The Study of Terrain Simulation Based on Fractal

DENG FANG
College of Computer and Information
Zhejing Wanli University
Ningbo, 315100
CHINA
http://www.zwu.edu.cn

Abstract: - In this paper the principle of fractal theory and fractal Brownian motion (FBM) are analyzed in detail. The Diamond-Square algorithm is introduced. And it introduces the process of generating three dimensional terrain, cloud and sky by this algorithm. There also analyses the physical meaning of FBM's two parameters H and σ . Through the analysis of the effect of different parameters, there is proposed a method of controllable fractal terrain based on parameters. Simulated a controlled fractal scene via VC ++ and OpenGL. And the same time, points the limitation of this algorithm. But as this algorithm's generating speed is faster, it also has validity and practicability.

Key-Words: - Fractal, FBM, Diamond-Square, 3D terrain, OpenGL

1 Introduction

It is very important to simulate the nature environment in visual simulation. The successful simulation can enhance the simulation results to reach the purpose closing to the reality.

B.B. Mandelbrot puts forward the fractal dimension and the idea of fractal geometry in 1973[1]. Fractal has the meaning of rulelss and fragment. Fractal geometry is a kind geometry of researching non-rules. Because the non-rules is widespread in natural, the fractal geometry is called as description of the nature. Compared with the traditional geometry, fractal geometry has some characteristics as follows:

(1) Overall, the fractal image is irregular everywhere.

(2) At different scales, the graphics of the rules are the same.

With the study of the fractal geometry deeply, it is found that many natures have the fractal characteristics in the process of observation and analysis of nature. It can reflect accurately the nature's form by using the fractal geometry mathematical model to replace the traditional Euclidean geometry mathematical model.

The algorithm to generate realistic graphics is one important part in Computer Graphics. The nature simulation is one of the most challenging issues among the shows of the realistic images [1]-[3]. The algorithm is very complicated; it is difficult in a conventional way to describe. The modeling and rendering of nature has become a hot research and the difficulty in computer graphics. Compared with other natural, terrain is its own special: 1. the scope of space, wide field of vision; 2. Details of the surface terrain is complex, the surface of the ups and downs has the randomness.

Common terrain visualization technology can be broadly classified into three categories. The first is using curved surface to generating the 3D terrain. The second method is based on digital terrain model. The third method is using fractal geometry to generating terrain. In this paper, the third method is adopted.

In order to improve the realistic simulation of terrain，many domestic and foreign scholars have made an in-depth study. In 1968, Mandelbrot and Van Ness put forward a one-dimensional Gaussian random process, called “fractional Brownian motion (FBM)”. Because FBM describes a lot of nature phenomena primely, many scholars attempt to simulate the realistic terrain using FBM[4].

In this paper the terrain simulation technology based fractal is discussed. The method is simply. Terrain generates fast and the effect of simulation is reality.

2 Fractional Brownian Motion

2.1 Origin of the Brown Motion

Brown Motion is found by Robert Brown who is botanist when he studies farina in 1827. The motorial directions of farina can change at any moment, and its trajectory is irregular broken line. It is explained as a rule less and unstable random movement phenomenon arising from the atom colliding other atoms surrounding constantly and continuously.

Mandelbort and Ness bring forward the concept and model of FBM through the expansion of Brownian motion. FBM[5][6] laid the foundation for the fractional theory. And FBM has a strong practicality. It can be used to simulate the mountain, cloud, terrain and so on. Now FBM has been the theoretical basis of fractal simulation.

FBM is affine fractal method to simulate object's characteristic. It can get the height realistic of ground and other nature by the method. The fractal Brownian path can be used to simulate fractal curve, and two dimension random fractal Brownian height arrays can be used to simulate terrain.

2.2 FBM Mathematical Model

In 1923, German mathematician N. Wiener creates the FBM mathematical Model by statistics .He consider Brownian motion is a random process and it can be considered to be a cumulation of a series of Gaussian random variable. In other words, a number of random events of Gaussian distribution can be accumulated to the Brownian motion. FBM mathematic model is a random process which meets the following conditions[7]-[9].

FBM is a random process X defined in a probability space. $X:[0,\infty]->R$, and satisfy those conditions as following:

1: according to probability 1, $X(t)$ in a row, and $X(0)=0$.

2: $\forall t \geq 0$ and $h>0, X(t+h)-X(t)$ subject to the following distribution.

Set U is a real number at $(-\infty,+\infty)$, w is a range of random function, it belongs to a sample space W .So the ordinary FBM $B(u,w)$ can be defined as a real random function. Set h is a parameter, and $0 \leq h \leq 1$, B_0 is a arbitrary real number, and the FBM $B_h(u,w)$ which parameters is h, initial value is B_0 . $B_h(0,w)=b$

$$B_h(u,w)-B_h(0,w)=1/\Gamma(h+0.5)\times\{\int_{-\infty}^{0}[(u-s)^{h-0.5}-(-s)^{h-0.5}\}d_{B(s,w)}+\int_{0}^{u}(u-s)^{h-0.5}d_{B(s,w)}\}$$

when H=0.5, $B_h(u,w)$ is common Brownian motion. The random process meeting those relations is called FBM. FBM has the basis attribute of common Brownian motion. such as mathematics model and self-similarity.

FBM two dimension curved surface is the 1D definition's expand. Using the coordinate variable (x,y) replaces the time variable. The random variable X(x,y) can be seem as the height of point (x,y) in 2D curved surface.

FBM is an important random process in modern non-linear time series analysis. It can express most non-linear phenomena in the natural world effectively. So far, it is also the best random process which to describe the real terrain. Because of the different achieve methods [10]; FBM has several types of methods: Poisson faulting, Fourier filtering, midpoint displacement, successive random additions and summing band limited noises[11].

Mandelbrot is the first man who put forward the Poisson faulting; this method is suitable to ball surface generating of ball analogous subject, but the high time complex degree is its main defects.

The merit of the Fourier filtering is that it can control the frequency heft accurately, further more it can get the terrain in any undulation degree. Meanwhile its defect is that the final earth's surface result has its periodicity, and its vision effect is poor. Besides it, it is lacking of details controlling of local part and it is hard to change the fine degree of sampling.

The successive random additions is a flexible fractionize project, if it is necessary to use the point that fixed in the last step's fraction process, thereby a random variable which is subject to a certain distribution should be first added on all of the points. Normally, we can get new points through cutting in the linear or nonlinear value on the basis of last step's fraction process. The time complexity depends on the final distinguishing rate and terrain L's undulation degree.

Summing band limited noises is to repeat the signal which strictly restrained within the frequency scope, and every signal's range is stochastic changed, viz. noise. So that it is also called noise generating method. It is a way of model shaping on the basis of function; any point's fixing is independent from its abutting ones; it differentiates itself from other stochastic fractal arithmetic methods.

Midpoint displacement is standards fractal geometry, it's time complexity is N's linear. So it can be used to quick landscape generation. Since its nice visual effects and efficient algorithm feasibility, midpoint displacement becomes a representative method in terrain simulation.

2.3 Character of Brownian Motion

Brownian motion has some important characters. These characters have great significance for generating fractal dimension.

Character 1: the track of Brownian motion has the statistical self-similarity. This is the most essential character. It shows that the statistical character couldn't change whether time or space scale.

Character 2: Brownian motion is not only one dimension, but also N dimension. This depended on the value space is R or R^N.

Character 3: fractal Brownian motion is a non-stationary process. Although non-stationary process is not easy to deal with in mathematical, now it has been convinced of the effect that it is the non-stationary which makes the natural has so rapid change and complex result. Non-stationary is an important character of FBM.

Brownian motion is not only the three characters above. Those characters are related with the fractal terrain.

3 Random Midpoint Displacement

Midpoint displacement[12][13] is use of the process to simulate the terrain that interpolation within two or more points. This method has the capacity of high-speed and increasing the details of the shape. Its iterative formula is

$$f_{mid} = \frac{(f_i + f_{i+1})}{2} + \Delta \times Gauss() \qquad (1)$$

where :

$$\Delta_n = (\frac{1}{2})^{nH}\sigma \qquad (2)$$

f_i and f_{i+1} are a segment of the two endpoints. Δ is to control the disturbance of the current level. it determines the roughness of fractal outcome . Gauss() is a random function. H is spectrum exponent, N is topology dimension.

To simulate the terrain, H represents the complicated of natural terrain. With H increasing, the terrain becomes flat; With H falling, it becomes rough. And σ response the regional terrain characteristics of the ups and downs .As long as we have taken a regional characteristic parameters H and σ , we can use FBM simulate the regional terrain that has the multi-resolution and more details and levels.

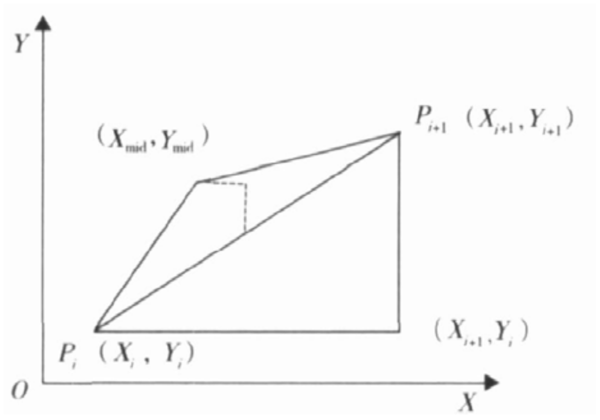
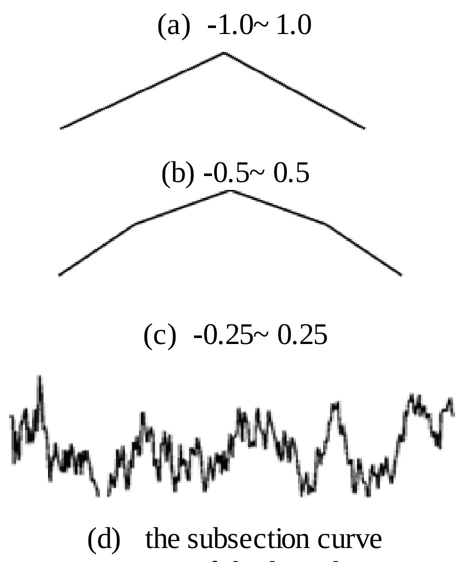


Fig .1 disturbed partition on middle of the line

3.1 One Dimension Midpoint Displacement

Assuming there is a line segment at x-axis. Its change range is [-1.0, 1.0]. First, assuming a random numerical range [-1.0,1.0].Getting a random value in this range, then moving the midpoint of the line segment away like the value. Showing in the Fig.2 (a).Now there are two line segments. the length is half of original. And the random numerical range is halved [-0.5,0.5].Similarly, getting the two line segments' random values in this range. And moving their midpoints like above showing in Fig.2 (b). Then repeating the process, you can get four midpoints, and you can see the effect image in Fig.2(c). The process can be realised by iteration. After a certain number of iterations, the effect image can be seen in Fig.2 (d).



(d) the subsection curve
Fig.2 map of the line change

The process can be described by a flow chart showing in Fig.3.

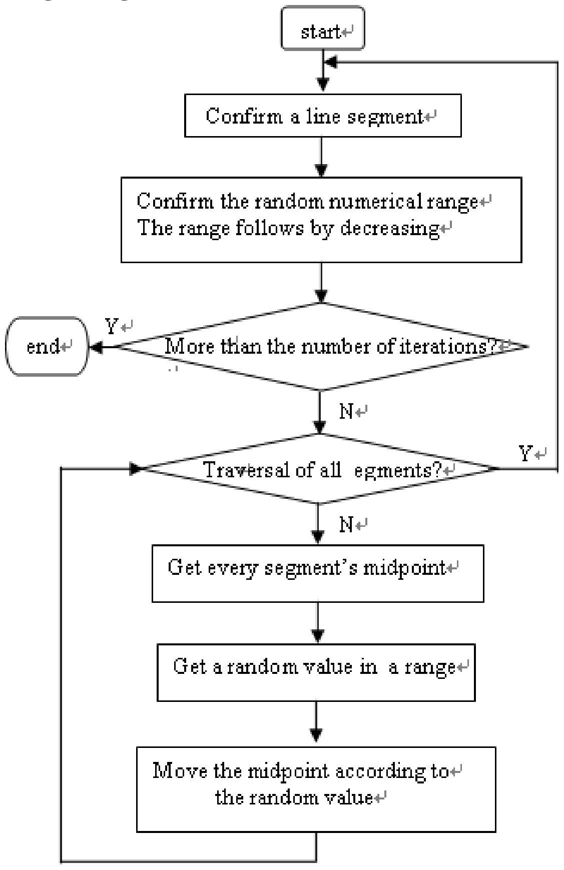

start
Confirm a line segment
Confirm the random numerical range
The range follows by decreasing
More than the number of iterations?
Y
end
N
Traversal of all egments?
Y
N
Get every segment's midpoint
Get a random value in a range
Move the midpoint according to the random value

Fig.3 Flow chart of 1D midpoint displacement

The process is recursive. It can use an iterative process to realize. Do the same operation to every line segment. When iterative times are enough, there can be an ideal graphic.

3.2 Two Dimension Midpoint Displacement

The one dimension midpoint displacement algorithm can be used to generate the ground when it used to rectangular plane. Shown in Fig.4 the method can be described as following:

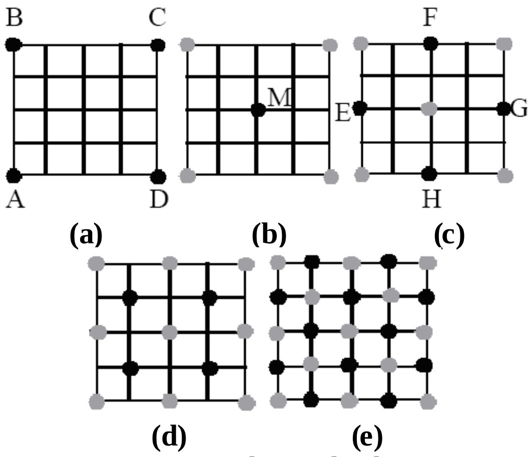


Fig.4 2D midpoint displacement

(1) four vertexes A,B,C and D were designated a height value at a plane in Fig.3(a).

(2) according to the average of four vertexes, we can get the high value of the midpoint M. we can use the formula

$hM = (hA + hB + hC + hD)/4 + \alpha$ .There α is a random.

(3) according to the average of A ,M and B, the high value of E can be got. According to the average of A ,M and D, the H's value will be got. According to the average of B, M and C, the F's value will be got. According to the average of C,M and D, the G's value will be got. That is

$$hE = (hA + hB + hM)/3 + \alpha$$
$$hF = (hA + hC + hM)/3 + \alpha$$
$$hG = (hB + hD + hM)/3 + \alpha$$
$$hH = (hC + hD + hM)/3 + \alpha$$

The α is a random but the four α can not same. But they are in a same range.

(4) according to the average of the small square AEMH's four vertexes, it obtains the midpoint's value and the edge midpoint's value of the small square. Then following the above rules to computer the midpoint and edge midpoint's value of the small square MFCG. And so on, to computer the small square EBFM and HMGD. And it also uses the formula above. Although the α is a random, its value range is smaller than step (3). This is for the needs of vision.

(5) repeat (4), the square grid is refined continually until they reach to the recursive depth.

We draw the results in the x-z plane, then in the y axis direction, a square grid's different point value corresponds to different height. So the mountain can be created.

3.3 Diamond-Square Algorithm

In order simulate the random terrain, we extend to three dimension space. In order to do this, there needs a 2D height value array. The array can map the index value (x, z) for the high value z. The array only needs to save the high value y. And the value of x and z can generate when the array is analysed real-time. A high image can be showed as a picture through appointing a colour to every high. We use Diamond-Square algorithm.

Diamond-Square algorithm [14] is based on the recursive call .The steps are following:

(1) Construction of a large empty 2D array, the array size usually is $2^n + 1$.We can see the array as a square. Then put the four vertexes as a same high

value. The four vertexes are appointed for the black spots. There is a 5×5 array in the Fig.4(a).

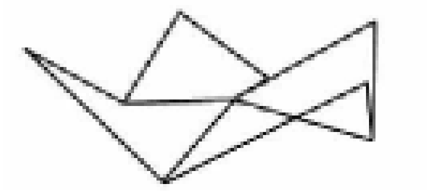
(2) diamond step: get the square with 4 points in the mid-point of the square to generate a random value. The mid-point is the intersection of two diagonal. The value of the mid-point is that the average value of 4 vertexes value plus a random value. Then we get a pyramid. When the grid has many square, it is similar to a diamond.

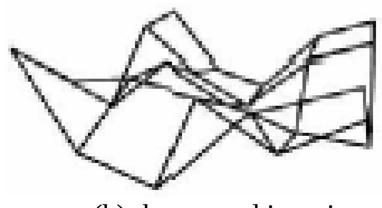
(3) Square step: get a pyramid form every 4 vertexes. Generating a random value from the corner of pyramid. The average value plus the random which is same to the diamond step. We can get every edge's middle value. There will be a square.

Like as, if there has been a seed square, and after a separate process, there is broken down to 4 squares. there will be 16 squares after the second process, 32 squares the third process ,and so on...... the square's number equal to 4^I , I is number of recursive. For the first time after the Diamond step, the middle value is called new value. The new value displays as a black point in Fig.4(b), other points display as grey.

To the Square step, the four pyramids intersect in the centre of the array. So to compute the four Diamond centre, in order to get the base of new value .the same to above, we can see it in Fig.4(c).

These are the first iteration. If we connect those nine points with line, there is a wire-frame surface. It shows in Fig.5 (a).


(a) the first iteration


(b) the second iteration

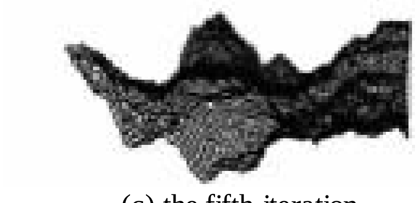

(c) the fifth iteration

Fig.5 the terrain with different iteration time

The next we restart from Diamond step. This time is different to the first iteration. One is there will be four squares. So it needs to compute four middle points. The other is to reduce the range of random from $(-1.0,1.0)$ to $(-0.5,0.5)$.As showing in the Fig.4(d),the four middle value of squares is black.

Then there is the square step again. There are twelve pyramid centres. And compute twelve new values as showing in Fig.4(e). Now there are 25 elements generated. Then it can arrive to the Fig.5(b).

Repeating the Diamond-Square algorithm, it can be broken down more time. Every time there will be more details. For example, we can see the terrain after the fifth iteration in Fig.5(c).

3.4 Improved Diamond-Square Algorithm

We can bring forward a improved algorithm in order to accelerate the speed of terrain's forming. The algorithm is following:

To construct a large empty 2D array. Look the array as a square. Along the diagonal the square is divide up two triangles. The mid-point value is that the average value of the four angles value plus the random.

Then to those triangles do the following:

Get the mid-point of hypotenuse of a triangle. Through the lines of the mid-point with those vertexes, the original triangle is divided into two small triangles. And take the operations above to the two triangles one by one. Until meeting the demands of partition precision.

The process is showed in Fig.6.

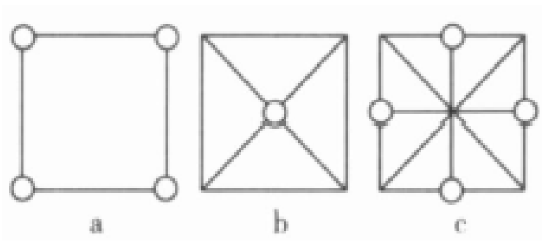


Fig.6 the improved Diamond-Square algorithm

3.5 Simulation of Cloud

The clouds is some kind of complex natural phenomena, there is no well-defined surface and limitation. The predecessors had put forward some generating model for some clouds data[15]-[17], for instance, the light reflection equation method on the bases of physics course, the model designing on the bases of particle system, and the Perline noise function method on the bases of texture mapping technology, same of these methods were based on complex mathematic or physical method, and time consuming; Some would bring some medial data, and have poor image resolution, and it can not meet the require of imaging emulation system.

Cloud data generated model based on random midpoint displacement algorithm is the more effective method. The process is very similar to generated terrain. The main difference is that the image draw in (x,z) plane, and generate different high value at the y axis . But because the require for nice changes on the height, extent, colour of sky and cloud is higher than the terrain, so we should adopt proper amendment to the algorithm[18][19].

We change the higher-field date into opaque date of the cloud. The lowest value point stands for the most bluest and the clearest part of the sky. The highest value point stands for the part where the clouds is thick in the sky, thus we can get a rounded picture of cloud texture.

The particular steps are as follows:

(1) to generate the basic clouds date through Diamond-Square;(2)to figure out the max. and the min of every parameter in the area;(3)to draw the colour of every point in the area according to clouds' colour that decided yet, C_n (the bluest) and the C_0 (the whitest).

The parameter H and σ is the key which decides the distribution of the higher-field increment. As the parameter H reflects the grads' changes of the fractal cloud curve, so we can limit the value range of fractal cloud which is under the colour space RGB, so as to get the changes of the cloud thickness. When H approaches to 1, the colour range of RGB will be wider. The other way round, the colour range of RGB will be narrower when the parameter approaching to 0.

When the σ is fixed, t he grids' changes of H in vertical line plays an important part in controlling the amount of the clouds, so it is possible to control the thickness of cloud by mastering the parameter H. The same time, when the H is fixed, the changes of σ is to control the distributing of fractal cloud.

4 Controlled Terrain Generations

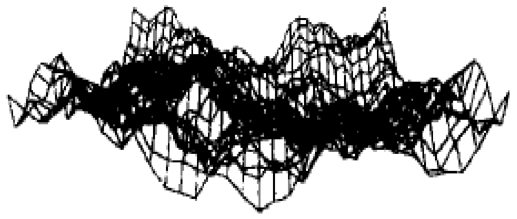
From the formula (2) we can find that the midpoint displacement Δ is relative to parameter H, σ and N. The variance σ is not a random variable. It's value can confirmed when iteration begins. It is fixed while the iteration is going along. It's influence is smaller with the H's influence. So it can be negligible.

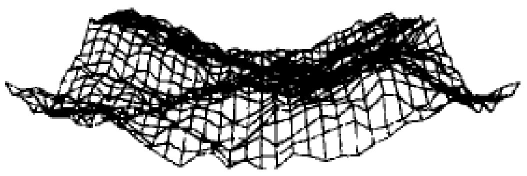
For $0 < H < 1$, so that $\frac{1}{2} < (\frac{1}{2})^H < 1$. And because $(\frac{1}{2})^H$ is reduce with the H's increasing. Under the variance σ is fixed and iteration time is unchanged, $\Delta_n = (1/2)^{nH}\sigma$ will reduce with the H's increasing. So the generated grid terrain will be smoothing with the H's increasing.

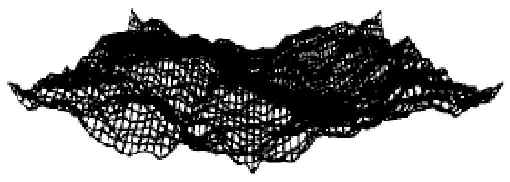
Because $\Delta_{m+1} = \Delta_m \times (\frac{1}{2})^H$, the Δ_n is regressive according to $(\frac{1}{2})^H$'s multiple. That is said iteration time N influxes directly the precision of terrain grid. With the increase of iteration time, the generated grids are finer.

The generated Gaussian distribution is different sequences due to different initial seed value according to the formula (1). And the corresponding random midpoint displacement $\Delta \times Gauss()$ is difference. So there will be generated difference fractal terrain [20].

Based on the above analyses, we know that H, N and seed directly affect the shape of the terrain. With H's increasing, the terrain will be smooth from rough. So it is possible to control the smooth degree through changing the parameter H. Iterative times N can affect reseau's fine degree, and we can receive a satisfied resolving rate by changing the iterative time N. Different initial seed value corresponding to the different shape of terrain, and control the outline and shape of terrain by setting different seed value. In this paper, we can use a dialog box to change those parameters. Controllable fractal terrain could be generated through controlling H, N and seed. In Fig.7 there are serial terrains model which is different parameter values.


H=0.6 N=5 seed=3


H=0.7 N=5 seed=6



H=0.7 N=6 seed=9
Fig.7 different terrains

Like above, we also can control the different cloud model via setting proper parameters.

5 Texture Mapping

The function of texture mapping in OpenGL[21] is to merge digital image and terrain model. Thus it can improve the terrain's expression, enhance the terrain's realistic. Texture mapping technology provides a powerful technology means for simulation. It has two steps that is texture generation and texture mapping.

(1) Picture Texture

Texture mapping is used from pictures. This method is that object surface is covered with arbitrary plane graphics or images, and formed a realistic colour patterns in the surface. The picture may be a flat image, and also may be a light map. This method is simple to implement. But it couldn't achieve realistic results because of the conditions restrict of image resolution, realistic, and so on.

(2) Fractal Texture

When using fractal data to generate texture, it needs high value. We can divide these height values into several ranges based on height. The values in every range use difference colour to draw. The method's advantage is that can get different shade colour texture through controlling the colour of texture. It is propitious to perform the graphics content well. It can reflect the level when the generated texture mapped to the 3D terrain. Peaks, valleys and ridges also can be showed.

5.1 Definition of the Texture

Texture[22] generally is defined above a square domain. We call it as texture space. Defined texture has two methods:

(1) using continuous method to define texture function. The function can be simulated the coarse cloth's texture.

(2) using discrete method to define texture function. This method is simpler. Its main idea is put the dates of texture image into a 2D array. In this paper, we use this method to define the digital image.

After getting the digital image, what we need do is to call the texture function to generate the texture image meeting the texture mapping's demand. If the digital image's width is W, its height is H, each pixel is 8 bits. the information of image data is stored into a 2D array bits. There can generate a 2D texture image by calling the function:

gluBuild2Dmipmaps(type,3,W,H,GL-RGB,GL_UNSIGNED_BYTE,bits).

5.2 Implement the Texture Mapping

The texture mapping technology affix a surface texture to object's surface by using 3D geometric coordinate relationship. It is divided into six steps that is defining texture, controlling texture, setting texture mapping way, activating texture mapping, defining texture coordinate and texture mapping.

On the basis of the texture definition, it merges the defined texture image and terrain model by using of texture mapping. In this paper, the texture mapping is Mipmaps mode. The basic idea is based on the appropriate size of the square to express similar to each pixel's mapping regional in the texture plane. In OpenGL, we can use those functions to realistic the mapping:

gluBulid2Dmipmaps(type,3,W,H,GL_RGB,GL_UNSIGNED_BYTE,bits); // define the texture array

glEnable(GL_TEXTURE_2D);// start texture mapping

glBindTexture(GL_TEXTURE_2D,filename);// merge the texture image and model.

6 Experiment Result

The result after using the Diamond-Square algorithm is a data model. When it has to go through a series of projection transformation and color rendering, the data model can be transform to a realistic image [23].

The program is adopted the Visual C++ 6.0 and OpenGL. The visual process is created through 3D transformation, blanking, light and material.

When the terrain model has been created, we use those functions of OpenGL to simulate the realistic preliminary terrain. Those functions include fractal model construct, blanking, setting light model, and so on...

We call the blanking function:

glEnable(GL_DEPTH_TEST) to blanking the generated fractal model.

The light model is a indispensable part in realistic image display. To creating the light model has relation with the normal vector of object surface. So it must determine the every surface normal vector. Then it needs fit those normal vectors.

Material and light is corresponding. The former reflects the geometry object's light reflection characteristics. The later reflects the light characteristics of projected the object surface.

Calling those functions to set light model:

glEnable(GL_LIGHTING); //start light

glLightModeli(GL_LIGHT_MODEL_TWO_SIZE,GL_TRUE); // set light model

glEnable(GL_COLOR_MATERIAL); //start colour track the material

The next we use texture mapping method to enhance the details. In Fig.8 there is the step of simulating the realistic terrain.

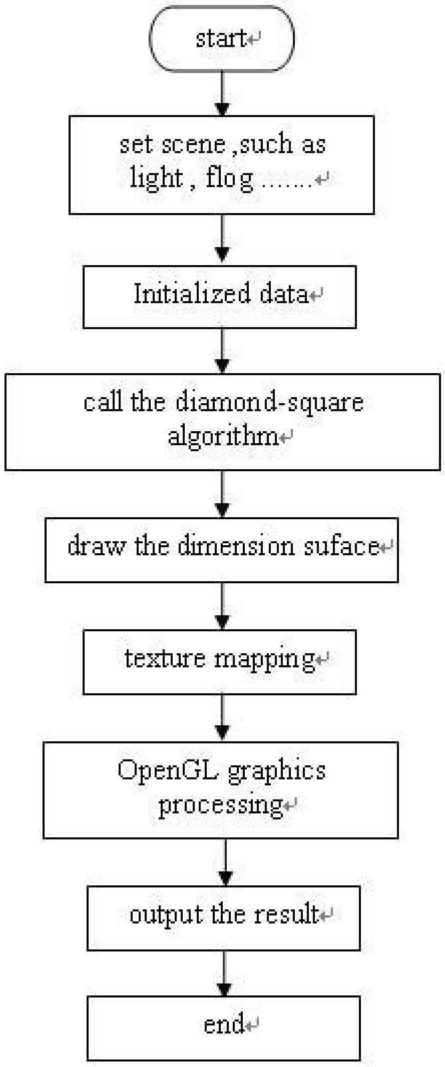

start
set scene ,such as light , flog ....
Initialized data
call the diamond-square algorithm
draw the dimension suface
texture mapping
OpenGL graphics processing
output the result
end

Fig.8 Flow chart of simulating terrain

The program starts form the surface after twice iteration. The surface is drawn by line. Connecting the array values with lines. It can use the triangles to draw the surface through dividing the square into two triangles. Figure 9 is a generated terrain image.

Adding sky and cloud texture to the terrain, there will generate a more complex terrain. Other more, we can change the parameters H, random seed and iteration time N to create different simulation terrain

The terrain image is simulated in the Fig.10 when its parameters as follows:

loud_H=0.7,cloud_N=3,terrain_H=0.7,terrain_N=3,surface_H=0.7,surface_N=3.

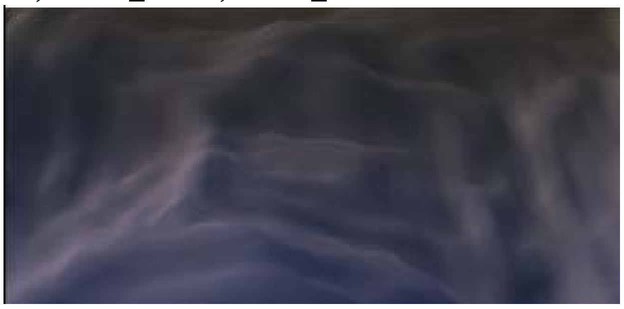


Fig.9 terrain effect

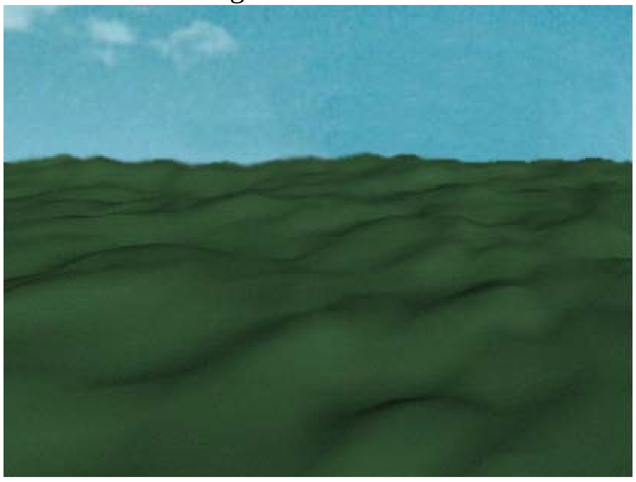


Fig.10 the 3D terrain

7 Conclusion

Fractal theory has been widely application in many fields. It has obvious advantages in the simulating the ruleless nature. In this paper the random midpoint displacement method based on FBM is discussed. Midpoint displacement is standard fractal geometry method. It can be used to generate the terrain quickly. It uses the interpolation between two points or more points to model in the subdivision process. But there is an obvious signs at the adjacent region. We call it as crease problem[24]. The crease couldn't disappear even adding more details. Although this is midpoint displacement method's short, but it's capacity of high speed and adding detail. It has been a useful fractal theory.

The algorithm implements its function depending on the recursive. Though the above discussion, we can see that the total number of the grid will be four times after a recursive. In other words, with the increasing of the recursive number, the grid will increase rapidly. So it occupies a large number of system resources. The running speed is slower than the last. There is a time chart of number different recursive in Fig.11. X axis represents the number of initial grid. And y axis represents the time in this figure[25]. From this figure, we can see that the program's running time is considerable while the grid number is very larger.

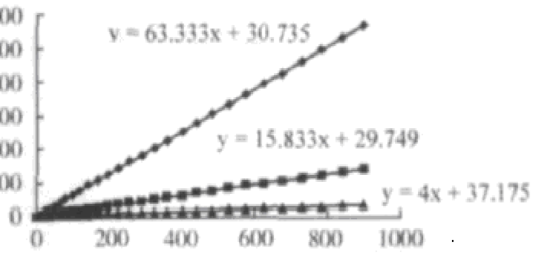


Fig.11 time in different recursive number

In order to display the complex terrain quickly, the method reducing the recursive times can be adopted. Generally the recursive time controlled within three times can meet all the requirements.

From the above analysis and examples proving, when the data is collected incomplete or even fewer, random fractal interpolation method renews the real structure of raw data a certain extent.

FBM is fit to creating the rough terrain. It has available in the Geographic Information System (GIS), Digital Model (DIM) and Virtual Reality (VR), especially in 3D games.

References:

[1] Mandelbrot B.B. *the Fractal Geometry of Nature*, San Francisco: W.H Freeman and Co,1982.

[2] Wang Dongshen,Cao Lie, *Chaos, Fractal and application*, University of Science and Technology of China Press,1995

[3] Qi Dongxu, *Fractal and its Computer generation*, Beijing Science and technology press, 1996

[4] Peng qunsheng, Bao hujun, Jin xiaogang, *the Algorithm Basic of Computer Realistic Image*, Science press,1999

[5] Falconer KJ, *Fractal Geometry: Mathematical Foundation and Application*, New York: Wiley, 1990

[6] Dai Qianwei, Peng Zhenbin and He Jishan, Fractal Interpolation Based on FBM, *the Chinese Journal of Nonferrous metals*, Vol 8,No.4,1998,pp.719-722

[7] Gavin S P Millier. The Definition and Rendering of Terrain Maps. *Computer Graphics*, Vol 20, No.4,1986, pp. 39-48

[8] J E Dudgon, R Gopalakrishnan.Fractal Based Modeling of 3D Terrian Surface. *Proc of IEEE*,Vol 84,No. 4,1996, pp.246-252

[9] Kenneth J. *Falconer Geometry Mathematical Foundations and Application*.NEU press, 2003

[10] Zhang jizhong, *Fractal* . TsingHua University Press, 1995

[11] QI Min, HAO Chongyang, TONG Mingan. Research of Modeling Method to 3D Multiresolution Terrain Based on Fractal, *Journal of Image and Graphics*, Vol 5, No.7,2000, pp.568-572

[12] Liang Jue, Wang qi, Lui kunliang, Lu quanhue. 3D Terrain Simulation Based on the Method of Random Midpoint Displacement, *Computer simulation*, Vol 1 ,No.22,2005, pp.213-215

[13] Li Yaohui, Zhou Lili, Generating of Simulate 3D Terrain of Virtual Reality, *Control & Automation*,Vol.22.No.10,2006, pp.280-282

[14] Chen Hongying, Xiong Hongyan, Mao Gefei, Computerized Implementation of a Fractal Interpolation Algorithm for Graph Simulation, *Microcomputer Application*, Vol 20,No.8,2004, pp.36-38

[15] Blinn J F, Light reflection Function for simulation of clouds and dusty surface. *Computer Graphics*, Vol 16,No.3,1982, pp.21-29

[16] Matthias U, Andrzej T , Cloud Simulation in Virtual Environments. *IEEE Visualization Proceedings MelBoume*,1998

[17] Qi Yue, Shen Xukun,Yin Miyi,Cheng Huiling. a Method of Rendering Clouds with Perlin Noise, *Journal of System Simulation* ,Vol 14 ,No.9 2002, pp.1204-1207

[18] Shi Jiandi, Jiang Yuming. Simulation of Dynamic Cloud Based on Fractal Geometry. *Computer Simulation*, Vol 23,No.4,2006, pp. 197-200

[19] Gu daqian, Fan Yin, Xu Ping, Gong Lin. Simulating Algorithm of Moving Clouds, *Meteorological Science and Technology*, Vol 34,No.2, 2006, pp. 99-103

[20] Li Qingzhong, Gao Xiurong. Study on Generating Method for 3-D Controllable and Realistic Terrain, *Journal of System Simulation*, Vol 20 No.11,2008, pp.2938-2941

[21] Dov of peace studio, *OpenGL high-level program and visualization system development*,.China WaterPower Press,2006

[22] Tang Bin. Research of the Texture Mapping Based on OpenGL, *Research and Exploration in Laboratory*., Vol 25,No.5,2006,pp.40-41

[23] Xue An, Ma Ai-nai,Li Tian-hong. A study on the Representation of Terrain Realistic Image

Based on OpenGL, *Journal of Image and Graphics*,Vol 6(A) No. 8 ,2001 pp. 801-805

[24] Mohsen Sharifi, Fatemeh Hshemi Golpaygani, Mehdi Esmaeli. A New Fractal-Based Approach for 3D Visualization of Mountains in VRML Standard,*Proceeding of the 2nd international conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. Singapore, IEEE Computer Society*,2004, pp. 100-105

[25] Liang Jun, Jiang Jilong, Zhao Xueliang.The Applicability Analysis of MPD Method in 3D Terrain Interpolation, *Science of Surveying and Mapping*.,Vol 32,No.3,2007, pp.44-46