Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ACADEMIC PRESS

Computer Vision and Image Understanding 89 (2003) 1–23

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Linear-time connected-component labeling based on sequential local operations

Kenji Suzuki,[a,*] Isao Horiba,[a] and Noboru Sugie[b]

[a] *Faculty of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi 480-1198, Japan*

[b] *Faculty of Science and Technology, Meijo University, Tempaku, Nagoya 468-8502, Japan*

Received 18 October 2000; accepted 19 November 2002

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.

Keywords: Labeling; Connected component; Linear time; Sequential local operation; One-dimensional table; Raster scan order

1. Introduction

One of the most fundamental operations in pattern recognition is the labeling of connected components in a binary image. The labeling algorithm transforms a binary image into a symbolic image in order that each connected component is

* Corresponding author. Present address: Kurt Rossmann Laboratories for Radiologic Image Research, Department of Radiology, The University of Chicago, 5841 South Maryland Avenue, MC 2026, Chicago, IL 60637, USA. Fax: 1-773-702-0371.

E-mail address: k-suzuki@hilum.bsd.uchicago.edu (K. Suzuki).
URL: http://xray.bsd.uchicago.edu/krl/.

1077-3142/03/$ - see front matter
doi:10.1016/S1077-3142(02)00030-9

assigned a unique label. This issue has a close connection with the connectivity of the connected components and cannot be resolved by mere parallel local operations [21]. This is a typical one requiring sequential operations [28,30].

Various algorithms have been proposed so far. They are classified into four classes; the following two classes are representatives and suitable for ordinary computer architectures:

(A) Algorithms [9,11] repeat passes through an image in forward and backward raster directions alternately to propagate the label equivalences until no labels change.

(B) Algorithms [7,8,10,16–18,22,28,30,37] 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 by some search. This is often performed by using a search algorithm such as the union-find algorithm [5,20,41,42]. The results of resolving are generally stored in a one-dimensional table. During the second pass, the provisional labels are replaced by the smallest equivalent label using the table.

The others are as follows:

(C) Algorithms [4,6,13,31,33–35,40] have been developed for the images represented by hierarchical tree structures [15,32,38], i.e., *n*-ary tree such as bintree, quadtree, octree, etc. The label equivalences are resolved by using a search algorithm such as the union-find algorithm. The algorithm in [4] can handle the images represented by both an array and hierarchical tree structures.

(D) Parallel algorithms [2,3,14,19,23,26,36,43] have been developed for parallel machine models such as a mesh connected massively parallel processor.

In addition, the hardware implementation of the above algorithms has been also studied [1,12,24,25,44].

The algorithms in the class (A) are relatively easy to implement in hardware because they are based on only sequential local operations and require no search algorithm. However, they require a large number of passes. The required execution time has not been clarified theoretically as yet: it may depend on the complexity of the connected components. In the algorithms in the class (B), since the label equivalences are resolved by using some search, the execution time may also depend 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 those in the class (B), i.e., where the image representation is an array. The algorithms in the class (D) are not suitable for ordinary computer architectures.

The execution time of the above conventional algorithms differs exceedingly among different images. In their applications, it should be estimated that the maximum execution time is required. This prevents them from applying to a wide range of real-time applications.

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. First, the determination of the connected components at which the conventional algorithms in the class (A) are weak, i.e., the maximum execution time is required for them, is stated. Then, it is described that the proposed algorithm can determine such connected components fast. Through experiments, the characteristics of the conventional algorithms and the proposed algorithm are made clear. By comparative evaluations with the conventional algorithms, the efficiency of the proposed algorithm is shown.

2. Conventional algorithm

2.1. Outline of the conventional algorithm

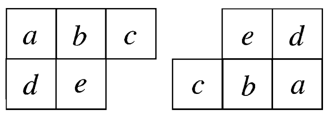
The conventional algorithm in [9] in the class (A) repeats passes through a binary image $b(x,y)$ in the forward and the backward raster directions alternately. Let us suppose that the binary image $b(x,y)$ consists of pixel values F_{O}, indicating objects, and F_{B}, indicating the background; that F_{O} and F_{B} are sufficiently high values ($F_{\mathrm{O}} < F_{\mathrm{B}}$); and that a provisional label m is initialized to one. First, the following sequential local operations in the forward raster scan order, called the forward scan, are performed using the mask shown in Fig. 1a for eight-connected components—see [29] for the definition

$$g(x,y) = \begin{cases} F_{\mathrm{B}} & \text{if } b(x,y) = F_{\mathrm{B}}, \\ m, (m = m+1) & \text{if } \forall\{i,j \in M_S\} g(x-i, y-j) = F_{\mathrm{B}}, \\ g_{\min}(x,y) & \text{otherwise}, \end{cases} \tag{1}$$

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

where $(m = m+1)$ indicates an increment of m, $\min(\cdot)$ 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 $b(x-1,y)$. In the above equation (1), the priority is given to the condition in the upper column. The condition in the first column means a case where the object pixel $b(x,y)$ is in the background. The condition in the second column means a case where the object pixel belongs to the object, and all of the neighboring pixels in the mask to the background. In this case, an initial provisional label is assigned to the object pixel. In the case of the third column, the minimum label among the provisional labels in the mask is assigned to the object pixel. Thus, the smaller label propagates on the connected component.

Then, the following operations in the backward raster scan order, called the backward scan, are performed using the mask shown in Fig. 1b,



(a) (b)

Fig. 1. Masks for the labeling of eight-connected components: (a) forward scan; (b) backward scan.

$$g(x,y) = \begin{cases} F_{\mathrm{B}} & \text{if } g(x,y) = F_{\mathrm{B}}, \\ \min[\{g(x-i,y-j)\,|\,i,j \in M\}] & \text{otherwise,} \end{cases} \tag{3}$$

where M indicates the region of the mask.

The forward and the backward scans are repeated alternately until no provisional labels change, and then the final labeled image in which each connected component is assigned a unique label can be obtained.

2.2. Problem

In the conventional algorithm, the labeling can be completed by performing plural scans. However, the number of scans required to complete the labeling has not been clarified theoretically as yet; it depends on the geometrical complexity of the connected components. It may take huge time to complete the labeling; this is a serious problem.

Fig. 2 illustrates the labeling of one sample, a stair-like connected component, using the conventional algorithm. When an object with a rectangular shape is put with a slight inclination, the obtained digital image frequently includes this kind of connected component. The labeling is completed by four scans, as shown in the figure. The provisional labels propagate from the top left to the right bottom—in the opposite direction in the case of the backward scan—on the connected components, by replacing neighboring connected pixel values with smaller ones. Plural scans depending on the geometrical complexity of the connected components are required to complete labeling. When the number of steps of the stair-like connected component is N, $2(N-1)$ scans are required. The labeling of the other sample, a spiral-like connected component, is illustrated in Fig. 3. Four scans are required to complete the labeling of this connected component, the number of turns of which is two, as shown in the figure. When the number of turns is N, $2N$ scans are required.

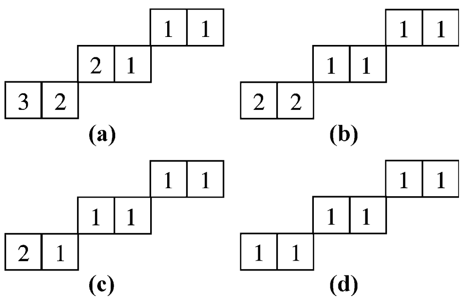


Fig. 2. Labeling of a stair-like connected component using the conventional algorithm: (a) after the first forward scan; (b) after the first backward scan; (c) after the second forward scan; (d) after the second backward scan.

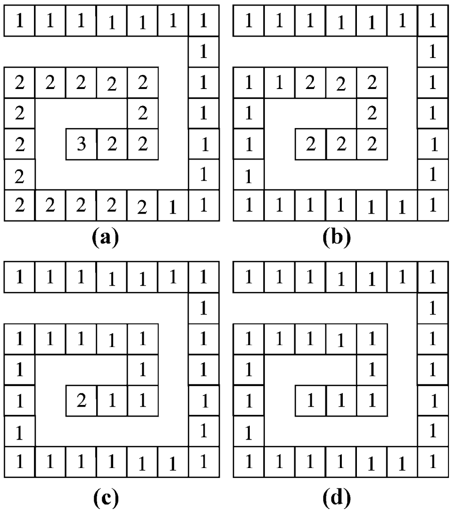


Fig. 3. Labeling of a spiral connected component using the conventional algorithm: (a) after the first forward scan; (b) after the first backward scan; (c) after the second forward scan; (d) after the second backward scan.

3. Fast connected-component labeling

3.1. Proposed algorithm

We have reported the preliminary version of the proposed algorithm briefly in [39]. In the conventional algorithm, since the provisional labels propagate only in a definite direction on the connected components, plural scans depending on the geometrical complexity of them are required. In the proposed algorithm, a one-dimensional table called the label connection table, which memorizes label equivalences, is used successively during operations. The provisional labels propagate not only on the connected components but also on the table. By this measure, the connectivity between the provisional labels at a geometrical distance can be reflected on the label-propagation; this can reduce the number of scans.

In the proposed algorithm, the forward and the backward scans are performed alternately successively using the label connection table; this leads to fast labeling. First of all, the label connection table is initialized as $T[F_O] = F_O$ and $T[F_B] = F_B$.

[*First scan*]

The following sequential local operations using the label connection table $T[m]$ are performed to assign the provisional labels and propagate them to the connected components:

$$g(x,y) = \begin{cases} F_B & \text{if } b(x,y) = F_B, \\ m, (m = m+1) & \text{if } \forall\{i,j \in M_S\} g(x-i, y-j) = F_B, \\ T_{\min}(x,y) & \text{otherwise,} \end{cases} \tag{4}$$

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

The first and second columns in the above equation (4) are the same as those in the conventional algorithm. In the third column, the minimum label among the provisional labels converted by the label connection table is assigned to the object pixel. Thus, smaller labels, on which the previously obtained connectivity is reflected, propagate on the connected component.

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

$$\begin{cases} \text{non-operation} & \text{if } b(x,y) = F_{\mathrm{B}}, \\ T[m] = m & \text{if } \forall\{i,j \in M_S\} g(x-i,y-j) = F_{\mathrm{B}}, \\ T[g(x-i,y-j)] = T_{\min}(x,y) & \text{if } g(x-i,y-j) \neq F_{\mathrm{B}}. \end{cases} \tag{6}$$

In the second column, an initial value is given to the label connection table. In the third column, the connectivity is stored in the table as the form that the provisional labels are equivalent to the minimum one. Thus, the previously obtained connectivity is reflected on the stored labels as smaller labels.

[*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:

$$g(x,y) = \begin{cases} F_{\mathrm{B}} & \text{if } g(x,y) = F_{\mathrm{B}}, \\ T_{\min}(x,y) & \text{otherwise}, \end{cases} \tag{7}$$

$$T_{\min}(x,y) = \min[\{T[g(x-i,y-j)] \mid i,j \in M\}]. \tag{8}$$

The label connection table is updated, simultaneously with the above operations, as follows:

$$\begin{cases} \text{non-operation} & \text{if } g(x,y) = F_{\mathrm{B}}, \\ T[g(x-i,y-j)] = T_{\min}(x,y) & \text{if } g(x-i,y-j) \neq F_{\mathrm{B}}. \end{cases} \tag{9}$$

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

$$g(x-i,y-j) \neq T_{\min}(x,y) \quad \text{if } g(x-i,y-j) \neq F_{\mathrm{B}}, \tag{10}$$

where $i,j \in M_S$.

The operations in the proposed algorithm can be performed by using the logic table shown in Appendix A. An efficient implementation can be realized by using it. The labeling of four-connected components [29] can be performed by using the mask shown in Appendix B.

3.2. Main features

The main features of the proposed algorithm are as follows:

(1) The algorithm is based on only sequential local operations; no search algorithm to resolve the label equivalences is required.

(2) The label equivalences are stored in a one-dimensional table; the connectivity is resolved by simply reading and writing the table during the scans.

As for the feature (1), since the search algorithm consists of complex procedures, it is performed generally in software. It takes much time to execute. The proposed algorithm, requiring no search algorithm, is expected to take less time. Furthermore, since the proposed algorithm is based on only sequential local operations, it is suitable for implementation in hardware. The conventional algorithms having the same feature as the feature (1) are the algorithms in [9,11] in the class (A). They may require a lot of scans, depending on the geometrical complexity of connected components, as described concretely in the previous section.

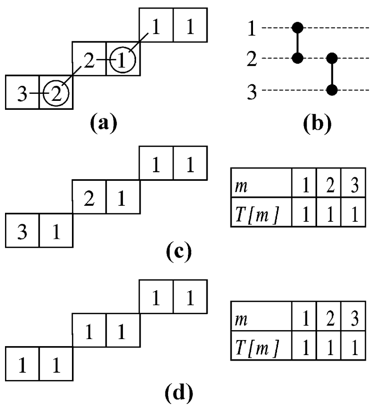
As for the feature (2), in the conventional algorithms, the label equivalences are stored in a two-dimensional table, a pair of one-dimensional tables or a one-dimensional table. In general, the algorithm using a one-dimensional table, like the proposed algorithm, can be implemented in a smaller size of hardware. The proposed algorithm requires just a small size of one-dimensional table handling the number of initial provisional labels. Since the connectivity is resolved by simply reading and writing the table in the proposed algorithm, the control circuit becomes very simple. The algorithms which resolve the connectivity in the same way as the proposed algorithm have not been proposed yet.

Thus, the proposed algorithm is expected to take less time and to be suitable for implementation in hardware.

3.3. Analysis of the proposed algorithm

3.3.1. Some results of labeling

Consideration is given to some results of labeling. Fig. 4 shows the result of labeling the stair-like connected component shown in Fig. 2. Fig. 4a shows the provi-



m	1	2	3
$T[m]$	1	1	1

m	1	2	3
$T[m]$	1	1	1

Fig. 4. Labeling of a stair-like connected component using the proposed algorithm (connection pattern I): (a) connectivity; (b) order of occurrences of the junctions; (c) after the first scan (forward scan); (d) after the second scan (backward scan).

sional labels assigned by the first forward scan of the conventional algorithm. ○ indicates the junction of two different provisional labels. On this connected component, the provisional labels 1 and 2 meet before 2 and 3 meet. Fig. 4b 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 shown in Fig. 4c. Since the provisional labels 1 and 2 meet first, the label 1 is written to the address ($T(2)$) of 2 of the label connection table. Then, the provisional labels 2 and 3 meet. The label 1, the result of conversion of the label 2 using the label connection table, is written to the address ($T(3)$) 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 shown in Fig. 4d. The labeling of the stair-like connected component having N steps is also completed by two scans.

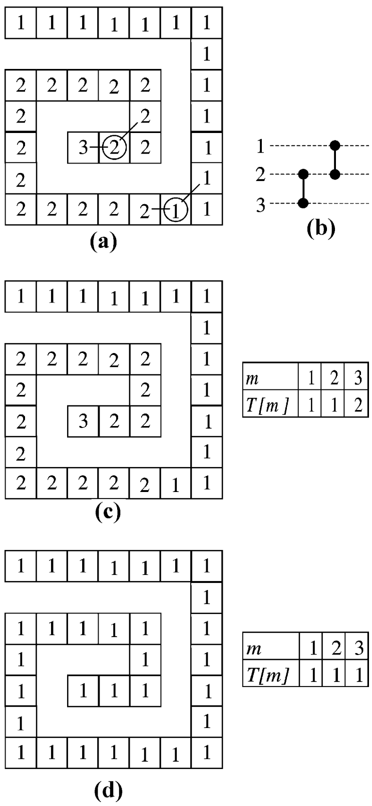


Fig. 5. Labeling of a spiral connected component using the proposed algorithm (connection pattern II): (a) connectivity; (b) order of occurrences of the junctions; (c) after the first scan (forward scan); (d) after the second scan (backward scan).

Fig. 5 shows the result of labeling the spiral-like connected component shown in Fig. 3. On this connected component, the provisional labels 2 and 3 meet before 1 and 2 meet. As compared with the stair-like connected component, the order of occurrences of the junctions is reversed. Therefore, the label connection table is completed after the second scan; i.e., both the operations in the backward raster scan order and the labeling are completed simultaneously. The labeling of the spiral-like connected component having N turns is also completed by two scans.

Thus, in the proposed algorithm, the labeling of the illustrated connected components can be completed by performing a definite number of scans, while the number depends on the geometrical complexity of image for the conventional algorithms.

3.3.2. Consideration of the number of scans

The labeling of the connected component shown in Fig. 5 is completed at the same time as the label connection table is completed. However, to complete the labeling of arbitrary connected components, one more scan after the completion of the table is required, as shown in Fig. 6.

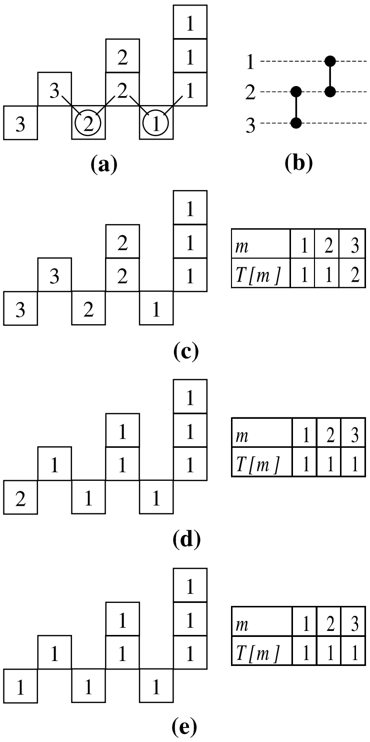


Fig. 6. Labeling of another connected component using the proposed algorithm (connection pattern II): (a) connectivity; (b) order of occurrences of the junctions; (c) after the first scan (forward scan); (d) after the second scan (backward scan); (e) after the third scan (forward scan).

Giving consideration to the order of occurrences of the junctions, the connected component composed of three provisional labels connected by two junctions constitutes a primary orderly label connection. We can make six binary permutations with three junctions consisting of three pairs of provisional labels 1–2, 1–3, and 2–3: the number of permutations of three junctions taken two at a time, ${}_3P_2$, is six. Possible six types of connected components in the primary orderly label connections are shown in Fig. 7. The results of labeling the first two of them, the connection patterns I and II, are shown in Figs. 4 and 5. Fig. 8 illustrates the results of labeling the other four types of connected components. The label connection tables for the patterns III and IV are completed after the forward scan and the backward scan, respectively, as shown in the figure. The label connection tables for the patterns V and VI are completed after either the forward scan or the backward scan. Therefore, six patterns can be classified into three classes: (i) the connectivity of the patterns I and III can be

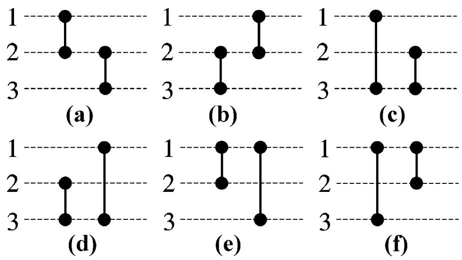


Fig. 7. Primary orderly label connections. From (a) to (f): connection patterns I, II, III, IV, V, and VI.

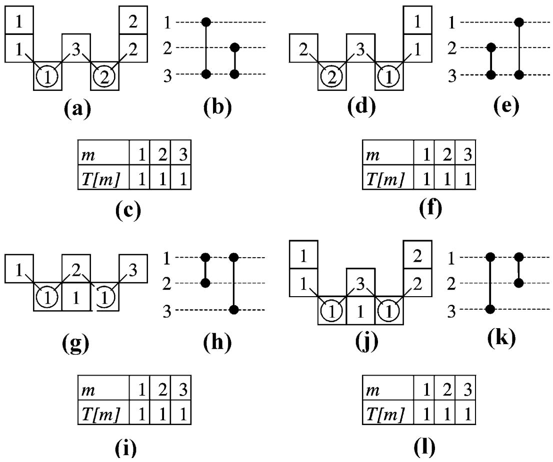
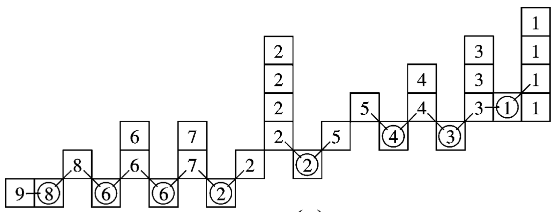


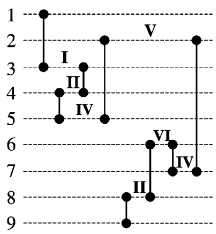
Fig. 8. Labeling of four connected components using the proposed algorithm (connection patterns III, IV, V, and VI): (a), (d), (g), and (j) Connectivity of the connection patterns III, IV, V, and VI, respectively; (b), (e), (h), and (k) order of occurrences of the junctions; (c) after the forward scan; (f) after the backward scan; (i) and (l) after the forward scan or the backward scan.

resolved after the forward scan; (ii) that of the patterns II and IV can be done after the backward scan; (iii) that of the patterns V and VI can be done after either the forward scan or the backward scan.

Any arbitrary connected component can be represented by the combination of these six primary patterns. One forward scan can resolve the connectivity of the patterns combined with the patterns I, III, V, and VI: on the patterns I or III, since one of the provisional labels at the ensuing junction (2 on the pattern I; 3 on the pattern III) has already met the minimum label, all labels of the label connection table, related to this pattern, should become the minimum label even in the case of the combination of these patterns. On the patterns V or VI, since the provisional labels, 2 and 3, meet directly the minimum label 1, all labels of the label connection table should become the minimum label. In the manner similar to the above consideration, the connectivity of the patterns combined with the patterns II, IV, V, and VI can be resolved after the backward scan.



(a)



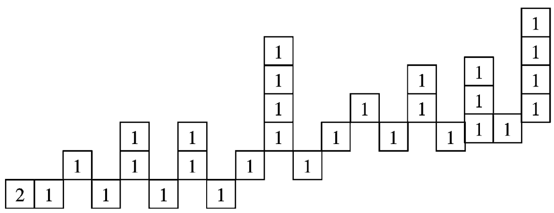
(b)

m	1	2	3	4	5	6	7	8	9
$T[m]$	1	2	1	1	2	6	2	6	8

(c)

m	1	2	3	4	5	6	7	8	9
$T[m]$	1	1	1	1	1	2	2	2	6

(d)



m	1	2	3	4	5	6	7	8	9
$T[m]$	1	1	1	1	1	1	2	2	6

(e)

Fig. 9. Sample of the connected component, the labeling of which is completed by four scans: (a) connectivity; (b) order of occurrences of the junctions; (c) after the first scan (forward scan); (d) after the second scan (backward scan); (e) after the third scan (forward scan).

However, the connectivity of the multiplex combination of these patterns cannot be always resolved after the backward scan. Such a pattern is illustrated in Fig. 9. This connected component consists of the connection patterns I, II, IV, V, and VI. During the first scan (forward scan), the initial provisional labels 1–9 are assigned to the pixels. After the second scan (backward scan), they are replaced by the provisional labels 1, 2, and 6, i.e., the provisional labels 3, 4, 5, 7, 8, and 9 do not exist any more on the connected component. Fig. 9d shows the label connection table after the second scan (backward scan). At this time, the connectivity of the provisional label 6 has not been resolved yet. After the third scan (forward scan), the label connection table is completed; the provisional label 2, however, remains on this connected component, as shown in Fig. 9e. Therefore, one more scan is required to complete labeling this connected component.

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

4. Experiments

4.1. Maximum number of scans

4.1.1. Results

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: 1000) with uniform random noise were transformed into 41 binary images by varying the threshold from 1 to 1000 with the step of 25. Fifty 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 2050 images was 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 any arbitrary images is completed by no more than four passes.

4.1.2. Analysis

The image, the labeling of which was completed by four scans, is analyzed. For convenience of analysis, the connected component in the image is reduced. The reduced connected component is shown in Fig. 9. This connected component consists of the connection patterns I, II, IV, V, and VI. As described in the previous section, four scans are required to complete labeling of this connected component. This case is very rare: only five images required four scans; 2045 images required no more than three scans.

4.2. Characteristics of algorithms

4.2.1. Maximum execution time

The maximum execution times of the conventional and the proposed algorithms are estimated on the basis of experimental measurements, and the comparative evaluation of the execution time is performed. We selected Haralick's algorithm in [9] from the class (A), Shirai's algorithm in [37], and Lumia's algorithm in [18] from the class (B) (here referred to as the conventional algorithms A, B, and C, respectively), which are well-known representatives, as the targets for comparison. The conventional algorithms have some improved versions. We compare with the original ones and clarify their characteristics, because we can estimate the efficiencies and characteristics of the improved versions from the information on the original ones.

The conventional algorithm B takes two passes: during the first pass, the label equivalences are stored in a one-dimensional table. Then, the label equivalences are resolved by using searching the table. During the second pass, the provisional labels are replaced by the final labels using the table. The execution time depends on the number of initial provisional labels. The conventional algorithm C takes two passes in top-down and bottom-up directions. The label equivalences are stored in a small local table at each row of an image. Searching the table and replacing the provisional labels are performed at every row. A small table whose size is enough to handle the provisional labels in two rows is required. It is the smallest of all algorithms in the class (B). This algorithm is advantageous in terms of memory efficiency. The execution time of this algorithm is dominated by that for searching the table. The execution time of the conventional algorithm A depends on the geometrical complexity of the connected components, as shown in the previous section.

The test images in the previous subsection were used in this experiment. The results of comparison of the CPU execution time on a workstation (UltraSPARC-II 300 MHz made by Sun Microsystems) are shown in Fig. 10. It is shown that the execution time of the proposed algorithm depends on the number of pixels in the connected components. The maximum execution times of the conventional algorithms A, B, and C were 4.33, 8.88, and 5.36 s, respectively. In contrast, that of the proposed algorithm was 0.18 s. The evaluation should include the worst case one, because the test images include the extremely complex connected components. Therefore, 0.18 s must be the maximum execution time of the proposed algorithm for this size of images on this computer.

Fig. 10b shows the execution time against the number of initial provisional labels. The figure shows that the execution time of the conventional algorithm B depends on the number of initial provisional labels. The evaluation should include the worst case execution time. Therefore, 8.88 s should be the maximum execution time of the conventional algorithm B for this size of images on this computer. The execution time of the conventional algorithm C is dominated by the execution time for searching the table. Since the test images include the connected components having extremely complex connectivity, 5.36 s should be the maximum execution time of the conventional algorithm C for this size of images. When the whole image is composed of the stair-like connected components shown in Fig. 2 the conventional algorithm A takes the

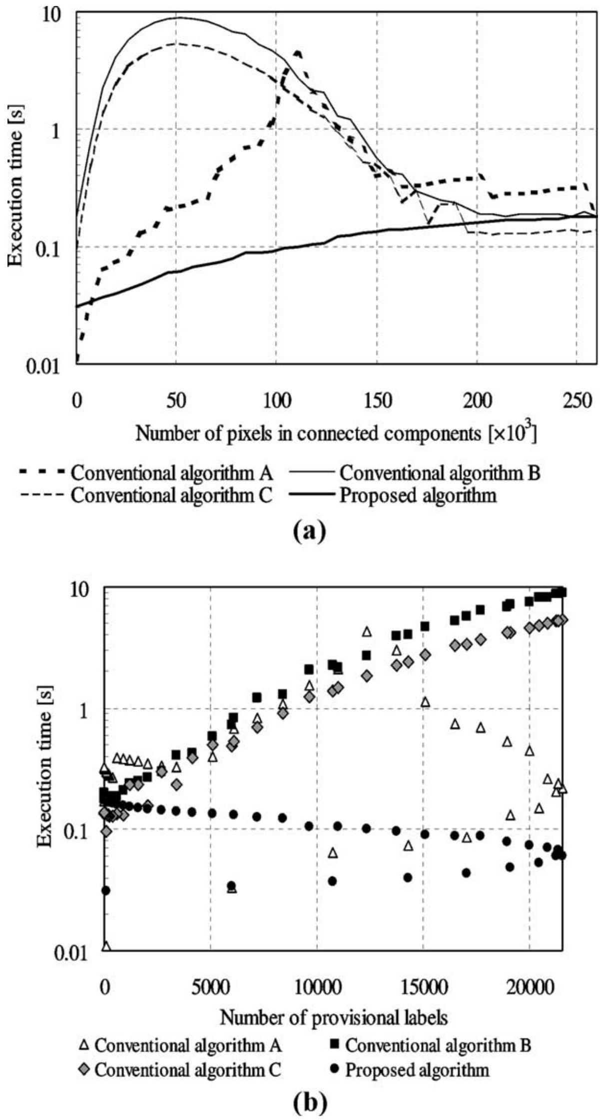


Fig. 10. Comparison of the execution times: (a) execution time against the number of pixels in connected components; (b) execution time against the number of initial provisional labels.

theoretical upper bound of the execution time. As a result of the measurement, it was 15.53 s. Therefore, it is the maximum execution time of the conventional algorithm A for this size of images. For the same image, the execution times of the conventional algorithms B and C and the proposed algorithm were 0.20, 0.24, and 0.11 s, respectively.

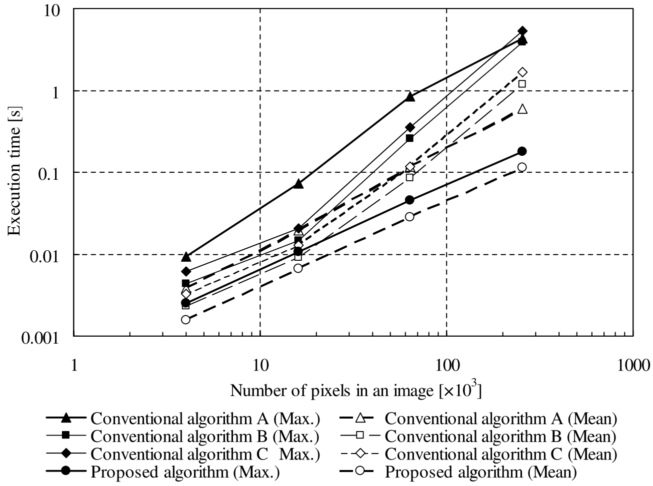


Fig. 11. Comparison of the execution times against the number of pixels in an image.

Table 1
Results of calculation of the parameters for Eq. (11)

Algorithm	a	b
Conventional algorithm A	4×10^{-10}	1.50
Conventional algorithm B	4×10^{-8}	1.90
Conventional algorithm C	4×10^{-9}	1.68
Proposed algorithm	5×10^{-7}	1.03

4.2.2. Characteristic against image size

By varying the size of images (sizes: 64×64, 128×128, 256×256, and 512×512 pixels), the execution times with 164 binary images were measured. The results are shown in Fig. 11. This figure shows that the execution time increases in proportion to the power of the number of pixels. In order to make quantitative the characteristics against the number of pixels I, the maximum execution time $t_{\max}$ was approximated by the following function using the least-square-fitting:

$$t_{\max} = aI^b, \tag{11}$$

where a and b denote parameters. The results are shown in Table 1. The execution times of the conventional algorithms A, B, and C and the proposed algorithm are directly proportional to the power of 1.5, 1.9, 1.7, 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.

4.3. Evaluation with various images

Fifty 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,

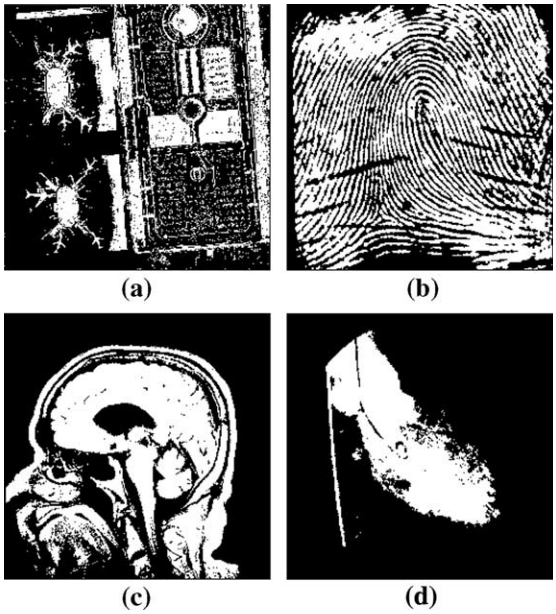


Fig. 12. Sample test images: (a) LAX from the USC image database; (b) FINGERPRINT from the SIDBA; (c) human head taken on MRI system; (d) human heart taken on DSA system.

Table 2
Comparison of the execution times

Algorithm	Execution time (s)		
	Max.	Mean	Min.
Conventional algorithm A	1.144	0.299	0.059
Conventional algorithm B	0.451	0.206	0.144
Conventional algorithm C	0.338	0.142	0.095
Proposed algorithm	0.131	0.076	0.038

which are 512×512 pixels in size, were used for evaluation. The images were transformed into binary images by using Otsu's threshold selection method in [27]. The samples of the natural and medical images are shown in Fig. 12. The results of evaluation are shown in Table 2. The proposed algorithm has completed labeling all images by no more than three scans. It has been demonstrated that the proposed algorithm is faster.

4.4. Comparisons with the improved versions

We compare the proposed algorithm with the improved versions of the conventional algorithms B and C.

4.4.1. Conventional algorithm B

The improved version of the conventional algorithm B (here referred to as the conventional algorithm B*) has been reported in [7,8]. The conventional algorithm B* is a two-pass algorithm. During the first pass, the label equivalences are stored in a two-dimensional bit table using a method for removing duplicate equivalences, called the improved triangular memory method. In the method, the image is divided into sub-images in order to reduce the size of the table and the execution time for making the table. Then, the label equivalences are resolved by using a label classification process. During the second pass, the final labels are assigned to the image using the resolved label equivalences. Before the improved triangular memory method was introduced, a one-dimensional table was used for storing the label equivalences. This was realized in software and not suitable for video-rate processing. By introducing the improved triangular memory method, a video-rate labeling has been actually implemented in a commercial device [16].

Before the improved triangular memory method was introduced, a one-dimensional table was used in software. This was not suitable for video-rate processing. By introducing the triangular table, video-rate labeling was in a commercial device.

We conducted theoretical comparisons with the conventional algorithm B*. For any labeling algorithms, the process assigning the provisional labels during the first pass and the process assigning the final labels during the final pass are always required. For convenience, we calculated the execution time excluding these two processes. According to [7,8,16], the maximum execution time of the conventional algorithm B* can be represented by

$$t_{\mathrm{C}} = (3L_0 \cdot \ell_d)t_{\mathrm{RW}} + (6L_0 - 12)t_{\mathrm{RW}} + (6L_0 - 12)N_S \cdot t_{\mathrm{MC}}, \tag{12}$$

where L_0 denotes the number of initial provisional labels, ℓ_d the upper limit on the number of initial provisional labels in a sub-image, t_{RW} the execution time for reading from or writing on a table memory, t_{MC} the time of a machine cycle of a microcomputer used for processing the label classification process, and N_S the number of machine cycles needed for one search in the label classification process. By substituting $\ell_d = 256$ and $N_S = 27$, according to [7,8,16], and assuming that $t_{\mathrm{MC}} = 2t_{\mathrm{RW}}$, this is the ordinary case, the above equation can be represented by the following equation:

$$t_{\mathrm{C}} = (1098L_0 - 660)t_{\mathrm{RW}}. \tag{13}$$

Since the conventional algorithm B* is stated in [7,8,16] as the algorithm for four-connected components, the execution time of the proposed algorithm for four-connected components should be calculated for fair comparison. In the proposed algorithm, the masks in Fig. 13 in Appendix A are used in this case. Only the first, second, third, sixth, and seventh columns in Tables 3 and 4 in Appendix A are used. For labeling of four-connected components, the label connection occurs only in the condition indicated at the seventh column. Those points are the difference between the algorithm for four-connected components and that for eight-connected components. Although the number of conditions decreases to five, the execution time does not change so much.

Assuming each condition occurs with the same probability, the following execution time is required for the first scan:

$$t_{P1} = \frac{6}{4}L_P \cdot t_{RW} + \frac{2}{4}L_P \cdot t_{MC} = \frac{10}{4}L_P \cdot t_{RW}, \tag{14}$$

where L_P denotes the number of pixels in all connected components in an image. The following execution time is required for each scan after the first scan:

$$t_{P2} = \frac{12}{4}L_P \cdot t_{RW} + 2L_P \cdot t_{MC} = \frac{28}{4}L_P \cdot t_{RW}. \tag{15}$$

Since the maximum number of scans is four, the total execution time of the proposed algorithm becomes the following equations:

$$t_P = t_{P1} + 3t_{P2} - 2L_P \cdot t_{RW} = \frac{86}{4}L_P \cdot t_{RW}. \tag{16}$$

The relationship between the number of initial provisional labels and the number of pixels in connected components can be represented by

$$L_P = N_{P0} \cdot L_0, \tag{17}$$

where N_{P0} denotes a parameter. By comparing among Eqs. (13), (16), and (17), and considering the fact that L_0 is generally more than several hundreds, it is shown that the proposed algorithm is advantageous in time efficiency when the following condition is fulfilled:

$$N_{P0} \leqslant 51.1 - 30.7/L_0 \cong 51. \tag{18}$$

In other words, the conventional algorithm B* is advantageous in the case of much simpler connected components where L_P is greater than 51 times of L_0. Since the mean for N_{P0}'s of all test images is about 15, this case is rare. Therefore, the proposed algorithm is superior to the conventional algorithm B* for most images.

We estimate the average case execution time by using real values measured from the test images. The mean values of L_0 and L_P among all test images were 1994 and 48,723, respectively. Substituting these values into Eqs. (13) and (16), the average case execution time of the conventional algorithm B* and that of the proposed algorithm become the following equations:

$$t_C = 2.19 \times 10^6 \cdot t_{RW}, \tag{19}$$

$$t_P = 1.05 \times 10^6 \cdot t_{RW}. \tag{20}$$

Therefore, it is shown that the proposed algorithm is approximately two times faster on average than that of the conventional algorithm B*.

Since we should design on the assumption of the worst case, we estimate the worst case execution time. The maximum value of L_0 and that of L_P among all test images were 21,555 and 260,100, respectively. Substituting these values into Eqs. (13) and (16), the worst case execution time becomes the following equations:

$$t_{C} = 23.7 \times 10^6 \cdot t_{RW}, \tag{21}$$

$$t_{P} = 5.6 \times 10^6 \cdot t_{RW}. \tag{22}$$

Therefore, it is shown that the worst case execution time of the proposed algorithm is approximately four times smaller than that of the conventional algorithm B*.

4.4.2. Conventional algorithm C

The improved version of the conventional algorithm C (here referred to as the conventional algorithm C*) has been reported in [10]. The conventional algorithm C* employs run-length encoding. Since the process assigning a provisional label to each pixel and the process reassigning labels at each row become more efficient by using the run-length encoding, the conventional algorithm C* should be faster than the conventional algorithm C.

We actually measured the execution time for performing the two processings that can be efficient by using the run-length encoding. As a result, the mean for execution times and the maximum execution time for the test images (size 512×512 pixels) were 0.21 and 0.23 s, respectively. While the conventional algorithm C took 1.67 and 5.36 s, the proposed algorithm took only 0.11 and 0.18 s, respectively. Thus, the execution time of the processings that can be efficient by using the run-length encoding is not dominated in the whole processing. Therefore, the proposed algorithm would be faster than the conventional algorithm C*.

In addition, we can employ the union-find algorithm with path compression as the search algorithm of the conventional algorithm C*. According to [41,42], the worst case number of search operations is $O(N_{O}^{2})$ where N_{O} denotes the number of union-find operations required for a target processing. If we use the union-find algorithm with path compression, the worst case number of search operations becomes $O(N_{O} \log N_{O})$. Therefore, the conventional algorithm C* using the union-find algorithm with path compression should be much faster than the conventional algorithm C*.

5. Concluding remarks

This paper has presented a fast algorithm for labeling connected components in binary images. The proposed algorithm is a quite simple: it is based on sequential local operations with one-dimensional table in the raster scan directions. This is suitable for implementation in hardware. Through comparative evaluations with the conventional algorithms, it has been demonstrated that the proposed algorithm takes less 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 2314 images is completed by no more than four scans, the mathematical proof of the maximum number of scans remains. We will perform experimental comparisons with the improved versions of the conventional algorithms. We plan in the near feature to implement the proposed algorithm in hardware.

Acknowledgments

The authors thank Prof. Yoshiaki Shirai of Osaka University, Prof. Shigeru Okabayashi, and Prof. Shin Yamamoto of Meijo University, Dr. Kouji Ueda of Nagoya Electric Works, and Dr. Hiroki Yoshioka of Aichi Prefectural University for their valuable suggestions. The authors also thank anonymous reviewers for their valuable comments. This work was supported in part by the Ministry of Education, Science, Sports and Culture of Japan under grant-in-aid for encouragement of young scientists.

Appendix A

Giving consideration to the movement of the mask, the provisional label at the position a in Fig. 1 propagates the positions b and d; the provisional label at the position b in Fig. 1 propagates the positions c and d. Therefore, the operations of the proposed algorithm can be represented by the logical table in Tables 3 and 4. 0 and 1 in the tables indicate F_B and F_O, respectively. * indicates 0 or 1, NOP non-operation, $\max(\cdot)$ an operator calculating the maximum value, and $\max 2(\cdot)$ an operator calculating the second maximum value. $\max T(\cdot)$ and $\max T2(\cdot)$ are defined as the following equations:

$$\max T(p,q) = \begin{cases} p & \text{if } \max(T[p],T[q]) = T[p], \\ q & \text{if } \max(T[p],T[q]) = T[q], \end{cases} \tag{A.1}$$

$$\max T(p,q,r) = \begin{cases} p & \text{if } \max(T[p],T[q],T[r]) = T[p], \\ q & \text{if } \max(T[p],T[q],T[r]) = T[q], \\ r & \text{if } \max(T[p],T[q],T[r]) = T[r], \end{cases} \tag{A.2}$$

Table 3
Logic table for the first scan

e	a	b	c	d	$g(x,y)$	Update of the label connection table
0	*	*	*	*	F_B	NOP
1	0	0	0	0	m	$T[m] = m$
1	0	0	0	1	$T[d]$	NOP
1	0	0	1	0	$T[c]$	NOP
1	0	0	1	1	$\min(T[c],T[d])$	$T[\max T(c,d)] = g(x,y)$
1	0	1	0	0	$T[b]$	NOP
1	0	1	0	1	$T[d]$	NOP
1	0	1	1	0	$T[c]$	NOP
1	0	1	1	1	$\min(T[c],T[d])$	$T[\max T(c,d)] = g(x,y)$
1	1	0	0	0	$T[a]$	NOP
1	1	0	0	1	$T[d]$	NOP
1	1	0	1	0	$\min(T[a],T[c])$	$T[\max T(a,c)] = g(x,y)$
1	1	0	1	1	$\min(T[c],T[d])$	$T[\max T(c,d)] = g(x,y)$
1	1	1	0	0	$T[b]$	NOP
1	1	1	0	1	$T[d]$	NOP
1	1	1	1	0	$T[c]$	NOP
1	1	1	1	1	$\min(T[c],T[d])$	$T[\max T(c,d)] = g(x,y)$

Table 4
Logic table for the scans after the first scan

e	a	b	c	d	$g(x,y)$	Update of the label connection table
0	*	*	*	*	F_B	NOP
1	0	0	0	0	$T[e]$	NOP
1	0	0	0	1	$\min(T[e], T[d])$	$T[\max T(e,d)] = g(x,y)$
1	0	0	1	0	$\min(T[e], T[c])$	$T[\max T(e,c)] = g(x,y)$
1	0	0	1	1	$\min(T[e], T[c], T[d])$	$T[\max T(e,c,d)] = g(x,y)$, $T[\max T2(e,c,d)] = g(x,y)$
1	0	1	0	0	$\min(T[e], T[b])$	$T[\max T(e,b)] = g(x,y)$
1	0	1	0	1	$\min(T[e], T[d])$	$T[\max T(e,d)] = g(x,y)$
1	0	1	1	0	$\min(T[e], T[c])$	$T[\max T(e,c)] = g(x,y)$
1	0	1	1	1	$\min(T[e], T[c], T[d])$	$T[\max T(e,c,d)] = g(x,y)$, $T[\max T2(e,c,d)] = g(x,y)$
1	1	0	0	0	$\min(T[e], T[a])$	$T[\max T(e,a)] = g(x,y)$
1	1	0	0	1	$\min(T[e], T[d])$	$T[\max T(e,d)] = g(x,y)$
1	1	0	1	0	$\min(T[e], T[a], T[c])$	$T[\max T(e,a,c)] = g(x,y)$, $T[\max T2(e,a,c)] = g(x,y)$
1	1	0	1	1	$\min(T[e], T[c], T[d])$	$T[\max T(e,c,d)] = g(x,y)$, $T[\max T2(e,c,d)] = g(x,y)$
1	1	1	0	0	$\min(T[e], T[b])$	$T[\max T(e,b)] = g(x,y)$
1	1	1	0	1	$\min(T[e], T[d])$	$T[\max T(e,d)] = g(x,y)$
1	1	1	1	0	$\min(T[e], T[c])$	$T[\max T(e,c)] = g(x,y)$
1	1	1	1	1	$\min(T[e], T[c], T[d])$	$T[\max T(e,c,d)] = g(x,y)$, $T[\max T2(e,c,d)] = g(x,y)$

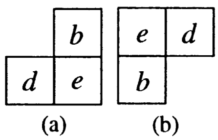


Fig. 13. Masks for the labeling of four-connected components: (a) forward scan; (b) backward scan.

$$\max T2(p,q,r) = \begin{cases} p & \text{if } \max 2(T[p], T[q], T[r]) = T[p], \\ q & \text{if } \max 2(T[p], T[q], T[r]) = T[q], \\ r & \text{if } \max 2(T[p], T[q], T[r]) = T[r]. \end{cases} \tag{A.3}$$

The efficient implementation in hardware can be performed using these logic tables.

Appendix B

The labeling of four-connected components can be performed by using the mask shown in Fig. 13.

References

[1] H.M. Alnuweirim, V.K. Prasanna, Parallel architectures and algorithms for image component labeling, IEEE Trans. Pattern Anal. Mach. Intell. 14 (10) (1992) 1024–1034.
[2] P. Bhattacharya, Connected component labeling for binary images on a reconfigurable mesh architectures, J. Syst. Architect. 42 (4) (1996) 309–313.
[3] A. Choudhary, R. Thakur, Connected component labeling on coarse grain parallel computers: an experimental study, J. Parallel Distrib. Comput. 20 (1994) 78–83.
[4] M.B. Dillencourt, H. Samet, M. Tamminen, A general approach to connected-component labeling for arbitrary image representations, J. ACM 39 (2) (1992) 253–280.
[5] C. Fiorio, J. Gustedt, Two linear time union-find strategies for image processing, Theoret. Comput. Sci. 154 (2) (1996) 165–181.
[6] I. Gargantini, Separation of connected component using linear quad- and oct-trees, in: Proc. 12th Conf. Numerical Mathematics and Computation, vol. 37, University of Manitoba, Winnipeg, 1982, pp. 257–276.
[7] T. Goto, Y. Ohta, M. Yoshida, Y. Shirai, High speed algorithm for component labeling, Trans. IEICE J72-D-II (2) (1989) 247–255, in Japanese.
[8] T. Gotoh, Y. Ohta, M. Yoshida, Y. Shirai, Component labeling algorithm for video rate processing, in: Proc. SPIE, Advances in Image Processing 804 (1987) 217–224.
[9] R.M. Haralick, Some neighborhood operations, in: Real Time/Parallel Computing Image Analysis, Plenum Press, New York, 1981, pp. 11–35.
[10] R.M. Haralick, L.G. Shapiro, in: Computer and Robot Vision, vol. I, Addison-Wesley, Reading, MA, 1992, pp. 28–48.
[11] A. Hashizume, R. Suzuki, H. Yokouchi, et al., An algorithm of automated RBC classification and its evaluation, BioMed. Engrg. 28 (1) (1990) 25–32.
[12] T. Hattori, A high-speed pipeline processor for regional labeling based on a new algorithm, in: Proc. Int. Conf. Pattern Recognition (NJ), 1990, pp. 494–496.
[13] J. Hecquard, R. Acharya, Connected component labeling with linear octree, Pattern Recog. 24 (6) (1991) 515–531.
[14] D.S. Hirschberg, A.K. Chandra, D.V. Sarwate, Computing connected components on parallel computers, Commun. ACM 22 (8) (1979) 461–464.
[15] C.L. Jackins, S.L. Tanimoto, Octrees and their use in representing 3D objects, Comput. Graphics Image Process. 14 (3) (1980) 249–270.
[16] M. Komeichi, Y. Ohta, T. Gotoh, T. Mima, M. Yoshida, Video-rate labeling processor, in: Proc. SPIE, Image Processing II 1027 (1988) 69–76.
[17] R. Lumia, A new three-dimensional connected components algorithm, Comput. Vision, Graphics, and Image Process. 23 (2) (1983) 207–217.
[18] R. Lumia, L. Shapiro, O. Zungia, A new connected components algorithm for virtual memory computers, Comput. Vision, Graphics, and Image Process. 22 (2) (1983) 287–300.
[19] M. Manohar, H.K. Ramapriyan, Connected component labeling of binary images on a mesh connected massively parallel processor, Comput. Vision, Graphics, and Image Process. 45 (2) (1989) 133–149.
[20] K. Mehlhorn, Data Structures and Algorithm 1: Sorting and Searching, Springer, Berlin, 1984.
[21] M. Minsky, S. Papert, Perceptron, MIT Press, Cambridge, 1969.
[22] S. Naoi, High-speed labeling method using adaptive variable window size for character shape feature, in: IEEE Asian Conference on Computer Vision, vol. 1, 1995, pp. 408–411.
[23] D. Nassimi, S. Sahani, Finding connected components and connected ones on a mesh connected parallel compute, SIAM J. Comput. 9 (4) (1980) 744–757.
[24] C.J. Nicol, Design of a connected component labelling chip for real time image processing, in: Proc. IEEE Asia-Pacific Conf. Circuits and Systems, Sydney, Australia, 1992, pp. 142–147.
[25] C.J. Nicol, A systolic approach for real time connected component labeling, Comput. Vision and Image Understanding 61 (1) (1995) 17–31.
[26] S. Olariu, J.L. Schwing, J. Zhang, Fast component labelling and convex hull computation on reconfigurable meshes, Image and Vision Comput. 11 (7) (1993) 447–455.

[27] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Systems Man Cybernet. 9 (1979) 62–66.
[28] A. Rosenfeld, Connectivity in digital pictures, J. ACM 17 (1) (1970) 146–160.
[29] A. Rosenfeld, A.C. Kak, Digital Picture Processing, vol. 2, second ed., Academic Press, San Diego, CA, 1982.
[30] A. Rosenfeld, J.L. Pfalts, Sequential operations in digital picture processing, J. ACM 13 (4) (1966) 471–494.
[31] H. Samet, Connected component labeling using quadtrees, J. ACM 28 (3) (1981) 487–501.
[32] H. Samet, The quadtree and related hierarchical data structures, Comput. Surveys 16 (2) (1984).
[33] H. Samet, Computing geometric properties of images represented by linear quadtrees, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-7 (2) (1985) 229–240.
[34] H. Samet, M. Taminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-10 (4) (1988) 579–586.
[35] H. Samet, M. Tamminen, An improved approach to connected component labeling of images, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, Florida, 1986, pp. 312–318.
[36] Y. Schiloach, U. Vishkin, An $o(\log n)$ parallel connectivity algorithm, J. Algorithms 3 (1982) 57–67.
[37] Y. Shirai, Labeling connected regions, in: Three-Dimensional Computer Vision, Springer, Berlin, 1987, pp. 86–89.
[38] S.N. Srihari, Hierarchical representations for serial section images, in: Proc. 5th Int. Conf. Pattern Recognition, 1980, pp. 1075–1080.
[39] K. Suzuki, I. Horiba, N. Sugie, Fast connected-component labeling based on sequential local operations in the course of forward raster scan followed by backward raster scan, in: Proc. 15th Int. Conf. Pattern Recognition, Barcelona, Spain, vol. 2, 2000, pp. 434–437.
[40] M. Taminen, H. Samet, Efficient octree conversion by connectivity labeling, in: Proc. SIGGRAPH 84 Conf., Minneapolis, MN, 1984, pp. 43–51 ACM.
[41] R.E. Tarjan, Efficiency of a good but not linear set union algorithm, J. ACM 22 (2) (1975) 215–225.
[42] R.E. Tarjan, J.V. Leeuwen, Worst-case analysis of set union algorithms, J. ACM 31 (2) (1984) 245–281.
[43] L.W. Tucker, Labeking connected components on a massively parallel tree machine, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, Florida, 1986, pp. 124–129.
[44] X.D. Yang, Design of fast connected components hardware, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Ann Arbor, MI, 1988, pp. 937–944.