IEEE ICIP 1989 Conference Proceeding Vol. 1 of 2, pp. 369-373, 1989

Fast Region Labeling with Boundary Tracking

Hironobu Takahashi

Tsukuba Research Center
SANYO Electric Co., Ltd.
2-1 Koyadai, Tsukuba-shi, Ibaraki 305, Japan

Fumiaki Tomita

Computer Vision Section
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba-shi, Ibaraki 305, Japan

ABSTRACT

We propose a new algorithm that labels the image in one raster scan by tracking the boundary simultaneously. This algorithm has several advantages. It does not require large buffers to record the connectivities of runs used by the normal two-raster scan algorithm. Since it uses only one raster scan, it can label faster than the two-raster scan algorithm. It also can detect the topological properties for each region (such as the existence of holes) at the same time as labeling the regions.

1 Introduction

When recognizing objects in the image, the most common way is to first segment the image into regions, and then to label the regions. All points of the same region are given the same label. This process is called region labeling. The normal labeling algorithm performs two raster scans. In the first scan, each horizontal run (continuous points in the same region and on the same line) is assigned a temporary label. The vertical connectivity between runs is also recorded in a table. In the second scan, label numbers are reassigned according to the records of the table (namely connected runs are merged). Another algorithm labels the region by propagation from a initially labeled point in the region. This algorithm, however, is very time consuming for serial computers. Raster scan type algorithms, therefore, are more commonly used [1].

In this paper, we propose a new algorithm that labels the image in one raster scan by tracking the boundary simultaneously. This algorithm has the advantages that it does not require the above mentioned propagation procedure, and it does not require memory for recording the connectivity of each run. Since it uses only one raster scan, it can label faster than the two-raster scan algorithm. At the same time it can detect the topological properties (such as the existence of holes) for each region.

2 Outline of Method

Points of special interest in a region are those whose left neighbour is not part of the same region. These points are on the boundaries of regions and initiate runs. They are labeled in a procedure which tracks the region boundaries. If we can correctly label them, the others points in the region on the same run can be labeled by propagation from left to right in the raster scan.

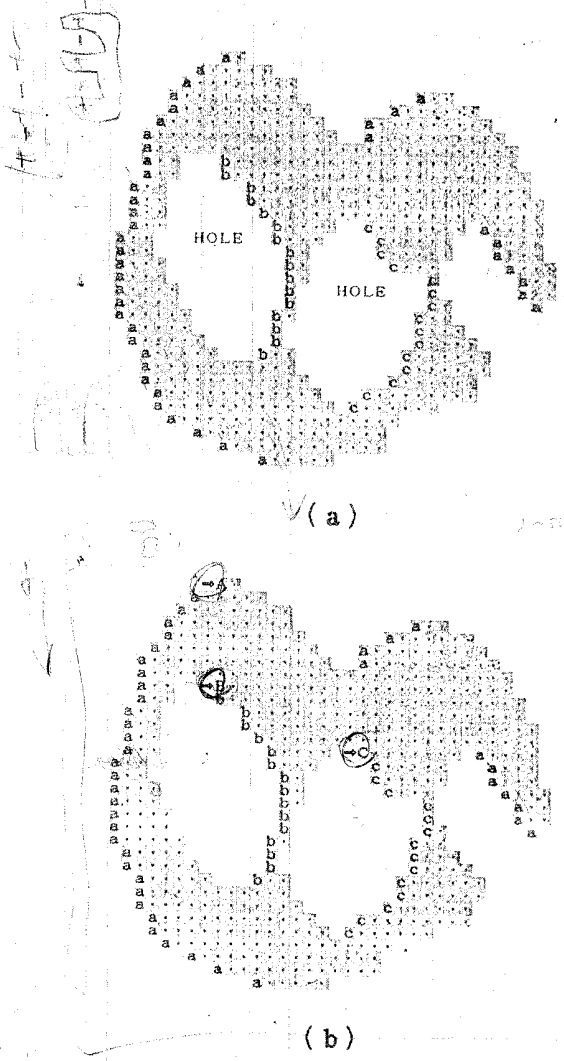


Figure 1: Points of special interest in a region: (a) Points whose left neighbors are not the part of the region; (b) The starting points of boundary tracking, which are shown by arrows.

These points are of two types corresponding to whether they are on the peripheral boundary of the region (shown as 'a' in Fig. 1a) or on the boundary of a hole (shown as 'b' and 'c' in Fig. 1a). These points are selected from the points obtained by tracking the peripheral and hole boundaries, respectively. Let us assume that the first point of a new region is found (shown as 'A' and by an arrow (⇒) in Fig. 1b). This point is given a new label.

From this starting point the peripheral boundary is tracked and the points which initiate runs (marked 'a' in Fig. 1a) are given the same label. Similarly from points such as 'B' and 'C' we can track the boundary of holes in a region and label the corresponding points b and c with the same region label. By propagating these label numbers to the other points by raster scanning, we can get the labeled region.

3 Description of Algorithm

Each point of the input image is initialized to one of the two values, '0' or 'R'. The background (in the case of binarized images) or edges (for edge images) are labeled '0'. Points to be labeled are set to 'R' (See Fig. 2a). By making 'R' a negative integer, the regions are labeled serially starting from '1'. A register stores the current label number and is initially set to '0'. We start a raster scan at the first line in the image. If a scanned point is labeled '0' or with a positive number (i.e. already labeled), the next point is scanned. If the value of the point is 'R', this point has not been labeled. We can distinguish three cases: the point is on the peripheral boundary, the point is on the hole boundary or the point is within the region. The case is determined by looking at the labels of previously scanned points above and to the left of the point and a labeling procedure is selected accordingly. In the following the procedure for 4-connected regions is outlined. The extension to 8-connected regions is then considered.

(a) **Tracking of peripheral boundary of region**
Consider the case shown in Fig. 3a in which the left point and the above point are both '0' (i.e. not belonging to a region). The current point is thus on the peripheral boundary and is the first point to be labeled of a new region. We increment the register, and get the label number of this region. After giving the current point the new label, we then track the peripheral boundary and label points whose left neighbour is outside of the region in the following way.

1. For each current point, a direction code, D, indicating the direction traversed to reach the current point, is determined as shown in Fig. 4 (The start point has this direction code set to 0).
2. The next point of the boundary is found by searching clockwise from the point (i.e. incrementing D) in the directions given by $D-1$ until a point with value 'R' (not yet labeled) or 'K' (already labeled region) is found. This becomes the next point of the boundary.

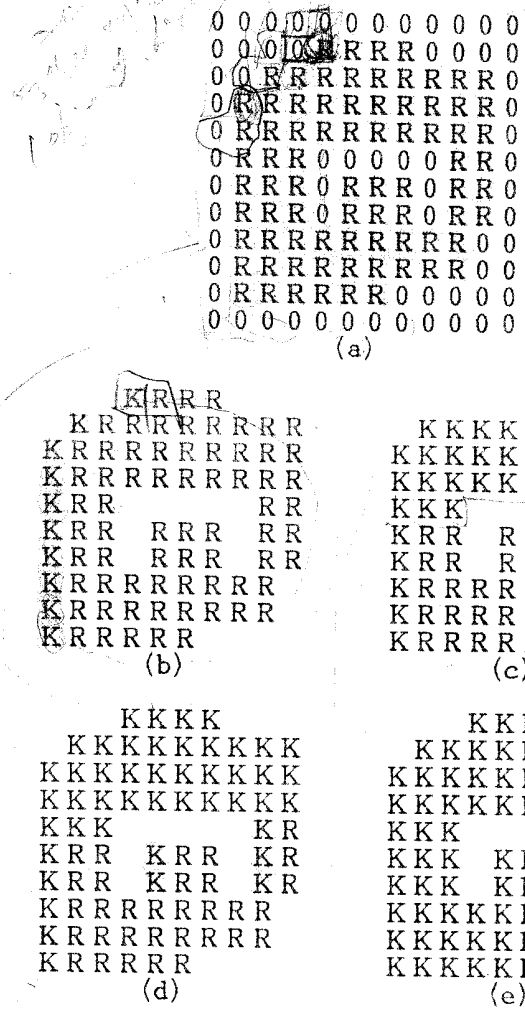


Figure 2: Examples of labeling a region with a hole.

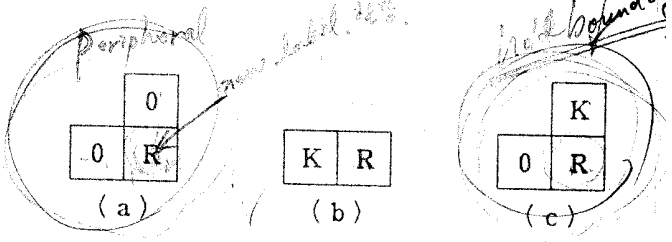


Figure 3: Conditions of labeling and tracking for 4-connected regions.

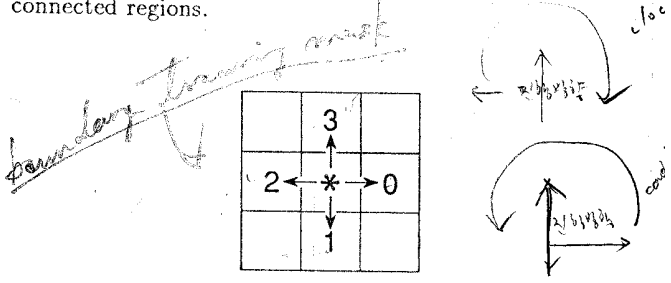


Figure 4: Direction codes from the current point (*) to the next point in 4-neighborhood boundary tracking.

3. The current point is only labeled if its left neighbour is outside of the region since these are points which initialize runs. The decision to label is determined by the directions from the previous point to the current point and from the current point to the next point. The conditions are described in Table 1. Basically the desired points are tracked from "below to above", when we track a peripheral boundary clockwise, and a hole boundary counterclockwise.

4. The direction code is updated, and the procedure is repeated until the boundary is fully tracked (i.e. the current point is the same as the start point and the direction to the next point is the same as the direction from the start point). The result of this boundary tracking and labeling is shown in Fig. 2b.

(b) **Propagation to points in the same horizontal run**

Consider the case shown in Fig. 3b in which the left point is labeled 'K' (> 0). The current point also belongs to the same region 'K' and is labeled accordingly. This results in the propagation of the same value to all unlabeled remaining points of the horizontal run. For example, the points are labeled by this procedure from Fig. 2b to Fig. 2c.

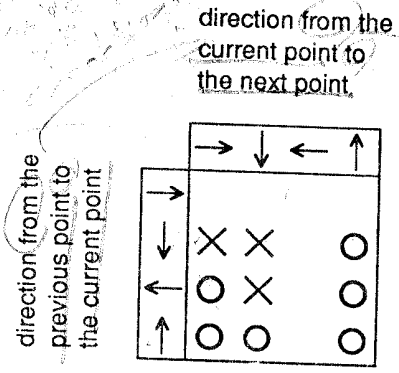
(c) **Tracking the boundary of a hole**

Consider the case shown in Fig. 3c in which its left point is '0' (i.e. an edge point or background), whereas the above point has already been labeled 'K'. The current point is also labeled 'K' as part of the same region (See point marked by an arrow in Fig. 2c). It is in fact on the inside boundary of the region (boundary of a hole), because if this point had been on the peripheral boundary of the region, it would have been labeled previously by the peripheral boundary tracking procedure. We then track the hole boundary and label the left side points in the same way as in procedure of the peripheral boundary tracking. By using the same direction codes and decision table, the hole boundary is tracked in the opposite direction. Only points whose left neighbour is the edge of a hole, will be labeled. This is illustrated in Fig. 2d.

The above procedures are repeated until the end of raster scan. As a result, all the regions are assigned serial numbers from 1 to N (the number of regions in the image), as shown in Fig. 2e.

There is a reason that we do not label some of the cases shown in Table 1. Let us try to label all points on the peripheral boundary of a region in Fig. 5a which has a hole and a thin portion of width 1. After labeling the peripheral boundary and raster scanning, we arrive at the point shown by an arrow (⇒) in Fig. 5b. This point is regarded as a new region. Not only we fail to find a hole but also we label part of the same region with different numbers.

Table 1: The conditions of boundary labeling for 4-connected regions (o: do label, ×: don't label, *blank*: don't care).



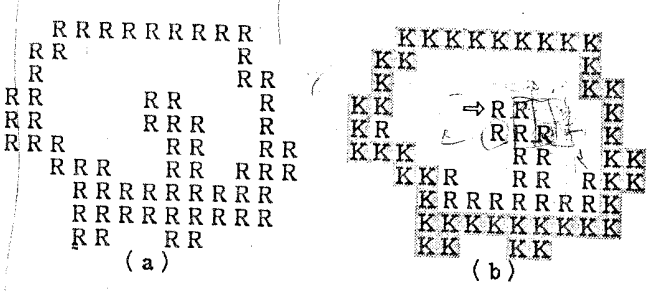


Figure 5: Labeling illegally: (a) Region which has a hole and a thin portion of width 1; (b) Result of labeling all points of the peripheral boundary.

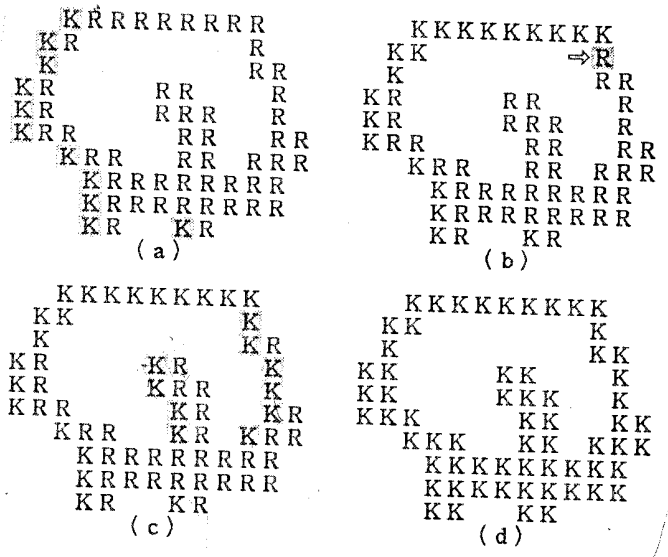


Figure 6: labeling legally.

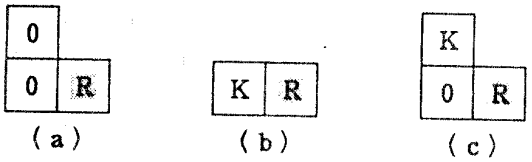


Figure 7: Conditions of labeling and tracking for 8-connected regions.

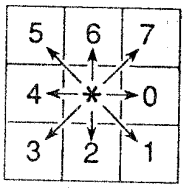


Figure 8: Direction codes from the current point (*) to the next point in 8-neighborhood boundary tracking.

Table 2: The conditions of boundary labeling for 8-connected regions (o: do label, ×: don't label, −: not exist, *blank*: don't care).

direction from the previous point to the current point \ direction from the current point to the next point	→	↘	↓	↙	←	↖	↑	↗
→						−	−	
↘	×	×	×	×		o	−	×
↓	−	×	×	×		o	o	−
↙	−	×	×	×		o	o	o
←	o	−	−	×		o	o	o
↖	o	o	−	×		o	o	o
↑	o	o	o	−	−	o	o	o
↗	o	o	o	o	−	o	o	o

Our algorithm, in this case, first labels the part of the peripheral boundary, as shown in Fig. 6a. After raster scanning, we arrive at the point shown by an arrow (⇒) in Fig. 6b, where we know the existence of a hole. By tracking the hole boundary, the part of the boundary is labeled, as shown in Fig. 6c. And, the remaining points in the region are labeled properly by the horizontal propagation by the end of the raster scan, as shown in Fig. 6d.

The procedure for 4-connected regions has been outlined above. We extend the procedure to the case of 8-connected regions. We change the boundary search pattern of Fig. 3 to that of Fig. 7, and we use the direction codes illustrated in Fig. 8. We track the boundary in a 8-neighbourhood, and label the boundary points according to Table 2.

4 Comparative Study

We compare the our algorithm and the normal two-raster scan algorithm with respect to memory accesses. We consider the case of labeling an image of a region which has m points and a boundary with n points. At first we estimate the case of a 4-connected region. By the normal algorithm [1], in the first raster scan, we need 2 memory *reads* (point itself, and the above point), and 1 memory *write* for each point in the region. In the second scan, we need 1 memory *read* and 1 memory *write*. In total, the normal algorithm requires $5m$ memory accesses. With our algorithm, we need 1 memory *read* and 1 memory *write* for each region point in the raster scan, and we need 2 memory *reads* on average for searching in the boundary tracking and 0.5 memory *write*. In total, our algorithm requires an average of $2m + 2.5n$ compared to the above $5m$ memory accesses. This represents a saving of $3m - 2.5n$ memory accesses. Because m is far greater than n in the normal images, our algorithm can label the images faster than the normal algorithm. For a 8-connected region, the normal algorithm requires a total of $7m$ memory accesses, (the first scan of 4 *reads* and 1 *write*, and the second scan of 1 *read* and 1 *write*). Our algorithm requires a total of $2m + 3.5n$ memory accesses, (raster scan of 1 *read* and 1 *write*, and boundary tracking of an average of 3 *reads* and 0.5 *write*). This represents a saving of $5m - 3.5n$ memory accesses.

5 Experimental Results

The algorithm was tested on a real scene image containing about 100 regions (Fig. 9). The elapsed CPU time of a SUN3/260 work station was measured and compared. For 4-connected regions, the results are shown in Fig. 10. Our

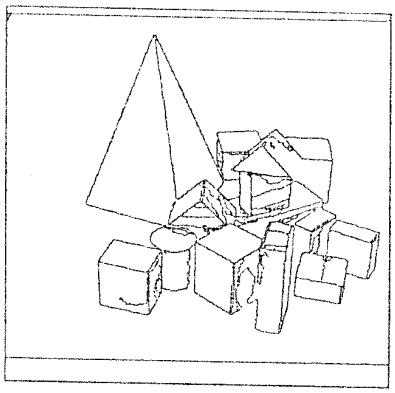


Figure 9: Sample edge image.

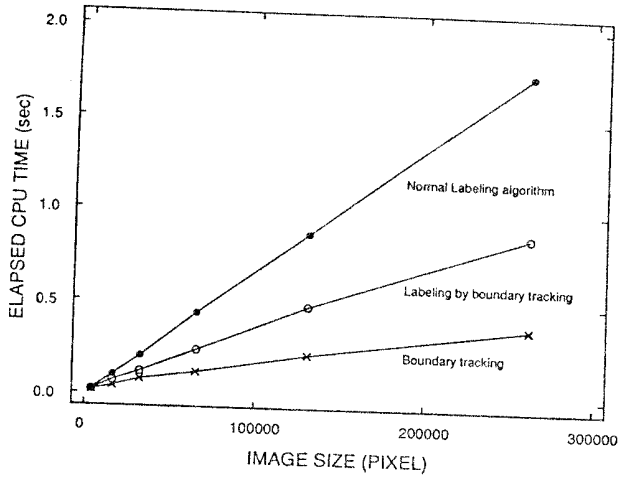


Figure 10: Comparison of experimental results in case of 4-connected regions.

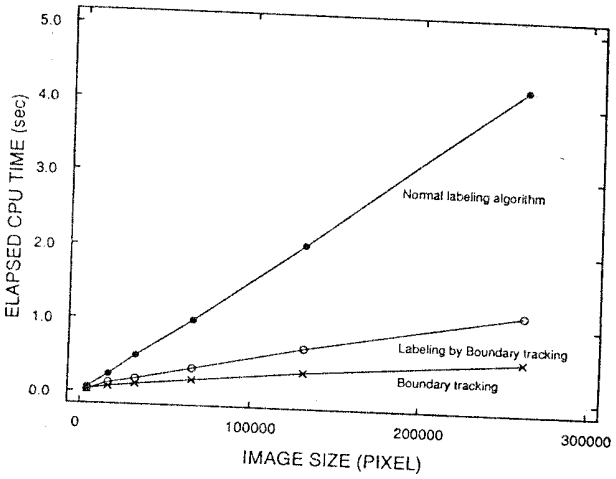


Figure 11: Comparison of experimental results in case of 8-connected regions.

algorithm is approximately twice as fast as the normal algorithm. For 8-connected region, the results are shown in Fig. 11. Our algorithm is approximately 3.5 times faster. Our method can, of course, be further simplified, if we are only interested in the boundary information and nor in labeling points within the region. This will save memory and CPU time. Fig. 10 and 11 show the improved performance of this simplified boundary tracking algorithm. It is approximately 4 times faster for 4-connected regions and about 8 times faster for 8-connected regions than the normal algorithm.

6 Conclusions

The proposed method using one raster scan and boundary tracking, has several advantages. The algorithm does not require large buffers to record the connectivities of runs used by the two-raster scan algorithm. Our algorithm labels the images in one raster scan effectively. It also can determine the boundary properties at the same time as labeling the regions. The algorithm can be implemented in hardware, and the prototype system is now being constructed.

ACKNOWLEDGMENTS

The authors are grateful to Mr. R. Cipolla for comments and criticisms which helped to improve the readability of this paper and to the members of the Computer Vision Section, ETL for useful discussion.

REFERENCES

[1] A. Rosenfeld and A.C. Kak, Digital Picture Processing, Academic Press, 1976.