# A Precise TIN Clipping Algorithm for Digital Mining Design of the Open-pit Coal Mine

**Jingchang Zhao** ( ltuzjc@126.com )
Liaoning Technical University https://orcid.org/0000-0003-0659-0681
**Fei Gao**
Liaoning Technical University
**Guangwei Liu**
Liaoning Technical University
**Dong Wang**
Liaoning Technical University

**Research**

**Keywords:** Grid Index, TIN Topology, One-time Edge-prior CDT Growth Algorithm, Precise TIN Clipping, Local Detail Features, Digital Mining Design, Open-pit Coal Mine

**Posted Date:** June 26th, 2020

**DOI:** https://doi.org/10.21203/rs.3.rs-37339/v1

# A Precise TIN Clipping Algorithm for Digital Mining Design of the Open-pit Coal Mine

Jingchang Zhao[1,2,*], Fei Gao[3], Guangwei Liu[2], Dong Wang[1]

1. *Institute of Technology and Equipment for Exploitation and Utilization of Mineral Resources, Liaoning Technical University, Fuxin 123000, China*; 2. *School of Mining, Liaoning Technical University, Fuxin 123000, China*; 3. *School of Mechanics & Engineering, Liaoning Technical University, Fuxin 123000, China*

**Abstract:** TIN clipping algorithm is one of the basic algorithms for digital mining design of open-pit coal mine based on TIN model. In this paper, a precise TIN clipping algorithm for digital mining design of open-pit coal mine is proposed. Based on the constructed grid index of the clipped TIN and the clipping polygon, the clipping polygon is embedded into the clipped TIN by interpolating the vertex elevation of the clipping polygon and calculating its intersections with the clipped TIN. Then, according to the reconstructed topology of the TIN triangles located inside (outside) the clipping polygon, generate the boundary of those two triangles set, and construct the TIN between the boundaries using the one-time edge-prior CDT growth algorithm, thus , achieve the precise TIN clipping keeping the clipped TIN's local detail features. The experiment results show that the algorithm proposed in this paper is efficient, stable in performance, and it has been applied to the digital mining design of the open-pit coal mine.

**Key words:** Grid Index; TIN Topology; One-time Edge-prior CDT Growth Algorithm; Precise TIN Clipping; Local Detail Features; Digital Mining Design; Open-pit Coal Mine

*Corresponding author. Tel: 86418-3350462, E-mail: lntuzjc@126.com

## 1. Introduction

Building the geological model of the deposit is the basis of realizing the digital mining design of open-pit coal mine. Considering that the coal deposit is a typical sedimentary deposit and the coal bearing stratum generally has a perfect stratiform structure, therefore, the DEM (Digital Elevation Model) described by TIN (Triangulated Irregular Network) is usually selected for the stratum geological models of the coal deposit.

In the subsequent digital mining design process, it is often necessary to clip the built geological stratum DEM according to the local mining or geological data update range. As mentioned above, the stratum DEM is described by TIN, therefore, it is of great significance to design and implement a TIN clipping algorithm for realizing the digital mining design of open-pit coal mine based on the geological stratum DEM.

TIN clipping includes surface/surface clipping and edge/surface clipping. Surface/surface clipping separates the TIN by calculating the intersection of the two TINs; edge/surface clipping uses a spatial curve to separate the TIN. Concerning surface/surface clipping, Maillot (1991) proposed a clipping algorithm based on the Sutherland-Hodgman polygon (Sutherland et al. 1974) to clip the triangles strip by the plane, and Lindenbeck et al. (2002) designed a TRICUT program based on RAPID (Robust and Accurate Polygon Interference Detection) (Gottschalk et al. 1997) and TRIANGLE library, the program achieved the mutual clipping of the TINs by computing their intersection lines. Hua et al. (2006) improved the algorithm proposed by Lindenbeck (2002), the algorithm implemented the collision detection between the surfaces by establishing the OBB (Oriented Bounding Box) tree of the TIN, and then calculating the intersection points of the intersecting triangles, finally, the fast geology model reconstruction after clipping was completed by normalizing the vertex coordinates. Li et al. (2008) simplified the intersection of the TINs into the intersection of the TIN and rectangular mesh, thus the clipping speed is increased. Compared with the mature surface/surface clipping algorithm, due to the complexity of the spatial curve, the algorithm arbitrary curve clipping surface cannot be implemented, resulting in the research of the edge/surface algorithm of TIN is not mature.

Zhong et al. (2010) proposed a projection strategy to clip the TIN. The algorithm firstly projects the TIN onto a two-dimensional plane, and then each edge of the clipping polygon is embedded into the TIN by the constrained Delaunay triangulation, the extra triangles outside the clipping polygon are deleted by the edge-triangle topological relationship, finally, the vertices elevation of the clipping polygon are interpolated by the edge-point topological relationship, the TIN to be clipped was generated. Yang et al. (2014) proposed a TIN clipping algorithm based on topology tracing, which mainly uses the spatial curve attached to the TIN to perform topology tracing along the TIN and separate the TIN along the tracing track, then the TIN is clipped.

In reference [7], the TIN is clipped by embedding the clipping polygon edges into the TIN and the triangle outside the clipping polygon is deleted by the topology of the edge-triangle. In reference [8], the TIN clipping is implemented by inserting the clipping polygon vertices into the triangles, “breaking” the edge or triangle according to the positional relationship of the vertex and the triangle, and reconstructing the TIN by the TIN’s topology. In short, neither the algorithm mentioned in reference [7] nor in the reference [8] takes into account of the local detail features of the TIN to be clipped, the TIN is not clipped “precisely” and it is easy to cause

lack fidelity of the clipped TIN. In this paper, a precise TIN clipping algorithm keeping the clipped TIN's local detail features for digital mining design of the open-pit coal mine is proposed, and the algorithm has been implemented with C#.NET and applied to the digital mining design practice of open-pit coal mine .

## 2. Algorithm Thought and Data Structure

### 2.1 Algorithm Thought

In order to improve the efficiency of the TIN clipping, firstly it's necessary to construct the grid index according to the range of the clipped TIN and the length of the triangle's edges in the TIN, then map the points, edges of the clipping polygon and the triangles of the clipped TIN to each cell of the grid index. After that, we can interpolate the clipping polygon's vertices' elevation based on the clipped TIN, compute the intersection of the clipping polygon and the triangles in the clipped TIN, that is to say, the clipping polygon is embedded into the clipped TIN, then determine the triangles set of the clipped TIN inside (outside) the clipping polygon according to the location relationship of the points and the polygon, and construct a "point-edge-triangle" topology of the triangles set, based on that, generate the boundary of the triangles set located inside (outside) the clipping polygon according to the adjacency of the edge and the triangle. Finally, a new boundary TIN between the clipping polygon and the triangles set boundary is generated, finally the TIN clipping is completed with separated the TIN to be clipped off from the clipped TIN by the topological relationship modification between the boundary edges and their adjacent triangles, **t**hus the TIN has been clipped precisely.

### 2.2 Data Structure of the Algorithm

The main data objects involved in the algorithm include: TIN, grid index, triangles, edges, vertices, etc. in TIN. The data structure defined by C#.NET is shown in Fig. 1.

| CDTIN |
|---|
| vert_List //Vertexes set<br>tri_ List //Triangles set<br>ed_List //Edges set |

| Tri |
|---|
| tri_ID //Triangle ID<br>vert_ID[3] //Vertexes' IDs array<br>ed_ID[3] //Edges' IDs array |

| Grid |
|---|
| gridCell_Arr[rowNum,ColNum] //Array of grid index cells<br>cellSize //Size of grid index cell<br>rows_Num //Row number of grid index cell array<br>cols_Num //Column number of grid index cell array |

| GridCell |
|---|
| rt_Coor[2] //Coordinate of the upper-right vertex<br>lb_Coor[2] //Coordinate of the lower-left vertex<br>row_ID //Row ID<br>col_ID //Column ID<br>vert_List //Incident vertexes set<br>ed_ List //Incident edges set |

| Vertex |
|---|
| vert_ID //Vertex ID<br>x,y,z //Coordinate of vertex<br>vert_Prop //Property of vertex: if vertex is on the boundary, vert_Prop=1; if vertex isn't on the boundary, vert_Prop= 2; if vertex is discrete vertex, vert_Prop=3<br>tri_ List //Set of triangles adjacent to a vertex<br>ang_Sum //Sum of all interior angles associated with the same vertex<br>gridCell_ID //Grid index cell ID |

| Edge |
|---|
| ed_ID //Edge ID<br>vert_ID[2] //Endpoints' IDs array<br>ed_Prop //Edge property: boudary edge, ed_Prop=1; inner constrained edge, ed_Prop=2; other edges, ed_Prop=3<br>left_triID //Left neighboring triangle ID<br>right_triID //Right neighboring triangle ID<br>grid_List //Neighboring grid index cells set<br>gridCellId_List //Grid index cells' IDs set |

Fig. 1 Data structure of the algorithm

## 3. The Algorithm

### 3.1 Establishing a Grid Index of the clipped

**TIN and the Clipping Polygon**

The purpose of establishing the spatial index of the clipped TIN and the clipping polygon is to realize rapid positioning of the spatial geometric elements and speed up the subsequent operation. Among the common spatial indexes, the grid index is an efficient, concise and easy to realize (Wu et al. 2001), its basic thought is to divide the minimum enclosing rectangle of the spatial geometric element set into a space composed of several small cells, then the spatial geometric elements to be processed are assigned to the corresponding grid cells according to the spatial positional relationship, thereby establishing a spatial index of the spatial geometric element set.

The specific steps to create a grid index of the clipped TIN are as follows:

(1) Determining the minimum enclosing rectangle of the clipped TIN;

If the maximum and minimum values of the $X$ and $Y$ direction coordinates of all the triangles in the clipped TIN are $X_{max}$,$Y_{max}$,$X_{min}$,$Y_{min}$, then the two vertices of the clipped TIN minimum enclosing rectangle main diagonal are determined: ($X_{min}$,$Y_{min}$), ( $X_{max}$,$Y_{max}$).

(2) Dividing the minimum enclosing rectangle into $l \times m$ grid cells according to the number and geometric features of the clipped TIN's triangles;

The size of the grid index cells determines the number of triangles associated with the grid cell. If the grid cell is too large or too small, the efficiency of the algorithm will be affected. In this paper, the grid cell size which determined by the experimenting is 1.3 times the average edge length of all the triangles in the clipped TIN.

Taking two integers ($i$, $j$) to uniquely identify each grid index cell. The values of $i$, $j$ respectively represent ID number of the grid index cell in $x$ and $y$ direction.

(3) Mapping the triangles of the clipped TIN to the grid cells according to the spatial position relationship between the triangles and the grid cells;

Assuming $x_{min}$, $x_{max}$, $y_{min}$, $y_{max}$ are the maximum and minimum coordinates in $x$ and $y$ directions of the one triangle of the clipped TIN, then the range of grid cells occupied by the triangle in $x$ and $y$ directions can be determined as follows:

$$i:\left[Int\left(\frac{x_{min}-X_{min}}{cellsize}\right)+1\right]\rightarrow\left[Int\left(\frac{x_{max}-X_{max}}{cellsize}\right)+1\right]$$

$$\mathrm{j}:\left[Int\left(\frac{y_{min}-Y_{min}}{cellsize}\right)+1\right]\rightarrow\left[Int\left(\frac{y_{max}-Y_{max}}{cellsize}\right)+1\right]$$

Applying the above method, a grid index of the clipping polygon can also be established according to the spatial position relationship between the polygon constituent edges and the grid cells.

After created the grid index of the clipped TIN and the clipping polygon, only vertices, edges, and triangles associated with the same cells performing point-to-triangle positional or edge-intersection tests are required in the course of the the clipping polygon vertices' elevation interpolation and the intersections calculation of the constituent edges of the clipping polygon and the clipped TIN's triangles, without the ergodic of all triangles in the TIN, thus the TIN clipping algorithm's efficiency is evidently enhanced..

**3.2 Embedding the Clipping Polygon into the Clipped TIN**

Clipping polygon embedding into the clipped TIN firstly needs to interpolate the vertices of the polygon based on the clipped TIN, then calculate the intersections of the line segments of the clipping polygon and edges of TIN's triangles, finally insert the intersections into the correct position of the clipping polygon's vertices sequence.

(1) Elevation interpolating of the clipping polygon's vertices;

When interpolating the clipping

polygon's vertices elevation based on the clipped TIN, determining which triangle the polygon's vertex falls in of the clipped TIN is the first problem to be solved. The unified grid index of the clipped TIN and the clipping polygon has been established in section 3.1 of this article. With the established grid index and the algorithm for judging the position relationship between points and triangles, which triangle the polygon vertex falls into can be quickly located by traversing the triangles adjacent to the same grid cell index with the interpolated polygon's vertex.

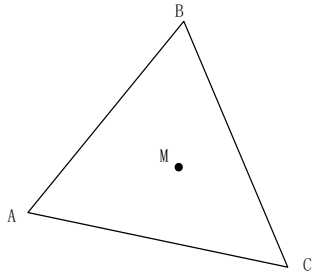



Fig. 2 Diagram of point and triangle position relation

Vector cross multiplication method is used to judge the relationship between the interpolated polygon's vertex and the triangles. As shown in Fig. 2, $\overrightarrow{MA}$, $\overrightarrow{MB}$ and $\overrightarrow{MC}$ are the vectors formed by the three vertices of $\Delta ABC$ and vertex $M$, the following rules can be used to determine whether the vertex is inside $\Delta ABC$:

1) One of the following conditions is satisfied, and the vertex $M$ is located inside $\Delta ABC$;

①

$\overrightarrow{MA} \times \overrightarrow{MB} > 0 \,\&\, \overrightarrow{MB} \times \overrightarrow{MC} > 0 \,\&\, \overrightarrow{MC} \times \overrightarrow{MA} > 0$;

② $\overrightarrow{MA} \times \overrightarrow{MB} < 0 \,\&\, \overrightarrow{MB} \times \overrightarrow{MC} < 0 \,\&\, \overrightarrow{MC} \times \overrightarrow{MA} < 0$;

2) One of the following conditions is satisfied, the vertex $M$ is located on the side of $\Delta ABC$;

① $\overrightarrow{MA} \times \overrightarrow{MB} = 0$;

② $\overrightarrow{MB} \times \overrightarrow{MC} = 0$;

③ $\overrightarrow{MC} \times \overrightarrow{MA} = 0$;

3) None of the above conditions are satisfied, the vertex $M$ is outside of $\Delta ABC$.

If the vertex is inside of the triangle, its elevation can be calculated using the plane equation formed by the three vertices of the triangle.

Assumed that the coordinates of the three vertices of $\Delta ABC$ are $(x_A, y_A, z_A)$, $(x_B, y_B, z_B)$, $(x_C, y_C, z_C)$, thus the three points can determine the plane vector $\vec{n}$ as follows:

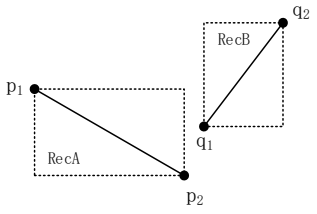
$$\begin{cases} \vec{n} = a\vec{i} + b\vec{j} + c\vec{k} \\ a = (y_B - y_A)(z_C - z_A) - (z_B - z_A)(y_C - y_A) \\ b = (z_B - z_A)(x_C - x_A) - (x_B - x_A)(z_C - z_A) \\ c = (x_B - x_A)(y_C - y_A) - (y_B - y_A)(x_C - x_A) \end{cases}$$

Then the elevation $z_M$ of the vertex $M$ to be interpolated is:

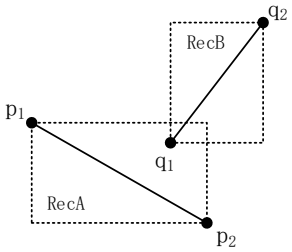
$$z_M = z_A - \frac{a(x_M - x_A) + b(y_M - y_A)}{c}$$

(2) Calculating the intersections of the clipping polygon and the clipped TIN

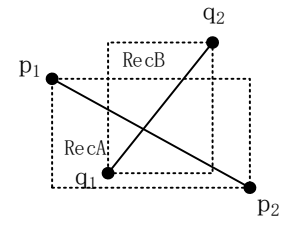
By using the established grid index, the triangles of the clipped TIN that may intersect the line segments of the clipping polygon are quickly determined, then the intersection of each component line segment of the clipping polygon and each edge of the triangle is calculated by using the line segment intersection algorithm (Sun et al. 2005).



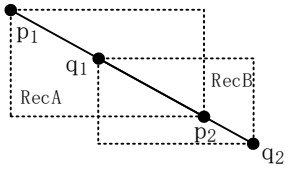

(a)




(b)




(c)




(d)

Fig. 3 Diagram of line segments intersection

There are three kinds of position relations between two line segments: coincidence, intersection and non-intersection. As shown in Fig. 3, the intersection of two straight line segments *p1p2*and *q1q2*is calculated as follows:

1) Rapid rejection detection

If the rectangle *RecA* with *p1p2* as the diagonal line and the rectangle *RecB* with *q1q2* as the diagonal line do not intersect, *p1p2* and *q1q2* must not intersect, otherwise, *p1p2* and *q1q2* may intersect.

Whether the two rectangles *RecA* and *RecB* intersect can be determined as follows:

If the expressions $RecA.minX \le RecB.maxX$, $RecB.minX \le RecA.maxX$, $RecA.minY \le RecB.maxY$, $RecB.minY \le RecA.maxY$ are held, then *RecA* and *RecB* intersect, otherwise, they do not intersect.

As shown in Fig. 3 (a), *RecA* and *RecB* do not intersect, *p1p2* and *q1q2* do not intersect, Fig. 3 (b) *RecA* and *RecB* intersect, but *p1p2* and *q1q2* do not intersect, Fig. 3 (c) *RecA* and *RecB* intersect, *p1p2* and *q1q2* intersect. It can be seen that the intersection of *RecA* and *RecB* cannot be a sufficient condition for the intersection of *p1p2* and *q1q2*, which needs further judgment.

2) Cross detection

When the two lines intersect, they must cross each other. As shown in Fig. 3 (c), the conditions for judging whether the line segments *p1p2* and *q1q2* intersect with the cross detection method are:

$$① \left(\overrightarrow{q_1p_1} \times \overrightarrow{q_1q_2}\right) \cdot \left(\overrightarrow{q_1p_2} \times \overrightarrow{q_1q_2}\right) < 0$$

$$② \left(\overrightarrow{p_1q_1} \times \overrightarrow{p_1p_2}\right) \cdot \left(\overrightarrow{p_1q_2} \times \overrightarrow{p_1p_2}\right) < 0$$

When the above two conditions are true, the two line segments must intersect.

3) Calculating the intersection of the line segments

After the rapid rejection test and the cross detection, the following methods are used to calculate the intersections for the line segments that are determined to intersect.

As shown in Fig. 3 (c), assuming the two endpoints' coordinates of the line segments *p1p2* and *q1q2* are *(x1, y1)*, *(x2, y2)*, *(x3, y3)*, *(x4, y4)*, and the coordinate of the intersection is:

$$\begin{cases} x_0 = \dfrac{d_1}{d} \\ y_0 = \dfrac{d_2}{d} \\ d_1 = b_2(x_2 - x_1) - b_1(x_4 - x_3) \\ d_2 = b_2(y_2 - y_1) - b_1(y_4 - y_3) \\ d = (x_2 - x_1)(y_4 - y_3) - (x_4 - x_3)(y_2 - y_1) \\ b_1 = (y_2 - y_1)x_1 + (x_1 - x_2)y_1 \\ b_2 = (y_4 - y_3)x_3 + (x_3 - x_4)y_3 \end{cases}$$

After calculating the plane coordinates of the intersection of the line segments of the clipping polygon and the triangle of the clipped TIN, the elevation of the intersection can be further calculated by the linear interpolation method (Sun et al. 2005).

While calculating the intersection of the clipping polygon and the clipped TIN, inserting the intersection into the clipping polygon vertices sequence using the distance method (Sun et al. 2005), then a new clipping polygon with that inserted intersection polygon vertices sequence is generated.

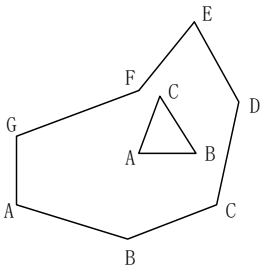
**3.3 Generating Boundary of the TIN to be Clipped Off**

After embedding the clipping polygon into the clipped TIN, it is necessary to generate the TIN boundary inside or outside the clipping polygon, so as to reconstruct the TIN between the clipping polygon and the TIN boundary in the subsequent process.

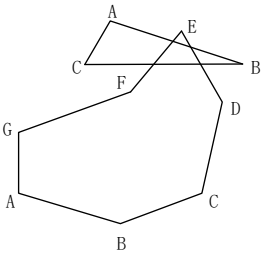
### 3.3.1 Obtaining the Triangles Inside/Outside of the Clipping Polygon

To get the triangles inside (outside) the clipping polygon, the first step is to determine the positional relationship (inside, intersecting, outside, as shown in Fig. 4) between the triangles of the clipped TIN and the clipping polygon.

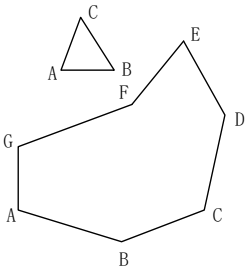
When the three vertices of the triangle are inside the clipping polygon, the triangle must be inside the clipping polygon (Fig. 4 (a)); however, when the three vertices of the triangle are outside the clipping polygon, it is not certain that the triangle must be outside the clipping polygon (as shown in Fig. 4 (b)), and at this point, further determination is needed to finally determine the positional relationship between the triangle and the clipping polygon according to whether it intersects with the clipping polygon or not.




(a)




(b)




(c)

Fig. 4 Diagram of the position relationship between the triangle and clipping polygon

The grid index and the algorithm to judge whether the line segments intersect (mentioned in section 3.2 of this article) can be used to determine whether a triangle intersects with a clipping polygon. The position relationship (inside, outside, on the edge) between the point and the polygon can be judged by improved ray method (Zhao et al. 2017), the steps are as follows:

(1) Judging the relationship between the point and the minimum enclosing rectangle of the polygon. If the point is outside the minimum enclosing rectangle of the polygon, it can be directly judged that the point is outside the polygon, otherwise step (2) is continued;

(2) According to the equations of the line segments of the polygons, determining whether the points are located on the edges of the polygons (in this paper, the triangle vertex is located on the edge of the polygon, which is treated as intersecting with the polygon);

(3) If the point is not on the edge of the polygon, calculate intersections of the ray emitted from the point and the polygon, count the number of the intersections. When the number of intersections is even, the point is inside the polygon, otherwise, it is outside the polygon, as shown in Fig. 5.

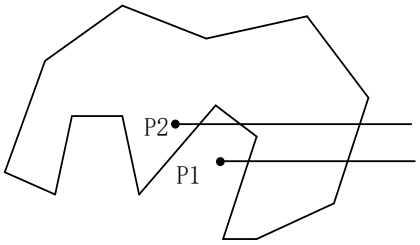


Fig. 5 Diagram of position relationship between point and polygon

The above method is applicable to judge the positional relationship between the point and the convex or concave polygon.

Whether the clipping polygon and the triangle intersect can be judged by the line segment intersection detection algorithm proposed in section 3.2.

After judged the position relationship between the point and the clipping polygon and whether the clipping polygon and the triangle intersect, the triangles inside (outside) the clipping polygon can be obtained.

### 3.3.2 Reconstructing the Topology of the Triangles Inside/outside of the Clipping Polygon

In order to generate the boundary of the triangles inside (outside) the clipping polygon, it is necessary to reconstruct the topology of those triangles according to the "edge-edge" and "edge-triangle" adjacency relationship.

Vertex aggregation and duplicate edges merging are the two main tasks in the course of reconstructing the TIN topology. In this paper, a TIN topology reconstruction algorithm based on Hash function and half-edge data structure is applied. Firstly, the Hash function is used to calculate the hash address of the vertex. When the vertex hash address has "conflict", the list combined with AVL tree is applied for vertex aggregation. During the vertex aggregating, an improved half edge data structure is used to build an index table of incident half-edge for each vertex to complete the duplicate edges merging. Then establishing "edge-edge" and "edge-triangle" adjacency relationship to reconstruct the TIN topology.

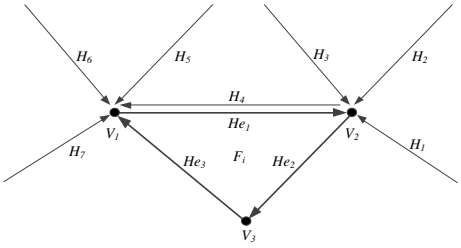



Fig. 6 Diagram of TIN's half-edge merged

The steps to reconstruct the TIN topology based on the Hash function and the half-edge data structure are as follows (Zhao et al. 2017):

(1) As shown in Fig. 6, read a triangle $F_i$ of the TIN;

(2) The hash addresses of the three vertices $V_1$, $V_2$, and $V_3$ of a triangle are calculated by the following Hash function:

$$Index = (\text{int})\left((\alpha X + \beta Y + \gamma Z)C + 0.5\right) \& T$$

In the above equation, $\alpha$, $\beta$, $\gamma$ are the coefficients of the coordinates of the vertex of the triangle, and the values of $\alpha$, $\beta$, $\gamma$ directly affect the performance of the Hash function. Through a large number of experimental studies, Jan et al. (2003) thought that $\alpha = 3$, $\beta = 5$, $\gamma = 7$ is more reasonable; $C$ is the proportional coefficient. In order to make full use of the computer storage and prevent overflow and increase the load factor value, $C$ value can be determined by the following steps:

1) Calculate the maximum coordinates of the triangle vertices, $X_{\max}$, $Y_{\max}$, $Z_{\max}$, then:

$$\xi_{\max} = \alpha X_{\max} + \beta Y_{\max} + \gamma Z_{\max}$$

2) $C=min\{C_1,C_2\}$, where $C_1\xi_{\max} \le 2^{32}-1$, $C_2 = 2^{32} - 2^k$, $T$ is the length of the hash table, which is generally the range of integers that can be expressed by the computer, and its value is between $(0, 2^k)$.

If the slot list corresponding to the hash address of each vertex in the hash table is not empty, then judge whether the current vertex

coincides with the vertex in the address slot list. If they are coincident, assign the ID value of the coincident vertex to the current vertex, otherwise, insert the current vertex into the slot list, and assign *num*+1 as ID to the current vertex (variable *num* is the number of non-coincident vertices of the TIN);

(3) In Fig. 6, the half-edge $He_1$ of the triangle $F_i$ contains the vertices $V_1$, $V_2$, both $V_1$ and $V_2$ have coincident vertices, therefore, it is necessary to find the partner half-edge with the same end points but opposite direction as $He_1$, then merge the half-edge $He_1$.

(4) To merge $He_1$, it is only necessary to find the partner half-edge of $He_1$ in the half-edge table with $V_1$ as the end point under the condition of whether the endpoint's ID is equal or not. In the TIN model shown in Fig. 6, the half-edges with $V_1$ as the endpoint are $H_4$, $H_5$, $H_6$, *and H7*. According to the condition of determining partner half-edge, $H_4$ is the partner half-edge of $He_1$;

(5) Update the half-edge table with the related vertices as the endpoint. As shown in Fig. 6, deleting the half-edge $H_4$ from the half-edge table with $V_1$ as the endpoint (each half-edge has at most one partner half-edge), while inserting the half-edge $He_3$ into the half-edge table with $V_1$ as the endpoint, at the same time inserting the half-edge $He_2$ into the half-edge table with $V_3$ as the endpoint;

(6) Insert the three half-edges $He_1$, $He_2$, $He_3$ of the current triangle $F_i$ into the half-edge set of the TIN.

Following the above steps (1) to (6) to traverse all the triangles inside (outside) of the clipping polygon, that is, completing the topology reconstruction of those triangles.

With the reconstructed topology, and the "edge-triangle" adjacency relationship, the boundary half-edges can be found, they have only one adjacent triangle. According to the "edge-edge" adjacent relationship, the boundary polygon of the triangles inside (outside) of the clipping polygon is traced.

### 3.4 TIN Clipping

#### 3.4.1 Reconstruct the Boundary TIN

After embedding the clipping polygon into the clipped TIN, reconstructing the topology and generating the boundary of the triangles inside (outside) the clipping polygon, it is necessary to reconstruct the boundary TIN between the clipping polygon and the boundary of the triangles inside (outside) the clipping polygon to complete the precise clipping of the TIN.

In the construction of the boundary TIN, this paper applies a one-time edge-prior CDT (Constrained Delaunay Triangulation) growth algorithm (Zhao et al. 2015). Firstly, the algorithm selects the edge whose number of adjacent triangles is less than 2 as the extended edge, and DT point which can form a Delaunay triangle with current extended edge is searched quickly in the point set by the minimum enclosing rectangle method, then the DT point and the current extended edge are formed into Delaunay triangle, repeat the above process until the number of adjacent triangles of all edges is 2, the construction of CDT is completed.

Certainly, the boundary TIN construction can also use such classical algorithms as Divide-conquer Algorithm (Chew et al. 1989), Two-phase algorithm (Sloan et al. 1993), Sweep-line algorithm (Domiter et al. 2008), etc.

#### 3.4.2 TIN Separating

There is still a topological relationship between the newly constructed boundary TIN and the clipped TIN, and the TINs are not completely separated.

Therefore, it is also necessary to modify the topological relationship between the boundary edges and their adjacent triangles of the clipped TIN, set the boundary edges' adjacent triangles' number to 1, then the topological association between the boundary

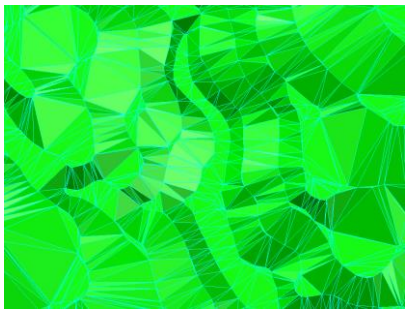
edges and the adjacent triangles has been deleted, thus the TIN to be clipped off is separated from the original clipped TIN, and the TIN clipping is completed.

## 4. Experiment and application

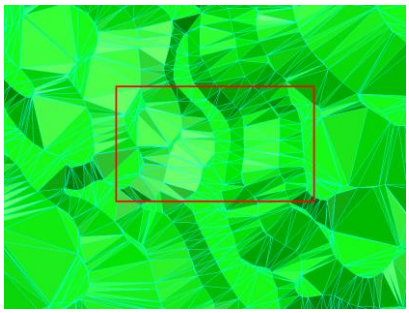
### 4.1 Algorithm experiment

Fig. 7 shows the experiment result of the TIN clipping algorithm considering the local detail features proposed in this paper. Fig. 7 (a) shows the original terrain TIN to be clipped, and the red rectangle in Fig. 7 (b) is the clipping polygon, Fig. 7 (c) shows the TIN clipping result when the local terrain details are neglected, and Fig. 7 (d) shows the TIN clipping result applying the precise clipping algorithm proposed in this paper, and the local terrain detail features are remained.

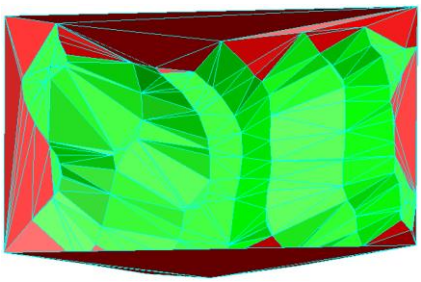
Comparing Fig. 7 (c) with Fig. 7 (d), it is found that the TIN is clipped when the local terrain detail features are neglected, resulting in obvious lack fidelity of the terrain at the clipping boundary, and the clipped TIN is seriously inconsistent with the original terrain TIN. When the algorithm of this paper is used to clip the TIN, because of the interpolation the clipping polygon's vertices and intersection calculation between the line segments of the clipping polygon and the triangle edge of the TIN , the local detail features of the clipped TIN are well preserved (Fig. 7 (d)).



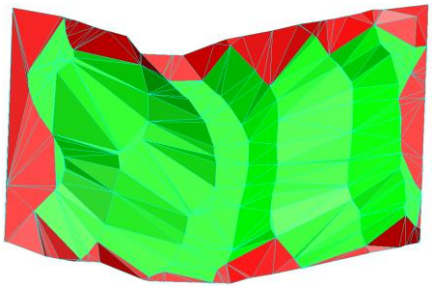
(a) original terrain TIN



(b) clipping polygon



(c) clipping result neglecting the local detail features



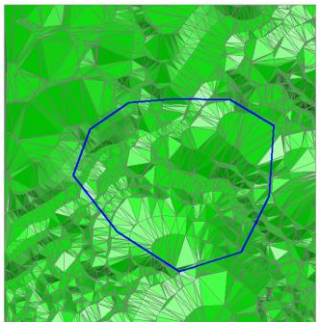
(d) clipping result of the proposed algorithm

Fig. 7 Experiment result of the proposed precise TIN clipping algorithm

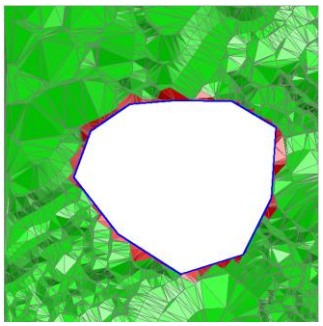
### 4.2 Algorithm Application

The algorithm proposed in this paper has been successfully applied in the digital mining design practice of the open-pit coal mine.

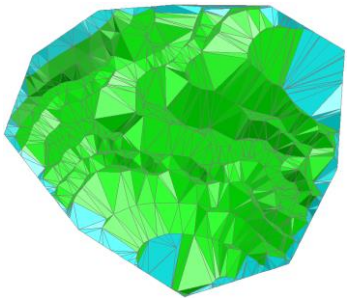
Fig. 8 shows the effect of applying the proposed algorithm to the local terrain DEM clipped. Fig. 8 (a) is the original terrain TIN model and the clipping polygon (blue), and Fig. 8 (b) is the clipping result of preserving the outside TIN of the clipping polygon, where the red part is the newly constructed TIN between the inserted clipping polygon and the boundary, and Fig. 8 (c) it is a clipping result that preserves the inside TIN of the clipping polygon.



(a) Original terrain TIN and clipping polygon



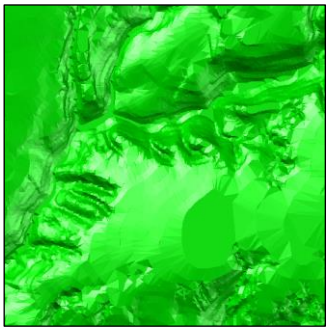
(b) Terrain TIN outside clipping boundary



(c) Terrain TIN inside clipping boundary

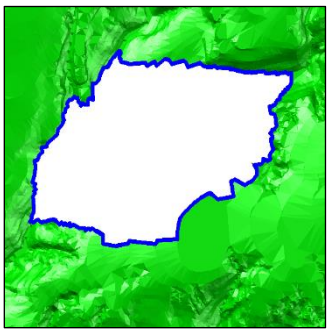
Fig. 8 Terrain DEM clipping

Fig. 9 shows the effect applying the algorithm proposed in this paper to the open-pit coal mine dump design, according to the boundary of the designed dump bench TIN, precisely clip the original terrain TIN, and merge designed dump bench TIN with the clipped terrain TIN.

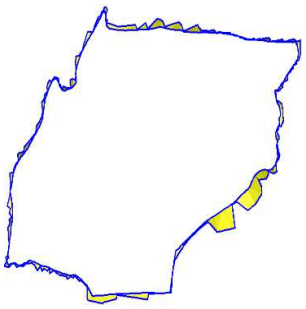


(a) Original terrain TIN

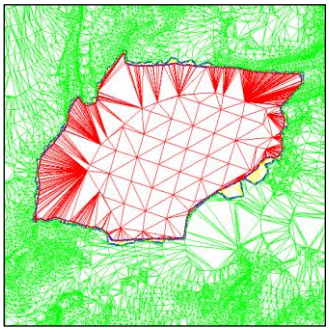
(b) Designed dump TIN and the boundary polygon

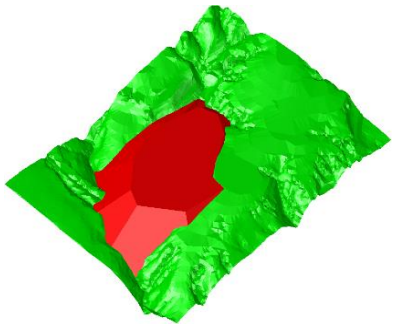


(c) Precise clipping of terrain TIN



(d) Boundary TIN





(e) Clipped terrain TIN and designed dump step TIN merged (f) rendering effect of merged TINs

Fig. 9 Dump site design of open-pit coal mine applying the proposed TIN clipping algorithm

## 5 Conclusion

(1) As the basis of improving the efficiency of vertex elevation interpolation, intersection calculation, topology reconstruction and boundary TIN construction, the grid index of TIN and the clipping polygon is established by taking 1.3 times of the average edge length of triangles in the TIN as the grid unit size;

(2) The Hash function and the improved half-edge data structure are applied to the vertex aggregation and half-edge merging of TIN to realize the fast reconstruction of TIN topology;

(3) Based on the reconstructed TIN topology, according to the position relationship between triangles and the clipping polygons, the boundary of triangles located inside (outside) the clipping polygon is obtained, and a one-time edge-prior CDT construction algorithm is applied to construct the boundary TIN, and the TIN to be clipped off is separated from the clipped TIN by modifying the edge-triangle adjacent relationship;

(4) Experimental results show that the algorithm proposed in this paper can retain the local details and achieve the precise clipping of the TIN;

(5) The algorithm proposed in this paper has been successfully applied in the digital mining design of open-pit coal mine, with stable operation and high time efficiency.

## References


Maillot P G (1991) Three dimensional homogeneous clipping of triangle strips. ARVO J. Graphics Gems II. New York: AP professional.

Sutherland I E, Hodgman G W (1974) Reentrant polygon clipping. Communications of the ACM 17(1):32-42.

Lindenbeck C, Ebert H, Ulmer H, Lavorante L P, Pflug R (2002) TRICUT: A program to clip triangle meshes using the rapid and triangle libraries and the visualization toolkit. Computers & Geosciences 28(7): 841-850.

Gottschalk S, Lin M C, Manocha D (1997) RAPID:Robust and Accurate Polygon Interference Detection. http://www.cs.unc.edu/～geom/OBB/OBBT.html.

Hua W H, Deng W P, Liu X G, Shang J G (2006) Improved Partition Algorithm between Triangulated Irregular Network. Earth Science-Journal of China University of Geosciences 31(5): 619-623.

Li J W, Li J G (2008) Rapid Rectangular Grid-Based Clipping Algorithm of Surface. Microcomputer Information 24(27):157-159.

Zhong J M, Guo X Z, Li Y (2010) Precise clipping algorithm for 3D triangulated irregular network. Computer Engineering and Applications 46(16):204-206+231.

Yang Y, Li ZL, Pan M (2014) Clipping Algorithm for Triangulated Irregular

Network Based on Topology. Geography and Geo-Information Science 30(3):21-24.

Wu L, Liu Y, Zhang J, Ma X J, Wei Z Y, Tian Y (2001) Geographic Information System: Principle, Method and Application. Beijing: Science Press.

Sun J G, Hu S M (2005) Basic Course of Computer Graphics. Beijing: Tsinghua University Press.

Zhao J C, Gao F, Liu G W, Bai R C, Wang D (2017) TIN Topological Reconstruction Algorithm Based on Hash Function and Half-edge Data Structure. Application Research of Computers 34(12):3689-3692+3700.

Jan H, Martin K, Vaclav S (2003) Hash function and triangular mesh reconstruction. Computer & Geosciences (29): 741-751.

Zhao J C, Bai R C, Liu W, Liu G W (2015) An Edge-prior DEM Construction Algorithm for Open-pit Coal Mine. Journal of China Coal Society 40(8):1827-1833.

Chew L P (1989) Constrained Delaunay triangulations. Algorithmica 4(1):97-108.

Sloan S W (1993) A fast algorithm for generating constrained Delaunay triangulations. Computers & Structures 47(3):441-450.

Domiter V, Žalik B (2008) Sweep-line algorithm for constrained Delaunay triangulation. International Journal of Geographical Information Science 22(4):449-462.

## Funding

This research was supported by: (1) the Foundation of the Basic Scientific Research Projects of Liaoning Higher Education Institutions under Grant No. LJ2017FAL015; (2) the National Natural Science Foundation of China under Grant No. 51974144.

## Conflicts of interest/Competing interests

These no potential competing interests in the paper. The author confirm that the content of the manuscript has not been published for publication elsewhere.

## Availability of data and material

All data in AutoCAD format are available from the corresponding author by request.

## Code availability

All the codes of the algorithm in C#.NET are available from the corresponding author by request.

## Authors' contributions

Jingchang Zhao: Contributed to the algorithm designing, programming, manuscript writing and revision.

Fei Gao: Contributed to the data processing, program testing.

Guangwei Liu, Dong Wang: Contributed to the manuscript revision.

# Figures

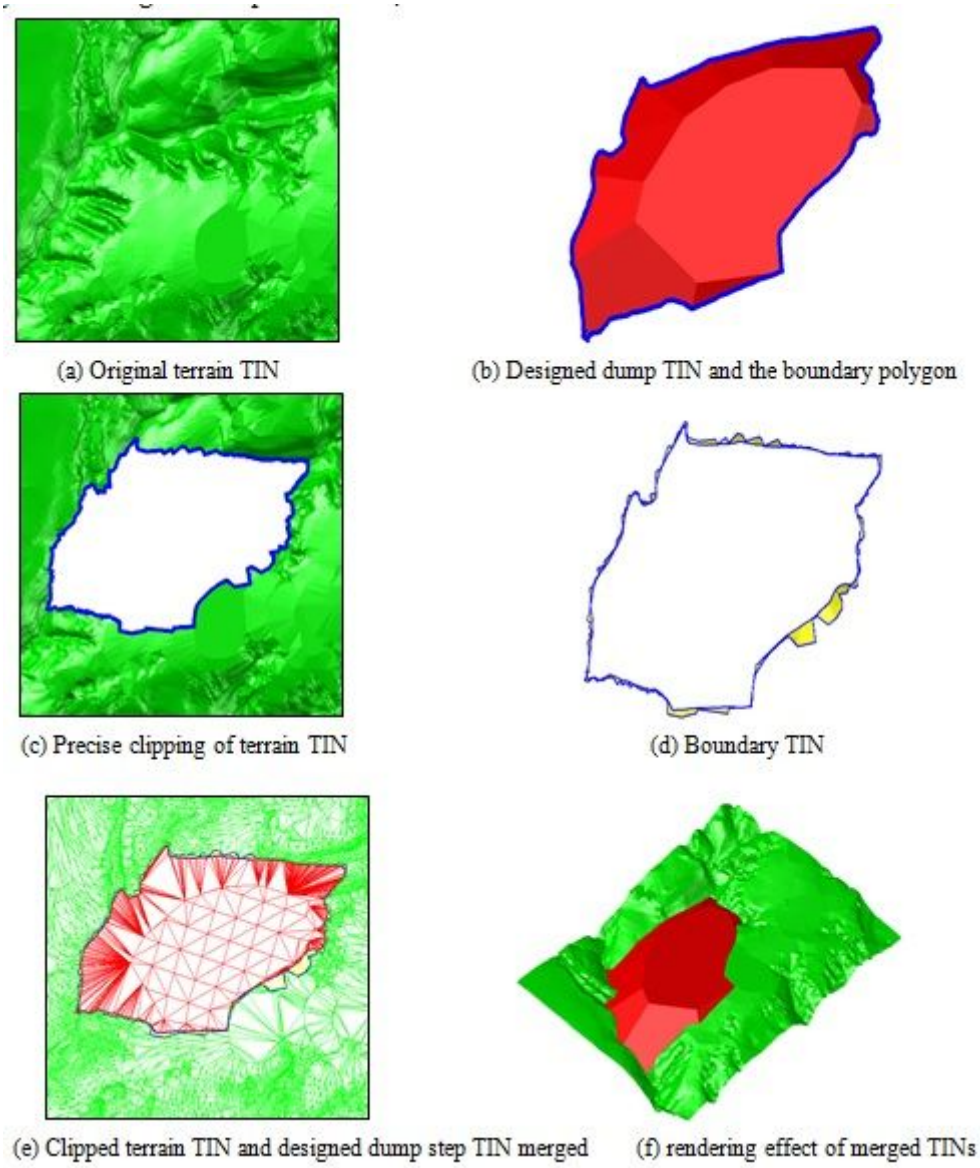



**Figure 1**

Dump site design of open-pit coal mine applying the proposed TIN clipping algorithm

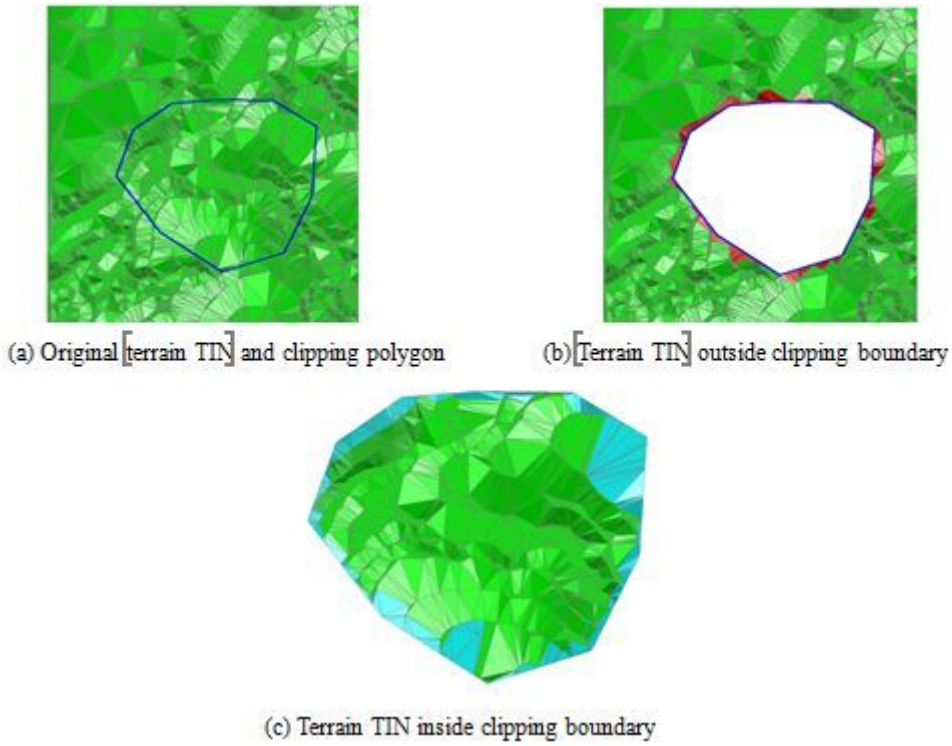



**Figure 2**

Terrain DEM clipping

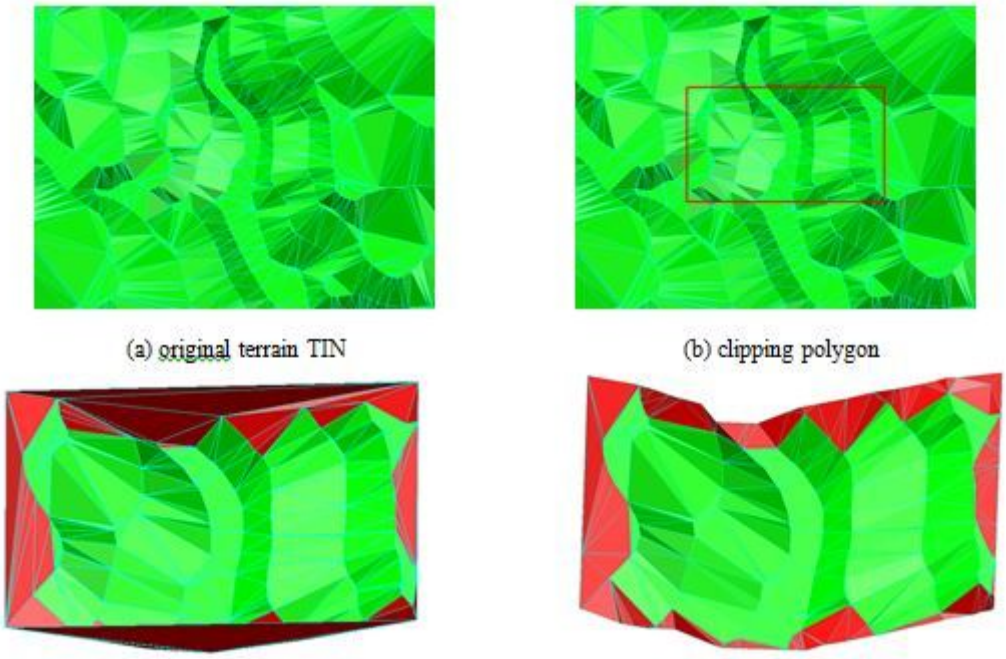
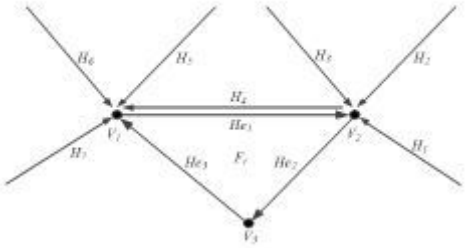


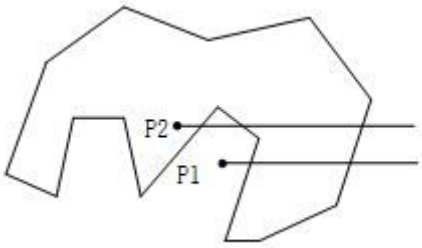

**Figure 3**

Experiment result of the proposed precise TIN clipping algorithm



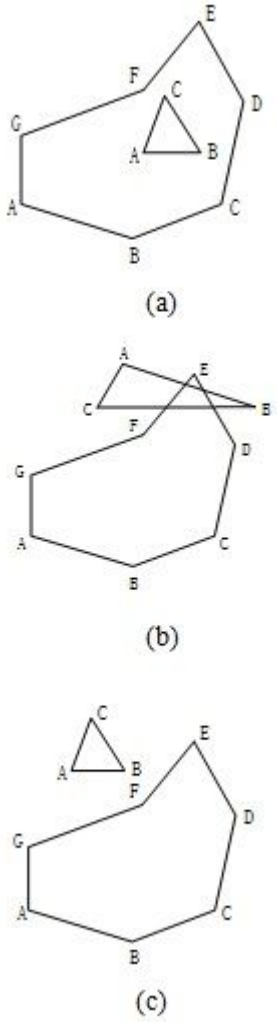
## Figure 4

Diagram of TIN’s half-edge merged



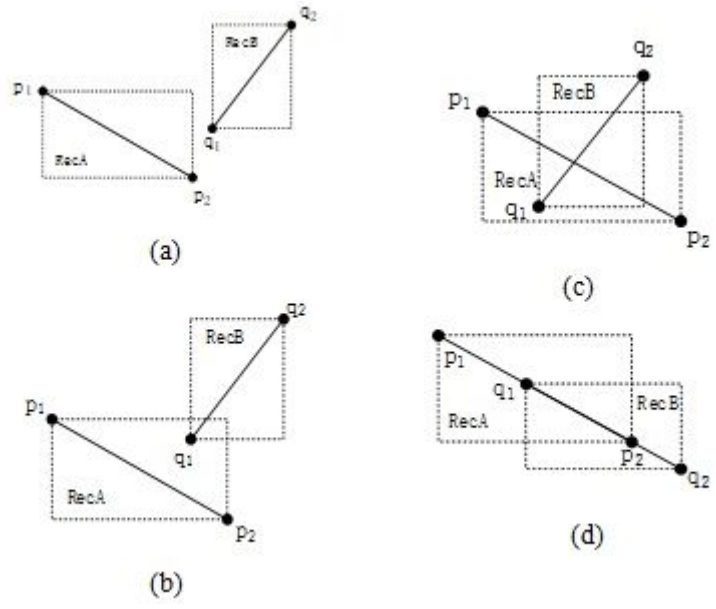

## Figure 5

Diagram of position relationship between point and polygon



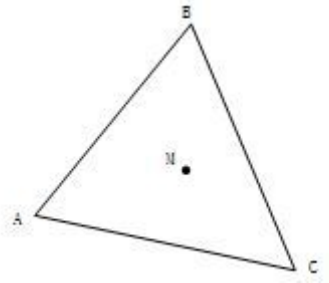

## Figure 6

Diagram of the position relationship between the triangle and clipping polygon



## Figure 7

Diagram of line segments intersection




## Figure 8

Diagram of point and triangle position relation

| CDTIN | Tri |
|---|---|
| vert_List //Vertexes set<br>tri_ List //Triangles set<br>ed_List //Edges set | tri_ID //Triangle ID<br>vert_ID[3] //Vertexes' IDs array<br>ed_ID[3] //Edges' IDs array |

| Grid | GridCell |
|---|---|
| gridCell_Arr[rowNum,ColNum] //Array of grid index cells<br>cellSize //Size of grid index cell<br>rows_Num //Row number of grid index cell array<br>cols_Num //Column number of grid index cell array | rt_Coor[2] //Coordinate of the upper-right vertex<br>lb_Coor[2] //Coordinate of the lower-left vertex<br>row_ID //Row ID<br>col_ID //Column ID<br>vert_List //Incident vertexes set<br>ed_List //Incident edges set |

| Vertex | Edge |
|---|---|
| vert_ID //Vertex ID<br>x,y,z //Coordinate of vertex<br>vert_Prop //Property of vertex: if vertex is on the boundary, vert_Prop=1; if vertex isn't on the boundary, vert_Prop= 2; if vertex is discrete vertex, vert_Prop=3<br>tri_ List //Set of triangles adjacent to a vertex<br>ang_Sum //Sum of all interior angles associated with the same vertex<br>gridCell_ID //Grid index cell ID | ed_ID //Edge ID<br>vert_ID[2] //Endpoints' IDs array<br>ed_Prop //Edge property: boudary edge, ed_Prop=1; inner constrained edge, ed_Prop=2; other edges, ed_Prop=3<br>left_triID //Left neighboring triangle ID<br>right_triID //Right neighboring triangle ID<br>grid_List //Neighboring grid index cells set<br>gridCellId_List //Grid index cells' IDs set |

## Figure 9

Data structure of the algorithm