THE ACQUISITION OF MAP INFORMATION FROM SCANNING MAP IMAGES FOR GIS

Dr. Xiangyang Hao
Zhengzhou Institute of Surveying and Mapping
66 Middle Longhai Road
PRC—450052 Zhengzhou, Henan
Peoples Republic of China

KEY WORDS: GIS, Recognition, Extraction, Algorithms, Graphics, Scanner.

ABSTRACT

The acquisition of geographic information is an indespensable process in the creation of GIS. The existed topographical maps are the most important source of geographic information. The automatic recognition and extration of map information from scanning map images are the key steps and have the great significance in the creation and application of GIS. In this paper, the methodology and algorithms of recognizing and extracting hatched polygons which stand for residential areas on maps are put forword and discussed. Run length smearing method in four directions is described and realized for the segmentation of hatched polygons. Other processes such as shrinking, expansion, thinning, open graphics deletion and distinguishing the hatched polygons from others also be discussed in detail. Experiments show that the correct recognition ratio reaches about 96%.

1. INTRODUCTION

It is indespensable to acquire geographic information in the creation of GIS. The existed maps are the most important one among various geographic information sources. Map digitizing is necessary in the process of inputting all sorts of map information into computers. The traditional way of map digitizing is sampling the feature points of the map symbols one by one on the board of digitizer manully. In order to keep the accuracy of digitized points, it is required to aim at each point as strictly as possible with the mouse. Because of the enormous amount of information on topographic maps, it is strenous and inefficient with this way of map digitizing. The appearance of scanner offers a novel way for the acquisition of map information. The digital map images, which can be obtained by scanning maps, make it possible to acquire map information intelligently or automatically.

The strategy of automatic acquisition of map information is as follows: a) Obtaining the binary map images by directly scanning maps or transforming the gray map images. b) Recognizing and extracting map symbols one sort by one sort. c) Deleting the recognized map symbols from the copy of original binary map images for the recognition and extraction of the succeeding sorts of map symbols. d) Overlapped displaying the recognition results with the original map images and interactive editting. Because of the limit of pages, only the procedures and algorithms of automatic recognition and extraction of the residential areas are discussed in this paper. Residential areas with heavier populations are usually shown as hatched polygons on the topographical maps. To automatically extract this kind of graphics directly form the complicated topographical maps, sophisticated image processing technology must be applied. The algorithms and procedures are described in the following sections.

2. RUN LENGTH SMEARING ALGORITHMS

One of the important steps of recognizing residential sections is to segment the hatched polygons into blocks and then distinguish them from other sorts of map graphics. The run—length smearing (RLS) method is an effective way for this purpose. There are only two kinds of pixels, black pixels and white pixels, on binary map imges. By assuming a white pixel represented by 0 and black pixel by 1, we define the run length smearing tranforms in horizontal, vertical and two diagonal directions as follows.

Defination 1 Suppose $A(i_1,j)$ and $B(i_2,j)$ are two black pxels in row j, if $|i_1-i_2|\leqslant n$, then turn an arbitrary pixel $C(i,j)$ between A and B (i. e. $i_1<i$

$<i_2$) into a black pixel. This kind of transform is called horizontal RLS transform and signed as $H_n(i,j)$.

Defination 2 Suppose $A(i,j_1)$ and $B(i,j_2)$ are two black pixels in column i, if $|j_1-j_2|\leqslant n$, then turn an arbitray pixel $C(i,j)$ between A and B (i. e. $j_1<j<j_2$) into a black pixel. This kind of transform is called vertical RLS transform and signed as $V_n(i,j)$.

Defination 3 Suppose $A(i_1,j_1)$ and $B(i_2,j_2)$ ($i_2-i_1=j_1-j_2$) are two black pixels in the diagonal direction of down-left to up-right, if $|i_1-i_2|\leqslant n$, then turn an arbitrary pixel $C(i,j)$ between A and B (i. e. $i_1<i<i_2$) into a black pixel. This transform is called down-left to up-right diagonal RLS transform and signed as $D_n(i,j)$.

Defination 4 Suppose $A(i_1,j_1)$ and $B(i_2,j_2)$ ($i_2-i_1=j_2-j_1$) are two black pixels in the diagonal direction of up-left to down-right, if $|i_1-i_2|\leqslant n$, then turn an arbitrary pixel $C(i,j)$ between A and B into a black pixel. This kind of transform is called up-left to down-right diagonal RLS transform and signed as $C_n(i,j)$.

For the original binary map images $I(i,j)$ shown in Figure 1, the corresponding results of its $H_n(i,j)$, $V_n(i,j)$, $D_n(i,j)$ and $C_n(i,j)$ transforms (e. g. $n=10$) are respectively shown in Figure2, Figure 3, Figure 4 and Figure 5. Then a new image $P(i,j)$ is obtained from

$$P(i,j) = (H_n \cap V_n \cap D_n) \cup (H_n \cap V_n \cap C_n) \cup (H_n \cap D_n \cap C_n) \cup (V_n \cap D_n \cap C_n)(i,j)$$

An example of $P(i,j)$ is shown in Figure 6, which is correspondant to $I(i,j)$ shown in Figure 1.

In image $P(i,j)$, all of the hatched polygons are segmented into black bolcks. However, some Chinese characters which is much similar to hatched polygons in shape are segmented into blocks too. Other parts of the image remain the same as in the original map image.

3. SERIES SHRINKING TRANSFORMS

As shown in Figure 6, image $P(i,j)$ includes black blocks as well as lines and other graphics. By performing series shrinking transforms, the graphics except black blocks can be deleted from image $P(i,j)$. Shrinking tramsfrom is defined as follows: For a black pixel, if more than two among its 8 neigh-

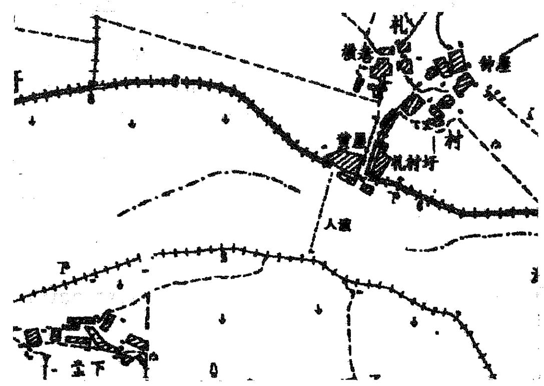

Fig. 1 The original map image $I(i,j)$

Fig. 2 The result of $H_n(i,j)$ transform for $I(i,j)$

Fig. 3 The result of $V_n(i,j)$ transform for $I(i,j)$

Fig. 4 The result of $D_n(i,j)$ transform for $I(i,j)$

Fig. 5 The result of $C_n(i,j)$ transform for $I(i,j)$

Fig. 6 The result of $P(i,j)$ for $I(i,j)$

bor pixels are white pixels, then trun it into a white pixel. We signed a shrinking transform as $S(i,j)$. Generally speaking, the lines in map images are more than one pixel in width because of the scanning resolution. Therefore, $P(i,j)$ should be performed n times of shrinking transforms $S_n(i,j)$ in order to delete other graphics. The result of $S_n(i,j)$ $(n=3)$ for $P(i,j)$ is shown in Figure 7.

Fig. 7 The result of $S_n(i,j)(n=3)$ for $P(i,j)$

4. SERIES EXPANSION TRANSFORMS

The image $S_n(i,j)$ consists of only the shrinked black blocks. To restore the original shapes and sizes of these black bolcks, series expansion transforms should be performed to image $S_n(i,j)$. Expansion transform is defined as follows: For a black pixel, let all of its neighbour white pixels be black pixels. Expansion transform is signed as $E(i,j)$. For the same reason as series shrinking transforms, $S_n(i,j)$ should be performed n times expansion transforms $E_n(i,j)$ in order to restore the original shapes and sizes of the black blocks as in image $P(i,j)$. The result of $E_n(i,j)$ $(n=3)$ for image $S_n(i,j)$ is shown in Figure 8. The black blocks in image $E_n(i,j)$ represent some other graphics such as Chinese characters, small closed graphics and lines near the residential sections besides the hatched polygons. A new image $Q(i,j)$ can be obtained from

$$Q(i,j) = (S_n \cap I)(i,j)$$

where $I(i,j)$ refers to the original image. The result of $Q(i,j)$ is shown in Figure 9.

Fig. 8 The result of $E_n(i,j)(n=3)$ for $S_n(i,j)$

Fig. 9 The result of $Q(i,j)$

5. THINNING AND OPEN GRAPHICS DELETION

Although there are fewer graphics in image $Q(i,j)$

than in original image I(i,j), it is still difficult to automatically recognize and extract residential sections directly form Q(i,j). The graphics in Q(i,j) are thick ones which are not easily manipulated. In this case, thinning transform should be performed to reduce the graphics to minimal sets of pixels representing invariants of the graphics geometrical shapes. There are many algorithms for binary image thinning process. The algorithms used in our system can be briefly described as follows. Every time the pixel meeting the following conditions are deleted (i. e. turning their image values to 1s) from image Q(i,j).

a) $I(i,j)=1$;

b) $I(i,j-1)=0 \parallel I(i,j+1)=0 \parallel I(i-1,j)=0 \parallel I(i+1,j)=0$;

c) $C(i,j)=1$.

where I(i,j) refers to the image value of pixel (i,j) and C(i,j) to the 0−1 modes of its neighbour pixels in clockwise direction of pixel (i,j). After performing a few times of this process, all of the graphics in Q(i,j) is thinned. Suppose n represents the times of thinning process, we sign the thinning transforms as $T_n(i,j)$. The result of $T_n(i,j)$ is shown in Figure 10.

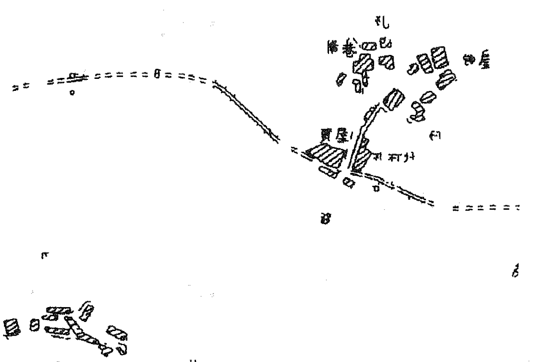

Fig. 10 The result of $T_n(i,j)(n=3)$ for Q(i,j)

We know that the residential areas are closed polygons on maps. But there are many open graphics in image $T_n(i,j)$ and they should be deleted. In order to delete the open graphics, the pixels which meet the following conditions are need to be changed into white pixels.

a) $I(i,j)=0$;

b) $C(i,j)=1$.

Thus a new image R(i,j) is obtained by deleting open graphics from Tn(i,j). The result of R(i,j) is shown in Figure 11.

6. TRACING POLYGONS

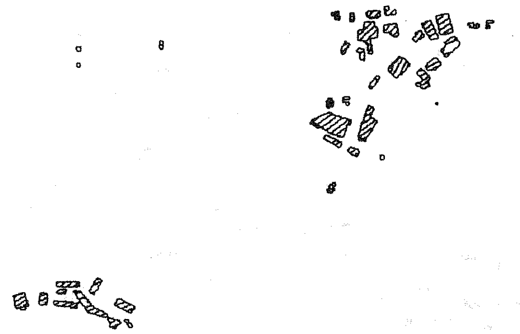

Fig. 11 The result of R(i,j)

As shown in Figure 11, image R(i,j) consists of closed polygons exclusively. Since all of the resindential sections are included in these polygons, it is a key step to trace these polygons automatiocally. The procedures of polygon tracing is briefly described as follows.

a) Searching a black pixel row by row and column by column as the first border pixel of a polygon.

b) Successively tracing other border pixels of the polygon anticlockwisely until coming back to the first border pixel.

Thus we can obtain all of the border pixels of the traced polygon and their image coordinates are supposed to be $(x_i,y_i)(i=1,\cdots,n)$. After completing the tracing process of a polygon, the pixels on its border and in its interior area should be deleted from image R(i,j) to avoid retracing this polygon. The border pixels can be easily deleted by changing them to white pixels. However, the deletion of its interior black pixels needs a much complicate process, which mainly includes following steps.

a) Determining the minimum row y_{min} and the maximum row y_{max} of the traced polygon.

$$y_{min} = min(y_i; i \in [1,n]);$$
$$y_{max} = max(y_i; i \in [1,n]).$$

b) For an arbitrary row y ($y_{min} \leqslant y \leqslant y_{max}$), searching the border pixels with vertical image coordinate y from the polygon border pixes set $(x_i,y_i)(i=\in[1,n])$ and thus obtaining a pixel set (x_i,y), $(i\in[1,k])$.

c) Reordering the pixel set $(x_i,y), i\in[1,k]$, according to the horizontal image coordinate. If $i<j$, $(i,j\in[1,k])$, then $x_i<x_j$.

d) Determining the start pixel and the end pixel of a horizontal segment in the polygon. There may be several horizontal segments in a row.

e) Turning the pixels on the segments into white ones.

In this way, the pixels in the polygon can be deleted from image R (i, j). It is possible that a few polygons link to one another by single lines. These linked polygons are traced as a complex polygon and it is necessary to seperate them into independant polygons for the automatic recognition of hatched polygons. The approach to seperating the linked polygons can be related as following two steps.

a)Removing the pixels on single line linking polygons. The pixels on single line linking polygons are traced twice in polygon tracing process. If there are two pixels, say (x_i, y_i) and (x_j, y_j), among the border pixel set of a polygon with the same position, then these pairs of pixels are removed from the border pixel set. This condition can be expressed by following formula.

$$\begin{cases} x_i = x_j \\ y_i = y_j \end{cases}$$

where $i,j \in [1,n], i \neq j$.

b)Regrouping the remained border pixels into independant polygons.

7. RECOGNITION OF HATCHED POLYGONS

Among the ploygons in image R (i, j), there are some other loops such as symbols of control points and parts of some Chinese characters besides the hatched polygons. A decisive step to recognize the hatched polygons is to exclude the polygons that do not represent residential sections. By analysing the charaterics of polygons in image R (i, j), we can classify them according to three criteria.

a)The circumstance of a polygon. Generally speaking, a hatched polygon is with a longer circumstance, which is measured by the number of border pixels, than other polygons. Suppose n stands for the number of border pixels of a polygon, then the polygons can be classifed into two groups, large polygons and small polygons. If n is larger than a threshhold N, e. g. N = 35, then the polygon is classifed as a large polygon. If n is less than N, then the polygon is classifed as a small one. Only the large polygons have the chance to be recognized as hatched polygons.

b)The number of black pixels inside a polygon in image R (i, j). A hatched polygon is with some black pixels inside it in image R(i,j). On the contrary, the paygons standing for symbols of control points and some Chinese characters do not include any black pixels inside them. Suppose n_b refer to the number of black pixels inside a polygon in image R(i,j), then the polygons can be classified into two groups, ploygons with black pixels and ploygons without black pixels. If $n_b > 0$, then the polygon is one with black pixels. If $n_b = 0$, then the polygon is one without black pixels. The polygons without black pixels inside them in image R (i, j) are definetely not hatched plygons.

c)The ratio of black pixels inside a polygon in original image I(i, j). There are still some graphics of Chinese characters that can not be distinguished by the above two approaches from hatched polygons in image R(i, j). It is discovered that most graphics of Chinese characters are with a higher ratio of black pixels inside the polygons in original image I (i, j) than that of hatched polygons. Suppose r stand for this ratio, then r can be obtained by following formula.

$$r = n_b / n$$

where n_b and n respectively refer to the numbers of black pixels and of total pixels inside a polygon in image I(i, j).

The ratio of black pixels inside a hatched polgyon mainly ranges from 0. 3 to 0. 6 and that of other polygon ranges from 0. 6 to 0. 9. The distribution of r is shown in Figure 12.

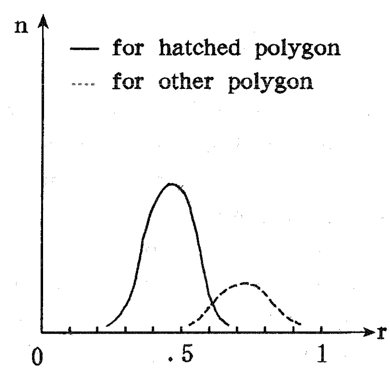

Fig. 12 The distribution of the ratio of black fixels inside polygons

As shown in Figure 12, the ratio of black pixels for hatched polygons and other polygons have obviously diffrent ranges. By choosing a threshold R = 0. 6, most of the hatched polygons can be seperated from others. The result of polygon recognition is shown in Figure 13. Although all of the hatched polygons can be correctly recognized with the method described above, a few polygons standing for Chinese character are misrecognized as hatched polygons at the same time. These misrecognized polygons should be removed by interctive editing. Forturately, the operation of removing misrecognized polygons is easy to carry out. After the process of extracting the vertexes of the recognized polygons, the hatched polygons in vector form can be used in geographic information systems.

At last, a process of deleting the recognized hatched polygons from the original image I(i, j) should be performed. It is very helpful to recognize other kinds of map information.

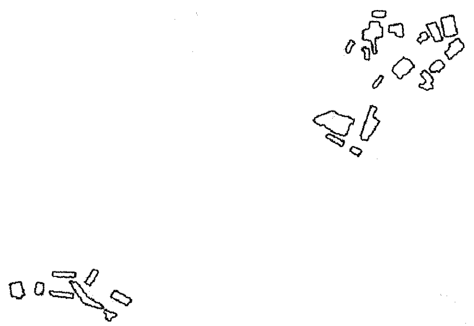

Fig. 13 The result of recognized polygons

8. EXPERIMENTS AND CONCLUSIONS

In order to verify the feasibility and the efficiency of the method proposed in this paper, we arbitrarily select a 1 : 50000 topographical map as experimental data. The scanning resolutions in both horizontal and vertical directions are 250dpi. This piece of map includes 834 hatched polygons as well as many other kinds of graphics. Among the hatched polygons on this map, 829 hatched polygons are correctly recognized and extracted with the proposed method. The ramained 5 hatched polygons are rejected because they are wrongly drown with unclosed borders on original map. Besides the recognized hatched polygons, 26 polygons standing for parts of Chinese characters are misrecognized as hatched polygons. The final ratio of correct recognition of hatched polygons is 96%.

Experiments show that the automatic recognition method for hatched polygons put forward in this paper is effective and feasible. It can reach a high ratio of correct recognition as about 96%. The misrecognized ones are exclusively parts of chinese characters. This means that a even higher recognition ratio of hatched polygons may be reached if this method is performed to maps of western countries since the western texts on maps are more diffirent from hatched polygons. The method has a strong capability of anti — interference. The hatched polygons linking to other graphics in original scanning image can be correctly recognized and extracted. According to the recoginition strategy of layer by layer, it is important to remove the recoginized map symbols from the original map images because it will be helpful to extract other layers of map information afterwards.

References

Capson, D. W. 1984. An improved algorithm for sequential extraction of boundary from a reaster scan. Computer Vision, Graphics, and Image Processing, 28(109).

Fulford, M. C. 1981. The FASTRAK automatic digitizing system. Pattern Recognition, 14(1).

Wahl, F. M. etal. 1982. Block Segmentation and text extraction in mixed text/image documents, Computer Graphics and Image Processing, 2 (6) pp375—390.