COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 30, 32–46 (1985)

Topological Structural Analysis of Digitized Binary Images by Border Following

SATOSHI SUZUKI*

Graduate School of Electronic Science and Technology, Shizuoka University, Hamamatsu 432, Japan

AND

KEIICHI ABE†

Department of Computer Science, Shizuoka University, Hamamatsu 432, Japan

Received December 16, 1983

Two border following algorithms are proposed for the topological analysis of digitized binary images. The first one determines the surroundness relations among the borders of a binary image. Since the outer borders and the hole borders have a one-to-one correspondence to the connected components of 1-pixels and to the holes, respectively, the proposed algorithm yields a representation of a binary image, from which one can extract some sort of features without reconstructing the image. The second algorithm, which is a modified version of the first, follows only the outermost borders (i.e., the outer borders which are not surrounded by holes). These algorithms can be effectively used in component counting, shrinking, and topological structural analysis of binary images, when a sequential digital computer is used. © 1985 Academic Press, Inc.

1. INTRODUCTION

Border following is one of the fundamental techniques in the processing of digitized binary images. It derives a sequence of the coordinates or the chain codes from the border between a connected component of 1-pixels (1-component) and a connected component of 0-pixels (background or hole). The border following technique has been studied deeply, because it has a large variety of applications, including picture recognition, picture analysis, and image data compression [1–10].

The purpose of this paper is to propose border following algorithms with a sort of topological analysis capability. If one wants to convert a binary picture into the border representation, then he can extract the topological structure of the image with little additional effort by using the algorithms presented here. The information to be extracted is the surroundness relation among the two types of borders: the outer borders and the hole borders. Since there exists one-to-one correspondence between an outer border and a 1-component, and between a hole border and a 0-component, the topological structure of a given binary image can be determined.

Several works have been reported on the topological structural analysis of binary pictures using raster scan and labeling [1, 10, 11]. An alternative for such analysis is to use border following. If an image processing system utilizes the border following for some purpose and at the same time needs to analyze the topological structure of

*Present Affiliation: Yokosuka Electrical Communication Laboratory, Nippon Telegraph and Telephone Public Corporation.

†Department of Computer Science, Faculty of Engineering, Shizuoka University, 3-5-1 Johoku, Hamamatsu 432, Japan.

0734-189X/85 $3.00

an input image, then this approach would be attractive. Few studies, however, have been devoted to this subject. The existing methods [6, 7] are rather complicated and time consuming, so that they do not seem advantageous over other structural analysis methods [1, 10, 11] which do not use border following.

In this paper, we first present an algorithm which can extract the topological structure of a given binary image. This algorithm is an extended version of the border following algorithm [1] which discriminates between outer borders and hole borders. The extensions are: (1) to put a unique mark on each border rather than to adopt the same marking procedure for every border (border labeling); and (2) to add a procedure for obtaining the parent border of the currently followed border (see Definition 4). With this algorithm we can extract the surroundness relation among the borders, which corresponds to the surroundness relation among the connected components. If a binary image is stored in the form of the borders and the surroundness relation is extracted by this algorithm, some simple image processing can be done without restoring the original image. Thus the method offers an effective way of storing binary image data.

Next we show a modified version of the first algorithm which follows only the outermost borders of a binary image (i.e., the outer borders which are not surrounded by holes). When we want to ignore the 1-components which are surrounded by other 1-components, this modified algorithm gives us a quick, sequential method of counting the 1-components or shrinking each 1-component to one point.

2. BASIC CONCEPT AND NOTATIONS

In this paper only digital binary pictures sampled at points of rectangular grids are considered. Though we will follow the general terminology and notations such as in [1], we would like to define and clarify some concepts and notations not so widely established.

The uppermost row, the lowermost row, the leftmost column, and the rightmost column of a picture compose its frame. Pixels with densities 0 and 1 are called the 0-pixel and the 1-pixel, respectively. Without loss of generality, we assume that 0-pixels fill the frame of a binary picture. We assume also that we can assign any integer value to a pixel during the processing. The pixel located in the ith row and the jth column is represented by the row number and the column number (i, j). We adopt the following coordinate system: the row number i increases from top to bottom; the column number j from left to right. A picture having the density value f_{ij} at a pixel (i, j) is denoted by $F = \{f_{ij}\}$.

A 1-component and a 0-component are the connected components of 1-pixels and of 0-pixels, respectively. If a 0-component S contains the frame of the picture, we call S the background; otherwise, a hole.

It is well known that in order to avoid a topological contradiction 0-pixels must be regarded as 8- (4-) connected if 1-pixels are dealt with as 4- (8-) connected. We will say "in the 4- (8-) connected case" when we deal with 1-pixels as 4- (8-) connected and 0-pixels as 8- (4-) connected. When we do not specify the type of the connectivity explicitly, we mean that the argument is valid for both types of the connectivity.

The border and the surroundness among connected components are defined as follows.

DEFINITION 1 (border point). In the 4- (8-) connected case, a 1-pixel (i, j) having a 0-pixel (p, q) in its 8- (4-) neighborhood is called a border point. It is also described as "a border point between a 1-component S_1 and a 0-components S_2," if (i, j) is a member of S_1 and (p, q) is a member of S_2.

DEFINITION 2 (surroundness among connected components). For given two connected components S_1 and S_2 in a binary picture, if there exists a pixel belonging to S_2 for any 4-path from a pixel in S_1 to a pixel on the frame, we say that S_2 surrounds S_1. If S_2 surrounds S_1 and there exists a border point between them, then S_2 is said to surround S_1 directly.

DEFINITION 3 (outer border and hole border). An outer border is defined as the set of the border points between an arbitrary 1-component and the 0-component which surrounds it directly. Similarly, we refer to the set of the border points between a hole and the 1-component which surrounds it directly as a hole border. We use the term "border" for either an outer border or a hole border. Note that the hole border is defined as a set of 1-pixels (not 0-pixels) as well as the outer border.

The following property holds for the connected components and the borders.

PROPERTY 1. For an arbitrary 1-component of a binary picture its outer border is one and unique. For any hole its hole border (the border between that hole and the 1-component which surrounds it directly) is also unique.

Proof. This property holds apparently, because any 1-component or hole is surrounded directly by one connected component.

We define the parent border and the surroundness among borders.

DEFINITION 4 (parent border). The parent border of an outer border between a 1-component S_1 and the 0-component S_2 which surrounds S_1 directly is defined as:

(1) the hole border between S_2 and the 1-component which surrounds S_2 directly, if S_2 is a hole;

(2) the frame of the picture, if S_2 is the background.

The parent border of a hole border between a hole S_3 and the 1-component S_4 which surrounds S_3 directly is defined as the outer border between S_4 and the 0-component which surrounds S_4 directly.

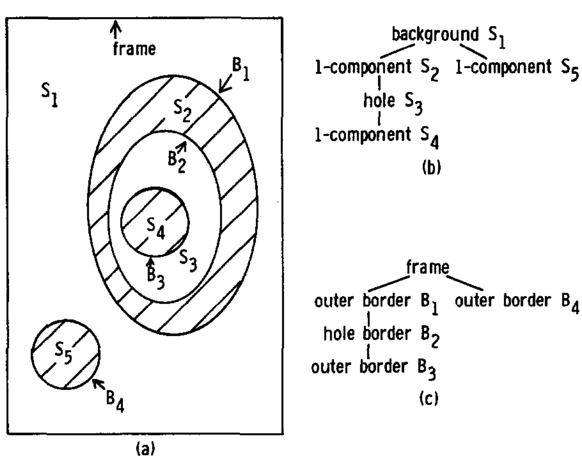

FIG. 1. Surroundness among connected components (b) and among borders (c).

DEFINITION 5 (surroundness among borders). For two given borders B_0 and B_n of a binary picture, we say that B_n surrounds B_0 is there exists a sequence of border $B_0, B_1, \ldots, B_n$ such that B_k if the parent border of B_{k-1} for all k $(1 \leq k \leq n)$.

From Definitions 2 through 5, we see that the following property holds between the surroundness among connected components and the surroundness among borders (Fig. 1).

PROPERTY 2. For any binary picture, the above two surroundness relations are isomorphic with the following mapping:

a 1-component	$\leftrightarrow$	its outer border;
a hole	$\leftrightarrow$	its hole border between the hole and the 1-component surrounding it directly;
the background	$\leftrightarrow$	the frame.

3. THE BORDER FOLLOWING ALGORITHM FOR TOPOLOGICAL ANALYSIS

We present a border following algorithm for topological structural analysis. This extracts the surroundness relation among the borders of a binary picture. First we give an informal explanation of the algorithm.

ALGORITHM 1. Scan an input binary picture with a TV raster and interrupt the raster scan when a pixel (i, j) is found which satisfies the condition for the border following starting point of either an outer border (Fig. 2a) or a hole border (Fig. 2b). If the pixel (i, j) satisfies both of the above conditions, (i, j) must be regarded as the starting point of the outer border. Assign a uniquely identifiable number to the newly found border. Let us call it the sequential number of the border and denote it by NBD.

Determine the parent border of the newly found border as follows. During the raster scan we also keep the sequential number LNBD of the (outer or hole) border encountered most recently. This memorized border should be either the parent border of the newly found border or a border which shares the common parent with the newly found border. Therefore, we can decide the sequential number of the parent border of the newly found border with the types of the two borders according to Table 1.

Follow the found border from the starting point, marking the pixels on the border. The border following scheme is the classical one [1–3]; the distinguished feature of our algorithm is the marking policy.

(a) If the current following border is between the 0-component which contains the pixel $(p, q + 1)$ and the 1-component which contains the pixel (p, q), change the value of the pixel (p, q) to $-$NBD.

(b) Otherwise, set the value of the pixel (p, q) to NBD unless (p, q) is on an already followed border.

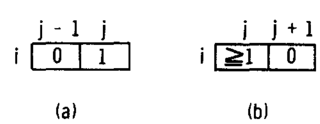

FIG. 2. The conditions of the border following staring point (i, j) for an outer border (a) and a hole border (b).

TABLE 1

Decision Rule for the Parent Border of the Newly Found Border B

Type of B \ Type of the border B' with the sequential number LNBD	Outer border	Hole border
Outer border	The parent border of the border B'	The border B'
Hole border	The border B'	The parent border of the border B'

The conditions (a) and (b) prohibit the pixel (p, q) from being the border following starting points of the already followed hole border and outer border, respectively. The positive value NBD and the negative value, $-$NBD, correspond to the labels "l" and "r" of the border following algorithm in [1], respectively.

After following and marking the entire border, resume the raster scan. When the scan reaches the lower right corner of the picture, the algorithm stops.

Let us explain the process of the algorithm by an example depicted in Fig. 3. The input binary picture is given in (a). When the raster scan reaches the circled 1-pixel, we find that it satisfies the condition for the starting point of an outer border (Fig. 2a). The sequential number 2 is assigned to the border, because the number 1 is set aside for the frame of the picture. We also find out from Table 1 that its parent border is the frame.

Then the border following starts and all the pixel values on that border are changed to either 2 or -2, as shown in (b). After following the entire border, raster scan resumes.

The next border following starts at the circled pixel in (b), which satisfies the condition for the starting point of a hole border (Fig. 2b). Note that a point having a negative value cannot be a starting point of border following. The parent border of this border (with the number 3) is found to be the outer border with the number 2. During the border following, two pixels with value 2 are changed to -3 by the marking policy (a), and two pixels with value 1 are changed to 3 by the policy (b). All the other already visited pixels (with value 2) are unchanged. The steps go on in a similar manner and we obtain the final results (e). The formal description of Algorithm 1 is given in Appendix I.

The following properties show the validity of Algorithm 1.

PROPERTY 3. The leftmost pixel (i, j) of the uppermost row of any 1-component S_1 satisfies the condition shown in Fig. 2a; the border B followed from the pixel (i, j) is an outer border (i.e., the 0-component S_2 which $(i, j - 1)$ belongs to surrounds S_1); after this outer border has been followed, the same outer border is never followed again.

Proof. See Appendix II.

PROPERTY 4. The pixel (i, j) whose right-hand neighbor $(i, j + 1)$ is the leftmost pixel of the uppermost row of any hole S_1 satisfies the condition shown in Fig. 2b;

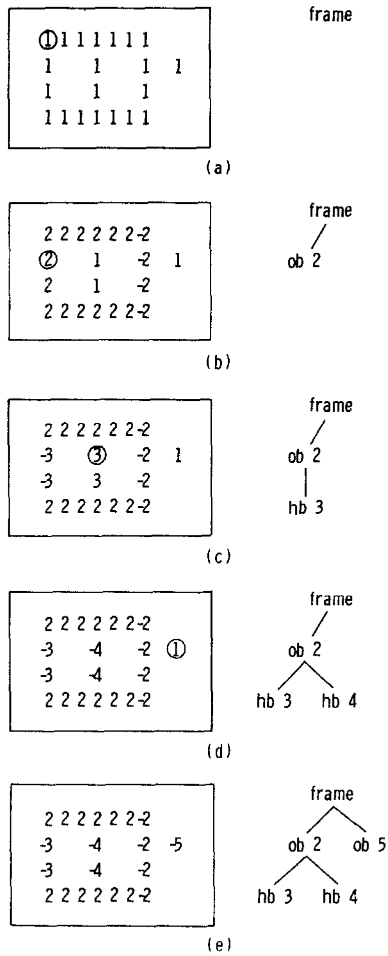

FIG. 3. An illustration of the process of Algorithm 1. The left-hand figures show the pixel values and the right-hand the extracted structures among borders (ob: outer border, hb: hole border). The circled pixels are the starting points of border following.

the border B followed from the pixel (i, j) is a hole border (i.e., the 1-component S_2 which contains (i, j) surrounds S_1); after this border has been followed, the same hole border is never followed again.

Proof. This proof is similar to that of Property 3.

PROPERTY 5. The border following starting points of Algorithm 1 have a one-to-one correspondence to the 1-components or the holes.

Proof. This property is derived immediately from Properties 3 and 4.

PROPERTY 6. When the parent border of a particular border B is to be determined, the variable LNBD contains the sequential number of either the parent border of B or a border which shares the same parent with B.

Proof. See Appendix III.

(a)

```
            10        20        30        40
 1|
 2|  22:  333-   4444444444444444+   555=  66#
 3|21:   31-    417¥           771+    51=  61#
 4|2:    3-   447¥               774+   5=   6#
 5|:    3-   41¥    8888888888?    71+   5=   #
 6|:   31-  41¥   88199999999918?   71+ 51=   #
 7|   31-  41¥   819&         991?   71+  51=
 8|   3-   4¥   81&             91?   7+   5=
 9|  31-  4¥   81&   AAAAAAAA@   91?   7+  51=
10|  3-  41¥ 81&    A1BBBBBBB1@   91?  71+  5=
11|  3-  4¥  81&  AAB        BBA@  91?  7+  5=
12| 3-   4¥  8&   A*           B@   9?  7+   5=
13| 3-  4¥  81&  A*   CCCCCC*   B@  91?  7+  5=
14| 3-  4¥  8&  A1*  C1DDDD1*   B1@  9?  7+  5=
15| 3-  4¥  8&  A*  C1      D1   B@  9?  7+  5=
16| 3-  4¥  8&  A*  C*       D   B@  9?  7+  5=
17| 3-  4¥  8&  A*  C*  EE*  D*  B@  9?  7+  5=
18| 3-  4¥  8&  A*  C*  E1*  D*  B@  9?  7+  5=
19| 3-  4¥  8&  A*  C*  EE*  D*  B@  9?  7+  5=
20| 3-  4¥  8&  A*  C*       D*  B@  9?  7+  5=
21| 3-  4¥  8&  A*  C1*     D1*  B@  9?  7+  5=
22| 3-  4¥  8&  A1*  C1DDDD1*   B1@  9?  7+  5=
23| 3-  4¥  81&  A*   CCCCCC*   B@  91?  7+  5=
24| 3-   4¥  8&   A*           B@   9?  7+   5=
25|  3-  4¥  81&  AAB*       BBA@  91?  7+  5=
26|  3-  41¥ 81&    A1BBBBBBB1@   91?  71+  5=
27|  31-  4¥   81&   AAAAAAAA@   91?   7+  51=
28|   3-   4¥   81&             91?   7+   5=
29|   31-  41¥   819&         991?   71+  51=
30|*   31-  41¥   88199999999918?   71+ 51=   *
31|*    3-   41¥    8888888888?    71+   5=   *
32|F*    3-   447¥               774+   5=   G*
33|F1*   31-    417¥           771+    51=  G1*
34| FF*   333-   4444444444444444+   555=  GG*
35|
            10        20        30        40
```

(b) Structured borders

Border no.	Outer border (1) /Hole border (0)	Coordinates of the border following starting point	Parent border no.
1	0	(1, 1)	0
2	1	(2, 3)	1
3	1	(2, 8)	1
4	1	(2, 15)	1
5	1	(2, 35)	1
6	1	(2, 41)	1
7	0	(3, 17)	4
8	1	(5, 18)	7
9	0	(7, 18)	8
10	1	(9, 19)	9
11	0	(11, 19)	10
12	1	(13, 20)	11
13	0	(15, 20)	12
14	1	(17, 22)	13
15	1	(30, 2)	1
16	1	(30, 44)	1

FIG. 4. Topological structural analysis among borders by Algorithm 1: (a) The resulting picture, "A", "B", "C", "D", "E", "F", and "G" denote $10, 11, \ldots, 15$, and 16, respectively. "·", ":", "−", "+", "=", "#", "¥", "?", "&", "@" represent $-1, -2, \ldots, -9, -10$, respectively. "*" denotes values smaller than -10; (b) The extracted structure among borders.

Algorithm 1, as well as the border following algorithm [1] which discriminates between outer borders and hole borders, has the following advantages in the applications as suggested by Properties 2 through 5.

(1) We can count the 1-components and the holes in a binary picture and find its Euler number.

(2) We can shrink every 1-component or every hole into one pixel, by representing a 1-component by the border following starting point of its outer border and a hole by the right-hand neighbor of the starting point of its hole border. The former and the latter locate at the leftmost in the uppermost row of the 1-components and the hole, respectively.

(3) We can perform (1) or (2) depending on the features derived from the border: for example, we can shrink only the 1-components whose perimeters are greater than a given threshold, or which have their framing rectangles with the areas larger than a specified value.

Algorithm 1 also has the following advantages as seen from Properties 2 and 6.

(4) We can extract the surroundness relation among the connected components in a binary picture. It can be applied to picture search in an image database or feature extraction of pictures.

(5) The border representation with topological structure derived by Algorithm 1 would be effective for a storing method of pictures. This is because some simple image processing can be executed without restoring the original picture. Such image processing includes: to obtain some features from the borders, e.g., the perimeters and the areas of the components; to analyze the topological structure, e.g., whether a 1-component is adjacent to the background or not, or whether it has more than n holes or not; and to extract or delete connected components or holes depending on their geometrical or topological features.

Figure 4 shows an example of the representation of topological structure among borders derived by Algorithm 1.

4. THE BORDER FOLLOWING ALGORITHM FOR EXTRACTING ONLY THE OUTERMOST BORDERS

We can modify the algorithm shown above so that it follows only the outermost borders of a binary picture (i.e., the outer border between the background and a 1-component).

ALGORITHM 2. We explain here only the difference of the algorithm from Algorithm 1. (1) We start the border following only at the points such that the condition for the border following starting point of an outer border (Fig. 2a) holds and LNBD ≤ 0 when the raster scan reaches there. (2) The marking policy is the same as that of Algorithm 1 except that the values "2" and "-2" are substituted for the values "NBD" and "$-$NBD," respectively. (3) We keep the value LNBD of the nonzero pixel encountered most recently during the raster scan. Every time we begin to scan a new row of the picture, reset LNBD to zero.

The rationale for Algorithm 2 is as follows. The parent border of an outermost border is the frame of the picture. Therefore, the border point (i, j) immediately at the right of a 0-pixel is on an outermost border if and only if (i, j) satisfies either of the following conditions (1) or (2).

(1) All the pixels $(i, 1), (i, 2), \ldots, (i, j-1)$ are 0-pixels.

(2) The border point (i, h) which has been encountered most recently during the TV raster scan is on an outer border and the pixel $(i, h+1)$ belongs to the background.

Since we kept the value LNBD of the nonzero pixel encountered most recently during the TV raster scan, the above conditions (1) and (2) can be checked by the conditions "LNBD = 0" and "LNBD = -2," respectively, without following the other borders than the outermost ones.

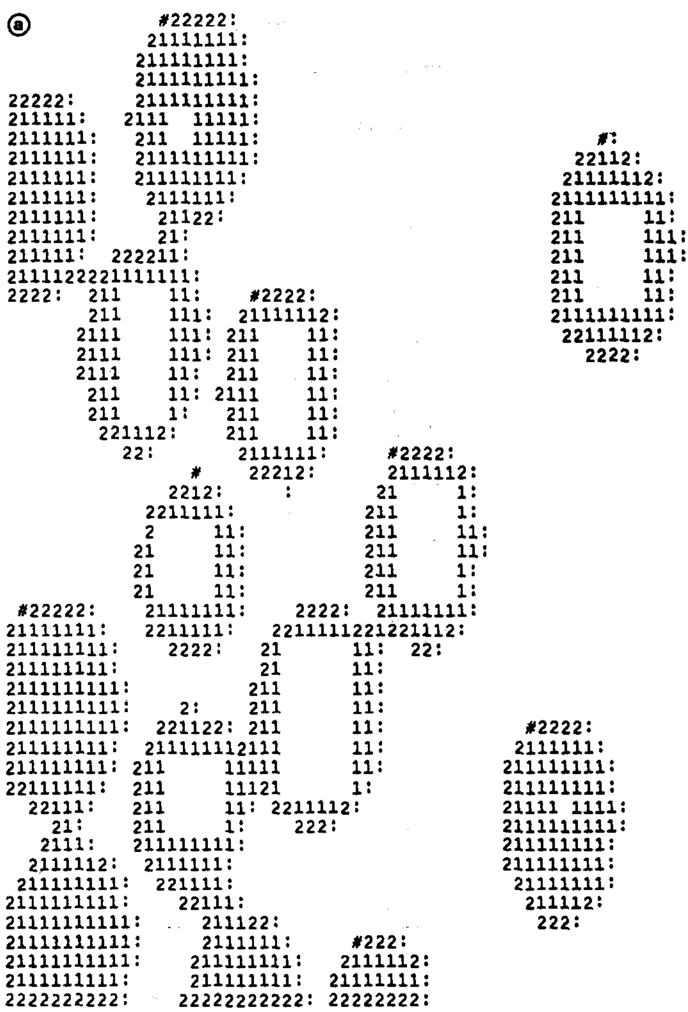

FIG. 5. The results of shrinking blobs (a) and line patterns (b) by Algorithm 2. "#" represents the border following starting point of an outermost border; ":" denotes -2.

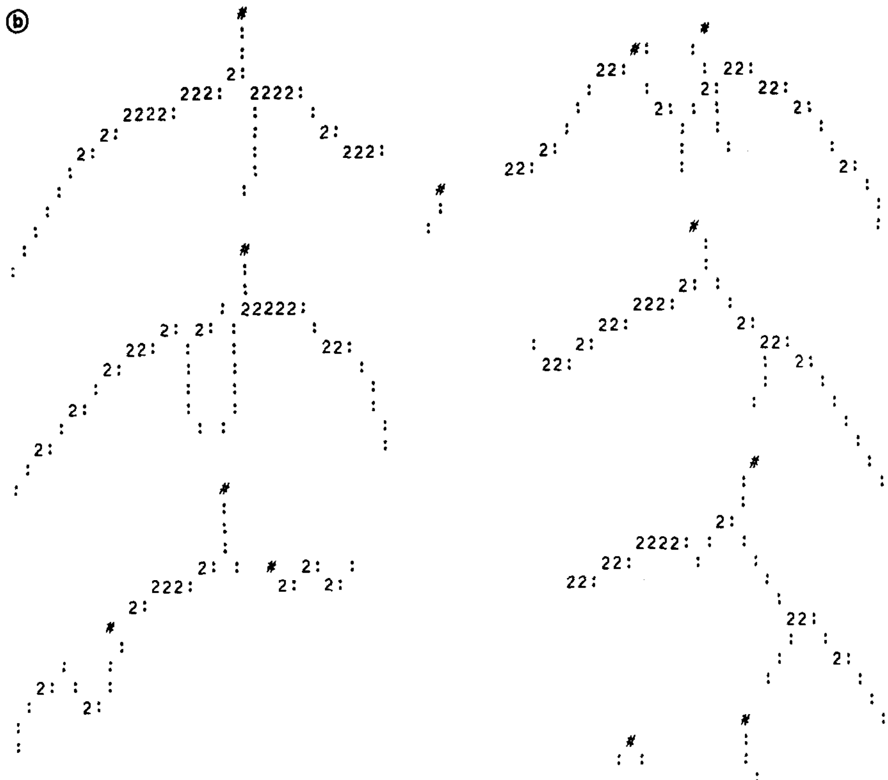

FIG. 5—*Continued.*

PROPERTY 7. Algorithm 2 follows only the outermost borders in an input binary picture and each outermost border is followed only once.

Algorithm 2 is effectively used for 1-component counting and shape analysis if there exists no 1-component surrounded by another 1-component, because it does

TABLE 2

Computation Time[a] of Different Shrinking Algorithms

Algorithm	Data	
	Blobs (60×66)	Line pattern (120×120)
Algorithm 1[b]	0.42 s	1.23 s
Algorithm 2	0.36 s	1.22 s
Component labeling	0.33 s	1.22 s
Yokoi's method[b] [12]	0.66 s	2.36 s
Rao's method [13]	93.18 s	503.64 s

[a] The CPU time measured on a FACOM 230-45S.
[b] The function which is not required for shrinking was removed.

not follow any hole border and consequently works faster. It is also useful if all the 1-components surrounded by other 1-components are noise or the elements not to be processed. In Table 2 we compare the processing time of several algorithms for 1-component shrinking [1, 12, 13] when a general-purpose sequential digital computer is used. The input pictures, which have no 1-component surrounded by other 1-components, are shown in Fig. 5: one is blobs with optional holes and the other is a line pattern. The shrinking method by Algorithm 2 is almost as fast as the component labeling. If all the 1-components surrounded by other 1-components should be disregarded, it would be most suitable, as the component labeling cannot discriminate in itself the outermost components from the components to be ignored.

5. CONCLUSION

In this paper, we showed a way to analyze the topological structure of binary images by border following. This is an extension of the border following algorithm which discriminates between the outer borders and the hole borders of a binary picture. We also presented a method for counting the 1-components and extracting the borders of a binary picture when it is desirable to disregard all the 1-components but the outermost 1-components. This is a modified version of the first algorithm. These methods are quick and effective if a sequential digital computer is used. We are now investigating their application to document processing, such as flowchart recognition.

APPENDIX I: THE FORMAL DESCRIPTION OF ALGORITHM 1

When we say "the neighborhood" in this algorithm, we mean the 4- (8-) neighborhood in the 4- (8-) connected case. Let an input picture be $F = \{f_{ij}\}$. Set initially NBD to 1 (the frame of F forms a special hole border and gets the sequential number 1; NBD stands for the sequential number of the current border). Scan the picture with a TV raster and perform the following steps for each pixel such that $f_{ij} \neq 0$. Every time we begin to scan a new row of the picture, reset LNBD to 1.

(1) Select one of the following:

(a) If $f_{ij} = 1$ and $f_{i,j-1} = 0$, then decide that the pixel (i, j) is the border following starting point of an outer border, increment NBD, and $(i_2, j_2) \leftarrow (i, j-1)$.

(b) Else if $f_{ij} \geq 1$ and $f_{i,j+1} = 0$, then decide that the pixel (i, j) is the border following starting point of a hole border, increment NBD, $(i_2, j_2) \leftarrow (i, j+1)$, and LNBD $\leftarrow f_{ij}$ in case $f_{ij} > 1$.

(c) Otherwise, go to (4).

(2) Depending on the types of the newly found border and the border with the sequential number LNBD (i.e., the last border met on the current row), decide the parent of the current border as shown in Table 1.

(3) From the starting point (i, j), follow the detected border: this is done by the following substeps (3.1) through (3.5).

(3.1) Starting from (i_2, j_2), look around clockwise the pixels in the neighborhood of (i, j) and find a nonzero pixel. Let (i_1, j_1) be the first found nonzero pixel. If no nonzero pixel is found, assign $-$NBD to f_{ij} and go to (4).

(3.2) $(i_2, j_2) \leftarrow (i_1, j_1)$ and $(i_3, j_3) \leftarrow (i, j)$.

(3.3) Starting from the next element of the pixel (i_2, j_2) in the counterclockwise order, examine counterclockwise the pixels in the neighborhood of the current pixel (i_3, j_3) to find a nonzero pixel and let the first one be (i_4, j_4).

(3.4) Change the value f_{i_3, j_3} of the pixel (i_3, j_3) as follows:

(a) If the pixel $(i_3, j_3 + 1)$ is a 0-pixel examined in the substep (3.3), then $f_{i_3, j_3} \leftarrow -\text{NBD}$.

(b) If the pixel $(i_3, j_3 + 1)$ is not a 0-pixel examined in the substep (3.3) and $f_{i_3, j_3} = 1$, then $f_{i_3, j_3} \leftarrow \text{NBD}$.

(c) Otherwise, do not change f_{i_3, j_3}.

(3.5) If $(i_4, j_4) = (i, j)$ and $(i_3, j_3) = (i_1, j_1)$ (coming back to the starting point), then go to (4); otherwise, $(i_2, j_2) \leftarrow (i_3, j_3), (i_3, j_3) \leftarrow (i_4, j_4)$, and go back to (3.3).

(4) If $f_{ij} \neq 1$, then LNBD $\leftarrow |f_{ij}|$ and resume the raster scan from the pixel $(i, j + 1)$. The algorithm terminates when the scan reaches the lower right corner of the picture.

APPENDIX II: THE PROOF OF PROPERTY 3

We show the proof only for the 8-connected case. For the 4-connected case, we can prove the property in a similar way.

LEMMA 1. *By step* (3) *of Algorithm* 1, *only border B between* 1-*component S_1 containing pixel (i, j) and the* 0-*component S_2 containing pixel $(i, j - 1)$ ($(i, j + 1)$ if B is a hole border) is followed; all the border points on B are visited.*

Proof. If S_1 consists of just one 1-pixel, Lemma 1 holds apparently. Suppose that S_1 consists of more than one 1-pixels. Let (p, q) and (h, k) be an arbitrary border point followed and the border point visited after (p, q), respectively. As the pixel (h, k) is connected to the pixel (p, q) and all the 0-pixels examined in the substep (3.3) belong to 0-component S_2, the pixels (p, q) and (h, k) belong to the border B. Thus the first statement is true.

The second statement is valid for the border following method in [3], which is the same as step (3) of Algorithm 1 except for the way of marking. Q.E.D.

LEMMA 2. *When the border B between the* 1-*component S_1 containing pixel (i, j) and the* 0-*component S_2 containing $(i, j - 1)$ ($(i, j + 1)$ if B is a hole border) is followed by step* (3), *the value of any border point (p, q) on B is changed to negative if and only if its right-hand neighbor $(p, q + 1)$ belongs to S_2.*

Proof. If S_1 consists of just one 1-pixel, Lemma 2 holds apparently. Suppose that S_1 consists of more than one 1-pixels. It is enough for the proof of Lemma 2 to show the following three propositions.

(1) The value of any border point (p, q) is not changed to negative if its neighbor $(p, q + 1)$ is not a 0-pixel. This proposition is derived immediately from substep (3.4).

(2) The value of any border point (p, q) is not changed to negative if its neighbor $(p, q + 1)$ is a 0-pixel not belonging to S_2. This is because all the 0-pixels examined in substep (3.3) belong to S_2.

(3) The value of any border point (p, q) is changed to negative if its neighbor $(p, q+1)$ is a 0-pixel belonging to S_2. Let (h, k) be the first found 1-pixel when the 8-neighbors of (p, q) are searched for clockwise starting from $(p, q+1)$. It means that the pixel (h, k) is on the border B. From the correspondence between the crack following and the border following [2], the pixel (h, k) is visited before (p, q) when the border B is being followed. The border following algorithm [2] is a bit different from step (3) of Algorithm 1 in the 4-connected case. However, the difference does not affect the validity of the above statement. Thus the pixel $(p, q+1)$ is examined by substep (3.3). From substep (3.4), the value of pixel (p, q) is changed to negative. Q.E.D.

LEMMA 3. *During the TV raster scan of Algorithm* 1, *the first scanned pixel* (i, j) *in a particular* 1-*component* S_1 *satisfies the condition shown in Fig.* 2a. *Moreover, the pixel* (i, j) *lies on the outer border between* S_1 *and the* 0-*component* S_2 *which contains pixel* $(i, j-1)$.

Proof. Let us prove the first statement. Because the pixel (i, j) is scanned first in the 1-component S_1, the values of the already scanned pixels in its 8-neighborhood are 0; that is, $f_{i-1,j-1} = f_{i-1,j} = f_{i-1,j+1} = f_{i,j-1} = 0$. In Algorithm 1 the change of the value of a pixel is restricted to in the border following step (3). Since (i, j) is the first encountered pixel of the 1-component S_1, its value must be 1.

Next we prove the second half of the lemma. We assume that the 0-component S_2 is surrounded by 1-component S_1, and show that this assumption leads to a contradiction. From Definition 2, there should be a pixel in S_1 on any 4-path from $(i, j-1)$ to a pixel on the frame of the picture. If we consider a specific 4-path which goes straight upward, this contradicts the fact that (i, j) is the first visited pixel in S_1 during the TV raster scan. This implies that the pixel $(i, j-1)$ is a member of the 0-component which surrounds S_1. Q.E.D.

Now in order to prove Property 3 we only have to show that the other pixels on border B than (i, j) which satisfy the condition shown in 2a or b cannot be a starting point for following B again. We can see that no pixel except (i, j) satisfies the condition in Fig. 2a as follows. From Lemma 1 and step (3) of Algorithm 1, after following border B from the starting point (i, j) all the pixel values on B are changed to values other than 1, and will never be reset to 1.

Let S_2 be the 0-component containing pixel $(i, j-1)$. The following reasoning shows us that if a border point (p, q) on border B satisfies the condition in Fig. 2b, the pixel $(p, q+1)$ is a 0-pixel not belonging to S_2 (i.e., the border to be followed from the pixel (p, q) is different from B). From Lemma 2 and the step (3) of Algorithm 1, after following the border B from the starting point (i, j) any border point (p, q) whose neighbor $(p, q+1)$ belongs to S_2 is negative and will never be reset. That concludes the proof of Property 3.

APPENDIX III: THE PROOF OF PROPERTY 6

We will show the proof for the 8-connected case, as we could prove for the 4-connected case in a similar way.

The following lemma is derived from Properties 3 and 4.

LEMMA 4. *During the raster scan of Algorithm* 1, *the border following starting point of an arbitrary border B is visited after the border following starting point of the parent border of B has been scanned.*

Proof. Let (i, j) be the border following starting point of the border B.

First, suppose that B is an outer border between the 1-component S_1 and the 0-component S_2 which surrounds S_1. From Property 3 the pixel $(i, j-1)$ belongs to S_2. From Property 4, the border following starting point (p, q) of the border B_1 between S_2 and the 1-components which surrounds S_2 (i.e., the border B_1 is the parent border of B) has been visited before $(i, j-1)$ during the TV raster scan.

Suppose that B is a hole border between the hole S_3 and the 1-component S_4 which surrounds S_3. From Property 4, the pixel (i, j) belongs to S_4. From Property 3, the border following starting point (p, q) of the border B_2 between S_4 and the 0-component which surrounds S_4 (i.e., the border B_2 is the parent border of B) has been visited before (i, j) during the TV raster scan and cannot be identical to the pixel (i, j). Therefore, the lemma holds.

Now we prove Property 6. The proof is divided to the following two cases:

(1) Suppose that we found the border following starting point (i, j) of an outer border B between a 1-component S_1 and the 0-component S_2 which surrounds S_1. LNBD is equal to the value of the first met 1-pixel (i, h) when we look over the pixels leftward from $(i, j-1)$ if such a pixel exists (Fig. A.1a); otherwise LNBD equals 1, which is the sequential number of the frame. In the latter case, apparently LNBD is the sequential number of the parent border of B. We will show that the same holds in the former case. Property 3 implies that (i, h) is not in S_1 and the pixels between $(i, h+1)$ and $(i, j-1)$ belong to S_2. By Definition 4, (i, h) lies either on the parent border of B or an outer border which shares the common parent with B. Denote the border by B'. If B' is the parent border of B, it should have been followed already as seen from Lemma 4. If B' is an outer border, Property 3 shows that it has also been followed. Therefore, from Lemma 2 in Appendix II and Algorithm 1 the value of the pixel (i, h) was changed to the negative of the sequential number of B' and has never been reset since. As LNBD has been set to the value of (i, h) in step (4) of Algorithm 1, this proves the property in the case (1).

(2) Next consider the case that we found the border following starting point (i, j) of a hole border B between a hole S_3 and the 1-component S_4 which surrounds S_3. As $(i, j+1)$ belongs to the hole, to the left of (i, j) (including (i, j)) there exists at least one 1-pixel whose left-hand neighbor is a 0-pixel. Let (i, h) be the first met such pixel when we examine the pixels leftward from (i, j) (Fig. A.1b). The pixel (i, h) has already been followed when the pixel (i, j) is visited during the raster scan of Algorithm 1. This is derived from Lemma 4 if (i, h) is on an outer border and from Property 4 if it is on a hole border. Therefore, LNBD equals the value of the already followed 1-pixel between (i, h) and (i, j), which belongs to S_4. It cannot be the sequence number of B, because (i, j) is the border following

(a) (b)

FIG. A-1. The relation between the border following starting point (i, j) of an outer border (a) or a hole border (b) and the last encountered border point (i, h).

starting point of B and the parent border of B is determined before LNBD is set to the sequence number of B. This implies that LNBD of the step (2) of Algorithm 1 shows the sequential number of a border on S_4 other than B, that is, the sequential number of the parent border of B or the border which shares the parent with B. That concludes the proof of Property 6.

ACKNOWLEDGMENTS

The authors are very grateful to Professor T. Fukumura, Professor J. Toriwaki, and Professor S. Yokoi of Nagoya University and their colleagues for their fruitful discussions. The authors also thank Dr. Y. Shirai of Electrotechnical Laboratory for valuable advice. The comparative experiment was performed using the image processing software package SPIDER developed at Electrotechnical Laboratory. This work is partly supported by Scientific Research Grant-In-Aid from the Ministry of Education, Science and Culture, Japan.

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed., Vol. 2, Academic Press, New York, 1982.
2. A. Rosenfeld, Connectivity in digital pictures, *J. Assoc. Comput. Mach.* **17**, 1970, 146–160.
3. S. Yokoi, J. Toriwaki, and T. Fukumura, An analysis of topological properties of digitized binary pictures using local features, *Comput. Graphics Image Process.* **4**, 1975, 63–73.
4. Z. Li, S. Yokoi, J. Toriwaki, and T. Fukumura, Border following and reconstruction of binary pictures using grid point representation, *Trans. Inst. Electron. Commun. Eng. Japan [Part D]*, **J65**, 1982, 1203–1210.
5. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Sci., Rockville, Md., 1982.
6. T. H. Morrin, II, Chain-link compression of arbitrary black–white image, *Comput. Graphics Image Process.* **5**, 1976, 172–189.
7. T. Agui and I. Iwata, Topological structure analysis of pictures by digital computer, *Trans. Inst. Electron. Commun. Eng. Japan [Part D]*, **J62**, 1979, 333–340.
8. T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin, 1977.
9. T. Agui, H. Iitsuka, and M. Nakajima, Image processing for urban map utilizing pyramid hierarchical data, *Trans. Inst. Electron. Commun. Eng. Japan [Part D]*, **J65**, 1982, 1243–1249.
10. R. L. T. Cederberg, Chain-link coding and segmentation for raster scan devices, *Comput. Graphics Image Process.* **10**, 1979, 224–234.
11. D. L. Milgram, Constructing trees for region description, *Comput. Graphics Image Process.* **11**, 1979, 88–99.
12. S. Yokoi, J. Toriwaki, and T. Fukumura, A sequential algorithm for shrinking binary pictures, *Trans. Inst. Electron. Commun. Eng. Japan [Part D]*, **J62**, 1979, 537–542.
13. C. V. K. Rao, B. Prasada, and K. R. Sarma, A parallel shrinking algorithm for binary patterns, *Comput. Graphics Image Process.* **5**, 1976, 265–270.