FORMALIZATION AND APPLICATIONS OF TOPOLOGICAL RELATION OF CONTOUR LINES

Tao Wang [a]

[a] School of Resource and Environment Sciences, Wuhan University, Luoyu Road 129, Wuhan, 430072, China
mapwang@tom.com

KEY WORDS: GIS, Automation, Triangulation, Algorithms, Contour, Topological Relation, DEM/DTM

ABSTRACT:

This paper proposes a novel formalization framework of topological relation of contour lines and elaborates the applications based on it. Our idea is concentrated on the continuity of the spatial proximity and the direction among contour lines. Constraint Delaunay TIN on the contour lines is employed to acquire proximal relation of neighbouring contour lines. We define proximal relation between two contour lines c_1 and c_2 if there are TIN edges whose two end nodes are on them respectively. The proximal relations are distinguished into 0-order ($N_0(c_1, c_2)$) if the involved contour lines have same elevation, and 1-order if they have a difference of one elevation interval. To 1-order proximal relation, we consider the elevation increasing direction and further define $N_1(c_1, c_2)$ if c_1 is higher than c_2, and $N_{-1}(c_1, c_2)$ if c_2 is higher than c_1. When two contour lines have 1-order proximal relation, their vector directions are updated to follow the rule that left area of each contour line is higher than itself and the right is lower. We describe an efficient method to implement this operation. The benefits of the proposed idea are demonstrated experimentally on the time-consuming and error-prone tasks of assigning elevation value to the contour lines and automatic connection of broken parts of contour lines resulted from vectorizing the raster map.

1. INTRODUCTION

Contour is one of the most important tools to represent geomorphological information in map and GIS. However, it is long known that acquiring vector contour data by digitizer is a labour-intensive and error-prone task. The procedure of semi- or full-automatic vectorizing scanning map improves this work greatly. But due to the quality of scanning and the actual representation of topographic maps, lots of contour are broken and cannot ensure the completeness by the existing automatic approach. At the same time only the position of contour is got, and the other equally important information-elevation of contour lines-has to be input manually, which is still tedious and possible to produce error.

Previous research has been concentrated on constructing some tree structures to get spatial relation of the disjoint contour line and some of them are used to label elevation value to vectorized contour lines. Sircar (Sircar, 1991) took the contour map as a graph and converted it into an oriented tree in which the node denotes the region enclosed by adjacent contour lines, the root is the region enclosed by lowest contour lines and between two regions sharing same contour the higher one is son-node of the lower one. The depth of node is proportional to its elevation. (Guo, 1995; Hao, 2001; Roubal, 1985; Wang, 2002; Wu, 1995; Zhai, 1996) took contour lines as nodes in the contour tree and set the lowest contour lines as the root. There is an edge between two nodes if their corresponding contour lines are adjacent. There are other tree structures mentioned in reference which incorporate the contour lines and the region between them. Figure 1(b) and 1(c) shows two kinds of contour tree of contour map in 1(a). The algorithms for constructing the contour tree can be grouped into two kinds: the ones based on raster that employs image dilatation or erosion and the ones based on vector which uses polygon-in-polygon test algorithm to determine the relation between contour lines. However, both need much manual pre-processing that can be reduced further and cannot ensure the consistency when there are broken contour lines which is often appeared in topographic maps.

This paper first proposes a novel formalization framework to describe the topological relation of contour lines, which is constructed based on TIN of contour lines. Finally the benefit is demonstrated by automatically labelling elevation value to contour lines in which the broken contour lines are processed without manual work and further the broken parts are automatically connected.

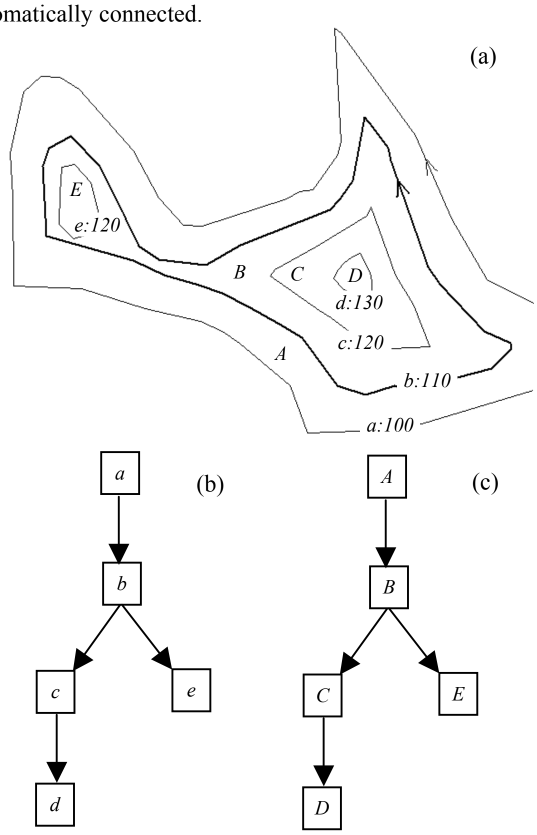


Figure 1 Contour map and corresponding contour trees.

2. CONSTRUCTION OF TOPOLOGICAL RELATION OF CONTOUR LINES

Most research used contour trees to represent the relation among contour lines. In the tree structure, node denotes contour line or the region between two neighbouring contour lines. If two nodes are connected in the tree, the corresponding contours or regions are adjacent consequently (figure 1). We can see that this kind of adjacency is continuous in the interested area and starting on one contour a sequential traversal can be done over the entire map. By this observation, the degree of automation can be improved by passing local known information to the rest part of the map considering the inherent property between neighbouring contour lines. On the other hand, the elevation of point on the contour lines is all same and the two sides are either higher or lower than the contour's elevation. Then, the local region that is monotone at the normal direction of a contour line can be determined at some degree, and further this property illustrates contour line is directional at 2-d plane, exactly, it is "left high and right low" or "left low and right high". In vector data model, the direction is denoted by the points' order constituting contour line. This paper takes "left high and right low" direction, as contour lines *a* and *b* shown in figure 1(a).

2.1 Triangulation on Contour Lines

Triangulated Irregular Network (TIN) is a very useful tool in spatial analysis of GIS and among them Delaunay Triangulation (DT) which is the dual of Voronoi Diagram has wonderful characteristics and is most frequently used to get proximal relation between geographic objects. In this paper, constrained Delaunay Triangulation (CDT) is employed to get the spatial relation between neighbouring contour lines.

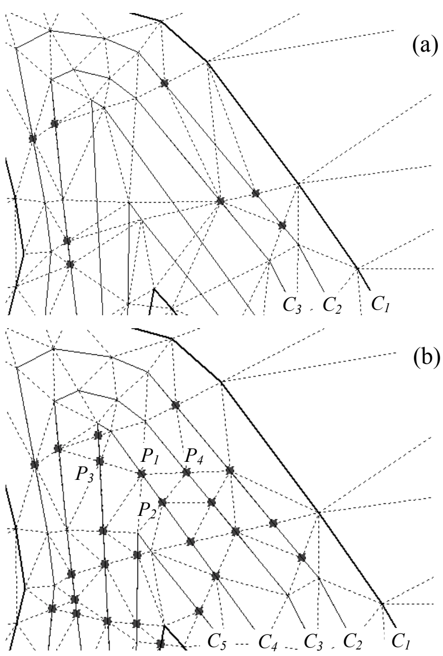


Figure 2 Iterative construction of CDT (the black nodes are inserted by the algorithm)

The algorithm for constructing CDT has two steps. First, all nodes of all contour lines are used as scattered points to construct DT. In this step an incremental approach is employed. Then, the edges of contour lines are inserted one by one, during which intersection between contour line's edge and DT edge may arose. There are various strategies to deal with this situation and this paper uses the one elaborated by Tsai (Tsai, 1993) which inserts the intersecting point into the contour lines at intersection. There is one situation that should be noted is in this research the CDT is on multiple objects, after a latter contour line's edge is inserted and the intersection is handled, the affected DT edges may intersect former inserted contour line's edges, as in figure 2(a) when inserting C_3 after C_1 and C_2 intersection arose again. In order to eliminate all possible intersections, the second step is iterated to insert new intersection points until there is not any intersection (as in figure 2(b)).

2.2 Topological Relation of Contour Lines

Contour line is one dimension curve and the topological relation between them only can be disjoint. But considering the geomorphologic semantics the relation model should be extended to incorporate the geometric information and the elevation information at the same time.

First the edges of CDT on contour lines are classified into three classes. The first type includes edges whose two end nodes on same contour line and the two nodes' corresponding nodes of the contour line are adjacent in order, as edge P_1P_2 in figure 2, we name it as object edge; the second includes edges whose two end nodes on same contour line and the two nodes' corresponding nodes of the contour line are not adjacent in order, as edge P_1P_3 in figure 2, we name it as flat edge; the third type includes edges whose two end nodes on different contour lines, as edge P_1P_4 in figure 2, we name it as linking edge.

The elevation of two contour lines c_1 and c_2 is denoted as $H(c_1)$ and $H(c_2)$. The elevation interval is denoted as $\triangle H$.

Definition 1 Proximal Contour Line. Two contour lines c_1 and c_2 are proximal if there are linking edges whose two nodes are on them respectively. Further:

If $H(c_2)-H(c_1)=0$ is true, then denote: $N_0(c_1)=c_2$, $N_0(c_2)=c_1$.

If $H(c_2)-H(c_1)=\triangle H$ is true, then denote: $N_1(c_1)=c_2$, $N_{-1}(c_2)=c_1$.

We call the two contour lines are proximal contour pair if they satisfy relation N_0 or N_1 or N_{-1}, alternatively denote:

$$N_k(c_1, c_2) = \begin{cases} 0, & c_1 \; and \; c_2 \text{ are 0 - order proximal} \\ -1 \; or \; 1, & c_1 \; and \; c_2 \text{ are 1 - order proximal} \end{cases}$$

We denote M(•) as a modal function and further define $M(N_0)=0$(0-order proximal) which corresponds contour pair with equal elevation, $M(N_1)=M(N_{-1})=1$(1-order proximal) which corresponds contour pair with $\triangle H$ difference. There exist:

1. If contour lines c_1, c_2 and c_3 satisfy: $N_1(c_2)=c_1$, $N_1(c_2)=c_3$ (or $N_{-1}(c_2)=c_1$, $N_{-1}(c_2)=c_3$, then: $N_0(c_1)=c_3$, $N_0(c_3)=c_1$.
2. If contour lines c_1, c_2 and c_3 satisfy: $N_0(c_1)=c_2$, $N_1(c_2)=c_3$ (or $N_{-1}(c_2)=c_3$), and c_1 and c_3 are on the same side of c_2, then: $N_1(c_1)=c_3$ (or $N_{-1}(c_1)=c_3$)

As in figure 1(a), $H(b)-H(a)=H(c)-H(b)=H(d)-H(c)=\triangle H$, $H(c)=H(e)$, then the following is true:

$N_1(a)=b$; $N_{-1}(b)=a$, $N_1(b)=c$, $N_1(b)=e$; $N_{-1}(c)=b$, $N_0(c)=e$, $N_1(c)=d$; $N_{-1}(d)=c$, $N_{-1}(e)=b$, $N_0(e)=c$.

To two contour lines c_1 and c_2, if $N_k(c_1)=c_2$ (k=0, 1 or –1) is not true, then denote $N_k(c_1, c_2)=0$.

Definition 2 Valid Node. Assume one of the nodes, *P*, of a linking edge is on contour c_1 and the other one node of this linking edge is on contour c_2, then we call *P* is a valid node of c_1 to c_2. As in figure 2(b), P_1, P_2 and P_3 are valid nodes of contour c_4 to contour c_5. P_1 and P_2 is a valid nodes of contour c_4 to contour c_3. But P_3 is not a valid node of contour c_4 to c_5.

Definition 3 Degree of Proximity. If the count of linking edge whose two nodes are on contour lines c_1 and c_2 respectively is N_{e12}, the count of valid nodes of c_1 to c_2 is N_{v12} and the count of valid nodes of c_2 to c_1 is N_{v21}, then the degree of proximity of c_1 to c_2 is:

$$\mathrm{D}(c_1, c_2) = N_{e12} / N_{v12}$$

And the degree of proximity of c_1 to c_2 is:

$$\mathrm{D}(c_2, c_1) = N_{e12} / N_{v21}$$

And the degree of proximity of contour pair (c_2, c_1) is:

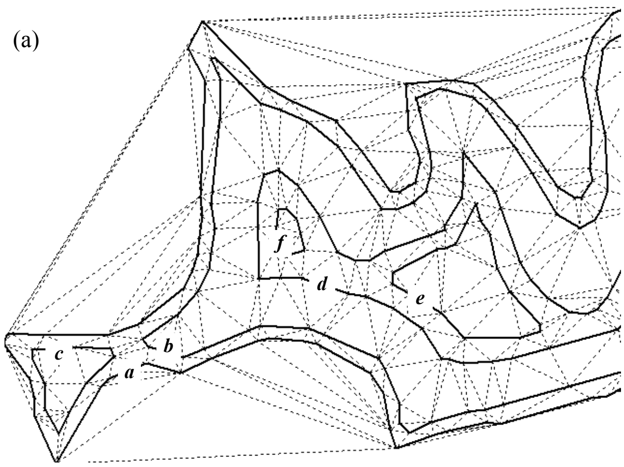
$$\mathrm{D}c_2c_1=\mathrm{Max}(\mathrm{D}(c_1, c_2), \mathrm{D}(c_2, c_1))$$

2.3 Construction of Topological Relation of Contour Lines

By construction of topological relation of contour lines, we mean the relation should be stored explicitly or can be retrieved directly. We take contour lines as the nodes of contour tree structure. After the CDT is constructed on the contour map, this relation is not difficult to be obtained by traversing the triangulation network and counting the third type edges. The node structure corresponding contour line which is used to record the topological relation in the implementation is as below. It has three array items which record the contour lines who have relation N_1, N_{-1} and N_0 with this contour line respectively. Some pairs of contour lines who are 0-order proximal may not be detected directly by traversal, e.g. the contour lines with one elevation interval difference above the saddle regions. According to previous section, this relation can be derived from existing ones. Just as in figure 3, figure 3(b) is the recorded relation table of figure 3(a), and there is not third type CDT edges between contour *e* and *f*, however we can see that it is true that $\mathrm{N}_1(d)=e$, $\mathrm{N}_1(d)=f$, *e* and *f* are on the same side of *d*, then we have $\mathrm{N}_0(e)=f$, $\mathrm{N}_0(f)=e$.

```
stuct Node
{
long *HighNeighbor;
long *LowNeighbor;
long *EquaNeighbor;
}
```

(a)



(b)

Node	HighNeighbor	LowNeighbor	EquaNeighbor
a	b c		
b	d	a	c
c		a	b
d	e f	b	
e		d	f
f		d	e

$\mathrm{H(e)-H(d)=H(d)-H(b)=H(b)-H(a)= \Delta H}$

$\mathrm{H(e)=H(f),\ H(b)=H(c)}$

Figure 3 Construction and management of topological relation of contour lines

As mentioned above, contour lines are directional. Any meaningful operation on contour should be based on the consistency of the direction of contour lines over the entire map. As a matter of fact the direction is a relative concept on this point and in order to adjust the contour lines to "left high and right low", the default direction setting in this paper, there must be a reference contour line. The reference contour line either has different elevation or has the same elevation and has been adjusted to the default direction. Regarding this property the algorithm implemented in our research has two stages which handle different situations sequentially. The detailed algorithm is demonstrated in figure 4.

Input: contour map and CDT on it. Output: contour map with the right direction.	
Ⅰ	1 Set *a* is the contour that need be adjusted. If a is null then go to Ⅱ, otherwise go to 2;
	2 Set P_{a1} and P_{a2} are two sequential nodes on *a*. If P_{a2} is null then *a* is passed waiting for process in Ⅱ, otherwise go to 3;
	3 $P_{a1}P_{a2}$ is a CDT object edge. Set P_{k1} and P_{k2} are two nodes who form triangles with $P_{a1}P_{a2}$ on either side and on other contour lines;
	4 If P_{k1} or P_{k2} is on contour line *b* and *b* satisfies $\mathrm{N}_0(a) \neq b$(assuming P_{k1}), go to 5, otherwise go back to 2;
	5 If $\mathrm{N}_1(a)=b$ and P_{k1} is on the right of $P_{a1}P_{a2}$, or $\mathrm{N}_{-1}(a)=b$ and P_{k1} is on the left of $P_{a1}P_{a2}$, go to 6, otherwise go back to 1;
	6 Reverse the order of points chain of *a*. go back to step Ⅰ-1.
Ⅱ	1 Set *a* is the contour that need be adjusted. If *a* is null then break iteration and exit, otherwise go to 2;
	2 Find two sequential nodes P_{a1} and P_{a2} on *a* where exists a third node P_{bn} on a processed contour line *b* forming triangle with edge $P_{a1}P_{a2}$. If find nothing then go back to 1;
	3 Find two sequential nodes P_{b1} and P_{b2} on *b* where exist a third node P_{am} on *a* forming triangle with edge $P_{b1}P_{b2}$. If find nothing then go back to 1;
	4 If P_{bn} is on the left (right) of $P_{a1}P_{a2}$ and P_{am} is on the left (right) of $P_{b1}P_{b2}$ then set a as processed, otherwise reverse the order of points chain of *a*. go back to Ⅱ-1.

Figure 4 The two-step process of adjusting contour lines to default direction

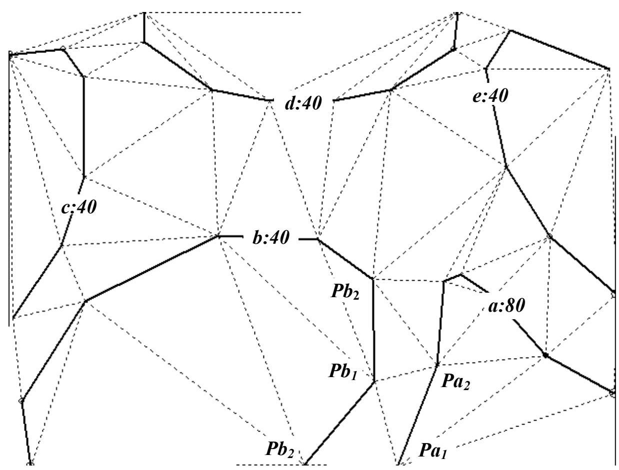


Figure 5 Adjustment of equal-elevation contour lines' direction

The second stage of the process handles the situation where a contour line's all neighbor contour lines are on the same elevation. As in figure 5 contour lines *a*, *b* and *e* are adjusted by the first stage, *c* and *d* are passed to and processed in the second stage.

3. AUTOMATIC CONTOUR LABELLING

Based on above relation model and the inherent properties of contour lines it is possible to propagate the known local information (elevation) to entire contour map. Elaboration in last section is based on the assumption that all involved contour lines have set the right elevation. It is better to extend the above definition at elevation monotone region of the map.

Lemma if $Dc_1c_2 > D_s$ and c_2 is on the left (right) of c_1, then $N_1(c_1)=c_2$ ($N_{-1}(c_1)=c_2$), and further $H(c_1)+\triangle H=H(c_2)$($H(c_1)-\triangle H=H(c_2)$). Where Dc_1c_2 is the degree of proximity of contour pair (c_1, c_2) and D_s is the threshold value.

The process of automatic labelling elevation value to contour line is as following.

1. Construct CDT on the contour map and compute the degree of proximity
2. Interactively labelling two contour lines c_1 and c_2 with the right elevation value. Saddle in the contour map produces ambiguous when propagate elevation using above rule. Thus the two setting contour lines must be lower than the lowest saddle. Adjust the c_1 and c_2 to the right direction as mentioned in section 2.3.
3. Every adjusted contour line is used as a referent contour line once. Get all contour lines that have relation of 1-order proximity with the referent contour line *a*, label the left ones and the right ones whose degree of proximity with the referent contour line is great than D_s with $H(a)+\triangle H$ and $H(a)-\triangle H$ respectively. If *a* is a closed contour line and the adjusting ones on the right side are closed too, label them with $H(a)$. Adjust the direction of them. Reiterate this step until all contour lines is labelled with elevation.

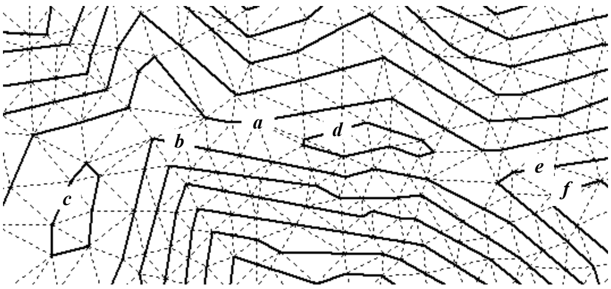


Figure 6 Ambiguous situation when automatically labelling contour lines

At some situation it may not be enough by interactively setting only two contour lines as local known information. As in figure 6, assuming contour line *a* is a reference and *b* and *c* are on its left, then *b* and *c* will be labelled as $H(a)+\triangle H$, but it is wrong if *a* and *b* are one contour line broken by the bound actually. So if we label some contour line, e.g. *f*, as the highest contour, then after the automatic process reaches *f* a reversing process can be triggered to check if there is wrong.

There is other situation like local pits that cannot get the right result when using above approach because the above approach takes that the whole map has not negative region. This problem can be dealt with the strategy as mentioned last paragraph by specifying the contour line at which elevation increment is becoming negative.

4. CONNECTING BROKEN PARTS OF CONTOUR LINES

There are broken contour lines in typical topographic maps caused by restricted space in map and the intersections with other features and annotations. The broken contour lines would become even more after the process of scanning and automatic recognition. Connecting the broken parts into one contour manually is evidently not an elegant solution. The automatic approach of this task includes two steps: identify the broken parts of one contour line first and connect them in sequential order secondly. Based on the topological relation the *n* parts $\{a_i \mid i = 1, n\}$ that are bona-fide one contour line satisfy three conditions.

1. All parts have equal elevation.
2. Every part at most has one end node at the boundary of the contour map. And of all parts, the count of end nodes at the boundary of contour map does not exceed two.
3. All parts have same topological relation (N_1, N_{-1} or N_0) with one contour line or multiple contour lines who are 0-order proximal (N_0).

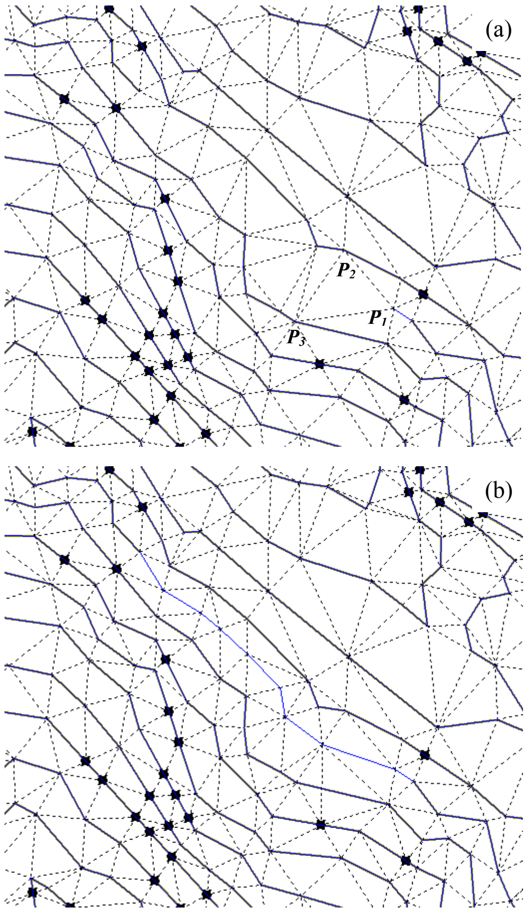


Figure 7 The connection of broken contour lines

The second step is the connection of broken parts of one contour line, which is performed in CDT. There may be various situations of the set of the broken parts. Figure 7 illustrates one of them and the connection process is designed as below.

1. To one part in the set, assume one of its nodes, P_1, which is not at map boundary. The distances of P_1 to the two neighbor contour lines are d_1 and d_2. Compute the ratio r of d_1 to d_2.
2. Get a triangle, if there is, one of whose vertices are P_1 and the other two are from the involved two neighbor contour lines. Interpolate a point on the facing edge of P_1 by the ratio got above. Insert the interpolated point in the broken part and CDT and let it be a new start. If there is not a triangle with that property then the interpolation reaches the map boundary or another broken part which joins current part to one.
3. Repeat above until the current set is connected into one contour line.
4. Repeat above until there is not broken part.

In figure 7(a) there are two broken parts of one contour line and interpolation is started at P_1. Triangle $P_1P_2P_3$ that is formed by three linking edges and involved three objects is used to interpolate a new point on edge P_2P_3. The thin solid line is interpolated to connect the two broken parts in figure 7(b).

5. CONCLUSION AND DISCUSSION

This paper designs a novel formalization framework of topological relation of contour lines based on the continuity of proximal relation between contour lines and the direction of contour lines. It is more practical because it only need to consider the adjacent contour lines. The algorithm implemented to construct the relation is more robust and efficient than raster algorithm and previous vector ones for it integrates the situation that there are often broken contour lines in topographic maps.

We believe there are other interesting applications of this formalization framework than automatic contour labelling and the connection of broken contour lines in CDT as derivation of terrain features and terrain classification. Further research can consider the geometric characteristics of contour lines for these 1-d curves implicate some characteristics of the 3-d topography and it may be of benefit to find the situation of local pits in the automatic contour labelling process and design more flexible algorithm for connecting the broken parts of contour lines at more complex situations.

ACKNOWLEDGEMENT

The suggestions and encouragements of Professor Hehai Wu and Professor Tinghua Ai at Wuhan University are gratefully acknowledged.

References

Ai, T.H., 2000. A constrained Delaunay partitioning of areal objects to support map generalization. Journal of WTUSM. 25(1), pp. 35-41. (In Chinese)

Guo, Q.S., 1995. Intelligent approach for constructing hierarchical structure of contour. Journal of WTUSM, 20(Supplement), pp. 60-75. (In Chinese)

Hao, X.Y., 2000. An automated approach to acquiring the heights of contour based on topological relation for GIS. In: Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, China, pp. 4b.3-8

Jones, C.B., etc, 1995. Map generalization with a Triangulated Data Structure. Cartography and Geographic Information System. 22(4), pp. 317-331.

Roubal, J., etc, 1985. Automated contour labelling and contour tree. In: AutoCarto 7, pp. 499-509.

Sircar, J.K, 1991. An automated approach for labelling raster digitised contour maps. Photogrammetric Engineering and Remote Sensing, 57(7), pp. 965-971.

Tsai, J.D., 1993. Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. Int. J. Geographical Information Systems. 7(6), pp. 501-524.

Wang, Y.M., 2002. An algorithm for automated labeling and checking vector-based contour maps. Chinese Journal of Computers. 25(9), pp. 976-981. (In Chinese)

Wu, H.H., 1991. Map database system. Surveying and Mapping Publishing House, Beijing. (In Chinese)

Wu, H.H., 1995. Construction of contour tree. Journal of WTUSM, 20(Supplement), pp. 15-19. (In Chinese)

Zhai, J.S., etc, 1996. Automatic recognition of depth contour's value, ACTA GEODAETICA et CARTOGRAPHICA SINICA, 25(4), pp.272-276. (In Chinese)