Fast Connected-Component Labeling Based on Sequential Local Operations in the Course of Forward Raster Scan Followed by Backward Raster Scan

Kenji Suzuki*, Isao Horiba*, and Noboru Sugie**
*Faculty of Information Science and Technology, Aichi Prefectural University
**Faculty of Science and Technology, Meijo University
E-mail: {k-suzuki, horiba}@ist.aichi-pu.ac.jp, sugie@meijo-u.ac.jp

Abstract

This paper presents a fast algorithm for labeling connected components in binary images based on sequential local operations. A one-dimensional table, which memorizes label equivalences, is used for uniting equivalent labels successively during the operations in forward and backward raster directions. The proposed algorithm has a desirable characteristic: the execution time is directly proportional to the number of pixels in connected components in an image. By comparative evaluations, it has been shown that the efficiency of the proposed algorithm is superior to those of the conventional algorithms.

1. Introduction

One of the most fundamental operations in pattern recognition is the labeling of connected components in a binary image [1]. The labeling algorithm transforms a binary image into a symbolic image in order that each connected component is assigned a unique label. Various algorithms have been proposed so far. They are put into four broad classes:

(A) Algorithms repeat passes through an image in forward and backward raster directions alternately to propagate the label equivalences until no labels change [2][3].

(B) Algorithms perform two passes: during the first pass, provisional labels are assigned to connected components; the label equivalences are stored in a one-dimensional or a two-dimensional table array. After or during the first pass, the label equivalences are resolved using a search algorithm such as the UNION-FIND algorithm. During the second pass, the provisional labels are replaced by the smallest equivalent label using the table [4]-[7].

(C) Algorithms [8]-[10] have been developed for the images represented by hierarchical tree structures, i.e., n-ary tree such as bintree, quadtree, octree, etc. The label equivalences are resolved using a search algorithm such as the UNION-FIND algorithm.

(D) Parallel algorithms [11]-[13] have been developed for parallel machine models such as a mesh connected massively parallel processor.

The algorithms in the class (A) are relatively easy to implement on hardware because they are based on only sequential local operations. However, they require a large number of passes. The required execution time has not been clarified theoretically as yet: it depends on the complexity of the connected components. In the algorithms in the class (B), since the label equivalences are resolved using a search algorithm, it is not easy to implement on hardware; it takes, in general, enormous time. The execution time also depends on the complexity of the connected components. In the algorithms in the class (C), the execution time is much more efficient, but the worst case one, however, is the same as that required for the image representation as an array. The algorithms in the class (D) are not suitable for ordinary computer architectures.

In this paper, a fast algorithm for labeling the connected components, based on sequential local operations using one-dimensional table, is proposed. The proposed algorithm combines ones in the classes (A) and (B), leading to fast computation, and is suitable for ordinary computers. By comparative evaluation with the conventional algorithms, the efficiency of the proposed algorithm is shown.

2. Fast connected-component labeling

2.1. The proposed algorithm

In the proposed algorithm, one-dimensional table called the label connection table, which memorizes label equivalences, is used successively during operations. The forward and the backward scans are performed alternately successively using the label connection table; this leads to fast labeling.

Let us suppose that the binary image $b(x,y)$ consists of pixel values F_O, indicating objects, and F_B, indicating the background; that F_O and F_B are sufficiently high values ($F_O < F_B$); that a provisional label m is initialized to one;

0-7695-0750-6/00 $10.00 © 2000 IEEE

and that the label connection table is initialized as $T[F_O]=F_O$, $T[F_B]=F_B$.

[First scan]

The following sequential local operations using the label connection table are performed in the forward raster scan order to assign the provisional labels and propagate them to the connected components.

$$g(x,y)=\begin{cases} F_B & \text{if } b(x,y)=F_B \\ m,(m=m+1) & \text{if } \forall\{i,j\in M_S\}\ g(x-i,y-j)=F_B \\ T_{\min}(x,y) & \text{otherwise} \end{cases}, \quad (1)$$

$$T_{\min}(x,y)=\min[\{T[g(x-i,y-j)]\,|\,i,j\in M_S\}] \quad (2)$$

where *(m=m+1)* indicates an increment of m, min(.) an operator calculating the minimum value, and M_S the region of the mask except the object pixel, i.e., *b(x-1,y-1)*, *b(x,y-1)*, *b(x+1,y-1)*, and *b(x-1,y)* for 8-connected. In the above equation (1), the priority is given to the condition in the upper column.

The label connection table is updated, simultaneously with assigning the provisional labels, as follows:

$$\begin{cases} non-operation & \text{if } b(x,y)=F_B \\ T[m]=m & \text{if } \forall\{i,j\in M_S\}\ g(x-i,y-j)=F_B \\ T[g(x-i,y-j)]=T_{\min}(x,y) & \text{if } g(x-i,y-j)\neq F_B \end{cases}. \quad (3)$$

[Scans after the first scan]

After the first scan, the backward scan and the forward scan are performed alternately. The following operations using each mask of its own are performed in each raster scan order:

$$g(x,y)=\begin{cases} F_B & \text{if } g(x,y)=F_B \\ T_{\min}(x,y) & \text{otherwise} \end{cases}, \quad (4)$$

$$T_{\min}(x,y)=\min[\{T[g(x-i,y-j)]\,|\,i,j\in M\}] \quad (5)$$

where M denotes the region of the mask, i.e., *g(x+1,y+1)*, *g(x,y+1)*, *g(x-1,y+1)*, *g(x+1,y)*, and *g(x,y)* for the backward scan. The label connection table is updated as follows:

$$\begin{cases} non-operation & \text{if } g(x,y)=F_B \\ T[g(x-i,y-j)]=T_{\min}(x,y) & \text{if } g(x-i,y-j)\neq F_B \end{cases}. \quad (6)$$

The forward and backward scans are repeated until no provisional labels change, i.e., until the following stopping condition is not fulfilled, and then the final labeled image is obtained.

$$T[g(x-i,y-j)]=T_{\min}(x,y) \ \text{ if } g(x-i,y-j)\neq F_B \quad (7)$$

where $i,j\in M_S$.

2.2. Analysis of the proposed algorithm

Figure 1 shows the result of labeling a stair-like connected component. (a) shows the provisional labels assigned by the algorithms without using the label connection table, i.e., the result after the first pass of the conventional algorithms in the classes (A) and (B). A circle indicates the junction of two different provisional labels. On this connected component, the provisional labels 1 and 2 meet before 2 and 3 meet. Figure 1(b) shows the order of occurrences of the junctions in the forward raster scan order. After the first scan of the proposed algorithm, the provisional labels are assigned to the connected component, as Figure 1(c) shows. Data of 1 is written to the address of 2 of the label connection table. Then, the provisional labels 2 and 3 meet. Data of 1, the result of conversion of 2 using the label connection table, is written to the address of 3 of the table, and the label 1 is assigned to the pixel of the connected component. Thus, the label connection table for this connected component, memorizing that the provisional labels 1, 2 and 3 are equivalent to the label 1, is completed. After the second scan, the labeling is completed, as Figure 1(d) shows. The labeling of the stair-like connected component having N steps are also completed by only two scans.

Giving consideration to the order of occurrences of the junctions, the connected component composed of three provisional labels connected by two junctions is a primary orderly label-connection. The number of permutations of this, ${}_3P_2$, is six. Possible six connected components in the primary orderly label-connections are shown in Figure 2. Any arbitrary connected component can be represented by the combination of these six patterns. These patterns can be classified into three groups: (i) the connectivity of the patterns I and III can be resolved by the forward scan; (ii) that of the patterns II and IV can be done by the backward scan; (iii) that of the patterns V and VI can be done by either the forward scan or the backward scan.

If the pattern is a simple combination of these six patterns, the labeling is completed by the same manner. As the proof is straightforward, it is omitted due to the limitation of space. However, to complete labeling the multiplex combination of the patterns I and/or III and the patterns II and/or IV, both forward and backward scans are required. To complete the labeling of such patterns, the following four scans are required: the forward scan, the backward scan, the forward scan, and either the forward or the backward scan.

Through the above consideration, it has been shown that at least four scans are required to complete labeling arbitrary connected components. However, since to derive the theoretical upper bound of the number of scans required to complete labeling is of difficulty, it will be shown experimentally in the next section.

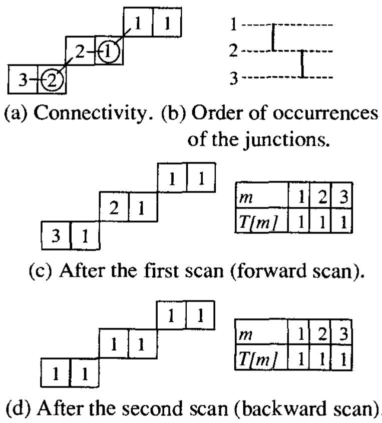

Figure 1. Labeling of a stairs-like connected-component using the proposed algorithm.

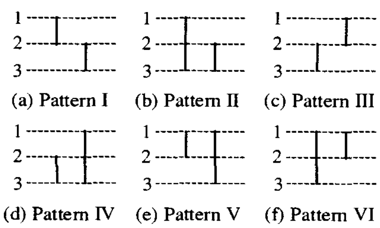

Figure 2. Primary orderly label connections.

3. Experiments

3.1. Characteristics of the proposed algorithm

3.1.1. Maximum number of scans. In order to examine the maximum number of scans, the following test images were prepared: the images (size: 512×512 pixels; maximum level of the gray scale: 1,000) with white uniform random noise were transformed into 41 binary images by varying the threshold from 1 to 1,000 with the step of 25. 50 sets of these test images were made by changing the noise. This kind of images is appropriate for the severe evaluation of the labeling algorithms, because the connected components in them have the complicated geometrical shapes and the complex connectivity.

The labeling of these 2,050 images were performed by the proposed algorithm. As a result of this experiment, the labeling of all of the images was completed by no more than four scans. This result indicates that the labeling of almost arbitrary images is completed by four passes.

3.1.2. Characteristic against the number of object pixels. We selected Haralick's algorithm in [2] from the class (A) and Shirai's algorithm in [4] from the class (B), which are well-known representatives, as the targets for comparison (here refereed to as the conventional algorithm A and B, respectively). The results of comparison of the CPU execution time on a workstation (UltraSPARC-II 300MHz made by Sun microsystems) are shown in Figure 3. The maximum execution time of the conventional algorithms A and B are 4.33 and 8.88 seconds, respectively. In contrast, that of the proposed algorithm is 0.19 seconds. It is the maximum execution time of the proposed algorithm for this size of images on this computer. We discovered that the most burdensome connected component to the conventional algorithm A is the stair-like connected-component in Figure 1. If the number of steps is N, $2(N-1)$ passes are required. When the whole image is composed of the stair-like connected components, the conventional algorithm A takes the theoretical upper bound of the execution time. As a result of the measurement, it was 15.53 seconds.

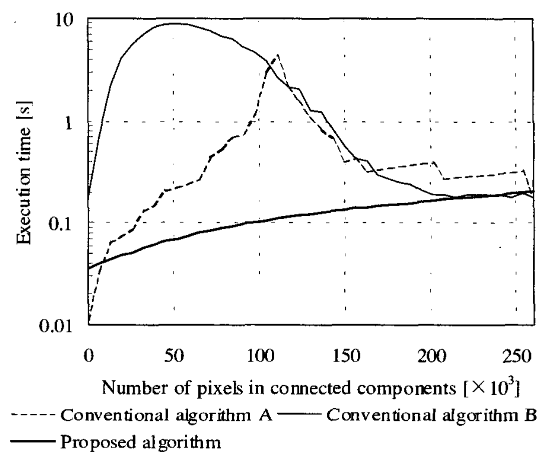

Figure 3. Execution time against the number of pixels.

3.1.3. Characteristic against image size. By varying the size of images (64×64, 128×128, 256×256, and 512×512 pixels in size), the execution times with 164 binary images were measured. The results are shown in Figure 4. In order to make quantitative the characteristics against the number of pixels I, the maximum execution time t_{max} was approximated by the following equation using the least-square-fitting:

$$t_{\max} = aI^{b} \qquad (8)$$

where a and b denote parameters. As a result, The execution time of the conventional algorithms A and B, and the proposed algorithm is directly proportional to the power of 1.5, 2.0, and 1.0 of the number of pixels, respectively. This shows that the proposed algorithm has a desirable characteristic: the execution time does not increase so much even in the case of a larger image.

3.2. Evaluation with various images

50 natural images from the SIDBA (Standard Image Data Base) and the image data base of the USC (University of Southern California), and 50 medical images, which are 512x512 pixels in size, were used for evaluation. The images were transformed into binary images by using Otsu's threshold selection method in [14]. The results of comparison of the execution time are shown in Table 1. The proposed algorithm has completed labeling all images by no more than three scans. The proposed algorithm takes the least execution time of all.

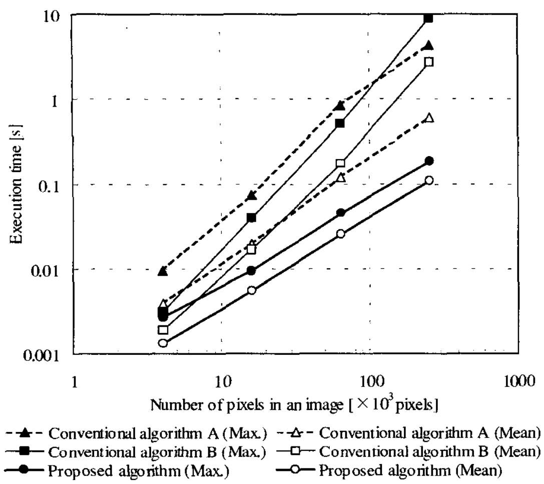

Figure 4. Execution time against the number of pixels.

Table 1. Comparison of the execution time

	Execution time [s]		
	Max.	Mean	Min.
Conventional algorithm A	1.144	0.299	0.059
Conventional algorithm B	0.451	0.206	0.144
Proposed algorithm	0.129	0.078	0.035

4. Concluding remarks

This paper has presented a fast algorithm for labeling connected components in binary images. The proposed algorithm is based on sequential local operations with one-dimensional table in the raster scan directions. This is suitable for implementation in hardware. Through comparative evaluation with the conventional algorithms, it has been demonstrated that the proposed algorithm takes the least execution time. It has been shown experimentally that the proposed algorithm has a desirable characteristic: the execution time is directly proportional to the number of pixels in connected components in an image. Although it has been shown that the labeling of 2,314 images is completed by no more than four scans, the mathematical proof of the maximum number of scans remains.

5. Reference

[1] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, 2nd ed., Vol. 2, Chap. 10, Academic Press, 1982.

[2] R.M. Haralick, "Some neighborhood operations," in *Real Time/Parallel Computing Image Analysis*, M. Onoe, K. Preston, Jr., and A. Rosenfeld Eds., Plenum Press, New York, pp. 11-35, 1981.

[3] A. Hashizume, R. Suzuki, H. Yokouchim, et al., "An algorithm of automated RBC classification and its evaluation," *Bio Medical Engineering*, vol. 28, no. 1, pp. 25-32, 1990.

[4] A. Rosenfeld and J.L. Pfalts, "Sequential operations in digital picture processing," *Journal of ACM*, Vol. 13, No. 4, pp. 471-494, Oct. 1966.

[5] Y. Shirai, "Labeling connected regions," in *Three-Dimensional Computer Vision*, Springer-Verlag, pp. 86-89, 1987.

[6] T. Hattori, "A high-speed pipeline processor for regional labeling based on a new algorithm," *Proc. ICPR'90*, NJ, USA, pp. 494-496, June 1990.

[7] R. Lumia, L. Shapiro, and O. Zungia, "A new connected components algorithm for virtual memory computers," *Computer Vision, Graphics, and Image Processing*, Vol. 22, No. 2, pp. 287-300, 1983.

[8] H. Samet, "Connected component labeling using quadtrees," *Journal of ACM*, Vol. 28, No. 3, pp. 487-501, July 1981.

[9] H. Samet, "Computing geometric properties of images represented by linear quadtrees," *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol. PAMI-7, No. 2, pp. 229-240, 1985.

[10] J. Hecquard and R. Acharya, "Connected component labeling with linear octree", *Pattern Recognition*, Vol. 24, No. 6, pp. 515-531, 1991.

[11] A. Choudhary and R. Thakur, "Connected component labeling on coarse grain parallel computers: an experimental study," *Journal of Parallel and Distributed Computing*, pp. 78-83, Jan. 1994.

[12] P. Bhattacharya, "Connected component labeling for binary images on a reconfigurable mesh architectures," *Journal of Systems Architecture*, Vol. 42, No. 4, pp. 309-313, 1996.

[13] M. Manohar and H.K. Ramapriyan, "Connected component labeling of binary images on a mesh connected massively parallel processor," *Computer Vision, Graphics, and Image Processing*, Vol. 45, No. 2, pp. 133-149, 1989.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems Man & Cybernetics*, Vol. 9, pp. 62-66, Jan. 1979.