Layering-based Breakpoint Handling in Contour Line Extraction

CHEN Dan  LONG Yi  CAI Jinhua

1 Introduction

Automatic recognition of geographic maps is one of the important research projects in GIS. It may resolve many fundamental problems, such as acquisition of data, database establishing, and automatic understanding of geographic spatial data, through integrating artificial intelligence with digitizing image processing technique. As the crucial element reflecting topographical structures, contour lines have always been the key unit in such researches, due to their vast data amount and complicated configurations. Presently, how to process contour line’s interruption and conglutination and how to capture its elevations quickly and credibly are still not thoroughly settled. This paper primarily deals with the first problem, that is, how to exactly link broken contours.

Although the brown plate map, one product of the whole product flow, is perhaps more difficult to get than the topographic map with full elements, it has more obvious advantages and less break points than a brown map extracted from a scanned map through color separation. Thus it should be regarded as the main data source of this recognition project. The primary reason for contour lines’ rupture on this brown plate is the existence of other brown elements. In order to get continuous contours, these broken lines must be linked. However, in previous handling methods the diversity of those interpretations caused by the complex topographical configurations has not been considered well, and correspondingly this problem has not been well solved. Therefore, this paper holds it necessary to take other elements as referent layers.

Project supported by the Scientific Research Foundation of Wuhan University (No. 40071071).

CHEN Dan, postgraduate, School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China.
E-mail: cd_chendan@sina.com
during the course of connection and fully use the information they provide. Several most familiar configuration properties of contour lines on the brown plate are discussed, when they encounter other thematic symbols (dry land, gully, cliff and elevation annotation etc.). And a layering-based method is proposed, aiming at adopting different operators to different factors.

2 Previous breakpoints handling methods

Although many developments in this field have been made in recent years, it should also be clearly seen that most experimental data are obtained by use of continuous contour lines. And even the handling arithmetic is effective only when the breakpoints are caused by the color separation or other simple factors, while less consideration is given to how to match and connect breakpoints caused by the contours’ encountering topographical symbols.

2.1 Connection method based on Freeman’s code

This method may deduce the code of the prolongation part of a line from the codes of its neighboring points. Prolong the broken contours, if the intersection point of prolongations of two lines are found, then the connection is accomplished. Obviously, this method is only effective to short and straight lines, and invalid to devious ones.

2.2 Connection method based on morphological transformation

This method is mainly a research on mathematical morphological method and theory, so the objects it can recognize are relatively simple. To those complicated ruptures, it needs to take lots of experiments in order to seek and construct suitable structural elements or arithmetic. Meanwhile, this method is based on mathematical morphological dilation and shrink operations, do not consider the mutual constraint relationships among these broken contour lines.

2.3 Connection method through maximal clique graph search based on relational structural constraints

This method changes the task of breakpoint connection to seek the best and maximal clique. It gets the final result with the help of physical constraints. It has certain automatic error-detection competence, but need a great deal of time. Consequently, it needs more efforts to study how to decompose breakpoint candidates, introduce proper physical restrictions, dwindle the clique graph, then realize more efficacious global broken contour connection. Furthermore, it still needs manual post-treatment to amend errors.

2.4 Connection method based on topology

This method collects vertexes and breakpoints of every contour from vector data to construct Delaunay TIN. It holds that any point in the triangulation has certain influence areas. By correctly choosing the size of grid, any breakpoint’s corresponding point must be contained in its area. Then examining all the points in it, we can judge and find out its accurate corresponding point, and realize their connection by linking them. This method based on vector data needs none of line scanning, so it is faster than that based on raster. But it also needs to study how to find appropriate size of grid in different circumstances and how to perfect the principle of connection.

3 Presentation of layering-based breakpoint handling

Although many scholars are engaged in designing one universal method to connect broken contours, such an approach are usually complicated and cannot satisfy the intention of linking, because the reasons for any rupture should be taken as the foundation of accurate connection. We should take different
matching operators for different reasons.

Since many factors on the brown plate map will cause the break of contours, in this paper only several most familiar ones are discussed, such as elevation annotation, gully, dry land, and the circumstance under which contours are too dense to be displayed continuously, etc. The former three elements should be extracted from basic map, and regarded as the reference layers. Among them, because the terrain elevation is one of the important parts of geographical information, and the crucial attribute of contour lines, many developments about automatic recognition of elevation annotation have been made. On the other hand, gully and dry land are still the difficult problems of recognition due to their complex structures. However, it is very significant to extract them, respectively, from map. For one thing, they are topographical symbols and belong to the field of recognition and extraction of map elements; for another, their existence surely brings some difficulties to this research, but it also provides certain information to entirely and exactly extract contour lines.

On the big scale maps, contour line is usually broken along the sides of a dry land symbol. Under this circumstance, inclination is almost close to a right angle between the line linking two corresponding breakpoints and the short line of dry land symbol there (Fig. 1). This rule can help us construct suitable linking operator. Moreover, under another condition, certain fraction of a contour cannot be displayed when this part overlaps with cliff symbol. But it can be said that the trend of this cliff symbol simultaneously reflects the trend of that broken line (Fig. 2). If we want to exactly get the information about contours, we must base our algorithm on the trend of these topographical symbols. Specific algorithm will be described below. Here we do not consider single-line gullies, because they cannot make the contour break and contours can be extracted accurately by using proper algorithms.

4 Data preparation and thematic information extraction

The input map is scanned in grey level 500dpi and binarized, from which we choose a more complex region as the experimental data (Fig. 3). It contains the thick lines (i.e. the master lines), the ordinary lines and several thematic elements needed. Because the method for extracting elevation annotation is relatively maturer, here we assume that this information has been extracted already and stored in S3.bmp. We will mainly deal with the process of other elements.

Firstly, since the thickness of thick line is far bigger than that of ordinary line, and its length is commonly longer than thematic symbols, we can separate it from others by the process including a series of erosion transformations, which will interrupt symbols other than those thick ones, thinning operation, and eliminating short lines. Meanwhile, the question of conglutination between different elements (for instance, the conglutination between tubers of cliff and contour, and that be-
tween short lines of dry land and contours) is solved by calculating the number of none zero pixels around each pixel on the thinned image. If the number is greater than three, then this seed pixel must be an intersect point among different elements. We can interrupt their conglutination by setting the value of this pixel as 0. Then we record the data containing thinned thick lines in S1.bmp, which is also a binary image.

Secondly, after we subtract thick lines from the base image, we have to take several same steps as thinning, interrupting conglutination and eliminating short lines less than a threshold to extract the ordinary lines and store them in S2.bmp, because they are longer than other symbols. Although sometimes the cliff symbol is as long as or a little longer than some ordinary lines, it may be well solved by the interrupting conglutination process which turn them into lots of very short lines.

Thirdly, on the remnant base image, we may easily distinguish dry land and cliffs according to their different length. So we can directly extract them relatively by thinning and eliminating shorter lines, and then record them relatively in S4 and S5.

Table:

| Breakpoint ID | Line ID containing this breakpoint | Row-column number of this breakpoint | Row-column number of its first neighboring point | Row-column number of its second neighboring point |
|---------------|-----------------------------------|-------------------------------------|-----------------------------------------------|-----------------------------------------------|

In this way, we can directly and quickly obtain breakpoint information from these files during the course of breakpoint handling, without search the whole vector data. The reason why we record two neighboring points is that we need to avoid some distortions at the end of several lines derived from unsatisfactory thinning effect. If it happens, we then abandon that breakpoint and regard the trend of two neighboring points as the true trend of that part of line.

5.2 Connecting breakpoints of thick lines

There are mainly two factors causing the rupture of thick lines, one is the elevation annotation and the others are some topographic symbols. Accordingly, we take the following algorithm which should be effective and direct.

1. Call over one breakpoint A from ‘BreakPoint1.txt’, judge the distance to its first neighboring point, and construct a sector domain S with angle $\theta$ and length $L$. If its distance is greater than certain threshold, the orientation of $S$ is defined according to the trend of the connecting line between $A$ and its first neighboring point, otherwise, according to that of the connecting line between two neighboring points of $A$. In order to simplify this process and avoid searching point in whole image, we regard polygon ACED as the approximate area of $S$ (Fig. 4), and it will not affect the accurate extrac-
tion. Then, if any point $P$ satisfies the following conditions, it must be in $S$ and store its row and column number in $SArray[n][2]$. We only show the circumstance when $x_A \neq x_B \neq x_P$, and the other circumstance is easy to be displayed.

![Fig. 4 Judge the relationship between a point and a sector](image)

$$\begin{align*}
x_P &\leq \max[x_A, x_C, x_D] \quad \text{and} \quad x_P \geq \min[x_A, x_C, x_D] \\
y_P &\leq \max[y_A, y_C, y_D] \quad \text{and} \quad y_P \geq \min[y_A, y_C, y_D] \\
\theta &\leq \theta/2 \\
L &\leq L
\end{align*}$$

where

$$\begin{align*}
a &= \left| \arctan(K_{BA} - K_{AP})/(1 + K_{BA}K_{AP}) \right| \\
K_{BA} &= (y_B - y_A)/(x_B - x_A) \\
K_{AP} &= (y_A - y_P)/(x_A - x_P)
\end{align*}$$

And we store the row and column number of the points lying on the line $AF$ in $LArray[m][2]$.

② Judge whether there is any pixel on annotation image $S_3$ satisfying the following condition:

$$S_3(LArray[i][0], LArray[i][1]) = 1$$

If there is one, stop judging and it illustrates that this rupture must be caused by elevation annotation. Considering that there are no different annotations which will be placed together in a region where two thick lines are very close, and also there are no annotation that will be placed in a region where other topographic symbols exist, in a word, under this circumstance, there will be no breakpoints of other lines in $S$, we may search the corresponding point $P$ of $A$ in ‘BreakPoint1.txt’, by judging whether there is any breakpoint in this file belonging to $S$, that is,

$$P = SArray[i][0], \quad P_y = SArray[i][1]$$

and repeatedly expanding this region until we find it.

Usually, if the value of $L$ is a little more than the average length of the elevation annotations, we can make it at once. And then connect the two points, uniform the ID of the two lines to which the two breakpoints belong, and subtract 1 from the number of the lines.

Meanwhile, set these two breakpoints’ ID as 0 in ‘BreakPoint1.txt’ in order to identify them from others. Then call over another point from ‘BreakPoint1.txt’, go on to find its corresponding point from first step. On the contrary, if there are no pixels on $S_3$ satisfying that condition, we have to do the next.

③ And then, continue to judge whether there is any pixel on dry land image $S_4$ satisfying:

$$S_4(LArray[i][0], LArray[i][1]) = 1$$

If there is one, we construct another sector domain $S'$ whose orientation is different from the previous one and its angle is bigger. We build it along the trend that is vertical to the short line of dry land there (Fig. 5), and extract a set of points of the other side of this symbol lying in this region. Subsequently, judge whether there are any point in ‘BreakPoint1.txt’ lying in this set. If there is one, it must be the corresponding point of $A$. Otherwise, go on to the next steps.

④ Judge whether there is pixel on cliff image $S_5$ satisfying

$$S_5(LArray[i][0], LArray[i][1]) = 1$$

If there is one, it shows that this rupture must be caused by cliff symbols. Prolong the connecting line of $A$ with its neighboring point $B$, find the intersect point $P$ of this prolongation and cliff symbol in sector domain $S$, and take it as the next point of $A$, then confirm the second point along the cliff symbol in the trend of prolongation. Namely, we extract two points $Q$ and $Q'$ from cliff along two sides of $P$ on the line, and $|PQ| = |PQ'|$. We decide which one is the next point of $P$ by relationship between $|AQ|$ and $|AQ'|$. This relationship provides an orientation, and we should trace and extract points along the cliff symbol in this orientation. If $|AQ| > |AQ'|$, then $Q$ is the right point. Otherwise, $Q'$ is what we want.
In Fig. 6, we should trace points along the trend of PQ'. And whenever we extract a new point, we still need to simultaneously judge whether there is any point at both sides of this symbol, which lies in BreakPoint1.txt. If there is one, it must be the corresponding point of A, then stop tracing. Uniform the ID of the two lines to which the two breakpoints belong, add those extracted points to them, and subtract 1 from the number of the lines. Set these two breakpoints' ID as 0 in 'BreakPoint1.txt'. Then call over another point from 'BreakPoint1.txt', go on to find its corresponding point from the first step. However, there is an exception to which such an algorithm does not fit (Fig. 7). Under this condition, we may not find the other breakpoint even we get to the end point of this cliff symbol. At this time, we should adopt the similar method when contour line encounters the dry land symbol.

Fig. 5 Construction of another sector
Fig. 6 Illustration of contour orientation
Fig. 7 One kind of special cliff orientation

⑤ Exception handling. For some circumstances except for the above, it is surely due to few ruptures deriving from several too thinner ordinary lines. Because the distance is so short between a pair of breakpoints in this circumstance, generally less than five pixels, we can directly find the other point in 'BreakPoint1.txt' that lies in S.

The whole flow may be seen in Fig. 8.
5.3 Connecting breakpoints of ordinary lines

The flow of this step is approximate to the above five steps. But there is another factor that will also cause the interruptions, that is, the density is too high to display contours completely. We should add a new handling operator before step ③.

If the rupture is caused by elevation annotation, the algorithm mentioned above is unsuitable. Because ordinary lines are far denser than thick lines, it will bring forth error if we search the other breakpoint directly by expanding S. Here, we should refer to the external information, that is, there must be a thick line nearby this broken ordinary one. Hence, we first dilate the region of the annotation that cause the break, and extract its boundary of outside polygon, then search whether there is any points in 'BreakPoint2.txt' belonging to this range. If there is a set of points BP satisfying this condition, it must be all the breakpoints of several ordinary lines round this annotation. In Fig. 9,

\[ BP = \{ P_i, i \in [1, 8] \} \]

Fig. 9 Illustration of the relationship between the breakpoints of ordinary lines around an annotation

Then continue to search whether there is any point in 'BreakPoint1.txt' belonging to this area as well. Usually there must be two points A and B satisfying this request, which are breakpoints of the thick line broken by this annotation and in the middle of these ordinary lines. Subsequently, as long as we build the relationship between breakpoints of thick lines and their neighboring ordinary ones, we can fluently finish the matching task.

Arbitrarily choose one point \( P_m \) from \( BP \), and calculate its distance \( Dis_{P_mP_j} \) to all the other points, where

\[ Dis_{P_mP_j} = \sqrt{(X_{P_m} - X_{P_j})^2 + (Y_{P_m} - Y_{P_j})^2}, \quad (1 \leq j \leq 8, j \neq m) \]

Here, \((X_{P_m}, Y_{P_m})\) is the coordinate of point \( P_m \), and \((X_{P_j}, Y_{P_j})\) is the coordinate of point \( P_j \). Then dispart \( BP \) according to the relationship between \( Dis_{P_mP_j} \) and \( \frac{1}{2} | AB | \). If \( Dis_{P_mP_j} \leq \frac{1}{2} | AB |, j \) must lie on the same side of the annotation as \( P_m \). Otherwise, the point must be on the other side. In Fig. 9, \( P_1, P_3, P_5 \) and \( P_7 \) are in the same group, and others in the other group. Then respectively set these points in both groups in order according to their value of \( X \) or \( Y \), and match them one by one according to their order. Thus the linking task is finished basically.

In addition, we have to consider the interruption caused by too high density, which also need to construct and refer to the relationship like the above to complete right matching. The special steps are very similar to the above. The difference is that it has to refer to two thick lines around the ordinary lines.

Connecting effects of each kind of lines are shown in Fig. 10 and Fig. 11. To show them clearly, we make each line wider than it originally is.

![Fig. 10 Thick lines](image)

![Fig. 11 Ordinary lines](image)

6 Conclusions

This method regards different elements as different referent layers, and adopts different arithmetic operators for different factors causing rupture. It may avoid blindness and disadvantage of conventional methods.

(Continued on Page 72)