

A ‘Drift’ algorithm for integrating vector polyline and DEM based on the spherical DQG

Jiaojiao WANG, Lei WANG, Wenmin CAO, and Xuesheng ZHAO

Department of Remote Sensing & GIS, China University of Mining & Technology (Beijing), Beijing, China

E-mail: doublejoy@126.com

Abstract. The efficient integration method of vector and DEM data on a global scale is one of the important issues in the community of Digital Earth. Among the existing methods, geometry-based approach maintains the characteristics of vector data necessary for inquiry and analysis. However, the complexity of geometry-based approach, which needs lots of interpolation calculation, limits its applications greatly in the multi-source spatial data integration on a global scale. To overcome this serious deficiency, a novel ‘drift’ algorithm is developed based on the spherical Degenerate Quadtree Grid (DQG) on which the global DEMs data is represented. The main principle of this algorithm is that the vector node in a DQG cell can be moved to the cell corner-point without changing the visualization effects if the cell is smaller or equal to a pixel of screen. A detailed algorithm and the multi-scale operation steps are also presented. By the ‘drift’ algorithm, the vector polylines and DEM grids are integrated seamlessly, avoiding lots of interpolation calculating. Based on the approach described above, we have developed a computer program in platform OpenGL 3D API with VC++ language. In this experiment, USGS GTOPO30 DEM data and 1:1,000,000 DCW roads data sets in China area are selected. Tests have shown that time consumption of the ‘drift’ algorithm is only about 25% of that of the traditional ones, moreover, the mean error of drift operation on vector nodes can be controlled within about half a DQG cell. In the end, the conclusions and future works are also given.

1. Introduction

With the rapid development of spatial data acquisition technology and global economy, single type of spatial data has been unable to meet the growing demand on various applications. Large-scale, multi-resolution and multi-source data is being used in comprehensive analysis of many fields in order to make better decisions [1]. Multi-source data fusion meets visual effects, and serves for visual design and decision timely and accurately. Due to the approximation of the format of image and DEM data, the methods and applications for the integration of image and DEM has been relatively skilful, while that for the integration of vector data and DEM shows slower progress [2]. The existing methods to integrate vector data and DEM can be classified into two categories: texture-based approach and geometry-based approach [3]. Texture-based approach translates the vector data into texture, then maps it onto DEM, including texture mapping algorithm [3-4] and shadow volume algorithm [5-8]. It is widely used in most digital earth systems such as World Wind and ArcGlobe due to its relatively high efficiency, but it is hard to carry out multi-scale query, calculation and decision analysis and vector data shows distortion under zoom. Geometry-based approach generates the points of intersection to adapt vector data to DEM grid, including geometry-overlay algorithm [9-10] and



geometry-embed algorithm [11]. It maintains the original characteristics of vector data and takes advantage of secondary data to constrain and trim DEM grid [8]. During LOD representation, lots of geometric primitives have to be created, it is not suitable for the data sets at large-scale [6,12], which currently limits its wide applications and has become a bottleneck problem urgent to be solved.

In response to the issues described above, this paper adopts Degenerate Quadtree Grid (DQG) as basic framework. The idea of grid cell decomposition is introduced and a novel approach named ‘drift’ algorithm for adapting vector data to DEM grid is developed and greatly improves the integration efficiency. The efficiency and error analysis and conclusion are given in the end.

2. Principles of DQG tessellation and cell decomposition

The principle of DQG tessellation is as follows: Firstly, inscribed regular octahedron is selected as basis of sphere tessellation. Spherical surface will be divided into 8 regular spherical triangles. Secondly, for every spherical triangle, the midpoints of the 2 waist sides are connected to form a new latitude line. The midpoint of the bottom edge of spherical triangle and the midpoint of the new latitude line are connected to form a new longitude line. Thus, every spherical triangle can be divided into a new spherical triangle and 2 new spherical quadrilaterals, in which the spherical triangle will be tessellated according to the above steps and the spherical quadrilaterals will be tessellated according to conventional quadtree subdivision, and so on (figure 1, details in literature [13]).

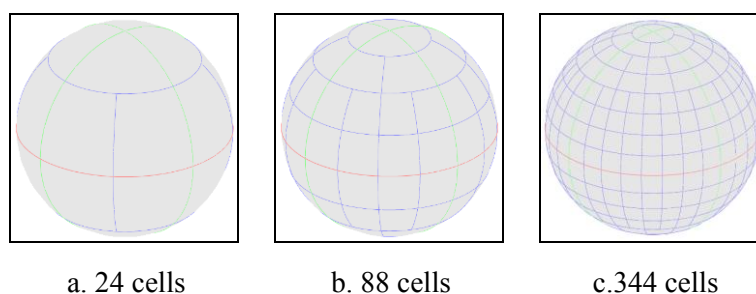


Figure 1. Hierarchical tessellation of spherical surface based on octahedron [12].

The idea of cell decomposition[14] is illustrated as follows: cell polygons stand for the polygon features, cell nodes stand for point features and accurate spatial location, cell boundaries would measure the relationship among cell polygons or nodes. Based on cell decomposition, drift operation on vector would be done when cell is smaller or equal to a pixel of screen, thus vector nodes could move to the location of cell nodes (figure 2), which could ensure ideal visual effects to some extent.

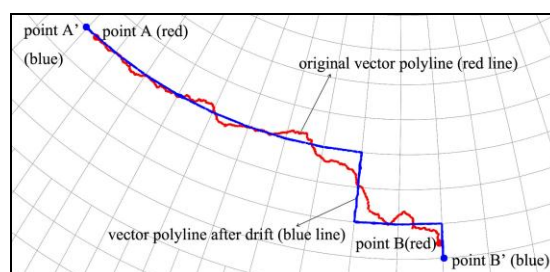


Figure 2. Drift operation on vector based on DQG cell decomposition.

3. A 'Drift' algorithm for vector polyline on DEM grid

3.1. Basic principle of drift algorithm

Firstly, the original vector polyline is divided into certain number of segments every of which is made up of two nodes; then for each segment, along the DQG cells gone through by the segment, we would find a new vector polyline which is approximate to the original segment in the spatial distance and direction by drift operation of the vector nodes. In order to ensure the visual effects, drift operation should meet the condition that the DQG cell is equal to or smaller than a pixel of screen. If DQG cell is larger than a pixel of screen, the traditional interpolation calculation would be adopted.

Drift operation would be done from the starting node to the end node of the original vector polyline until all of the nodes have been dealt with, thus the seamless integration of vector and DEM grid would be achieved. This approach is named 'drift algorithm'.

3.2. Detailed steps and flow chart

The detailed steps of drift algorithm are as follows:

Input: DQG DEM grid at certain level and original vector $V = [v_0, v_1, \dots, v_n]$

Step1 Find the cell GA in which starting point A of V falls;

Step2 Find the point ga closest to A in GA and add ga into new vector U ;

Step3 Drift operation on the other nodes of V , define B the point next to A ;

Step3.1 IF B falls in GA

Find the point gb closest to B in GA ;

IF the line connected by gb and the current end point uk of U goes through the diagonal, transition point P would be determined according to the type of DQG cell:

Case1: for pole triangle cell, P is the one of the cell nodes (figure 3-1)

Case2: for quadtree quadrilateral cell, P is the midpoint of diagonal (figure 3-2)

Case3: for non-quadtree quadrilateral cell, P is the point closest to the line connected by gb and uk (figure 3-3, 3-4, 3-5)

Step3.2 IF B does not fall in GA

Find the cell GB in which B falls;

Find all the adjacent cell nodes of point uk and define the set K ;

Find the cell node k_i along vector segment AB which is closest to the plane OAB (O is the center of the sphere) and add it into U :

IF k_i does not fall in GB , find all the adjacent nodes of k_i ;

Else k_i falls in GB , find the node gb in GB closest to B ;

IF $gb \neq k_i$, add gb into U ;

Else eliminate gb .

Step3.3 Loop until the remaining nodes in V have been completely traversed

Output: New vector $U = [u_0, u_1, \dots, u_m]$ which approximates to the original one

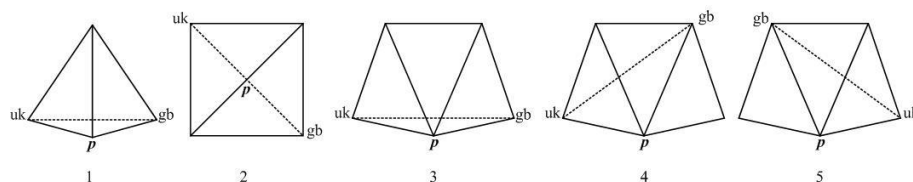


Figure 3. DQG pole triangle cell (1), quadtree quadrilateral cell (2), and non-quadtree quadrilateral cell (3-5).

Flow chart of drift algorithm shows as Figure 4.

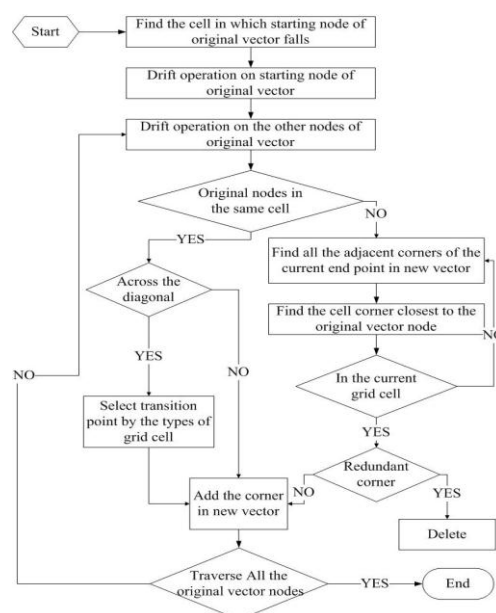


Figure 4. Flow chart of drift algorithm.

4. Experiment and analysis

The used terrain data is global GTOPO30 DEM distributed by USGS with the resolution 30'' and size 2.7Gb. The used vector data is China road polyline of DCW(Digital Chart of the World) produced by Defense Mapping Agency with the scale 1:1000,000 and the size 6.86Mb. The prototype system for data integration is implemented in the context of Visual C++ and OpenGL API. The subdivision level of DQG in this experiment is up to 13 corresponding to the resolution of GTOPO30 DEM.

4.1. Visualization of vector polyline on DQG DEM

In order to display the effect and error after drift operation on vector more clearly, subdivision level 8-13 was selected for comparison. Full view of China road polyline on DQG DEM at level 11 before and after drift operation is shown as figure 5. Figure 6 is local view (red line denotes original vector and the blue one denotes vector after drift) at level 8-13. It is obvious that vector polyline after drift was more and more approximate to the original one with subdivision level increasing, and the phenomenon of polyline going through DEM grid has been completely eliminated due to the connection of vector nodes by the cell edge or diagonal. It also demonstrated that drift algorithm achieved the good performance only when the DQG cell is equal to or smaller than a pixel of screen. The drift algorithm does not work well under any conditions.

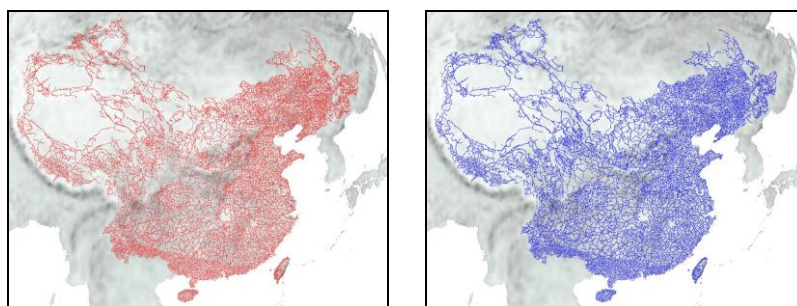


Figure 5. Full view of China roads before (left) and after (right) drift operation (level 11).

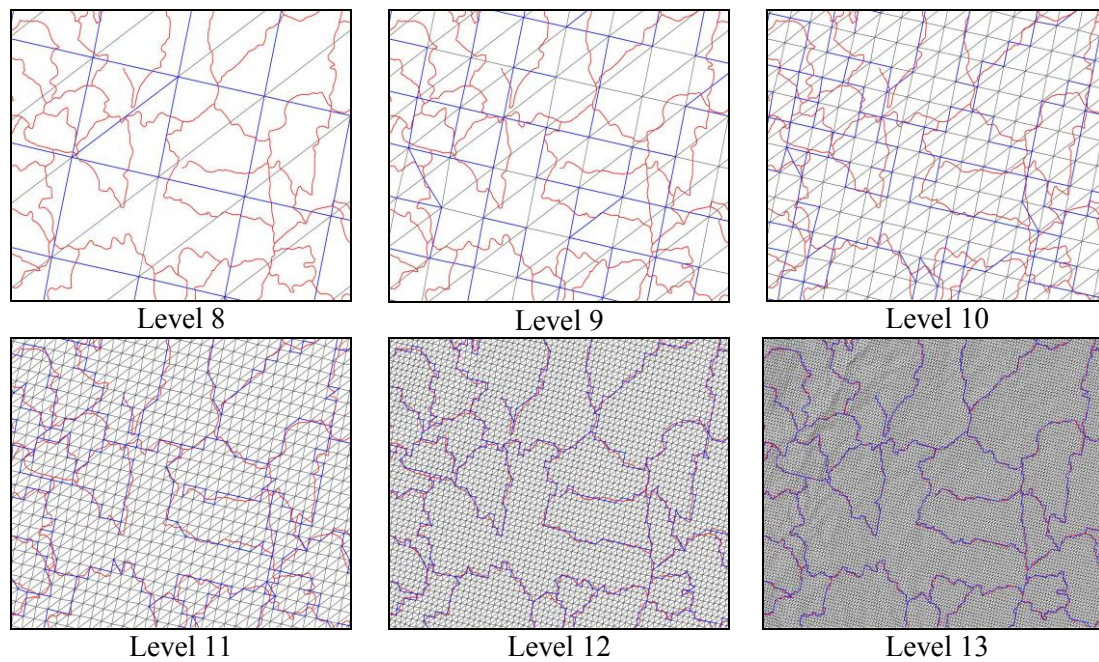


Figure 6. Local view of China roads before (red) and after (blue) drift operation.

4.2. Efficiency and error analysis of vector polyline on DQG DEM

Drift operation inevitably generates the error, and the following formula is used to compute the mean error produced by the drift operation:

$$E = \frac{1}{N} \sum_i^N |P_i - \bar{P}_i| \quad (1)$$

In equation (1), N denotes the number of original vector nodes, P_i and \bar{P}_i respectively denotes three-dimensional coordinates of one vector node (x, y, z) before and after drift operation. $|P_i - \bar{P}_i|$ denotes spherical distance between the original node and the new one after drift operation.

From table 1 and figure 7, it can be seen that time-consuming for drift algorithm is about 20%-30% of that for geometric interpolation at level 8-13, and increases obviously much more slowly than that for geometric interpolation with the level increasing. The mean error of the drift operation decreased with the level increasing, while the ratio of mean error to edge length of minimum DQG cell maintained about 0.5.

Table 1. Efficiency and error analysis of integrating China roads and DEM grid

Level	Efficiency (s)		mean error (km)	edge length of minimum cell (km)	Scale
	drift	interpolation			
8	0.155	0.589	13.981	27.642	1:10M
9	0.235	0.842	7.091	13.821	
10	0.376	1.324	3.523	6.911	
11	0.680	2.289	1.767	3.455	
12	1.302	4.400	0.885	1.728	
13	2.370	8.274	0.444	0.864	1:1M

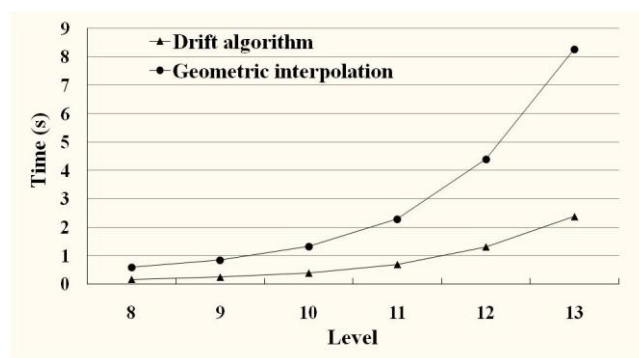


Figure 7. Efficiency comparison of drift algorithm and geometry interpolation approach.

5. Conclusions and future works

To overcome the low efficiency of geometric interpolation for large-scale vector data on DEM grids, a novel drift algorithm is developed to eliminate the mistaken representation of vector polyline going across DEM grid and achieves the adaptive integration. The main conclusions include as follows: ① This drift algorithm shows higher efficiency than traditional geometry-based approach, i.e. the time consuming approximately decreases 75% of that for geometric interpolation and increases much more slowly than geometric interpolation with the tessellation level increasing. ② The mean error by drift operation can be overall controlled in about half edge of DQG cell, which entirely meets the requirements for visualization under the condition of visual threshold.

This paper only focused on drift operation at individual tessellation level. Future works will focus on the LOD (levels of details) management, visualization, measurement, and spatial analysis for the global multi-scale spatial data sets (such as vector data, imaging data, and DEM data, *et al.*).

References

- [1] Carrara P 2008 *Int. Journal of G.I.S* **22** 781-799
- [2] Zhang X Q 2010 *Study of Lunar Surface Spatial Information Based on Discrete Grid* (Changchun: Jilin University)
- [3] Bruneton E and Neyret F 2008 *Eurographics* **27** 311-320
- [4] Döllner J 2005 *Exploring Geovisualization* ed J Dykes, A M MacEachren and M J Kraak (Pergamon: Elsevier Ltd.) Chapter 16 pp325-343
- [5] Schneider M, Guthe M and Klein R 2005 *Proc. of the 11th Int. Conf. on Virtual Systems and Multimedia(Ghent, Belgium)* p573-582
- [6] Schneider M and Klein R 2007 *Journal of WSCG* **15** 59-65
- [7] Yang L, Zhang L Q and Kang Z 2010 *Science China Information Sciences* **53** 1122-29
- [8] Vaaraniemi M, Treib M and Westermann R 2011 *19th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (Plzen)* (Vaclav Skala Union Agency) p41-48
- [9] Szenberg F, Gattass M and Carvalho P 1997 *Proc. of Brazilian Computer Graphic Symposium(Campos do Jordao)* vol231 (Los Alamitos: IEEE Comput. Soc) p103-110
- [10] Agrawal A, Radhakrishna M and JOSHI R 2006 *Proc. of WSCG (Plzen)* (Vaclav Skala Union Agency) p1-8
- [11] Schilling A, Lanig S, Neis P and Zipf A 2008 *3rd Int. Workshop on 3D Geo-Info (Seoul)* (Berlin:Springer) p109-126
- [12] Sun W B, Hu B L and Wang H B 2012 *Geography and Geo-Information Science* **28** 43-46
- [13] Cui M J, Zhao X S 2007 *Geography and Geo-Information Science* **23** 23-25
- [14] Zhou C H, Ou Y, and Ma T 2009 *Progress In Geography* **28** 657-662