Compressed Irregular Triangulation Network for Level_of_Detail Visualization

Bisheng Yang

GIS Divison, Department of Geography, University of Zurich-Irchel, Winterthurerstr. 190, CH-8057 Zurich, Switzerland. Tel +41 (0)1 635 5256; Fax +41 (0)1 635 6848;
E-mail: bisheng@geo.unizh.ch

Abstract

It is of vital importance to generate multi-resolution TIN model dynamically and efficiently in 3D visualization, virtual reality and GIS, due to the reason that multi-scale and large data volume to be processed. This paper proposes a method for compressing TIN model to create multi-resolution TIN model (LOD) dynamically. Here, multi-resolution TIN model means different level of details TIN model that can be transformed dynamically by the proposed solution, specifically based on TIN edge collapse and vertex split among different levels of TIN models.

First, edge collapse and vertex split is determined according to the minimum distance to an average plane. An edge fulfill this requirement will be collapsed, in the same time, the angle between two triangles is used as an additional criterion for this judgment. Secondly, the vertex tree model is proposed to manage and store the vertex relationship between different level of TIN models. The corresponding encoding method is proposed based on the bracketing method (Donaghey, 1980) -- each vertex is represented only with 0 or 1 to encode the position relationship in a tree. Thirdly, two new regulations for edge collapse and vertex split are defined to prevent the problem of surface foldovers during running time.

In the validity judgment of the edge collapse and vertex split, we extended the assessment to (a) time, (b) quality. The elevation RMS and triangle shape quality were proposed to evaluate the quality of a generated multi-resolution TIN model. In the proposed method, we do not store vertex dependency relationships explicitly, this is great helpful in reducing data volume. The experiments demonstrate that the proposed method can acquire reasonable balance between model quality and running time.

1.Introduction

Triangulated Irregular Network (TIN) is a popular representation for surface models in GIS, computer graphics and virtual reality (VR) because it has a simple data structure and can easily be rendered by common graphics hardware. High resolution representations of TIN can lead to huge data volumes and long computation times for rendering. Therefore, TIN representation should maintain a good trade-off between computation time, rendering speed and data storage where visualization is a key issue. Level_of_Detail (LOD) models provide an s much research has investigated models and algorithms for 3D compression, terrain visualization, model simplification, multi-resolution analysis, progressive transmission, and so on, for example hierarchy triangle (De Floriani 1989), progressive model (Hoppe 1996), triangle decimation (Schroeder 1992), edge collapse/split (Garland et al 1998). In these approaches many methods were proposed based on edge collapse/split for constructing LOD of a TIN (Xia et al 1997, EI-Sana et al 1999). The main difference between them is the encoding of the vertex dependency relationship when visualizing the model. The encoding of the vertex dependency relationship has a big impact on the rendering speed, data storage because it is the preconditions to judge the validity of edge collapse/split, Xia (1997) stores the explicit relationship of the vertex dependencies and EI-Sana (1999) improved the relationship by storing them implicitly. In fact, this approach has unnecessary restrictions (De Floriani et al 2001), and can prevent the further simplification of the original model. The method in this paper extends the method in EI-Sana (EI-Sana et al 1999), and adopts a new method to encode the relationship, which can reduce the data storage and running time. The main contribution in this paper is to develop an efficient algorithm to encode the vertex dependency relationship dynamically during the procedure of LOD construction based on edge collapse/split.

The method presented in this paper intends to extend the edge collapse and vertex split algorithm by using a new method to encode the topological relationships in TIN model. The solution further improves the vertex dependency relationships without storing vertex dependency relationship explicitly. The core of this development is on an efficient algorithm to encode the vertex dependency relationships dynamically for a multi-resolution TIN model construction based on edge collapse and vertex split.

Following an introduction of this section, the following sections are arranged as the following. The principle and concept of the approach and the algorithm of constructing vertex tree are proposed in Section 2. The data structure and implementation of the approach are thus presented in Section 3. This is followed by several experimental studies for testing the performance of the approach and evaluating the quality of the multi-resolution model in Section 4.

2. The Concept and Principles of the Proposed Method

The method proposed in this paper is derived based on the basic concept of edge collapse and vertex split algorithm, which is illustrated in figure 1. According to the principles of edge collapse and vertex split, four key issues will be resolved in order to propose an efficient solution for multi-resolution TIN model generation in this study. These are (a) to select candidate edges for collapse; (b) to judge the validity of edge collapse and vertex split; (c) to store and encode the vertex relationships, and (d) to evaluate the model quality.

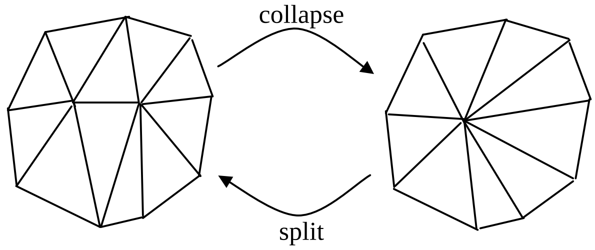


Figure 1. The concept of edge collapse and vertex split

Within the above four issues, the first three are related to the validity of the surfaces in the multi-resolution model and the performance of the model. The forth issue is related to the quality of multi-resolution model. In constructing a multi-resolution model, edge collapse and vertex split operation are very essential. For example, these will control whether an operation proceeds can be controlled according to the model accuracy or the triangle numbers in a multi-resolution model, such as the root mean square error RMS of elevations or the maximum error of point elevation in a terrain model. The procedure of constructing a multi-resolution model is as follow. Here, we supposing M_0 is the original model.

$$M_0 \longleftrightarrow M_1 ......M_i \longleftrightarrow M_n$$

The procedure from M_0 to M_n is the edge collapse process, and the reverse procedure is vertex split process. Evidently, each step operation from $M_i \longleftrightarrow M_{i+1}$ in the model, the edge collapse or vertex split is indispensable, the valid operation on edge collapse and vertex split is the precondition to avoid the triangle fold-over and retain the correctness of topological relationship. This is the foundation for further operations on the model.

Regulations on valid of vertex split and edge collapse

There have been many methods proposed for selecting candidate edges for collapse, for example, the quadric metric error method (Garland *et al*, 1997), the minimum energy method (Hoppe, 1996**),** and the method of distance to average plane (figure 2) (Schroeder *et al*, 1992). The quadric metric error and minimum energy methods considered the global accuracy of a TIN model, and a higher graphics quality can be acquired, and with a cost of lower time performance and larger data volume. The minimum energy method is relatively difficult implement (Lindstrom *et al*, 2002). Compared with these, the method of distance to average plane is relatively easier to implement and has a better time performance. When processing the vertex split and edge collapse, it is necessary to set up a set of regulations on validity of edge collapse and vertex split, since an invalid edge collapse or vertex split will lead to the problem of triangle foldovers (figure 3-b). In order avoid an error triangle in the multi-resolution model; the regulations on the validity judgment of edge collapse and vertex split are also need to be defined. Vertex dependency relationships were used to prevent surface foldovers in the pervious algorithms, such as by Xia *et al* (1997). The original regulation proposed as: vertex c can be collapsed to vertex p, only when the vertexes $p_0,..,p_n$ are present as the neighbors of p and c for display. However, the rigorousness of the regulation can be further studied. As illustrated in figure 3(a), the vertex neighborhood relationship conforms to the regulations. Nevertheless, the triangle foldovers still is inevitable when vertex c is collapsed to p (figure 3(b)).

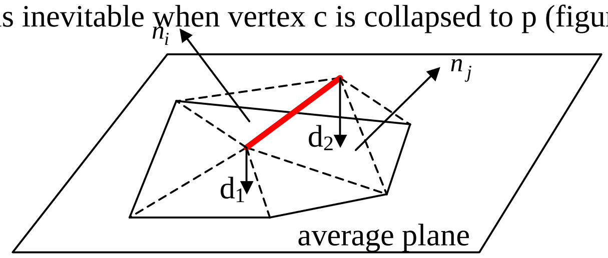


Figure 2 The method of distance to the average plane

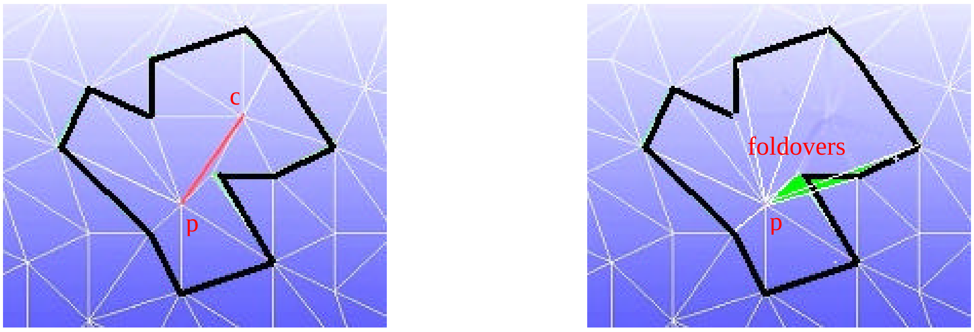
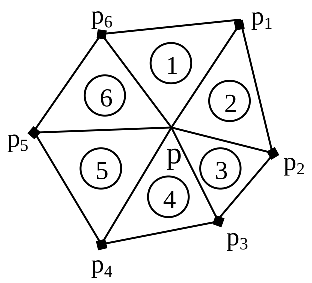
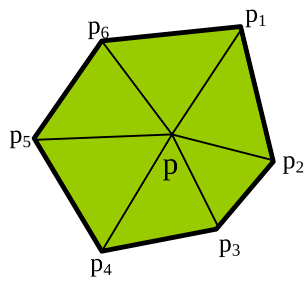


Figure 3 An example of invalid edge collapse operation

In order to overcome the deficiency based on the existing regulation and to avoid the problem of triangle fold-over, the new regulations are proposed. In order to describe the regulations clearly, we first define the following several concepts.

- Adjacent vertexes of a vertex: it is a vertex set (p_1,.., p_6), each of them and the vertex consists of one edge in the model (Figure 4(b)).
- Adjacent triangles of a vertex: it is a triangle sets (1, 2, …, 6), all the triangles in the sets have a common vertex p (Figure 4(a)).
- Influence region of a vertex: it is the polygon constructed by all the adjacent triangles of the vertex (polygon p_1, ..., p_6) (Figure 4(b)).





4(a) Adjacent triangle 4(b) Adjacent region

Figure 4 A description of adjacent triangle and adjacent region

Based on the above concepts, the following regulations are defined for checking the validity of vertex split and edge collapse during a running time.

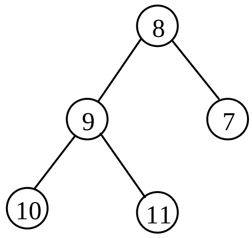
Regulation 1: Only when the influence region of the vertex v_b is a convex polygon, can the vertex v_b be collapsed to vertex v_a;
Regulation 2: The vertex v_b can be split safely only if the vertexes of the influence region are in a lower hierarchy in the vertex tree compared with that of vertex v_b.

The regulations are strict and can prevent the surface foldovers. Both **regulations 1** and **2** forms a rigorous mathematical foundation for edge collapse and vertex split, they are the precondition to avoid error triangle in forming multi-resolution TIN model. According to the workflow of the approach, whenever the edge collapse and vertex split, the local topological relationship among vertex, edge and triangle will be changed. Here, we state the topological relationship specifically as the vertex relationship. It is necessary to maintain the vertex relationship during a running time in order to retain the correctness of multi-resolution model. This is the precondition to recover multi-resolution model from a lower resolution to a higher one. Efficiently managing and storing them are of vital importance to the running time and data storage volume for an approach (EI-Sana et al 1999).
Because one vertex and its adjacent vertexes may be collapsed at different running time, there is a dependency relationship among them. The dependency relationship among vertexes is a basis for judging the validity of edge collapse and vertex split. If the dependency relationship is stored explicitly, a large data volume will be required. In order to storage and encode these vertex relationships efficiently, a data structure of vertex tree and corresponding encoding method are developed in this study. The vertex tree data structure is used to manage and store the vertex

relationship, and the encoding method is developed based on bracketing method (Donaghey, 1980), which proves a concise structure for the representation of vertex relationship in a tree (Park *et al*, 2001).

The vertex tree

Figure 5 illustrates an encoding result of an original graph. Based on the general bracketing methods, the graph can be represented with a nested pair of brackets. The left bracket can be denoted as 0, and the right bracket as 1. The tree structure, including vertex and connectivity and hierarchy relationship, can be decomposed into a set of nested brackets. The connectivity and hierarchy relationship of vertex in the tree can be recovered from the brackets sets. The encoded result of a tree consists of a set of brackets (bitstream of 0 and 1) and reordered vertices. Therefore, the connectivity of the vertex can be encoded with 2 bits per vertex (Park *et al*, 2001).



brackets ((( ) ( )) ( )) vertex sets 8 9 7 10 11 bitstream 0 0 0 1 0 1 1 0 1 1

Figure 5. Encoding vertex relationship by the bracketing method

Because continuously vertex split or edge collapse can happen during a running time, large amount of vertex relationship needs to be managed. For example, one edge is collapsed, the relationship between one vertex (parent) with two sub-vertex (children) will be generated. In order to encode and store the vertex relationship among the nodes of the tree by using the bracketing method, a virtual vertex is inserted as the root node of the tree. When we construct multi-resolution model of the original model, one of important factors is to encode the correct vertex dependency relationship in order to avoid the problem of triangle foldovers in a multi-resolution model.

During a procedure of vertex split and edge collapse, the vertex relationship in the multi-resolution model at a certain running time can be represented, as illustrated in figure 6 as an example. Various resolution models of the original model can be adjusted by selecting different density of the vertexes in the vertex tree (A or B in figure 6). However, random splitting the vertex and collapsing the vertex can cause an error (triangle foldovers) in the representation of the multi-resolution model, for example, the case illustrated in figure 3(b). Therefore, the edge collapse or vertex split during a running time must conform to the **Regulations 1** and **2**. Based on the bracketing methods, the vertex tree can be encoded by a set of recorded vertexes and a set of bracket (bitstream of 0 and 1). The bracket set records the implicit connectivity and hierarchy among the recorded vertexes in the vertex tree. The bracket set is the foundation of valid vertex split based on **regulation 2**. Because the minimum unit in the vertex tree is a binary tree (one parent with two children), the vertex traversing algorithm is easy to implement with depth traversing or width traversing in the tree. The connectivity and hierarchy among the vertexes in the vertex tree can be encoded dynamically by traversing the bracket sets.

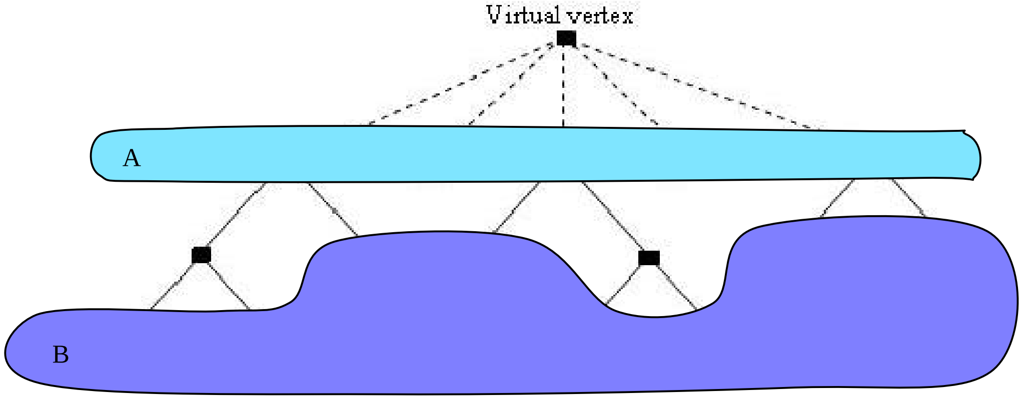


Figure 6 The vertex hierarchy relationship in a vertex tree

3. Implementation of the proposed method

Data structure

Data structure has a great impact on the efficiency of an algorithm or model. According to the above analysis, the original model is represented by a TIN model. In order to facilitate the selection of the candidate edges for collapse and the calculation the distance of a vertex to an average plane, the following vertex, the triangle and vertex tree structures are proposed to fulfill the operational requirements of the proposed method in this study.

```
Class Vertex
{
  Int  m_nIdcode; // the id of the vertex
  Triangle* m_pTriangles; // adjacent triangles
   Vertex*   m_pNeighborVetexes; //adjacent vertex
   Int       m_nLayerinTree;//the layer in the vertextree
   double     m_dDistancetoAveragePlane;
   // the distance to the average plane
//functions begin here……….

};
```

```
Class Triangle
{
// the id of three vertexes in the triangle
   int  m_nCode1;
   int  m_nCode2;
   int  m_nCode3;
  double m_nNormal[3];// the normal of
// the triangle.
// functions begin here …………

};
```

```
Class Vertex tree
{
  Int* m_pVertexSets; // the vertex sets
   byte* m_pbitstream// the bracketing sets (0001…..1)
   Int m_nHeight of tree // the height of vertex tree
//functions begin here……….

};
```

Compared with the vertex tree and vertex structure, the data structure of a triangle is of much simple. In the vertex data structure, the adjacent triangles and adjacent vertexes of the vertex are ordered in counter-clockwise, for acquire the influence region (the polygon) of the vertex more efficiently. The variable m_nLayerinTree is the height of the vertex in the vertex tree. The height of the root vertex in the vertex tree is zero, and the variable m_dDistancetoAveragePlane is used to record the distance of the vertex to the average plane the adjacent vertexes compose. Here, the data structure of vertex tree is developed according to bracketing methods. The vertex sets and bitstream with 0 and 1 are the foundation to represent the connectivity among vertexes. The vertex, triangle and vertex tree data structure form the foundation of the proposed method.

Criterion on edge collapse and vertex split

The operation of edge collapse and vertex split is divided into two steps. The first one is candidate the selection of edges; and the second step is judgment of validity in edge collapse and vertex split operation. The candidate edges are chosen according to the distance to the average plane. The edge with the minimum distance will be collapsed first. According to the above proposed data structure, the influence region of the vertex can be acquired easily once the vertex is selected. The average plane of the influence region can be easily calculated according to the proposed method in Schroeder *et al* (1992), and the distance can be easily calculated by the formula of point to plane. According, the distance to average plane of the adjacent vertexes can be acquired easily. For one edge <p_1, p_2>, supposing Dis_1 and Dis_2 are the distance to the vertexes p_1 and p_2 on the average plane of the influence region. The distance of the edge <p_1, p_2> to its average plane is $(Dis_1+Dis_2)/2$. By using this method, the distance of all the edges to their average plane can be ranked by the distance value in a stack structure. The edge with the minimum distance will be popped from the stack firstly. Although the criterion based on the distance is viable, the other criterion is added to control the quality of model when an edge collapse. In figure 2, the edge in red color is a common edge of two triangles. If the edge is collapsed, the two triangles will disappear. Supposing the normal of the two triangles are n_i and n_j respectively, the angle between the two triangles is

$$q = Arc\cos\left(\frac{n_i \bullet n_j}{|n_i| \bullet |n_j|}\right) \qquad (1)$$

During the procedure of construct a multi-resolution model, the angle between two triangles can be an additional criterion for the selection of candidate edges. For example, the 75^0 can be set as a threshold. If the angle is beyond the threshold, the selected edge is invalid for an edge collapse. This criterion is useful to keep the feature lines, such as geomorphology features like ridge lines, in the original model. Once the candidate edge is chosen, the next step is to judge the validity of edge collapse. The influence region of a vertex to be collapsed can be easily acquired by the data structure. It is clear that an influence region is a polygon, Moreover, the vertexes in the polygon are ordered in counter-clockwise. It is easy to estimate whether the polygon is convex or concave accordingly. Based on the **Regulation 1**, if the influence region is a concave polygon, the vertex is invalid for collapse. Otherwise, the surface foldovers will generate in the model, and it will lead to a wrong topological relationship and can be noticed visually in a 3D visualized TIN model.

According to the above analysis, an edge can be collapsed if the following all criterions are conformed, such as the distance to the average plane, validity and angle. If one edge is collapsed, the new vertex is not introduced to the original model, alternatively one of them is chosen as the final vertex. The criterion is that if one of them is a boundary vertex, the other vertex will be collapsed. Otherwise, the vertex with less distance to the average plane will be collapsed.

When one edge is collapsed, the bracketing method will be applied to encode the parent-child relationship (one parent with two child). The recorded vertexes and brackets will be inserted into the vertex tree. Each of these relationships consists of a vertex tree. The vertex tree records the transformation procedure of original model, which is indispensable for further edge collapse, or the recovery of original model by vertex split. Figure 7 demonstrates the vertex tree and encoding results at a particular running time.

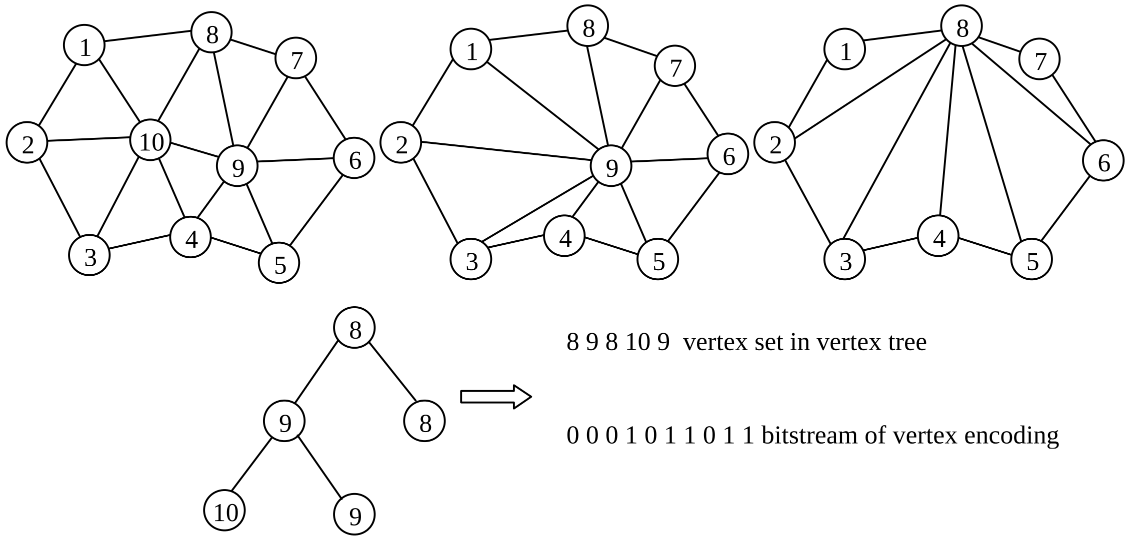


Figure 7. The edge collapse procedure and vertex encoding in a vertex tree

Figure 7 illustrates the edge collapse and the encoding result of a vertex tree. The vertex sets in the vertex tree is 8 9 8 10 9. Supposing the accuracy of the model is lower than the threshold, many of the vertexes need to be split to improve the quality of model. According to the vertex tree, both vertex 8 and vertex 9 can be split. It is clear that the split of vertex 9 must happen before that of vertex 8. Otherwise, the topological relationship in the original model will be changed. When the vertex 9 is to be split, the adjacent vertexes of vertex 9 will be traversed to check and see if these adjacent vertexes conform to the **Regulation 2**. With the vertex traversing algorithm in the vertex tree, the result is that vertex 8 is one of the adjacent vertexes of vertex 9, and vertex 8 in the vertex tree has a higher position than vertex 9. The condition does not conform to **Regulation 2**, consequently the split of vertex 8 must happen before that of vertex 9.

When the edge collapse and vertex split, the balance of the vertex tree will be changed. Each vertex in the vertex tree corresponds to a certain influence region. A good balance of the vertex tree means the edge collapse and vertex split happens among different regions of the original model, and the overlap of these regions occupies a smaller area within the original model. Therefore, the balance of the vertex tree is related to the shape of triangle, a better balance means a less chance in generating long and thin triangles in the model. To create a reasonable balance of the vertex tree, edges, as many as possible, should be collapsed at the each layer of the vertex tree, as a result the continuous edge collapse in a small local region can be avoided.

4. Experiment Study and Analysis

The proposed method is implemented with C^{++} language in Windows XP operation system platform. One region with a total 87,152 triangle and 44,003 vertexes is selected for this a case study. The experiment is tested in a Dell laptop with the configuration CPU Pentium IV 2.0 GHz, RAM 256 Mb.

Different resolution models are acquired based on the proposed method in this paper. As a step further to the past algorithms related to the model simplification, here we consider not only time performance but also evaluation of the model quality. In fact, the quality of multi-resolution terrain model has an important impact on the visual effects and calculation accuracy of model, such as visibility calculation, 3D visualization, 3D queries, and others. For a terrain model, accuracy of vertex elevation is an important factor to the quality of the terrain model. The elevation RMS of a terrain model is a viable indicator to evaluate its quality. In our approach, the elevation RMS of terrain models in various resolutions is calculated to evaluate the model quality. Table 1 shows a summary of the results of the study area.

Table 1 A summary of the results of terrain model in various resolutions

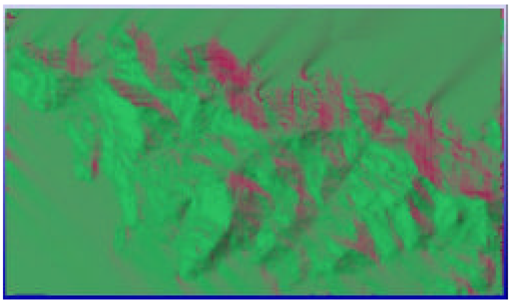
Vertex Number	Triangle Number	RMS (meter)	Time (seconds)
39,091(90%)	77,327(89%)	1.587	7.34
28,562(65%)	56,270(65%)	2.245	13.08
26,032(58%)	51,209(59%)	3.313	28.43
20,937(48%)	41,020(47%)	5.302	48.52
4,927(11%)	9,000 (10%)	10.124	72.12

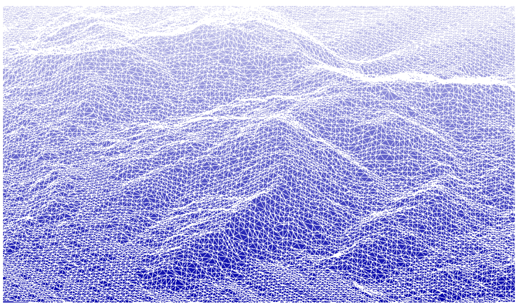
Table 1 summarizes the results of terrain models in various resolutions, and it shows the vertex and triangle number, elevation RMS, and running time of our proposed method.
The experimental results demonstrate that the simplified terrain model, based on the proposed method in this study, can keep higher elevation accuracy and a better time performance. For the scale level_3 TIN model, the triangle number is 59% of the original model, and the corresponding elevation accuracy (RMS) is 3.313 (compared with 1.587 of the original model), and running time is 28.43 seconds. Figure 9 illustrates the 3D visualization result of multi-resolution terrain model, and figure 8(a) and 8(b) shows the approximated or simplified model of the original terrain model with different triangles respectively, including gray model and wire frame model.

Because of edge collapse and vertex split in the running time, long and thin triangle may be generated. The shape of the triangles also has a visual effect on the final results. In our approach, the minimum unit of element is triangle, and long and thin triangle is considered to be a low quality in terms of graphics shape. As in Gueziec (1995), equation 2 is adopted to evaluate the quality of triangle in quantities. s_Δ is the area of the triangle, and l_0, l_1, l_2 is the length of three edges of the triangle.

$$quality = \frac{4\sqrt{3}s_\Delta}{l_0^2 + l_1^2 + l_2^2} \qquad (2)$$

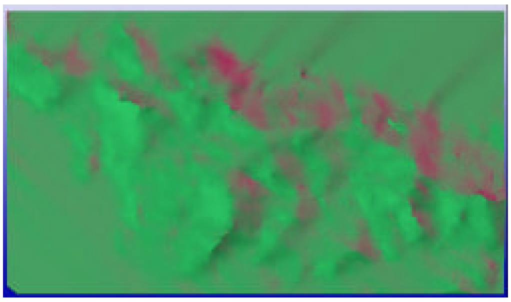
In equation 2, the value of quality is from 0 to 1. For an equilateral triangle, the value of quality is 1. The lower the quality value, the lower the quality of the triangle. To evaluate the triangle quality in the different resolution model, the triangle number in different quality distribute section is calculated. Figure 10 illustrates the results on the quality of the triangles. The number of triangle in the simplified model is about 47% of that of original triangle (total triangle numbers = 41,020).

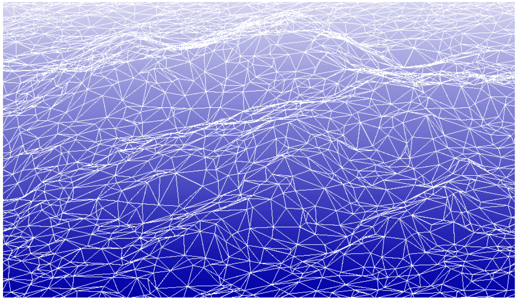




Gray model   Wire frame model

9(a) (triangle number = 56,270 (65%), vertex number = 28,562 (65%))





Gray model   Wire frame model

9(b) (triangle number = 9000 (10%), vertex number = 4,927(11%))

Figure 8 3D visualization result of multi-resolution model

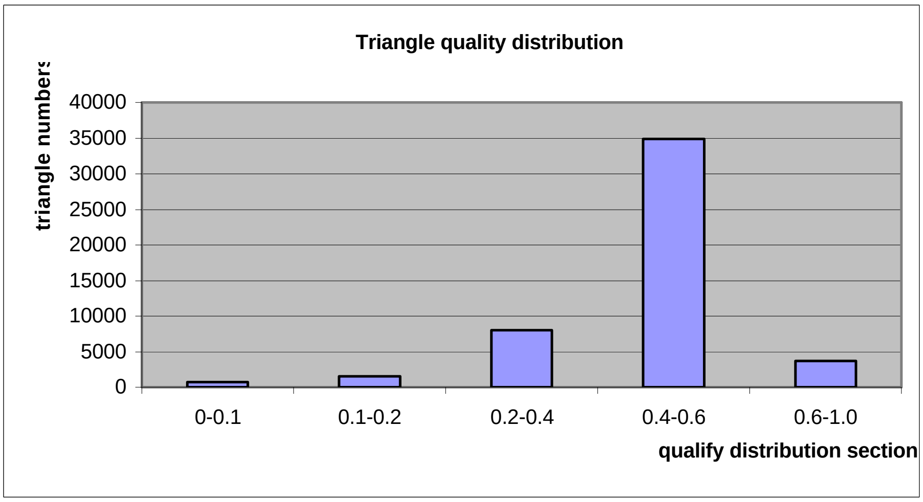


Figure 9 The triangle quality distribution in the multi-resolution model

Figure 9 demonstrates that the proportion of the triangle quality between 0.4 and 1.0 is over 60%, and the proportion of the triangle quality between 0.0 and 0.1 (lower quality) is about 0.2% (triangle numbers = 80). The lower quality triangle number only occupies in a very lower percent, so the triangle quality in the simplified terrain model is in a good quality. The results show that the proposed method in this study has a better ability in acquiring better graphics quality -- triangles.

Time Efficiency Analysis

In order to generate multi-resolution TIN model efficiency, it is vital of important that the time complexity of algorithm is reasonably low. In our method, the time costs includes two phrases. One is the initialization of edge pair; the other is the construction of vertex tree based on edge collapse or vertex split. We suppose the average adjacent vertexes of one vertex is 6, and the average adjacent triangles of one vertex is 10. After edge collapse operation, the adjacent triangles of the vertex reduce to 8. The triangle numbers about reduce by a factor of 4/5. For the edge collapse or vertex split operation can be finished with a cost *O(n)*. The whole time for constructing the vertex tree can be finished with a cost n + 4n/5 + 16n/25 +…= *O(n)*. The above ideal condition is based on selection the edge arbitrary. In actual running time, the edge is selected according to the distance to its average plane. All candidate edges are ordered according to the distance to the average plane, the edge with the minimum distance to the average plane will be collapsed firstly. So the edges can be stored in a heap structure for local updates. Storing the candidate edges in a heap requires *O(nlogn)*. So the whole time complexity of our method is *O(nlogn)*.

5. Conclusions

It is widely interested research issue to generate multi-resolution TIN model dynamically in computer graphics, virtual reality, scientific visualization and GIS. This paper presented a method for constructing multi-resolution TIN model dynamically. The multi-resolution TIN model here is in the sense that different level of details TIN model can be converted dynamically by the proposed solution, specially based on TIN edge collapse and vertex split among different levels of TIN models. First, edge collapse and vertex split is determined according to the minimum distance to an average plane. An edge fulfill this requirement will be collapsed, in the same time, the angle between two triangles is used as an additional criterion for this judgment. Secondly, the vertex tree model is proposed to manage and store the vertex relationship between different levels of TIN models. The corresponding encoding method is proposed based on the bracketing method -- each vertex is represented only with 0 or 1 to encode the position relationship in a tree. Thirdly, two new regulations for edge collapse and vertex split are defined to prevent the problem of surface foldovers during running time.

In the validity judgment of the edge collapse and vertex split, we extended the assessment from time performance only to (a) time, (b) quality. The elevation RMS and triangle shape quality were proposed to evaluate the quality of a generated multi-resolution TIN model. The quality indicators can also be used as the criterion to control the original terrain model simplification. In the proposed method, we do not store vertex dependency relationships explicitly, this is great helpful in reducing data volume and furthermore the validity judgment can be processed in the vertex tree in real time.

The experimental results demonstrated that that the newly proposed method in this study can acquire a reasonable balance between running time, data volume and the model quality -- in terms of elevation RMS and triangle shape quality. A further study on this proposed method will be on improving time performance and terrain feature preservation.

References

Donaghey, R., 1980. Automorphism on Ctalan trees and bracketing. *Journal of combinatorial theory,* B 28:75-90.

De Floriani, L., 1989. A pyramidal data structure for triangle-based surface description, *IEEE computer graphics and application,* vol. 8(2): 67-78.

De Floriani, L., Magillo, P., Puppo, E., 2000. VARIANT: A System for Terrain Modeling at Variable Resolution, *GeoInformatic,* Vol. 4(3): 287-315.

De Floriani, L., Magillo, P., 2001. Multiresolution Meshes, Principles of Multiresolution in Geometric Modeling - *PRIMUS01 summer school, Munich*, August 22-30, 2001, pp193-234.

Eck, M., et al., 1995. Multiresolution Analysis of Arbitrary Meshe, *Computer Graphics*, Vol.29, 173-182.

EI-Sana, J., Varshney, A., 1999. Generalized view-dependent simplification. *Computer graphics forum*, Vol.18(3):84-94

Garland, M., Heckbert,P., 1997. Surface simplification using quadric error metrics, *In SIGGRAPH'97*, pp209-216.

Garland, M., 1999. *Quadric-Based Polygonal Surface Simplication*, Ph.D. dissertation, Computer Science Department, Carnegie Mellon University, 210p.

Hoppe, H., 1996. Progressive meshes, *SIGGRAPH'96 Proceeding*, pp99-108

Hoppe, H., 1998. Smooth View-Dependent Level_of_Detail Control and its Application to Terrain Rendering, *IEEE Visualization 1998*, pp35-42.

Lindstrom, P., Koller, D., et al., 1996. Real-Time Continuous Level of Detail Rendering of Height Fields, *Computer Graphics*, Vol.20, 109-118.

Lindstrom, P., Pascucci, V., 2002. Terrain simplification simplified: A general Framework for view-dependent out-of-core visualization, *IEEE Transaction on visualization and computer graphics*, 8(3): 239-254

Park, D., Cho, H., Kim, Y., 2001. A TIN compression method using Delaunay triangulation, *International journal of geographical information science*, Vol.15(3):255-270.

Ribelles, J., Lopez, A., Belmonte, O., Remolar, I., and Chover, M., 2002. Multiresolution modeling of arbitrary polygonal surfaces: a characterization, Computer&graphics, 26:449-462.

Schroeder, W., Zarge,A.J., Du,W., 1992. Decimation of triangle meshes. *Computer Graphics,* 26(2):65-70

Xia, J., El-Sana J, Varshney, A., 1997. Adaptive real-time level-of-detail-based rendering for polygonal model. *IEEE Transactions on Visualization and computer graphics*,Vol.3(2):171-183