Method of Geometry Clipmap based on icosahedron

CHEN Mengyun, MENG Xin, PENG Xiaodong

Department of Information Simulation, National Space Science Center of China, Beijing 100190, *China*

Abstract: Large terrain rendering is hotspot in Computer Graphics, especially the spherical terrain rendering, which is more complicated in shape and data organization. Based on researching in spherical terrain rendering method, we propose a new spherical terrain rendering method using Geometry Clipmap. First construct an icosahedron model, and then split the icosahedron into ten main diamonds, and use spherical Sphere Quad Trees (SQT) model to ensure Geometry Clipmap can be used in every diamond, what else, grids around the main diamond are partitioned and composed to a bigger virtual diamond terrain in 3×3 size, which enlarge clipmaps' extent and solve the boundary crossing problem to a extent, and the Geometry Clipmap can be used in the big diamond. The result illustrates feasibility and effectivity of the method.

Key words: spherical terrain rendering, Geometry Clipmap, icosahedrons, Sphere Quad Trees (SQT)

CLC number: TP391.41 **Document code:** A

Citation format: Chen M Y, Meng X and Peng X D. 2014. Method of Geometry Clipmap based on icosahedron. Journal of Remote Sensing, 18(5): 1059-1071 [DOI: 10.11834/jrs.20143141]

1 INTRODUCTION

Massive terrain rendering has drawn considerable attention in graphic research. Particularly, spherical terrain rendering is more complicated than flat terrain rendering in terms of exploring space, mapping and data organization. However, many scholars are still focused on spherical terrain rendering.

The size and amount of data and the features of spherical geometry must be considered when designing a real-time spherical terrain rendering method. Previous studies developed real-time terrain simplification and split spherical and data organization methods. Current typical terrain simplification methods include the ROAM algorithm (Duchaineau, et al., 1997), Lindstroms continuous level of detail algorithm (Lindstrom, 1996, 1997, 2001), De Boer's geomipmapping algorithms (de Boer, 2000), and Geometry Clipmap (Losasso & Hoppe, 2004). Typical methods include spherical mesh with the organization's earliest and most conventional method of latitude and longitude mesh; a regular polyhedron is established with global grid model-based methods, such as Quaternary Triangular Mesh (QTM) (Dutton, 1996), Sphere Quad Trees (SQT) (Fekete & Treinish, 1996), and projection-based Icosahedron Snyder Equal Area (ISEA), which is based on the entire area projection of icosahedron (Snyder, 1992). In real-time terrain simplification, Geometry Clipmap has better coverage and faster updating shape efficiency than quadtree, ROAM, and other methods. However, only a few scholars have studied the capacity of Geometry Clipmap for spherical terrain rendering. Clasen and Hege (2006) presented a more classical spherical selection of Geometry Clipmap, which is in good agreement with the spherical shape, but each frame of terrain data resampling requires a relatively large real consumption. Bhattacharjee (2010) proposed the QTM and Geometry Clipmap combining method, which basically realizes Geometry Clipmap in spherical terrain. However, the regular octahedron subdivision of equal area is not satisfactory. Li, et al. (2010) used latitude and longitude with space division and other methods to achieve the approximate area of the space division of the deformation effect on large poles. However, they did not elaborate on how to resolve cross-border issues.

This study proposes a method based on terrain mapping Geometry Clipmap icosahedron partition to balance the global mesh in the area of the shape. The characteristics and specific algorithm of spherical subdivision methods are also discussed. The main problem in the icosahedral split with Geometry Clipmap is explored when combining virtual diamond solution to obtain preliminary simulation results.

2 ALGORITHM DESCRIPTION

Geometry Clipmap is based on the viewpoint of Level of Detail (LOD) simplification strategy, which consists of a set of nested viewpoint-centered square grids. Each square has the same structure, but LOD level increases with the triangular mesh constituting each square grid. Although Geometry Clipmap has advantages, it causes serious distortion in the polar regions of a spherical terrain when applied to the spherical surface using latitude and longitude grids. Therefore, a polyhedron split sphere is used to ensure that surface polyhedron clipmap obtains accurate results within each subdivision. However, terrain data from clipmap cannot normally transit when crossing the polyhedron faces

Received: 2013-06-05; **Accepted:** 2014-04-15; **Version of record first published:** 2014-04-22

First author biography: CHEN Mengyun (1987—), female, master candidate, she majors in computer graphics. E-mail: chenmengyun1987@163.com

because the sphere surface has different connectivity and planes. Bhattacharjee presented cross-border issues in the regular octahedron subdivision solution when clipmap is across the two faces of the octahedron; both faces establish a clipmap, drawing generated waste. The split in icosahedron has better performance than that in octahedron. Therefore, we used the Geometry Clipmap terrain rendering method based on the subdivision of the icosahedron. With five triangles surround each osahedron vertex, the terrain data cannot be updated correctly when the clipmap crosses the border vertex topology, making the split in icosahedron more difficult than that in octahedron.

Thus, we proposed a virtual diamond solution based on i cosahedral global subdivision with Geometry Clipmap effective combination. Briefly, vertex division and organizational methods are described with icosahedron, and the Geometry Clipmap is e xplored based on this data structure.

2.1 Spherical vertex division and organization

Global subdivision in icosahedron consists of three parts: conversion of space coordinates and vertex calculations, polyhedron subdivision and inverse projection computation of grid, and vertex projection computation.

2.1.1 Icosahedron subdivision of global grid

Our method was based on the spherical triangle quadtree model's isometric projection (Yuan, et al., 2009). The unit sphere has a radius of one, and two vertices coincide with the poles. The sphere's three vertices on the surface of the inner icosahedron's latitude and longitude are shown in Appendix Table. Cartesian coordinate values can be obtained through the translation. Twenty spherical trigonometric figures are formed by each side of icosahedron been projected. Each icosahedral surface projection plane of a spherical triangle is defined. The triangular projection plane is e venly divided into uniform triangular grids. The coordinates of the sphere vertices and edges are obtained by the inverse projection formula.

The triangular projection plane is evenly divided as below. The upper and lower triangular projection planes are combined into a diamond shape (Fig. 1), which is formed of 10 diamonds, where each diamond mesh is subdivided into n depth. Therefore, each side of the projection plane is divided into two $2n$ equal parts, forming parallel lines that intersect to generate grid vertices corresponding with the Cartesian coordinate system. Each diamond consists of $(2n+1) \times (2n+1)$ vertices. The formula for calculating the vertices on the spherical surface is to establish the relationship between the spherical triangle and the projection of spherical triangle, as shown in Fig. 2, P is any point on the sphere that is projected onto the projection plane called G, arc MPN project to projection plane generates $\boldsymbol{EF}$, and $\boldsymbol{EF}$ parallels to $\boldsymbol{BC}$. The ratio of the length of line segment $\boldsymbol{EF}$ to $\boldsymbol{EG}$ is x, and the ratio is equal to upper corner angle EOG to EOF. The ratio of the length of line segment $\boldsymbol{AB}$ to $\boldsymbol{AE}$ is y, and the ratio is equal to upper corner angle EOB to AOB (Yuan, et al., 2009). Each projection of the triangle's A, B, and C coordinate values are known, and P point coordinates can be derived from Eq. (1) and Eq. (2).

$$\boldsymbol{OE} = \boldsymbol{OA} + y(\boldsymbol{OB} - \boldsymbol{OA}) \quad (1)$$

$$\boldsymbol{OF} = \boldsymbol{OA} + y(\boldsymbol{OC} - \boldsymbol{OA}) \quad (2)$$

G coordinates are derived from Eq. (3).

$$\boldsymbol{OG} = \boldsymbol{OE} + x(\boldsymbol{OF} - \boldsymbol{OE}) \quad (3)$$

$\boldsymbol{OG}$ is normalized to multiply the radius of the sphere to o btain the coordinates of point P. The mesh vertices on the projection plane have been previously divided well, where $x = i/(i+j)$, $y = (i+j)/2n$ (i, j is the culmination of grid coordinates on the projection plane, n is divided depth).

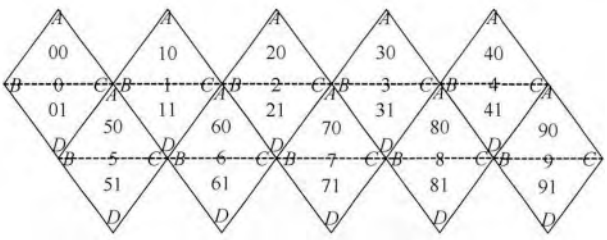


Fig. 1 Icosahedron of sphere expanded to 10 diamonds

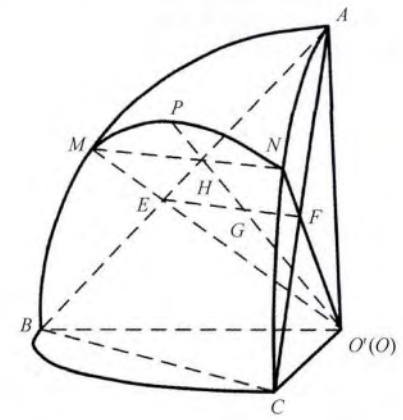


Fig. 2 Spherical trigonometry and perspective projection triangular plane (Yuan, et al., 2009)

2.1.2 Calculating coordinates of viewpoint relative to projection surface

Using the Geometry Clipmap view algorithm to generate a range centered on a uniform grid is necessary to identify the c oordinates of the viewpoint on the projection grid relative to the coordinates of the center, thus generating clipmaps.

As coordinates of each icosahedron face, point P is first c alculated from the sphere center of each face of the icosahedron, i. e., the shortest distance where the plane surface of an arbitrary point P is located. Calculating the coordinates of point P relative to the projection surface is easier than using the simplified isometric projection method. With the triangle above as an example, the relative position in the grid computation and sphere center point P is a connection point on projection plane P of projection point Q (Fig. 3). Two lines are made through point Q parallel to $\boldsymbol{AB}$ and $\boldsymbol{AC}$ separately and cross $\boldsymbol{AC}$ and $\boldsymbol{AB}$ on T and S points. Q points relative coordinate transform is found to calculate $\boldsymbol{AS}$, $\boldsymbol{AB}$, and $\boldsymbol{AC}$ and obtain the percentage of $\boldsymbol{AT}$. To simplify, the operation can be projected onto a plane that is n ormal to the largest component of the two-dimensional plane into two-dimensional calculations.

If $|\boldsymbol{AS}|/|\boldsymbol{AB}| = \alpha_1$,

$$\boldsymbol{OS} = \boldsymbol{OA} + \alpha_1(\boldsymbol{OB} - \boldsymbol{OA}) = \boldsymbol{OQ} + \beta_1(\boldsymbol{OC} - \boldsymbol{OA}) \quad (4)$$

Elimination of β_1 solution of equations can be obtained with

α_1. Similarly, let $|AT|/|AC| = \alpha_2$,

$$OT = OA + \alpha_2(OC - OA) = OQ + \beta_2(OB - OA) \quad (5)$$

Two simultaneous equations can be obtained with α_2. Multiplying by the number of vertices ranks, we can obtain the relative coordinates with Q points on the projection plane.

The relative coordinates of point Q are obtained to establish clipmap with center point Q, and the clipmap size is determined. The coordinates of each vertex in the clipmap projection plane can be identified by using the method in Section 2.1.1 of the vertex to the inverse sphere to form a spherical clipmap.

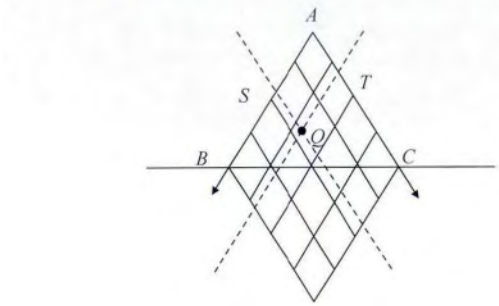


Fig. 3 Projection plane relative coordinate computing schematic

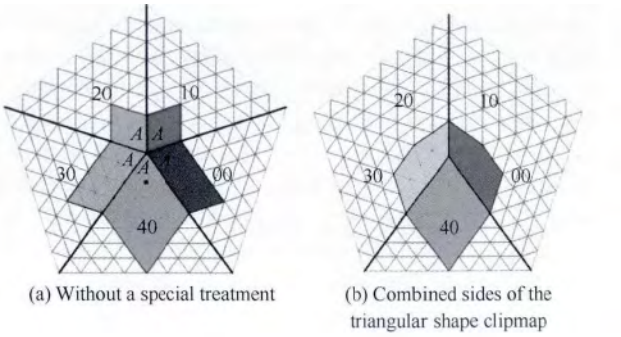
2.2 Realizing icosahedron spherical subdivision based on Geometry Clipmap rendering

2.2.1 Virtual diamond region algorithm

After the icosahedron expand to two-dimensional graphic in Fig. 1, establishing diamond mesh grid units to create clipmap centric viewpoint will be presented as a diamond. When clipmaps move within a diamond range, clipmap shapes become diamond-like. Around each of the four vertices of the four vertices are diamonds adjacent the periphery of the rectangular distribution is rectangular distribution type, the vertex of an acute and obtuse vertex both cases, acute apex acute angle into the surrounding four rhombus vertex coincides acute vertices and around two diamond obtuse vertex coincides in both cases, only the apex obtuse angle with the side of a diamond obtuse acute angle vertex and vertex coincides with a diamond case. When the diamond is across the clipmap boundary, clipmap vertices move to acute or obtuse apex around but do not maintain complete rectangle data structure which may generate data increase or decrease. Bhattacharjee established Geometry Clipmap based on octahedron, in which the diamond in the cone is a separate terrain. A diamond is created on each clipmap when a plurality of diamond within the cone draw clipmap is repeated. An icosahedron contains 10 diamonds. Hence, the applied method to the algorithm of the current study will consume largely, and icosahedron application will produce deformed cross-border, or worst around the pole where this method cannot be used.

In each vertex of the icosahedron, the center plane of five triangular projections may be formed around a pentagon. In the fourth diamond, for example, distribution analysis of diamond mesh vertices is shown below.

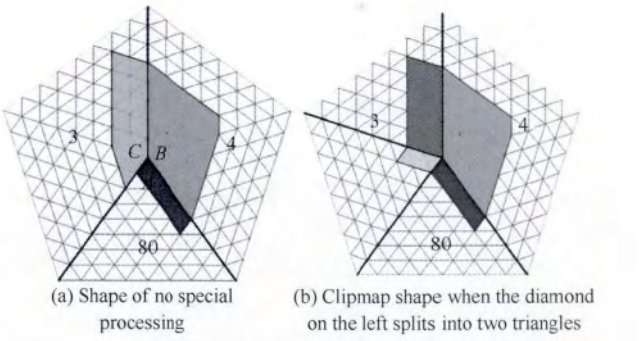
(1) When the clipmap center gets close to the diamond 4's vertex A, the acute angel vertex coincides with neighbor diamonds' A vertices. The clipmap shape would be like a five-pointed star such as Fig. 3(a) if do not deal with the crossing boundary problem. The increasing data lead to serious deformation. Merge diamond 2's top triangle 20 and diamond3' top triangle 30 to a new diamond, in the mean time, merge diamond 1's top triangle 10 and diamond 0's top triangle 00 to a new diamond as shown in Fig. 3(b). The clipmap would be rendered in new order, and the deformation is much less.



(a) Without a special treatment (b) Combined sides of the triangular shape clipmap

Fig. 4 Schematic of an acute angle 40 clipmap center near the apex

(2) When the clipmap center gets close to the diamond 4's vertex B, the obtuse angle vertex coincides with diamond 3's C vertex and diamond 8's A vertex. There would be big sawtooth shape caused by losing data and getting bigger as the clipmap size increases. If we split diamond 3 into 2 projection triangle 30 and 31 can satisfy the circumstance stated in last section as shown in Fig4(b). The situation is similar when clipmap center gets close to vertex C.



(a) Shape of no special processing (b) Clipmap shape when the diamond on the left splits into two triangles

Fig. 5 Schematic when clipmap center near the obtuse

(3) When the clipmap center gets close to the diamond 4's vertex D, the acute angle vertex coincides with diamond 8's C vertex and diamond 9's B vertex. The deformation of crossing boundary is similar with which is in Fig. 4(b), we could do nothing about it.

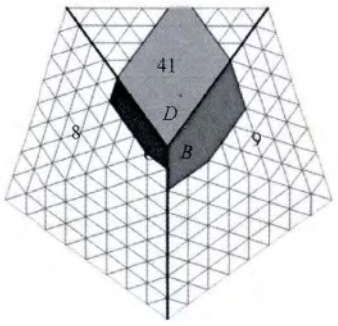


Fig. 6 Shape of clipmap center near NO. 41 obtuse

This study considers the diamond in Fig. 1 around a main diamond with special treatment, split or merge, and clipmap mesh that can better maintain the original shape of the resulting structure clipmap which expanded continuously active in clipmap range, resulting in strain in a tolerable range.

Taking the diamond clipmap center lays in as the main d iamond to construct 3 ×3 big diamond: Each diamond is consisted of two right triangles, which numbered as Fig. 1. As shown in Fig. 6, split the origin triangle and merge into new one. As shown in Fig. 6, take diamond4 as the center, the down left and the down right diamonds remain unchanged, the diamond 1 splits, 01 and 50, 00 and 10 merge into a new diamond. The diamond 3 splits, 30 and 20, 31and 70 merge into a new diamond, which satisfy circumstance in Fig. 3 and Fig. 4. When clipmap center moves in diamond 4, clipmap's activity restrict expands to size 3 ×3 diamond territory. Generate new big diamond territories d epend every little diamond, when clipmap center move to a nother little changes. The construct method differs between 5—9 diamonds and 0—4 diamonds. Diamond 5's top diamond remain and diamond 6 split into 60 and 61, and construct new diamonds with 21 and 71. Diamond 9 can split into 90 and 91 and construct diamonds with 41 and 81. Summary the construct virtual big diamond rules as show in Fig. 7 and Fig. 8.

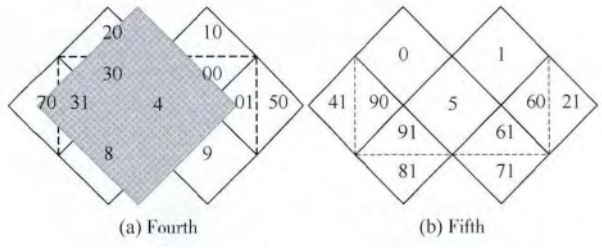


Fig. 7 Different diamond as the center of a virtual diamond mesh

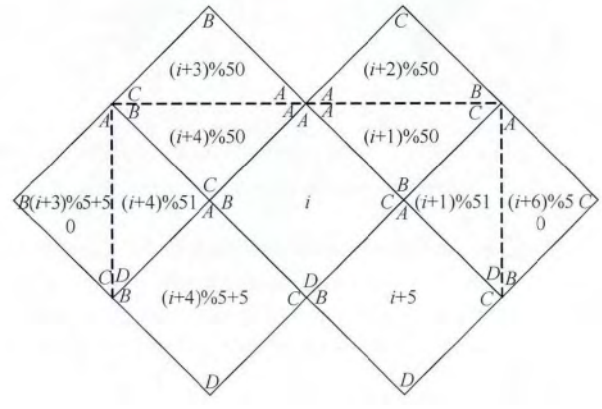


Fig. 8 Zero to fourth diamonds as the center of the diamond to create a virtual grid index number schematic

When initialize big diamond territory, construct 3 × 3 d iamonds. We need to build ten such diamonds territories. Each main diamond's four vertices' coordinates depend on triangle. As shown in Fig. 7, the diamond i, left down diamond $[i+4)\%5+5]$ and right down diamond $[i+5]$, which have the same vertices with the origin diamonds. The top left diamond's vertices $[A,B,C,D]$ are diamond $[i+3)\%5]$'s vertex B, diamond $[i+4)\%5]$'s vertex C, diamond $[i+4)\%5]$'s vertex B (the same with diamond $[i+3)\%5]$'s vertex C) and diamond $[i+4)\%5]$'s v ertex A(the same with diamond $[i+3)\%5]$'s vertex A).

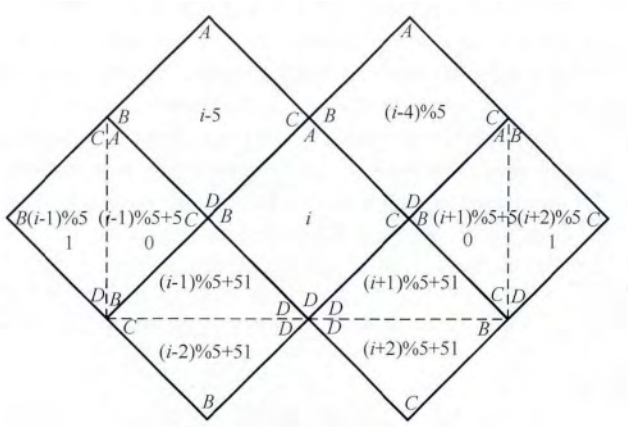


Fig. 9 Fifth to ninth diamonds as the center of the diamond to create a virtual grid index number schematic

Rhombus vertex generation is summarized in the region shown in Fig. 10. When the vertex coordinates of the corresponding region of small diamond are $i\in(0,n)$ and $j\in(n,2n)$, the vertices of the large diamond-shaped area relative coordinates are $I=i$, $J=j-n$, when $i\in(n,2n)$ and $j\in(0,n)$ when, $I=i-n$, $J=j$. The inverse projection in spherical coordinates is used when c alculating the relative coordinates.

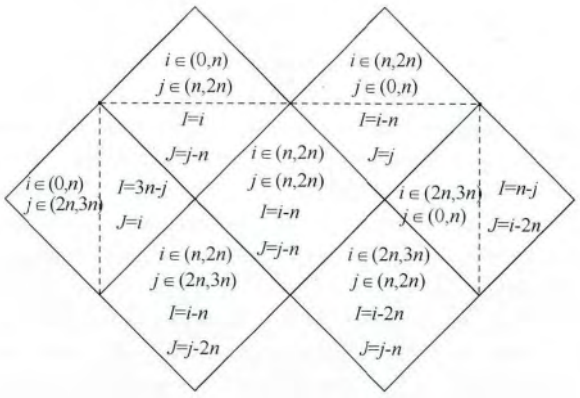


Fig. 10 Vertices generate a large diamond-shaped area within the virtual schematic

Generating this type of grid has an advantage. While the matching of the regular grid mesh shape with clipmap to solve cross-border problems of clipmap is no longer confined within the range of the maximum activity of a diamond. A diamond can cover the center of the seven diamonds, which is enough to cover half the globe. With these seven diamonds processing as a single piece of terrain rather than a separate seven, seven clipmaps do not need to be drawn seven times to improve efficiency.

2.2.2 Realizing Geometry Clipmap

Similar to the original Geometry Clipmap generating method, the method of generating clipmap is based on icosahedrons. A single terrain consists of a clipmap tree that is associated with the viewpoint, defining the highest level of clipmap that has the lowest resolution, which means the largest grid spacing, in which

each layer has the same number of vertices clipmap, each successively lower clipmap grid spacing. Different from the original algorithm, a diamond clipmap is not composed of a square grid.

Between two different grids, resolution will lead to cracks. Losasso and Hoppe's original algorithm in each layer of data is added to their height than the height of the point values that are also stored in the parent layer (rough layer). Mixing the vertex shader is highly desirable to the height of the fracture to eliminate the transition zone. Using vertex shader calculating the height value of transition region is complicated, we prefer to reduce resolution of more detailed clipmap's boundary so that to connect to the less detailed level. Vertex data update method is similar to the original algorithm in a wide range of diamond "L" zone updates. When clipmap crosses a large diamond range, memory data block should be updated.

3 SIMULATION RESULTS AND ANALYSIS

According to the above method, we use an ordinary PC containing Windows XP system, Visual Studio 2010, and DirectX9.0 to achieve a rendering algorithm. The machine is configured to 3.1 GHz Intel i5-2400 and GeForce GTX460 graphics cards. In experiments, elevation used 2500 × 1250 resolution global elevation map, clipmap size is 129 × 129, and the number of layers is six, in which the frame rate achieved 60 frames/s. Fig. 11 presents the simulation results that can be obtained when a diamond is in the range clipmap. When the clipmap rhombus generation rules and occurs within the cross, clipmap will produce some distortion but does not affect the observed effect.

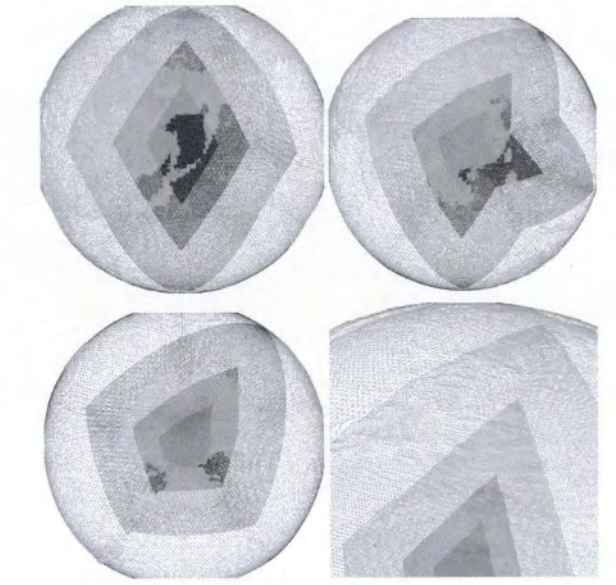


Fig. 11 Geometry Clipmap spherical terrain simulation results are different when the clipmap center moves at different circumstances

Cutting the two acute angles of the rhombus clipmap will yield a regular hexagon clipmap, which has the same distance in all directions to achieve approximately the same resolution (Fig. 12). This result is similar to Bhattacharjee's simulation results (Fig. 13).

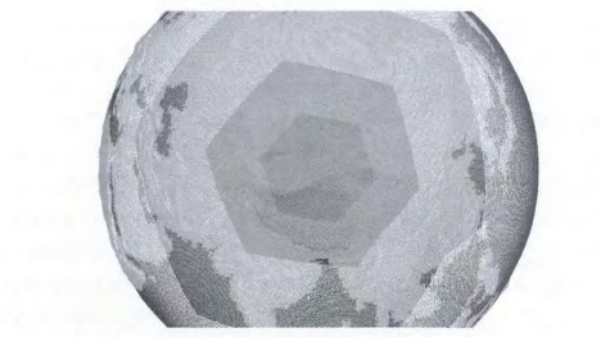


Fig. 12 Simulation results

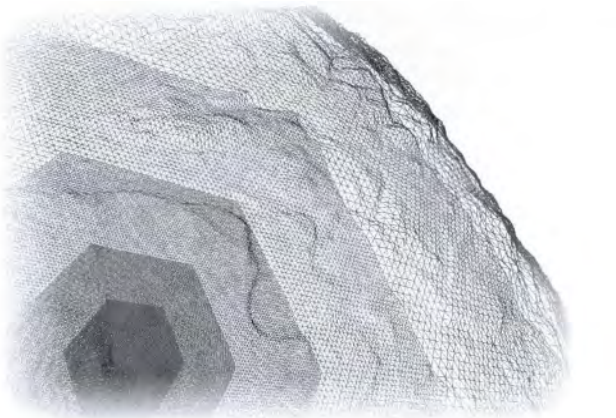


Fig. 13 Simulation results established by Bhattacharjee

4 CONCLUSIONS

In this study, we use icosahedron subdivision to build a global model of the triangular projection plane and to obtain uniform spherical mesh vertices by inverse projection calculations. We construct a new grid structure to the center of the diamond where the clipmap is centered around the diamond split and reassembled into a new virtual large area diamond. The seven regions are combined into a terrain treatment. To some extent across multiple diamond clipmaps that solve the problem of deformation, the degree of deformation is within tolerance range. Icosahedron-based global subdivision and Geometry Clipmap preliminary methods are combined.

This study is the first to apply Geometry Clipmap to icosahedron subdivision from spherical grid. In a fixed block-based diamond of 3 × 3 region of the aforementioned large-initialization, the main clipmap is away from the center of a diamond and to another piece of the main diamond stitching method because the large area diamond vertices are connected in a different order, which will result transitions when difference from one diamond block clipmap center moves to another block clipmap diamond-shape. Clipmap vertex data must have all updates. Frequent updates throughout the vertex data to draw clipmap rate have a certain influence. Therefore, the next step plan is to reduce strain design algorithms plurality while the clipmap diamond moves on the time, establishing the shape of clipmap with the same distance in each direction consistent resolution, to create a dynamic large diamond-shaped region from the center when the clipmap is transferred to another area of the triangle. The transfer

is based on the current location of the triangle area to construct a larger 3 × 3 diamond area and thus improve rendering efficiency.

REFERENCES

Bhattacharjee S. 2010. Real-time Terrain Rendering and Processing. International Institute of Information Technology Hyderabad, INDIA

Clasen M, Hege H C. 2006. Terrain rendering using spherical clipmaps // Proceedings of the Eighth Joint Eurographics / IEEE VGTC conference on Visualization. Switzerland: Eurographics Association Aire-la-Ville: 91 – 98 [DOI: 10.2312/VisSym/EuroVis06/091 – 098]

de Boer W H. 2000. Fast Terrain Rendering Using Geometrical Mipmapping [EB/OL]. [2013-09-20] http://www.flipcode.com

Duchaineau M, Wolinsky M, Sigeti D E, Miller M C, Aldrich C and Mineev-Weinstein M B. 1997. ROAMing Terrain: Real-TimeOptimally Adapting Meshes. Technical Report UCRL-JC-127870, Lawrence Livermore Nat1 Laboratory

Dutton G. 1996. Improving locational specificity of map data: a multi-resolution, metadata-driven approach and notation. International Journal of Geographical Information Science, 10(3): 253 – 268 [DOI: 10.1080/02693799608902078]

Fekete G and Treinish L A. 1990. Sphere quadtrees: a new data structure to support the visualization of spherically distributed data // Proceedings of the SPIE, Extracting Meaning from Complex Data: Processing, Display, Interaction, 1259: 242 – 253 [DOI: 10.1117/12.19991]

Li W X, Chen G, Huang B X. 2010. Proceedings of 2010 International Conference on Remote Sensing (ICRS 2010) Volume 3

Lindstrom P, Koller D, Ribarsky W, Hodges L F, Faust N and Turner G A. 1996. Real-time, continuous level of detail rendering of height fields // Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM: 109 – 118 [DOI: 10.1145/237170.237217]

Lindstrom P, Koller D, Ribarsky W, Hodges L F, Op den Bosch A and Faust N L. 1997. An Integrated Global GIS and Visual Simulation System. Graphics, Visualization, and Usability Center, Georgia Institute of Technology

Lindstrom P and Pascucci V. 2001. Visualization of large terrains made easy // Proceedings of Visualization. San Diego, CA, USA: IEEE: 363 – 574 [DOI: 10.1109/VISUAL.2001.964533]

Losasso F and Hoppe H. 2004. Geometry clipmaps: terrain rendering u sing nested regular grids // Proceedings of ACM SIGGRAPH. New York: ACM: 769 – 776 [DOI: 10.1145/1015706.1015799]

Snyder J P. 1992. An Equal-Area map projection for polyhedra globes. Cartographica, 29(1): 10 – 21 [DOI: 10.3138/27H7 – 8K88 – 4882 – 1752]

Yuan W, Zhuang D F, Yuan W and Liu J Y. 2009. Some essential questions in remote sensing science and technology. Journal of Remote Sensing, 13(1): 103 – 111 [DOI: 10.3321/j.issn:1007 – 4619.2009.01.014]

APPENDIX TABLE

Vertices of icosahedron

No.	Vertex 0		Vertex 1		Vertex 2		Center point	
	longitude/°	latitude/°	longitude/°	latitude/°	longitude/°	latitude/°	longitude/°	latitude/°
0	0.0	90.0	0.0	26.565	72.0	26.565	36.0	52.623
1	0.0	90.0	72.0	26.565	144.0	26.565	108.0	52.623
2	0.0	90.0	144.0	26.565	216.0	26.565	180 0	52.623
3	0.0	90.0	216.0	26.565	288.0	26.565	-108.0	52.623
4	0.0	90.0	288.0	26.565	0.0	26.565	-36.0	52.623
5	36.0	-26.565	0.0	26.565	72.0	26.565	36.0	10.812
6	108.0	-26.565	72.0	26.565	144.0	26.565	108.0	10.812
7	180.0	-26.565	144.0	26.565	216.0	26.565	180.0	10.812
8	252.0	-26.565	216.0	26.565	288.0	26.565	-108.0	10.812
9	324.0	-26.565	288.0	26.565	0.0	26.565	-36.0	10.812
10	72.0	26.565	36.0	-26.565	108.0	-26.565	72.0	-10.812
11	144.0	26.565	108.0	-26.565	180.0	-26.565	144.0	-10.812
12	216.0	26.565	180.0	-26.565	252.0	-26.565	-144.0	-10.812
13	288.0	26.565	252.0	-26.565	324.0	-26.565	-72.0	-10.812
14	0.0	26.565	324.0	-26.565	36.0	-26.565	0.0	-10.812
15	0.0	-90.0	36.0	-26.565	108.0	-26.565	72.0	-52.623
16	0.0	-90.0	108.0	-26.565	180.0	-26.565	144.0	-52.623
17	0.0	-90.0	180.0	-26.565	252.0	-26.565	-144.0	-52.623
18	0.0	-90.0	252.0	-26.565	324.0	-26.565	-72.0	-52.623
19	0.0	-90.0	324.0	-26.565	36.0	-26.565	0.0	-52.623

正二十面体全球剖分模型的 Geometry Clipmap 球面绘制

陈梦云，孟新，彭晓东

中国科学院国家空间科学中心 信息仿真室，北京 100190

摘　要: 地形绘制一直是图形学研究的热点问题，尤其是球面地形绘制，其在形状和数据组织方面比平面地形绘制更加复杂。在已有球面地形绘制算法的基础上，提出一种基于 Geometry Clipmap 的球面地形剖分与绘制方法。该方法以构建正二十面体球面网格为基础，将正二十面体划分为十个菱形区域，采用球面菱形网格的剖分，针对每个菱形区域的周边网格进行重新剖分和组合，形成一个虚拟的 3 ×3 的大菱形区域，扩大了 Clipmap 的活动范围，并在一定程度上解决了 Clipmap 的跨边界问题。实验结果表明了本文方法的可行性和有效性。

关键词: 球面地形绘制，Geometry Clipmap，正二十面体全球剖分，球面三角四叉树剖分模型

中图分类号: TP391.41　　**文献标志码**: A

引用格式: 陈梦云，孟新，彭晓东. 2014. 正二十面体全球剖分模型的 Geometry Clipmap 球面绘制. 遥感学报，18(5): 1059 – 1071

Chen M Y, Meng X and Peng X D. 2014. Method of Geometry Clipmap based on icosahedron. Journal of R emote S ensing, 1 8(5): 1059 – 1071　[DOI: 10.11834/jrs.20143141]

1　引　言

地形绘制一直是图形学研究的热点问题。尤其是球面地形绘制，相较于平面地形绘制在形状和数据组织方面都更加复杂，在绘制效果和效率方面还有很大的探索空间，目前仍被许多学者关注。

数据量的规模及球面的几何特点决定了设计一种实时的球面地形绘制方法至少需要考虑两方面的因素，一是实时地形简化方法，二是球面剖分与数据组织方法。目前典型的地形简化方法包括实时优化自适应网格 ROAM 算法(Duchaineau 等，1997) 和连续细节层次算法(Lindstrom 等，1997; Lindstrom 等，1996; Lindstrom 和 Pascucci 2001)，geomipmapping 算法(de Boer，2000)，以及 Losasso 和 Hoppe(2004) 提出的 Geometry Clipmap 算法等。典型球面剖分与组织方法包括最早、最常规的经纬网格划分法，基于正多面体建立全球网格模型的方法，如 Dutton(1996) 提出的四元三角网 QTM(Quaternary Triangular Mesh) 模型，Fekete 和 Treinish (1996) 提出的球面四叉树 SQT(Sphere Quad Trees) 模型，还有基于施奈德投影的整二十面体等积投影 ISEA(Icosahedron Snyder Equal Area) 模型(Snyder，1992) 等。在实时地形简化方法中，Geometry Clipmap 方法相较于四叉树、ROAM 等方法，具有更好的覆盖形状和较快的更新效率，但是由于算法本身主要针对平面地形，所以在全球的地形绘制的研究中应用较少，较经典的有 Clasen 和 Hege(2006) 选择的球形 Geometry Clipmap，在形状上很符合球形特征，但是地形数据每帧都需要实时重采样，计算量较大。Bhattacharjee 等人(2010) 提出的将 QTM 与 Geometry Clipmap 结合的方法，基本实现了 Geometry Clipmap 在球面地形中的应用，但正八面体剖分的等积性并不理想。李文庆等人(2010) 用等经纬度间隔空间划分的方法实现近似等面积空间划分的效果，在两极上形变较大，并且未阐述跨边界问题如何解决。为了使全球网格剖分在形状和等面积性上相对均衡，本文提出了基于正二十面体剖分的 Geometry Clipmap 地形绘制方法。

收稿日期: 2013-06-05; **修订日期**: 2014-04-15; **优先数字出版日期**: 2014-04-22

第一作者简介: 陈梦云(1987—)，女，硕士研究生，主要研究领域为计算机图学。E-mail: chenmengyun1987@163.com

2 算法描述

Geometry Clipmap 即可视区域剪裁,是一种基于视点的 LOD(Level of Detail) 简化策略,其由一组以视点为中心的规则方形栅格嵌套而成。其中各个方形栅格具有相同的结构,只是构成方形栅格三角网格随着各个方形栅格的 LOD 等级增加而增大。Geometry Clipmap 应用到球形地面上时,如果仍然使用经纬网格对球面进行剖分,绘制的地形块会在两极区域产生严重的变形。因此,目前常用的办法是采用正多面体对球面进行剖分,保证在多面体的每个剖分面内 Clipmap 具有比较正确的绘制效果。但是在跨越多面体的各个面时,由于球体表面的连通性和平面不同,Clipmap 的地形数据不能正常过渡。Bhattacharjee 给出了在正八面体剖分时跨边界问题的解决方法,即当 Clipmap 跨过八面体的两个面时,为两个面各建立一个 Clipmap,然而由此绘制产生了浪费。考虑到基于正八面体的剖分方法在球面网格的等积性方面表现不如正二十面体剖分法,因此本文采用基于正二十面体剖分的 Geometry Clipmap 地形绘制方法。然而,正二十面体的每个顶点周围是 5 个三角面,当绘制的地形跨边界时 Clipmap 的顶点拓扑关系会发生改变,进而导致在跨边界时地形数据无法正确更新,其困难程度超过了正八面体剖分时出现的问题。针对以上问题提出虚拟菱形解决方案,将基于正二十面体全球剖分与 Geometry Clipmap 有效结合。

2.1 球面顶点划分及组织方法

用正二十面体剖分全球网格方法包括空间坐标系的转换与顶点值计算,多面体的层次剖分与网格逆投影计算,顶点投影计算几个方面。

2.1.1 正二十面体剖分全球网格

在基于等角比例投影的球面三角四叉树剖分模型(袁文 等,2009) 基础上进行了改进,计算方法更加简单。定义单位球半径为 1,两个顶点分别与极点重合,球的内接正二十面体的 20 个面的 3 个顶点的经纬度见附录中表格,经过换算可得到直角坐标值。过球心对正二十面体的每条边在球面上做球心投影,将在球面上形成 20 个球面三角。定义正二十面体的 20 个面为球面三角在正二十面体上的投影三角平面。直接均匀剖分投影三角平面,形成均匀的三角网格,然后通过逆投影公式将每个三角的顶点以及边反算至球面,就可以得到球面顶点的坐标。

直接剖分均匀投影三角平面方法如下: 将上下两个投影三角平面合并成一个菱形,形成如图 1 的 10 个菱形,对每个菱形进行 n 深度剖分,即将投影平面的两条边剖分成 $2n$ 个相等部分,做平行线相交形成与直角坐标系一一对应的网格顶点。每个菱形由$(2n+1)\times(2n+1)$ 个顶点构成。将顶点反算至球面的逆投影公式: 建立球面三角和投影三角面的投影关系,如图 2,球心为坐标原点 O,P 为球面上任意一点,投影到投影三角平面上为 G,弧线 MPN 在投影平面上的投影为 EF,EF 平行于 BC。EG 与 EF 的长度比为 x,且等于角 EOG 与角 EOF 之比,线段 AE 与 AB 的长度比为 y,且等于角 EOB 与角 AOB 之比(袁文 等,2009) 。每个投影三角面的 A、B、C 坐标值已知,P 点坐标可以由以下公式推导得出:

$$\boldsymbol{OE} = \boldsymbol{OA} + y(\boldsymbol{OB} - \boldsymbol{OA}) \quad (1)$$

$$\boldsymbol{OF} = \boldsymbol{OA} + y(\boldsymbol{OC} - \boldsymbol{OA}) \quad (2)$$

G 点的坐标为:

$$\boldsymbol{OG} = \boldsymbol{OE} + x(\boldsymbol{OF} - \boldsymbol{OE}) \quad (3)$$

对 OG 进行归一化,再乘以球半径就可以得到球面上 P 点的坐标。投影平面上的网格顶点已经事先划分好,其中,$x = i/(i+j)$,$y = (i+j)/2n$(i、j 是投影平面上的顶点网格坐标,n 为划分深度) 。

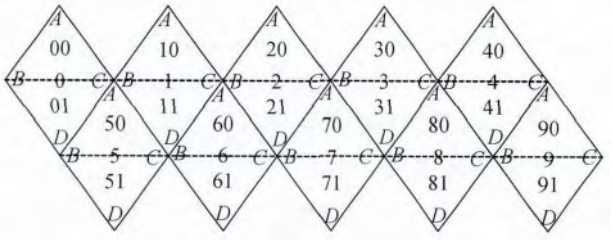


图 1 球内接正二十面体展开为 10 个菱形

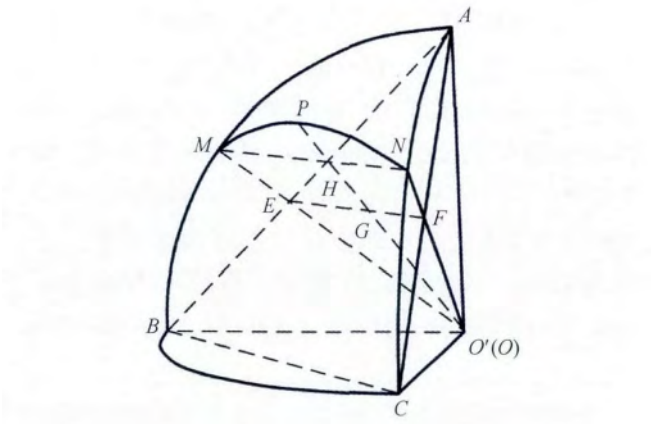


图 2 球面三角和投影三角平面的透视图(袁文 等,2009)

2.1.2 视点在投影面上相对坐标计算

为了满足 Geometry Clipmap 算法以视点为中心生成一定范围内的均匀网格,需要知道视点在投影网格上的相对坐标,以该坐标为中心,生成 Clipmaps。

由于已经知道正二十面体每个面的坐标,首先计算球面上 P 点到正二十面体每一个面的中心点的距离,距离最短的面即 P 点投影面所在的面。计算 P 点在投影面上的相对坐标,采用简化的等角比投影方法。以上三角为例,如图 3 所示,计算 P 点与球心的连线即 P 点在投影平面上的投影点 O 在网格中的相对位置。分别过点 O 做平行于 AB 和 AC 的平行线交 AC 和 AB 于 T 和 S 点。求 Q 点的相对坐标转化为求 AS 在 AB 中的百分比和 AT 在 AC 中的百分比,为简化运算可将平面投影到法线分量最大的 2 维平面转化为 2 维运算。设 $|AS|/|AB|=\alpha_1$,则有

$$OS = OA + \alpha_1(OB - OA) = OQ + \beta_1(OC - OA) \tag{4}$$

解方程组消去 β_1 可求得 α_1。同理设 $|AT|/|AC|=\alpha_2$,则有

$$OT = OA + \alpha_2(OC - OA) = OQ + \beta_2(OB - OA) \tag{5}$$

式(4)(5)联立,可求得 α_2。分别乘以行列顶点个数,即可得 Q 点在投影平面上的相对坐标。

求得 Q 点的相对坐标后,以 Q 点为中心建立 Clipmap,确定 Clipmap 的大小后,可知 Clipmap 的各个顶点在投影平面上的坐标,通过 2.1.1 节的方法将顶点反算至球面上,即可形成球面 Clipmap。

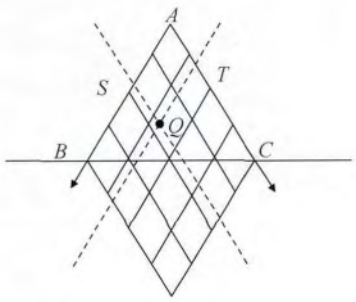


图 3 投影平面网格相对坐标计算示意图

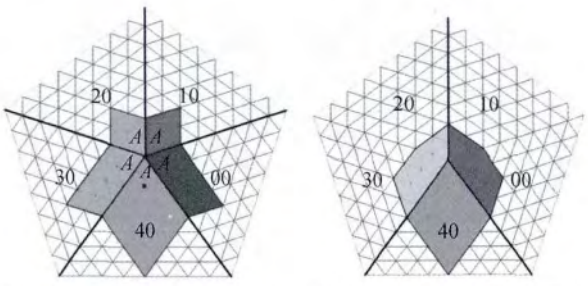
2.2 基于正二十面体球面剖分的 Geometry Clipmap 绘制实现

2.2.1 3×3 虚拟菱形区域算法

将正二十面体如图 1 展开后,以菱形为单位建立剖分网格,以视点为中心建立的 Clipmap 将不再是如原算法中的正方形而是菱形。当 Clipmap 在一个菱形范围内移动时,Clipmap 的形状是菱形。每个菱形 4 个顶点周边的相邻菱形分布与矩形 4 个顶点周围都是矩形分布不同,分锐角顶点和钝角顶点两种情况,锐角顶点分为锐角顶点与周围 4 个菱形的锐角顶点重合和与周围两个菱形的钝角顶点重合两种情况,钝角顶点只有与旁边一个菱形的钝角顶点和一个菱形的锐角顶点重合一种情况。当 Clipmap 跨越菱形边界时,Clipmap 移动到锐角顶点或钝角顶点周围,无法保持完整矩形的数据结构,可能会产生数据增加或缺失的现象。Bhattacharjee 在基于正八面体建立 Geometry Clipmap 时,将在视锥内的菱形当成独立的地形,基于每一个菱形建立一个 Clipmap,当视锥内有多个菱形时,绘制多次 Clipmap。由于本文中的正二十面体包含 10 个菱形,该方法运用到本文的算法中计算量巨大,并且在正二十面体应用中跨边界时会产生变形,尤其在极点周围,因此不能采用该方法。

以正二十面体的每个顶点为中心,由周围的 5 个投影三角平面可以构成一个五边形。以 4 号菱形为例,对菱形顶点周围的网格分布进行如下分析:

(1) 当 Clipmap 中心靠近 4 号菱形的 A 顶点时,该锐角顶点与周围的 4 个菱形的 A 顶点重合,如果不做任何处理跨越边界的 Clipmap 形状将变成如图 4(a) 的五角星形状,错误数据的增加导致的变形十分严重。若将 2 号菱形的上三角 20 和 3 号菱形的上三角 30 合并形成一个新菱形,同理将 1 号菱形的上三角 10 和 0 号菱形的上三角 00 合并形成一个新菱形,如图 4(b),跨越边界的 Climap 将遵循新菱形的网格绘制顺序,生成的 Clipmap 变形较之前情况好。

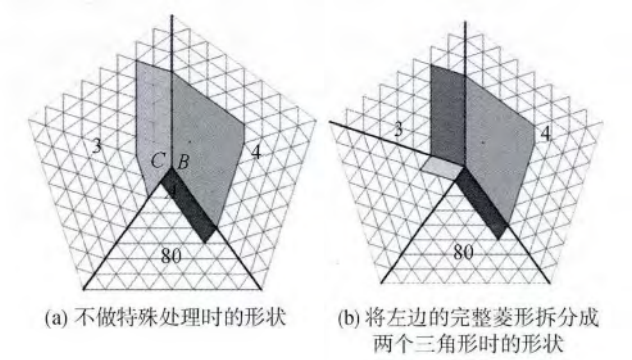


(a) Clipmap中心靠近三角面40的锐角顶点不做特殊处理 (b) 将两边的三角形分别合并后

图 4 跨越边界的 Clipmap 处理方案示意图

(2) 当 Clipmap 中心靠近 4 号菱形 B 顶点时,该钝角顶点和 3 号菱形的 C 顶点及 8 号菱形的 A 顶点重合,如果不做任何处理,由于数据缺失将在左下

角产生很大的锯齿并且随着 Clipmap 尺寸变大而变大。若把 3 号菱形拆分成两个投影三角面 30 和 31,不仅可以满足第 1 种情况中 30 与 20 进行合并,增加的数据结构使得 Clipmap 在跨越边界时产生的变形较小,如图 5(b)。当 Clipmap 中心靠近 4 号菱形右边的 C 顶点时情况类似。



(a) 不做特殊处理时的形状 (b) 将左边的完整菱形拆分成两个三角形时的形状

图 5 Clipmap 中心靠近钝角时处理示意图

(3) 当 Clipmap 中心靠近 4 号菱形 D 顶点时(图 6),该锐角顶点和 8 号、9 号两菱形的 C、B 顶点重合,Clipmap 跨越边界时产生的变形和图 4(b) 类似,可以不做菱形的合并或拆分。

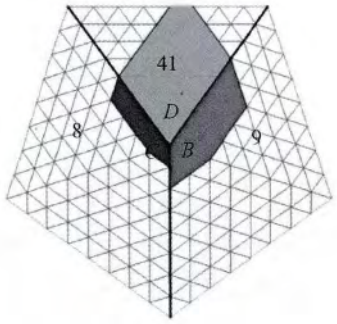
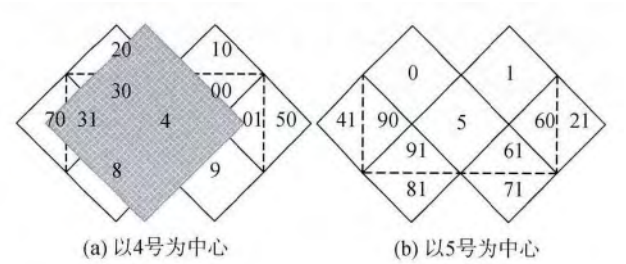


图 6 Clipmap 中心靠近 41 锐角时的形状

通过上述几种情况分析,本文认为将如图 1 中的某主菱形周围的菱形进行拆分或者合并处理,生成的 Clipmap 网格形状更能维持原始 Clipmap 的结构,扩大 Clipmap 连续活动的范围,产生的形变在可容忍范围内。

以 Clipmap 中心所在的菱形为主菱形构建 3×3 的大菱形区域如下: 每一个菱形由上下两个正三角形构成,如图 1 编号。如图 7 所示,以每一个菱形为中心,将周围的构成原始菱形的三角形拆分并拼接成新的菱形,虚线表示两三角形新拼接成菱形。图 7中以菱形 4 为中心,它的左下和右下方的菱形保持不变,右边的 1 号菱形拆分,01 与 50,00 与 10 分别形成一个新的菱形,左边的 3 号菱形拆分,30 与 20、31 与 70 分别形成新的菱形,同时满足图 4 和图 5 的情况。当 Clipmap 中心在 4 号菱形内移动时,Clipmap 的活动范围可以扩大到 3×3 菱形范围的 7 块菱形区域。依次以所有的菱形为中心生成新的大菱形区域网格,当 Clipmap 的中心移到另一菱形上时进行切换。以 5—9 号菱形为主菱形的构建网格区域方式与 0—4 号略有不同。5 号菱形的左上和右上菱形保持不变右边的 6 号菱形拆分成 60 和 61,分别与 21 和 71 形成新的菱形。左边的 9 号菱形可以拆分成 90 和 91 分别与 41 和 81 拼接成新的菱形。总结构成虚拟大菱形的三角面索引规律如图 8 和图 9 所示。



(a) 以4号为中心 (b) 以5号为中心

图 7 以不同菱形为中心建立的虚拟菱形网格

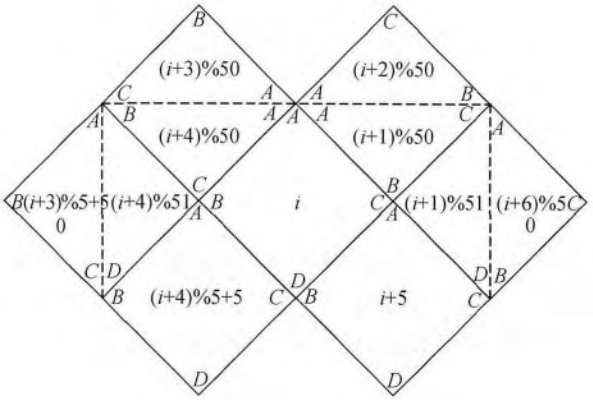


图 8 以 0—4 号菱形为中心建立虚拟网格的菱形索引编号示意图

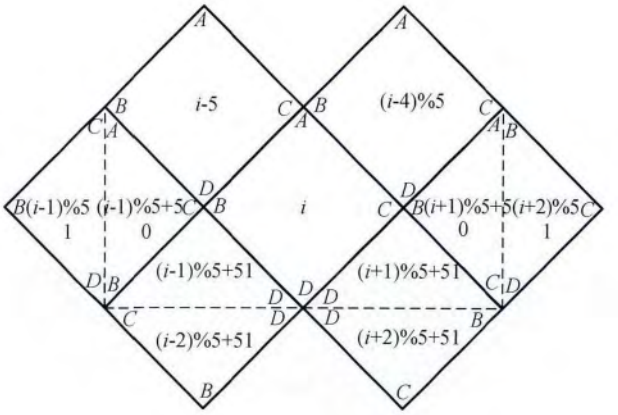


图 9 5—9 号菱形为中心建立虚拟网格的菱形索引编号示意图

初始化大菱形区域时以每个菱形为中心，向周围扩展 3 × 3 个菱形。由于周围的菱形组成具有特殊性，在初始化时需要针对每个菱形单独建立共 10 个 3 × 3 菱形区域。每个主菱形的上下左右顶点(A、B、C 和 D) 的坐标由拼接的三角形的顶点决定。如图 8 中，中心菱形 i 号、左下角菱形($i+4$) %5 +5 号、右下角菱形 $i+5$ 号的 A、B、C 和 D 顶点和原始菱形的相同，其左上角菱形的 A、B、C 和 D 顶点分别是($i+3$) %5号菱形的 B 顶点，($i+4$) %5 号菱形的 C 顶点，($i+4$) %5 号菱形的 B 顶点(与($i+3$) %5 号菱形的 C 顶点相同) ，($i+4$) %5 号菱形的 A 顶点(与($i+3$) %5 号菱形的 A 顶点相同) 。

大菱形区域内的顶点生成过程总结如图 10 所示，例如，当小菱形区域的顶点相对坐标 $i \in (0,n)$ 且 $j \in (n,2n)$ 时，大菱形区域内的顶点相对坐标为 $I=i$，$J=j-n$，当 $i \in (n,2n)$ 且 $j \in (0,n)$ 时，$I=i-n$，$J=j$。相对坐标的计算在逆投影反算球面坐标时用到。

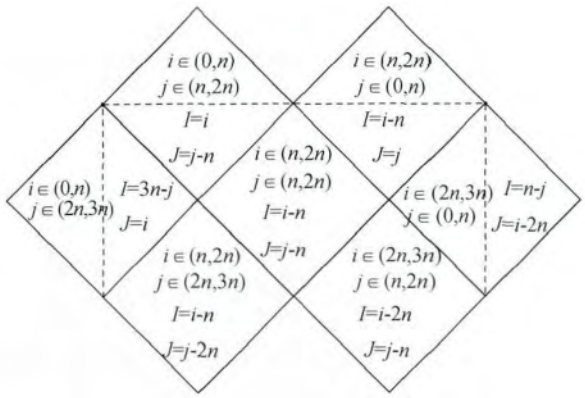


图 10 虚拟大菱形区域内顶点生成示意图

生成这种网格的优点是: 网格形状与 Clipmap 的规则网格匹配，一定程度上解决了 Clipmap 跨边界问题，最大活动范围不再局限于一个菱形内，可以覆盖到以一个菱形为中心的 7 个菱形，足以覆盖大半个地球。将这 7 个菱形当成整块地形处理，而不是单独的 7 块，因此不需要构建 7 个 Clipmap 绘制 7 次，在一定程度上提高了效率。

2.2.2 Geometry Clipmap 算法实现

基于正二十面体的 Clipmap 生成方法和原始的 Geometry Clipmap 算法相同。整块地形由一个与视点相关的 Clipmap 树组成，定义最高层级的 Clipmap 分辨率最低，也即网格间距最大，其中每一层 Clipmap 有相同的顶点数，每层 Clipmap 网格间距依次降低。与原始算法不同的是，每一个 Clipmap 由菱形构成，不再是正方形。

两层之间网格分辨率的不同将导致裂缝产生，Losasso 和 Hoppe 的原算法中每一层数据除了要存储自己的高度值以外还要存储该点在父层(粗糙层) 上的高度值。用顶点着色器混合两高度值得到过渡区域的高度值进行裂缝消除，计算较复杂，本文中采用的方法是将每一层 Clipmap 的较精细层的边界网格的外边界减少一倍分辨率，作为连接较精细层和较粗糙层的边界。顶点数据更新方法与原算法类似，在大菱形范围内进行“L”区域更新。当 Clipmap 跨越到大菱形范围外，则需要将整块内存数据更新。

3 仿真结果及分析

本文在普通的台式机上实现了基于 Geometry Clipmap 算法的球面地形绘制算法。实验环境为 Windows XP 系统，Visual Studio 2010 和 DirectX9.0，机器配置为 3.1 GHz Intel i5-2400，GeForce GTX460 显卡。实验中，高程采用 2500 × 1250 分辨率的全球高度图，Clipmap 大小为 129 × 129，层数为 6 层，顶点数为 99846，绘制帧速达到 60 帧/s。图 11 是实验仿真结果，可以看见当 Clipmap 在一个菱形范围内时，生成的 Clipmap 是规则的菱形，当发生跨界时，Clipmap 会产生一定变形但不影响仿真效果。

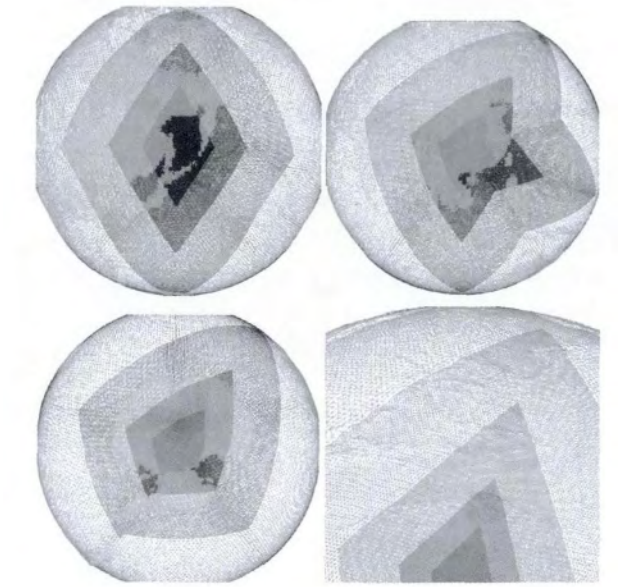


图 11 基于 Geometry Clipmap 的球面地形仿真结果

若将菱形 Clipmap 的两个锐角进行裁剪，可得到正六边形的 Clipmap，实现各个方向上相同距离分辨率近似相同，见图 12。结果与 Bhattacharjee 建立

的基于 QTM 的 Geometry Clipmap 地形绘制形状相似(图 13)。

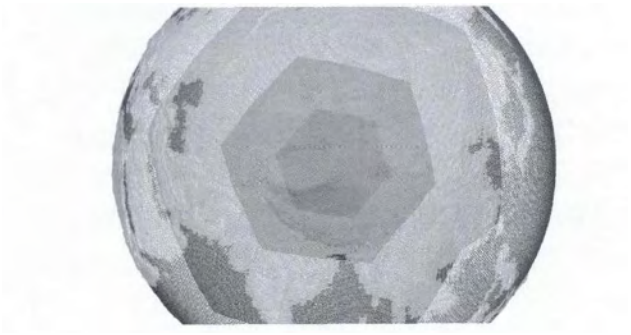


图 12 将菱形 Clipmap 进行裁剪后的仿真结果

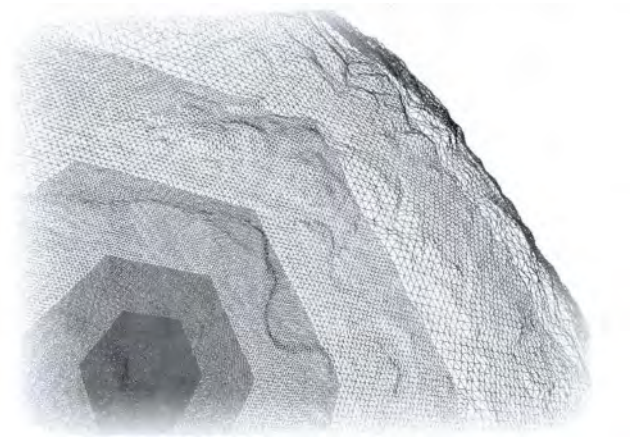


图 13 Bhattacharjee(2010)建立的仿真结果

4 结 论

采用正二十面体构建了全球剖分模型,对投影三角平面进行均匀网格划分,并通过逆投影计算得到球面顶点坐标。构建新的网格结构,以 Clipmap 中心所在的菱形为中心,将周围的菱形进行拆分并重新组合成新虚拟大菱形区域,将 7 块地形合并成 1 块处理。并在一定程度上解决了 Clipmap 跨多菱形的形变问题,变形程度在容忍范围内。将基于正二十面体进行全球剖分与 Geometry Clipmap 初步结合起来。

本文只初步实现了将 Gometry Clipmap 应用到正二十面体细分而成的球形网格。以一个固定的菱形块为主菱形的 3 × 3 大菱形区域是初始化时定义好的,当 Clipmap 中心离开一块主菱形移动到另一块主菱形时,由于大菱形区域的拼接方法不同,顶点的连接顺序也不同,将导致 Clipmap 的形状发生跳变,并且 Clipmap 的顶点数据必须全部更新,频繁地更新整个 Clipmap 的顶点数据对绘制效率有一定影响。因此,下一步工作是:设计算法减小 Clipmap 在多块菱形上移动时的形变,尽量使 Clipmap 的形状在各个方向上相同距离分辨率保持一致,建立动态的大菱形区域,当 Clipmap 中心从一个主菱形移动到另一个三角区域时,根据该三角区域当前所在的菱形区域动态地构建 3 × 3 的大菱形区域,提高绘制率。

参考文献(References)

Bhattacharjee S. 2010. Real-time Terrain Rendering and Processing. International Institute of Information Technology Hyderabad, INDIA

Clasen M, Hege H C. 2006. Terrain rendering using spherical clipmaps // Proceedings of the Eighth Joint Eurographics / IEEE VGTC conference on Visualization. Switzerland: Eurographics Association Aire-la-Ville: 91 – 98 [DOI: 10.2312/VisSym/EuroVis06/091 – 098]

de Boer W H. 2000. Fast Terrain Rendering Using Geometrical Mipmapping [EB/OL]. [2013-09-20] http://www.flipcode.com

Duchaineau M, Wolinsky M, Sigeti D E, Miller M C, Aldrich C and Mineev-Weinstein M B. 1997. ROAMing Terrain: Real-TimeOptimally Adapting Meshes. Technical Report UCRL-JC-127870, Lawrence Livermore Nat'l Laboratory

Dutton G. 1996. Improving locational specificity of map data: a multi-resolution, metadata-driven approach and notation. International Journal of Geographical Information Science, 10(3): 253 – 268 [DOI: 10.1080/02693799608902078]

Fekete G and Treinish L A. 1990. Sphere quadtrees: a new data structure to support the visualization of spherically distributed data // Proceedings of the SPIE, Extracting Meaning from Complex Data: Processing, Display, Interaction, 1259: 242 – 253 [DOI: 10.1117/12.19991]

Li W X, Chen G, Huang B X. 2010. Proceedings of 2010 International Conference on Remote Sensing (ICRS 2010) Volume 3

Lindstrom P, Koller D, Ribarsky W, Hodges L F, Faust N and Turner G A. 1996. Real-time, continuous level of detail rendering of height fields // Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM: 109 – 118 [DOI: 10.1145/237170.237217]

Lindstrom P, Koller D, Ribarsky W, Hodges L F, Op den Bosch A and Faust N L. 1997. An Integrated Global GIS and Visual Simulation System. Graphics, Visualization, and Usability Center, Georgia Institute of Technology

Lindstrom P and Pascucci V. 2001. Visualization of large terrains made easy // Proceedings of Visualization. San Diego, CA, USA: IEEE: 363 – 574 [DOI: 10.1109/VISUAL.2001.964533]

Losasso F and Hoppe H. 2004. Geometry clipmaps: terrain rendering using nested regular grids // Proceedings of ACM SIGGRAPH. New York: ACM: 769 – 776 [DOI: 10.1145/1015706.1015799]

Snyder J P. 1992. An Equal-Area map projection for polyhedra globes. Cartographica, 29(1): 10 – 21 [DOI: 10.3138/27H7 – 8K88 – 4882 – 1752]

袁文, 庄大方, 袁武, 刘纪远. 2009. 基于等角比例投影的球面三角四叉树剖分模型. 遥感学报, 13(1): 103 – 111 [DOI: 10.3321/j.issn:1007 – 4619.2009.01.014]

附录

表 正二十面体顶点定位

编号	顶点 0		顶点 1		顶点 2		中心点	
	经度/°	纬度/°	经度/°	纬度/°	经度/°	纬度/°	经度/°	纬度/°
0	0.0	90.0	0. 0	26.565	72.0	26.565	36.0	52.623
1	0.0	90.0	72. 0	26.565	144.0	26.565	108.0	52.623
2	0.0	90.0	144.0	26.565	216.0	26.565	180 0	52.623
3	0.0	90.0	216.0	26.565	288.0	26.565	-108.0	52.623
4	0.0	90.0	288.0	26.565	0.0	26.565	-36.0	52.623
5	36.0	-26.565	0.0	26.565	72.0	26.565	36.0	10.812
6	108.0	-26.565	72.0	26.565	144.0	26.565	108.0	10.812
7	180.0	-26.565	144.0	26.565	216.0	26.565	180.0	10.812
8	252.0	-26.565	216.0	26.565	288.0	26.565	-108.0	10.812
9	324.0	-26.565	288.0	26.565	0.0	26.565	-36.0	10.812
10	72.0	26.565	36.0	-26.565	108.0	-26.565	72.0	-10.812
11	144.0	26.565	108.0	-26.565	180.0	-26.565	144.0	-10.812
12	216.0	26.565	180.0	-26.565	252.0	-26.565	-144.0	-10.812
13	288.0	26.565	252.0	-26.565	324.0	-26.565	-72.0	-10.812
14	0.0	26.565	324.0	-26.565	36.0	-26.565	0.0	-10.812
15	0.0	-90.0	36.0	-26.565	108.0	-26.565	72.0	-52.623
16	0.0	-90.0	108.0	-26.565	180.0	-26.565	144.0	-52.623
17	0.0	-90.0	180.0	-26.565	252.0	-26.565	-144.0	-52.623
18	0.0	-90.0	252.0	-26.565	324.0	-26.565	-72.0	-52.623
19	0.0	-90.0	324.0	-26.565	36.0	-26.565	0.0	-52.623