual landscapes in game development can be targeted at terrains with different levels of detail and natural appearance features. However, relying only on fractal algorithms in the generation of virtual landscapes still has obvious limitations, therefore, exploring the combined use of fractal geometry and a variety of algorithms in order to further make the generation of landscapes more realistic and improve the generation effect.

2) In the game development, in order to simulate various natural processes such as erosion and

weathering in nature, it is also necessary to use algorithms that simulate natural erosion and weathering, such as hydraulic erosion, thermal erosion, mechanical erosion, weathering erosion, and other algorithms and techniques to simulate the effect of terrain erosion.
3) in the virtual world of the game virtual landscape generation to obtain natural realistic landscape effect which not only rely on the technology, the subjective aesthetics of the landscape scenario designers is also an important part of the evaluation of the virtual landscape generation of the impact of the game development, therefore, the generation of excellent quality and beautiful game landscape not only need to improve the generation technology, but also need to cultivate the designers of the aesthetics and the ability to quality.

Funding

Nature of the project: School-level teaching research and reform project of Beijing Union University: Research on practical teaching mode of Digital Game Design course in Digital Media art major (JJ2023Y040).

References

[1] Wohlgenannt, I., Simons, A., & Stieglitz, S. (2020). Virtual Reality. Business & Information Systems Engineering: The International Journal of WIRTSCHAFTSINFORMATIK, 62(5), 455-461.

[2] Rubio-Tamayo, J. L., Barrio, M. G., & García, F. G. (2017). Immersive environments and virtual reality: Systematic review and advances in communication, interaction and simulation. Immersive environments and virtual reality: Systematic review and advances in communication, interaction and simulation, 1(4).

[3] Roettl, J., & Terlutter, R. (2018). The same video game in 2D, 3D or virtual reality–How does technology impact game evaluation and brand placements?. PLoS ONE, 13(7).

[4] Li, Z., Cheng, Y., & Yuan, Y. (2018). Research on the Application of Virtual Reality Technology in Landscape Design Teaching. Kuram ve Uygulamada Egitim Bilimleri, 18(5), 1400-1410.

[5] Lv, Z., Li, X., & Li, W. (2017). Virtual reality geographical interactive scene semantics research for immersive geography learning. Neurocomputing, 100(254), 71-78.

[6] Paes, D., Arantes, E., & Irizarry, J. (2017). Immersive environment for improving the understanding of architectural 3D models: Comparing user spatial perception between immersive and traditional virtual reality systems. automation in Construction, 100(84), 292-303.

[7] Sun, B., Jiang, Y., Liu, Y., Wu, X., & Liu, Q. (2023). Rural Environmental Landscape Construction Based on Virtual Reality Technology. Sustainability (2071-1050), 15(23).

[8] Liu, X., & Pan, H. (2022). The Path of Film and Television Animation Creation Using Virtual Reality Technology under the Artificial Intelligence. Scientific Programming, 2022.

[9] Zhao, S., Abou-zeid, H., Atawia, R., Manjunath, Y. S. K., Sediq, A. B., & Zhang, X. P. (2021, December). Virtual Reality Gaming on the Cloud: A Reality Check. In GLOBECOM 2021-2021 IEEE Global Communications Conference (pp. 1-6). IEEE.

[10] Forlim, C. G., Bittner, L., Mostajeran, F., Steinicke, F., Gallinat, J., & Kühn, S. (2019). Stereoscopic Rendering via Goggles Elicits Higher Functional Connectivity During Virtual Reality Gaming. Frontiers in Human Neuroscience.

[11] Disztinger, P., Schlögl, S., & Groth, A. (2017, January). Technology Acceptance of Virtual Reality for Travel Planning. In Information and Communication Technologies in Tourism 2017: Proceedings of the International Conference in Rome, Italy, January 24-26, 2017 (p. 255). Springer.

[12] Maheswari, R., Gnanamalar, S. S. R., Gomathy, V., & Sharmila, B. (2018). Real Time Environment Simulation through Virtual Reality. International Journal of Engineering & Technology, 7(2.24), 404-406.

[13] Paavilainen, J., Korhonen, H., Alha, K., Stenros, J., Koskinen, E., & Mäyrä, F. (2017). The Pokémon GO Experience: A Location-Based Augmented Reality Mobile Game Goes Mainstream. In ACM SIGCHI annual conference on human factors in computing systems (pp. 2493-2498). ACM.

[14] Cao, S., Nandakumar, K., Babu, R., & Thompson, B. (2020). Game play in virtual reality driving simulation involving head-mounted display and comparison to desktop display. Virtual Reality, 24(3).

[15] Vogel, D., Lubos, P., & Steinicke, F. (2018, March). AnimationVR-Interactive Controller-Based Animating in Virtual Reality. In 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 781-782). IEEE.

[16] Chamilothori, K., Wienold, J., & Andersen, M. (2018). Adequacy of Immersive Virtual Reality for the Perception of Daylit Spaces: Comparison of Real and Virtual Environments. LEUKOS, 15, 203-226.

[17] Noueihed, H., Harb, H., & Tekli, J. (2022). Knowledge-based virtual outdoor weather event simulator using unity 3D. Journal of Supercomputing, 78(8).

[18] Mitsuhara, H., & Shishibori, M. (2020). Comparative experiments on simulated tornado experience via virtual reality and augmented reality. The Journal of Information and Systems in Education, 19(1), 21-31.

[19] Han, P. H., Hsieh, C. E., Chen, Y. S., Hsiao, J. C., Lee, K. C., Ko, S. F., ... & Hung, Y. P. (2017, July). AoEs: Enhancing teleportation experience in immersive environment with mid-air haptics. In ACM SIGGRAPH 2017 Emerging Technologies-International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2017 (p. a3). Association for Computing Machinery, Inc.