

# 逐次局所処理をともなう正方向および逆方向 ラスト走査による高速ラベル付け手法

鈴木 賢治<sup>†</sup> 堀場 勇夫<sup>†</sup> 杉江 昇<sup>††</sup>

2値画像中の別々の対象物(連結成分)にラベル付けをする手法は、パターン認識の基本的かつ重要な手法の1つである。本論文では、1次元テーブルを用いた逐次局所処理をラスト走査の正方向と逆方向に交互に行うことにより、ラベル付けを高速に行う手法を提案する。従来手法では、ラベルを図形上で伝播させることによりラベル付けを行うため、図形の幾何学的形状の複雑さにより実行時間が大きく異なるという問題点があった。これに対し本手法では、暫定ラベルの連結性を記憶するテーブルを逐次用いることにより、ラベルをテーブル内で伝播させ、幾何学的距離が離れたラベルの連結性を反映させる。これにより高速化を達成する。従来手法との比較評価実験により、各手法の特徴を明らかにし、本手法の高速性を示す。さらに、本手法の実行時間が画像サイズ(画素数)のオーダーであることを実験的に示す。

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

KENJI SUZUKI,<sup>†</sup> ISAO HORIBA<sup>†</sup> and NOBORU SUGIE<sup>††</sup>

This paper presents a fast algorithm for labeling connected components in binary images based on sequential local operations in the course of forward raster scan followed by backward raster scan. A one-dimensional table, which memorizes label equivalences, is used successively during the operations for uniting equivalent labels. 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 proposed algorithm outperforms the conventional algorithms in terms of the computational cost.

### 1. はじめに

2値画像中の別々の対象物(連結成分)にラベル付けをする手法は、パターン認識の基本的かつ重要な手法の1つである<sup>1),2)</sup>。この問題は図形の連結性の判定と密接な関係があり、単なる局所並列処理では解決できないことが示され<sup>3)</sup>、逐次的な処理を必要とする代表的な例の1つである<sup>4)</sup>。

ラベル付けには、これまでに多くの手法が提案されており、次の2種類が代表的である。

(A) 逐次局所処理により図形上でラベルを伝播させ、画像の走査を複数回行うことによりラベル付けを行う手法<sup>5),6)</sup>。

(B) ラベルの連結性を記憶する1次元もしくは2次元テーブルを利用し、画像走査中または走査後にテーブル解析を行うことにより、2回または複数回の画像走査によりラベル付けを行う手法<sup>4),7)~13)</sup>。

これらのほかに、(C) 画像を木構造で表現し、多進木を用いてラベルの連結性の判定を解決する手法<sup>14)~17)</sup>、(D) 並列計算機のための手法<sup>18)~21)</sup>、(E) 手法のハードウェア化<sup>22)~26)</sup>などの研究が行われている。

分類(A)の手法は、逐次局所処理に基づくためハードウェア化に適している。しかし、ラベルを図形上で伝播させるため、図形の幾何学的な複雑さによって多数回の画像走査を必要とし、その実行時間は膨大となる。分類(B)、(C)の手法では、ラベルの連結性の解析にUNION-FIND手法<sup>27),28)</sup>などが用いられるが、これは比較的複雑な処理であるためハードウェア化に適するものではない。さらに、この解析処理には大きな時間がかかり、これが図形の連結性の複雑さに依存する。このような従来手法では、処理対象とする図形

<sup>†</sup> 愛知県立大学情報科学部  
Faculty of Information Science and Technology, Aichi  
Prefectural University

<sup>††</sup> 名城大学理工学部  
Faculty of Science and Technology, Meijo University

により実行時間が大きく異なるため、これを前処理とする応用のためには、最大実行時間を想定しなくてはならないが、これがきわめて大きな時間であるため、オンラインリアルタイム処理が必要な分野への応用を妨げていた。

本論文では、1次元テーブルを利用した逐次局所処理によりラベル付けを高速に行う手法を提案する。本手法は、分類(A),(B)の手法を巧く組み合わせることにより、高速なラベル付けを達成するものである。ここでは、従来手法が不得意な図形とその問題点を述べ、それを解決する手法を提案する。従来手法との比較評価実験により、各手法の特徴を明らかにし、本手法の高速性を示す。さらに、本手法の実行時間が画像サイズ(画素数)のオーダーであることを実験的に示す。

2. 従来のラベル付け手法<sup>6)</sup>

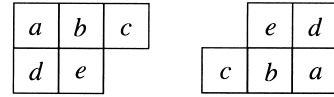
2.1 従来手法の概要

画像中の対象物を  $F_O$ 、背景を  $F_B$  に2値化した画像  $b(x,y)$  に対し、図1(a)に示すマスクをラスタ走査の順序で走査(正走査)しながら、次式の処理による暫定的なラベル付けを行う。ここで得られる画像は、完全なラベル付け画像(1つの対象物に対して1つのラベルが付けられている画像)ではなく、暫定的なラベルが付けられた画像である。ただし、 $F_O, F_B$  は十分大きな値( $F_O < F_B$ )とし、暫定ラベル値  $m$  は、たとえば1に初期化する。

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

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

ただし、 $(m = m + 1)$  は暫定ラベル値を1だけ増加させることを表し、 $\min(\cdot)$  は最小値を求める演算子、 $M_S$  はマスクの注目画素  $e$  を除く領域、式中の条件は上の行を優先して判断する。式(1)の1行目の条件は、注目画素  $b(x,y)$  が背景である場合を表している。2行目の条件は、注目画素が対象物であり、マスク中の近傍画素のすべてが背景である場合を表している。この場合、対象画素に初期の暫定ラベルが付けられる。3行目では、マスク内の暫定ラベル中で最も小さい暫定ラベルが付けられる。これにより、より小さな暫定ラベルが図形上を伝播する。次に、図1(b)に示すマスクをラスタ走査の逆の順序で走査(逆走査)しなが



(a) Forward scan. (b) Backward scan.

図1 ラスタ走査ラベル付け手法のマスク形状(8連結用)  
Fig. 1 Masks for labeling of 8-connected components.

ら、次式の処理を行う。

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

ただし、 $M$  はマスクを表す領域である。マスク内において異なる暫定ラベルが現れなくなるまで、これら正走査と逆走査を繰り返すことにより、完全なラベル付け画像が得られる。

2.2 問題点

従来手法では、複数回の画像走査を行うことにより、完全なラベル付けを行うことができるが、その走査回数が理論的におさえられていない。この走査回数は図形の幾何学的な複雑さに依存し、処理を完了するために膨大な時間がかかるという問題がある。

図2は従来手法によるラベル付けの例を示す図である。直線状の対象物が傾きをもって計測された場合このような図形となるため、通常の画像に頻繁に現れる。従来手法では、図に示すように4回の画像走査でラベル付けを完了する。暫定ラベルは、画像左上から右下(逆走査ではその逆)方向へ図形上を伝播し、より小さなラベルへと統合されていくが、1つのラベルに統合されるためには、図形の幾何学的形状に依存した複数回の画像走査が必要である。例のような階段状のパターンが  $N$  段であった場合、 $2(N-1)$  回の画像走査が必要である。図3に示す巻き数が2の渦状のパターンでは、4回の画像走査でラベル付けを完了する。渦の巻き数が  $N$  であった場合、 $2N$  回の画像走査が必要である。

3. 提案手法

3.1 手法

従来手法では、暫定ラベルが図形上を一定方向にしか伝播しないため、図形の幾何学的形状に依存して多数回の画像走査が必要であった。本手法では、暫定ラベルの連結性を記憶する1次元のテーブルであるラベル変換テーブルを逐次用いることにより、暫定ラベルを図形上ではなくテーブル内で伝播させ、ラベルの連

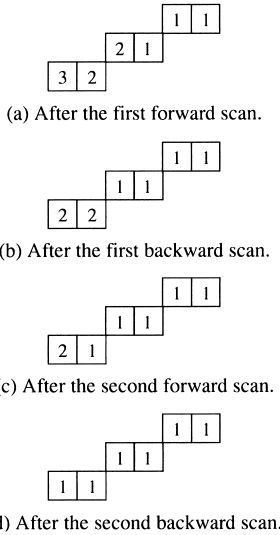


図2 従来手法によるラベル付けの例(階段状パターン)  
Fig. 2 Example of labeling of a stair-like connected-component by the conventional algorithm.

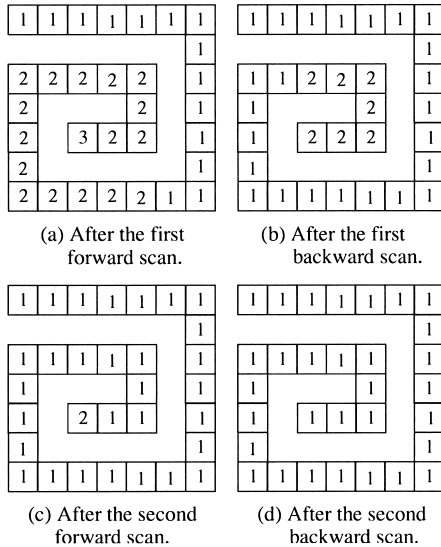


図3 従来手法によるラベル付けの例(渦状パターン)  
Fig. 3 Example of labeling of a spiral connected-component by the conventional algorithm.

結性を解決する(同一対象物内のラベルを1つのラベルに統合する)。これにより、幾何学的距離が離れた暫定ラベルの連結性を反映させることができるため、従来手法の問題であった画像の走査回数を抑えることができる。

本手法は、1次元テーブルを用いた逐次局所処理をラスト走査の正方向と逆方向に交互に行うことにより、ラベル付けを高速に行う。以下、本手法を説明

する。最初に、ラベル変換テーブルを  $T[F_0] = F_0$ ,  $T[F_B] = F_B$  に初期化する。

[ 1 回目の正走査 ]

1 回目の正走査では、次式のように、ラベル変換テーブル  $T[m]$  を利用しながら、暫定的なラベル付けを行う。

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

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

1, 2 行目は、従来手法と同じである。3 行目では、ラベル変換テーブルで変換された暫定ラベルのうち、最も小さい暫定ラベルが付けられる。これにより、それまでに合流(異なる暫定ラベルがマスク内で出会うこと)した暫定ラベルの連結性を、より小さな暫定ラベルとして反映させることができる。

画像への暫定的なラベル付けと同時に、次式のようにラベル変換テーブルを更新する。

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

2 行目では、テーブルに初期値が与えられる。3 行目では、合流した暫定ラベルがラベル変換テーブルに記憶される。

[ 1 回目の正走査以降の走査 ]

1 回目の正走査以降の走査では、逆走査、正走査を交互に繰り返し、次式の処理を行う。

$$g(x, y) = \begin{cases} F_B & \text{if } g(x, y) = F_B \\ T_{\min}(x, y) & \text{otherwise} \end{cases} \quad (7)$$

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

同時に、次式によりラベル変換テーブルを更新する。

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

画像走査の繰返しはマスク内の暫定ラベルが同じになるまで行い、最終的なラベル付け画像を得る。すなわち、次式の条件が満たされている間、画像走査を繰り返す。

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

ただし、 $i, j \in M_S$  である。

これらの演算は付録に示す演算論理により表現できるため、ハードウェア化の際には、これに基づくテーブルを用意することにより効率的な実現が可能である。また、ここでは8連結の連結成分のラベル付けを扱ったが、4連結の場合には付録に示すマスクを用いることにより可能である。

### 3.2 手法の位置付け

本手法の特徴を従来手法と比較することにより、本手法の位置付けを明確にする。本手法の主な特徴は以下である。

- (1) テーブル解析を行う必要がなく、逐次局所処理のみで実現されている。
- (2) ラベルの連結性の保存に1次元テーブルを使用し、連結性の解決に画像走査中に行うテーブルの単純な書き換えを用いている。

特徴(1)のテーブル解析は複雑な処理であるため、通常ソフトウェアにより実行されるが、この処理には大きな時間がかかる。このため、テーブル解析を行う必要がない本手法では高速性が期待できる。さらに、逐次局所処理のみで実現されている本手法は、ハードウェア化の際の回路構造が単純になるため、ハードウェア化に適している。特徴(1)と同じ特徴を持つ従来手法には分類(A)の手法<sup>5),6)</sup>がある。しかし、これらの手法では前章で具体的に述べたように、図形の幾何学的複雑さによって多数回の画像走査を必要とするため、この実行時間は膨大となるという問題点がある。

特徴(2)のラベルの連結性の保存には、2次元テーブル<sup>9),11)</sup>、2つの1次元テーブル<sup>4)</sup>、単一の1次元テーブル<sup>7),8),10),12)</sup>を用いる手法がある。本手法のように単一の1次元テーブルを用いる手法は、一般的に小さな回路規模で実現可能である。また、画像走査中に行うテーブルの単純な書き換えを連結性の解決に用いている本手法は、制御回路が大変簡単になり、パイプライン化も可能であるため、回路規模を小さくでき高速性も期待できる。このような方法により連結性を解決する手法はまだ提案されていない。

単一の1次元テーブルを用いる手法には、文献7)、10)の手法、文献8)の手法、文献12)の手法がある。

文献7)、10)の手法では、暫定ラベルの合流が起こるごとにテーブル探索を行う必要があり、さらにラベルの連結性の解決にテーブル解析を必要とする。このため、ハードウェア化やリアルタイム処理には不向きである。文献8)の手法では、画像の各ラインごとにテーブル解析を行う。このため、2本のライン上にある暫定ラベルの合流数分だけの小さなテーブルを用意すればよい。テーブルを用いる手法の中では最も小さなテーブルで実現できる。しかし、各ラインごとに、テーブル解析と画像へ付与した暫定ラベルの付け替えを行う必要があるため、演算時間は大変大きくなる。さらに、手続きが複雑になるためハードウェア化には適さない。文献12)の手法は、ラベルの連結性の解決にテーブル解析を必要とし、通常これはソフトウェアにより行われるため、この部分はリアルタイム処理に不向きである。さらに、ラベルの連結性を1次元テーブルに記憶する際、同じアドレスに2回以上の記憶が行われた場合、前に記憶された情報が壊れラベルの連結性が損なわれる。これに対処するため、テーブル解析後の画像走査の際にラベルの分離を検出し、分離があった場合には連結性を再度テーブルに記憶するという対策を行っている。しかし、この再記憶の際にも同様の現象が発生する可能性があるため、複数回の画像走査が必要である。この走査回数は図形の連結性の複雑さに依存し、理論的にも実験的にも示されていないという問題点がある。

以上のように、本手法は上述した2つの特徴を持ち、従来手法に比べ高速性が期待でき、ハードウェア化に適しているといえる。

### 3.3 手法の解析

#### 3.3.1 処理例

本手法によるラベル付けをいくつかの例に基づき考察する。まず、図2の図形の処理例を図4に示す。(a)は従来手法<sup>6)</sup>の正走査により付けられる暫定ラベルを表している。○は暫定ラベルの合流点である。この図形では、最初に暫定ラベル1と2が合流し、次に暫定ラベル2と3が合流する。(b)はこの図形におけるラベルの連結性をラベルの合流順に示したものである。本手法による1回目の正走査後、図形には(c)に示す暫定ラベルが付けられる。まず、暫定ラベル1と2が合流するため、ラベル変換テーブルのアドレス2には1が書き込まれる。次に、暫定ラベル2と3が合流する。暫定ラベル2をテーブル変換したラベルは1であるため、ラベル変換テーブルのアドレス3には1が書き込まれ、図形にもラベル1が付けられる。これにより、暫定ラベル1, 2, 3はラベル1で

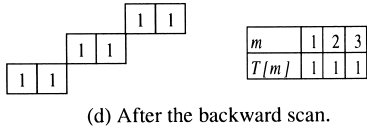
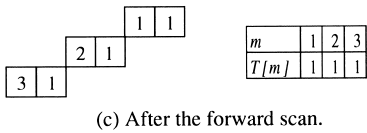
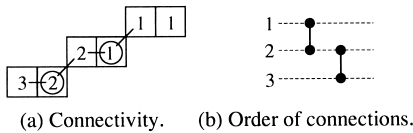


図4 提案手法による処理例(階段状パターン;合流パターン A)  
 Fig.4 Example of labeling of a stair-like connected-component by the proposed algorithm (Connection pattern A).

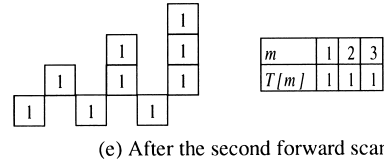
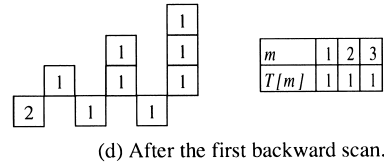
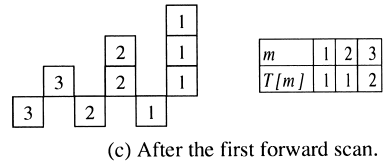
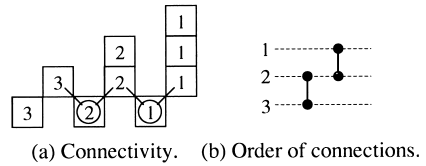


図6 合流パターン B を持つ別の図形の処理例  
 Fig.6 Example of labeling of another connected component with the connection pattern B by the proposed algorithm.

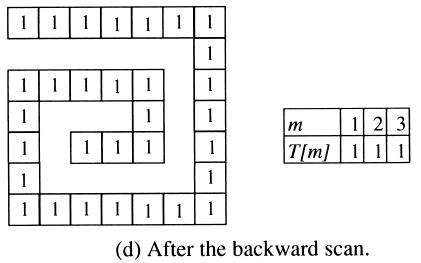
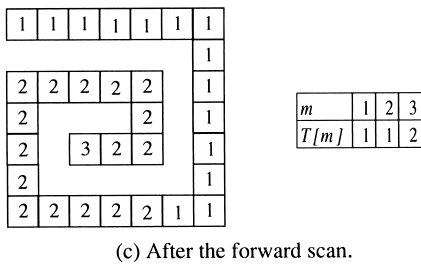
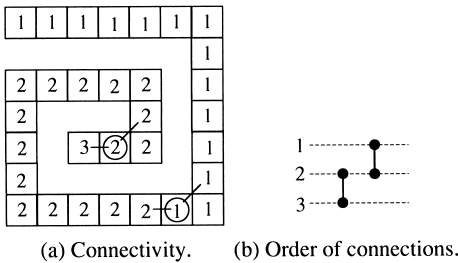


図5 提案手法による処理例(渦状パターン;合流パターン B)  
 Fig.5 Example of labeling of a spiral connected-component by the proposed algorithm (Connection pattern B).

あるとするラベル変換テーブルが完成する．この後の逆走査の終了後，(d)に示すようにラベル付けが完了する．本手法は  $N$  段の階段状パターンに対しても 2 回の画像走査でラベル付けを完了する．

次に，図 3 の図形の処理例を図 5 に示す．この図形では最初に暫定ラベル 2 と 3 が合流し，次に暫定ラベル 1 と 2 が合流する．図 4 の図形とは合流の順序が逆であるため，逆走査を行えばラベル変換テーブルが完成し，同時にラベル付けも完了する．本手法は渦の巻き数が  $N$  回の渦状パターンに対しても 2 回の画像走査でラベル付けを完了する．以上のように，従来手法<sup>6)</sup>では幾何学的形状の複雑さに比例した多数回の画像走査が必要な図形に対しても，本手法では一定の画像走査回数でラベル付けを完了することができる．

3.3.2 画像走査回数に関する考察

図 5 の図形では，ラベル変換テーブルが完成する走査の終了と同時にラベル付けも完了したが，一般的には図 6 に示すように，ラベル変換テーブルが完成した後，もう 1 回の画像走査が必要である．

ここで，2 回のラベル合流に連結された 3 つの暫定ラベルは，順序付きのラベル合流の最も基本的な要素であり，この順列は  ${}_3P_2$  であるため 6 通りである．基本的な順序付き合流の 6 パターンを図 7 に示す．図 4，

図5に示した2通り以外の残り4通りの処理例を図8に示す。図に示すように、合流パターンC, Dではそれぞれ、正走査後、逆走査後にラベル変換テーブルが完成する。合流パターンE, Fでは、正走査でも逆走査でもラベル変換テーブルが完成する。

任意の図形は、これら6通りの合流パターンの組合せにより表現できる。正走査によりラベルの連結性が解決できる合流パターン(A, C, E, F)では、これらのパターンが組み合わさっても、1回の正走査により連結性の解決が可能である。すなわち、合流パターンA, Cでは、後のラベル合流の前に、これと連結性を持つ暫定ラベル(合流パターンAでは2, 合流パターンCでは3)が最小値の暫定ラベルとすでに合流しているため、これら合流パターンが組み合わさっても、ラベル変換テーブルのデータは最小値となる。合流パターンE, Fは、大きな値を持つ暫定ラベル(2, 3)が最小値の暫定ラベル(1)と直接合流する合流パターンである。このため、これらの合流パターンが組

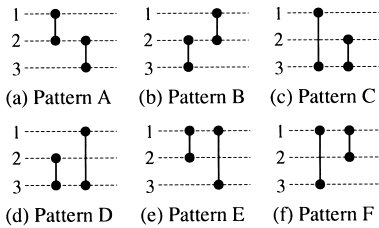


図7 基本的な順序付き合流の6パターン  
Fig. 7 Primary orderly label connections.

み合わさっても、ラベル変換テーブルのデータは最小値となる。

また、逆走査後にラベルの連結性が解決される合流パターン(B, D)では、通常の場合、逆走査におけるラベルの合流の順序は正走査時の順序とは逆になるので、正走査の場合と同様にこれらの合流パターンが組み合わさっても、ラベル変換テーブルのデータは最小値となる。

しかし、ラベルの合流の順序が正走査時の順序とは逆にならない場合がある。このことについて詳しく述べる。暫定ラベルの合流には、図9(a), (b)に示すような2種類の連結パターン( $\alpha, \beta$ )がある(付録の表3参照)。図9(a), (b)は正走査時のラベルの合流点を示している。逆走査時には、マスク形状および走査順の関係から、図9(c), (d)に示すように、ラベルは菱形で示す点で合流する。このため、逆走査時に合流の順序が正走査時の順序とは逆にならない場合がある。この例を図10に示す。このような場合においても、図に示したように、本手法では逆走査後にラベルの連結性が解決される。すなわち、正走査により大きな値を持つ暫定ラベル(3)が最小値としてテーブルに記憶されるため、逆走査ではどちらの順序で合流しても暫定ラベルは最小値となり、ラベル変換テーブルが完成する。

この順序の逆転は、連結パターン $\alpha$ の後に連結パターン $\beta$ が続いた場合にのみ起こる。走査の順序と幾何学的な位置関係から、 $\alpha\alpha, \beta\beta, \beta\alpha$ の順序では

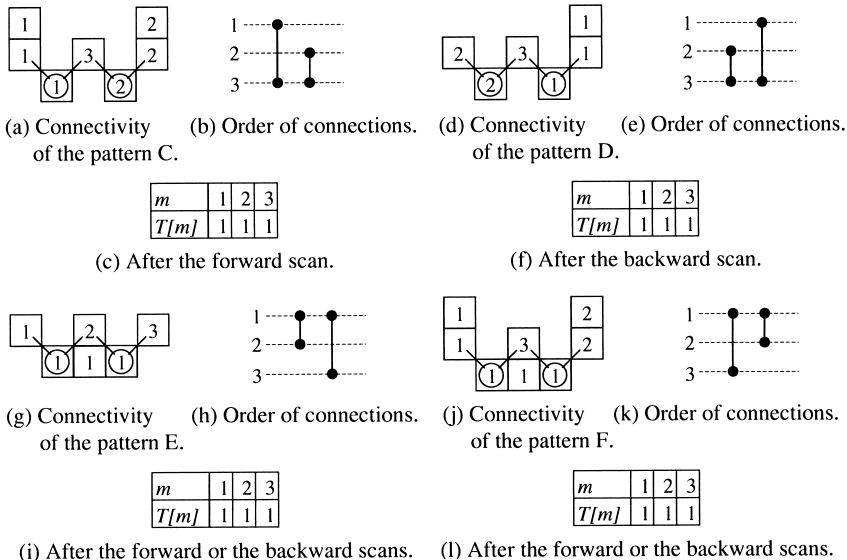


図8 提案手法による処理例(合流パターンC, D, E, F)  
Fig. 8 Examples of labeling by the proposed algorithm (Connection patterns C, D, E, and F).

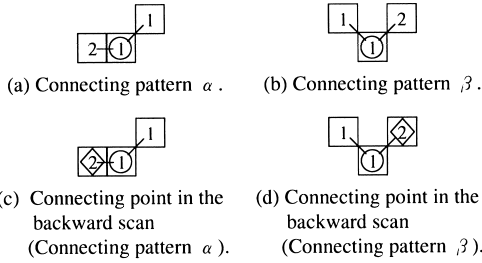


図9 正走査における合流点と逆走査における合流点の関係  
Fig.9 Relationship between the connecting point in the forward scan and that in the backward scan.

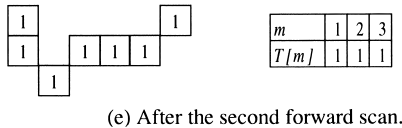
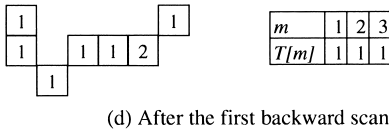
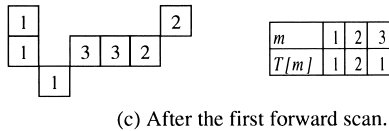
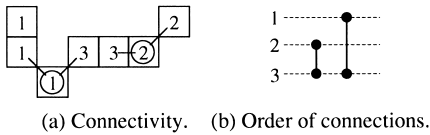


図10 提案手法による処理例(逆走査時のラベルの合流の順序が正走査時の順序とは逆にならない場合)  
Fig.10 Examples of labeling by the proposed algorithm (Connecting pattern  $\alpha$  followed by the connecting pattern  $\beta$ ).

起こりえない。順序の逆転は、 $\alpha\beta$ の順序でのみ起こることから、2つの合流点の前後関係でのみ生じるといえる。したがって、合流パターンが繰り返された場合については、この逆転現象を考える必要がないことが分かる。

これらの合流パターンが多重に組み合わせられた場合、正走査、逆走査の2回の走査ではラベルの連結性が解決されない場合がある。このような合流パターンの例を図11に示す。この図形では、最初に1~9までの初期暫定ラベルが付けられるが、逆走査後には1, 2, 6の暫定ラベルのみに統合される。すなわち、暫定ラベル3, 4, 5, 7, 8, 9は図形上にはない。図11(d)は逆走査後のラベル変換テーブルを示している。この時点ではまだ、暫定ラベル6の連結性が解決されて

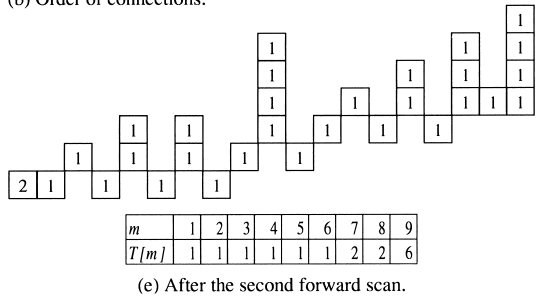
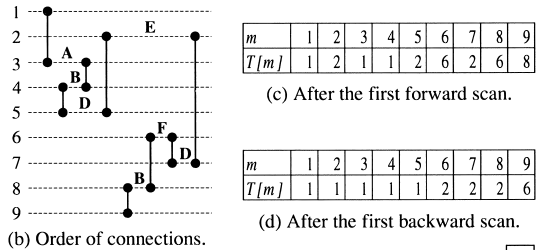
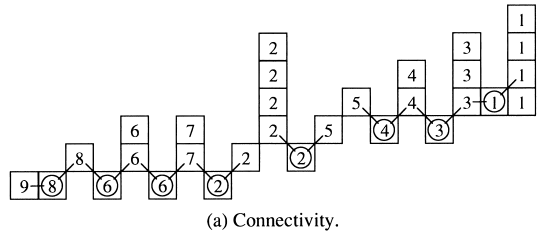


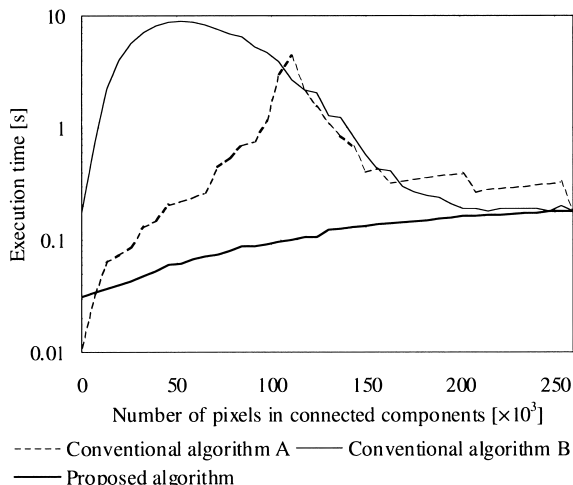
図11 提案手法による処理例(合流パターンが多重に組み合わせられた場合)  
Fig.11 Example of labeling by the proposed algorithm (Multiplex combination of the connection patterns).

いない。この後に正走査を行うことにより、図11(e)のようにラベル変換テーブルが完成する。しかし、この図形では1画素のみで暫定ラベル2が残る。したがって、さらに逆走査を行うことにより、この図形のラベル付けは完了する。以上より本手法の最大走査回数は、4回以上であることが分かる。しかし、任意の図形に対する最大走査回数の数学的証明は困難であるため、本論文では次章で実験的に示すことにする。

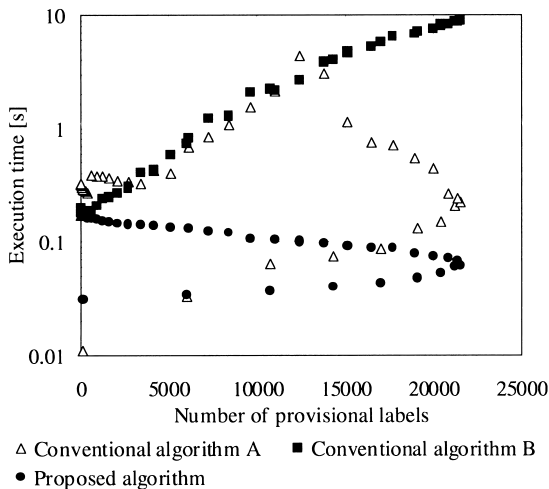
#### 4. 比較評価実験

##### 4.1 最大走査回数

本手法の最大走査回数を実験的に求めるため、白色一様ノイズを加えた画像(512×512画素, 1,000階調)に対し、閾値を0から1,000まで25ごとに变化させて2値化した画像41種を作成した。これを1セットとし、ノイズを変えて50セットを作成し、合計2,050種の画像に対してラベル付けを行った。このような画像は自然画像に比べ、複雑な幾何学的形状と複雑なラベル連結性を持つため、ラベル付け手法を厳しく評価



(a) Execution time against the number of pixels in connected components.



(b) Execution time against the number of provisional labels.

図 12 対象物の総画素数および初期暫定ラベル数に対する実行時間の比較  
 Fig. 12 Comparison of the execution time against the number of pixels in connected components and the number of provisional labels.

するのに適している．実験の結果，すべての画像において 4 回以下の画像走査でラベル付けを完了した．

4.2 各手法の性質

4.2.1 最大実行時間の想定

ラベル付けを前処理とする応用のために想定するラベル付けの最大実行時間を従来手法と本手法について求め，さらに各手法の性質を比較評価する．従来手法として，分類 (A) の代表的な手法である橋詰らの手法<sup>6)</sup> (従来手法 A と記す) と，分類 (B) の代表的な手法である白井らの手法<sup>7)</sup> (従来手法 B と記す) を用いた．従来手法 B では，最初に画像を走査して暫定的なラベル付けを行うと同時に，ラベルの連結性を 1 次元テーブルに巡回リストをなすように記憶する．この後にテーブル解析を行いテーブルを完成させ，このテーブルを用いた 2 回目の画像走査で最終的なラベル付けを行う手法である．各従来手法の実行時間は，従来手法 A では，先に述べたように図形の幾何学的形状の複雑さに依存する．従来手法 B では，暫定ラベルの合流が起こるごとにテーブル探索を行う必要があるため，実行時間は暫定ラベルの合流点数に依存し，最悪の場合は初期暫定ラベルの 2 乗のオーダーとなる．

白色一様ノイズを加えた画像に対し，従来手法および本手法によりラベル付けを行い，その実行時間を比較した．比較結果を図 12 に示す．各手法の実行時間は，ワークステーション (SUN 社製 UltraSPARC-II 300 MHz) 上での CPU 実行時間を用いた．従来手法 A では最大 4.33 秒，従来手法 B では最大 8.88 秒を要

するのに対し，本手法は最大 0.18 秒でラベル付けが完了しており最も高速である．本手法の実行時間は対象物の総画素数に依存し，評価にはこれが最大の画像が含まれている．したがって，本手法で想定すべき最大実行時間は，この画像サイズの画像，使用した計算機において，0.18 秒を想定すればよいことが分かる．

図 12 (b) は画像に最初に付けられる初期暫定ラベル数に対する実行時間を表している．ノイズ画像では暫定ラベルの合流点数が初期暫定ラベル数にほぼ比例する．このため，従来手法 B の実行時間は初期暫定ラベル数に依存していることが分かる．評価に用いた画像では，暫定ラベルの合流点数がほぼ最大に近いものを含むと考えられる．したがって，従来手法 B では 8.88 秒の最大実行時間を想定すればよいことが分かる．

従来手法 A では，図 2 のパターンを画像全体に敷き詰めた場合がこの画像サイズにおける理論的最大値である．これを測定したところ 15.53 秒であったため，従来手法 A ではこの時間を最大実行時間として想定すればよいことが分かる．なお，同じ画像において，従来手法 B，本手法では，それぞれ 0.20 秒，0.11 秒を要した．

4.2.2 画像サイズに対する性質

画像サイズを変化させ (64×64, 128×128, 256×256, 512×512 画素)，各 41 種類，合計 164 種類の画像に対して各手法の実行時間を測定した．結果を図 13 に示す．各手法共，画素数の累乗に比例して実行時間がかかっている．各手法による実行時間の最大値  $t_{max}$



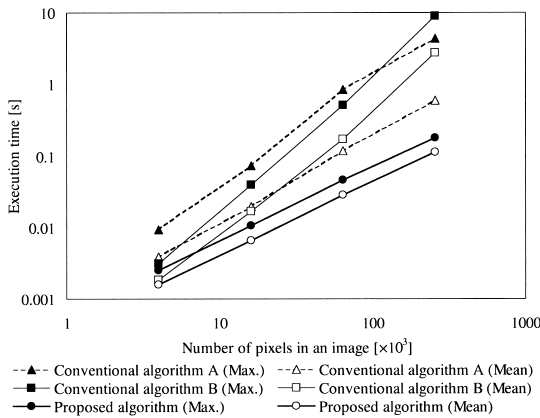


図 13 画像サイズ (画素数) に対する実行時間の比較

Fig. 13 Comparison of the execution time against the number of pixels in an image.

表 1 式 (11) のパラメータの算出結果

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

	a	b
Conventional algorithm A	$4 \times 10^{-10}$	1.50
Conventional algorithm B	$4 \times 10^{-8}$	1.90
Proposed algorithm	$5 \times 10^{-7}$	1.03

を次式の関数で最小 2 乗近似し, 画像中の総画素数  $I$  に対する依存性を調べた.

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

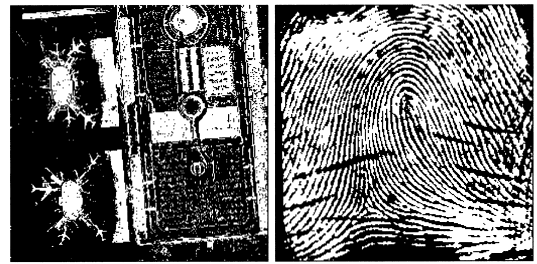
ただし,  $a, b$  はパラメータである. 結果を表 1 に示す. 従来手法 A では画素数の約 1.5 乗に比例, 従来手法 B では約 2 乗に比例, 本手法では画素数に比例していることが分かる. 本手法では画像サイズが大きくなっても実行時間は大きく増えないことを表している.

#### 4.3 様々な画像による評価

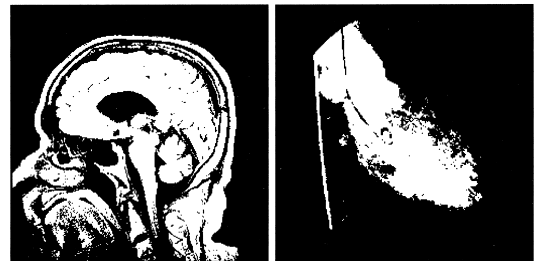
SIDBA (Standard Image Data Base) および USC (University of Southern California) の画像データベースから 50 種の自然画像, ならびに 50 種の医用画像の合計 100 種を評価用画像 (512×512 画素) として用いた. これらの画像を Otsu の判別分析法<sup>29)</sup>により 2 値化した. 自然画像および医用画像の例を図 14 に示し, 結果を表 2 に示す. 本手法は従来手法 B に比べ, 最大実行時間で 3.5 倍程度早いことが分かる. なお, 本実験に使用した 100 種の画像すべてにおいて, 本手法は 3 回以下の画像走査でラベル付けを完了した.

#### 5. おわりに

本論文では, 暫定ラベルの連結性を記憶する 1 次元



(a) LAX from the USC image data base. (b) FINGERPRINT from the SIDBA.



(c) Human head taken with MRI system. (d) Human heart taken with DSA system.

図 14 テスト画像の例

Fig. 14 Sample test images.

表 2 実行時間の比較

Table 2 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	0.131	0.076	0.038

テーブルを利用した逐次局所処理をラスタ走査の正方向と逆方向に交互に行うことにより, ラベル付けを高速に行う手法を提案した. 従来手法との比較評価実験により, 各手法において想定すべき最大実行時間を実験的に求め, 本手法によるラベル付けが高速であることを示した. さらに, 本手法の実行時間が画像サイズ (画素数) のオーダーであることを実験的に示した.

本手法は, 本論文中の 2,314 種の画像に対し, 最大 4 回の画像走査で完全なラベル付けが行えることを実験的に確認しているが, 数学的証明が今後の課題として残されている. 今後, 本手法のハード化を行っていく予定である.

謝辞 本研究に貴重なご助言をいただいた, 名城大学理工学部山本新教授, 名古屋電機工業 (株) 上田浩次氏, 画像を提供くださった (株) 日立メディコ技術研究所池田重之主任技師に深く感謝いたします. 本研究の一部は文部省学術フロンティア「量子情報論からの生命への接近」, 文部省科学研究費補助金奨励研究 (A), 日比科学技術研究助成金の支援による. また, 大

変重要かつ有益なご指摘をいただいた査読者に感謝いたします。

### 参 考 文 献

- 1) Rosenfeld, A. and Kak, A.C. (著) 長尾 真 (監訳): デジタル画像処理, pp.360-362, 近代科学社 (1978).
- 2) 鳥脇純一郎: 画像理解のためのデジタル画像処理 [ II ], pp.45-47, 昭晃堂 (1988).
- 3) Minsky, M. and Papert, S.: *Perceptron*, MIT Press, Cambridge (1969).
- 4) Rosenfeld, A. and Pfalts, J.L.: Sequential operations in digital picture processing, *J. ACM*, Vol.13, No.4, pp.471-494 (1966).
- 5) Haralick, R.M.: Some neighborhood operations, *Real Time/Parallel Computing Image Analysis*, pp.11-35, Plenum Press, New York (1981).
- 6) 橋詰明英, 鈴木隆一, 横内久猛, 堀内秀之, 山本真司: 赤血球自動識別アルゴリズムとその評価, *医用電子と生体工学*, Vol.28, No.1, pp.25-32 (1990).
- 7) 白井良明, 杉原厚吉: 連結領域を求める手法とそのハードウェア, *電子情報通信学会技術研究報告*, Vol.IE78-9, pp.29-33 (1978).
- 8) Lumia, R., Shapiro, L. and Zungia, O.: A new connected components algorithm for virtual memory computers, *Computer Vision, Graphics, and Image Processing*, Vol.22, No.2, pp.287-300 (1983).
- 9) 後藤敏行, 太田善之, 吉田真澄, 白井良明: パイプライン型高速ラベル付けアルゴリズム, *情報処理学会コンピュータビジョン研究会資料*, Vol.43, No.1, pp.1-8 (1986).
- 10) Shirai, Y.: Labeling connected regions, *Three-Dimensional Computer Vision*, pp.86-89, Springer-Verlag (1987).
- 11) 後藤敏行, 太田善之, 吉田真澄, 白井良明: 連結領域の高速ラベル付けアルゴリズム, *電子情報通信学会論文誌 ( D-II )*, Vol.J72-D-II, No.2, pp.247-255 (1989).
- 12) 奥山良幸, 小林芳樹, 武長 寛, 浅田和佳, 藤原和紀: ラスタ走査形ラベリングの高速化手法, *電子情報通信学会論文誌 ( D-II )*, Vol.J73-D-II, No.1, pp.36-45 (1990).
- 13) 直井 聡: 文字の形状特徴を利用した可変ウィンドウサイズによる高速ラベリング手法, *電子情報通信学会論文誌 ( D-II )*, Vol.J80-D-II, No.7, pp.1793-1801 (1997).
- 14) Samet, H.: Connected component labeling using quadtrees, *J. ACM*, Vol.28, No.3, pp.487-501 (1981).
- 15) Samet, H.: Computing geometric properties of images represented by linear quadtrees, *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol.PAMI-7, No.2, pp.229-240 (1985).
- 16) Samet, H. and Tamminen, M.: An improved approach to connected component labeling of images, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami, Florida, pp.312-318 (1986).
- 17) Hecquard, J. and Acharya, R.: Connected component labeling with linear octree, *Pattern Recognition*, Vol.24, No.6, pp.515-531 (1991).
- 18) Manohar, M. and Ramapriyan, H.K.: 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).
- 19) Olariu, S., Schwing, J.L. and Zhang, J.: Fast component labelling and convex hull computation on reconfigurable meshes, *Image and Vision Computing*, Vol.11, No.7, pp.447-455 (1993).
- 20) Choudhary, A. and Thakur, R.: Connected component labeling on coarse grain parallel computers: An experimental study, *Journal of Parallel and Distributed Computing*, Vol.20, pp.78-83 (1994).
- 21) Bhattacharya, P.: Connected component labeling for binary images on a reconfigurable mesh architectures, *Journal of Systems Architecture*, Vol.42, No.4, pp.309-313 (1996).
- 22) Yang, X.D.: Design of fast connected components hardware, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, MI, pp.937-944 (1988).
- 23) Hattori, T.: A high-speed pipeline processor for regional labeling based on a new algorithm, *Proc. Int. Conf. Pattern Recognition*, NJ, pp.494-496 (1990).
- 24) Nicol, C.J.: Design of a connected component labelling chip for real time image processing, *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*, Sydney, Australia, pp.142-147 (1992).
- 25) Alnuweiri, H.M. and Prasanna, V.K.: Parallel architectures and algorithms for image component labeling, *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol.14, No.10, pp.1024-1034 (1992).
- 26) Nicol, C.J.: A Systolic approach for real time connected component labeling, *Computer Vision and Image Understanding*, Vol.61, No.1, pp.17-31 (1995).
- 27) Mehlhorn, K.: *Data Structures and Algorithm 1: Sorting and Searching*, Springer,

表 3 演算論理 ( 1 回目の正走査 )

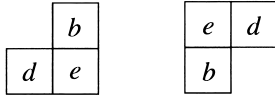
Table 3 Logic table for the first forward scan.

e	a	b	c	d	$g(x, y)$	Renewal of the label conversion 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	$\min(T[c], T[d])$	$T[\max T(c, d)] = g(x, y)$

表 4 演算論理 ( 1 回目の正走査以降 )

Table 4 Logic table for the forward and the backward scans except the first forward scan.

e	a	b	c	d	$g(x, y)$	Renewal of the label conversion 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	$\min(T[e], T[c], T[d])$	$T[\max T(e, c, d)] = g(x, y), T[\max T^2(e, c, d)] = g(x, y)$
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	$\min(T[e], T[c], T[d])$	$T[\max T(e, c, d)] = g(x, y), T[\max T^2(e, c, d)] = g(x, y)$
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	$\min(T[e], T[a], T[c])$	$T[\max T(e, a, c)] = g(x, y), T[\max T^2(e, a, c)] = g(x, y)$
1	1	0	1	1	$\min(T[e], T[c], T[d])$	$T[\max T(e, c, d)] = g(x, y), T[\max T^2(e, c, d)] = g(x, y)$
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	$\min(T[e], T[c], T[d])$	$T[\max T(e, c, d)] = g(x, y), T[\max T^2(e, c, d)] = g(x, y)$



(a) Forward scan. (b) Backward scan.

図 15 4 連結用マスク形状

Fig. 15 Masks for labeling of 4-connected components.

Berlin (1984).

- 28) Fiorio, C. and Gustedt, J.: Two linear time Union-Find strategies for image processing, *Theoretical Computer Science*, Vol.154, No.2, pp.165-181 (1996).
- 29) Otsu, N.: A threshold selection method from gray-level histograms, *IEEE Trans. Systems Man and Cybernetics*, Vol.9, pp.62-66 (1979).

付録 演算論理のテーブル化，4 連結用マスク

ラスタ走査に基づくマスクの移動を考えると，図 1 の位置 a の暫定ラベルは位置 b, d に伝播し，b の暫定ラベルは c, d に伝播する．このことを考慮することにより，本手法の演算は表 3，表 4 に示す演算論理により表現することができる．ただし，0, 1 はそれぞれ  $F_B, F_O$  を表し，\* は 0 または 1 を表し，NOP は演算しないことを表し， $\max(\cdot)$  は最大値をとる演算子， $\max 2(\cdot)$  は 2 番目に大きい値をとる演算子である．また， $\max T(\cdot)$ ， $\max T2(\cdot)$  は以下の式で定義される関数である．

$$\max T(p, q) = \begin{cases} p & \text{if } \max(T[p], T[q]) = T[p] \\ q & \text{if } \max(T[p], T[q]) = T[q] \end{cases} \quad (12)$$

$$\max T(p, q, r) = \begin{cases} p & \text{if } \max(T[p], T[q], T[r]) = T[p] \\ q & \text{if } \max(T[p], T[q], T[r]) = T[q] \\ r & \text{if } \max(T[p], T[q], T[r]) = T[r] \end{cases} \quad (13)$$

$$\max T2(p, q, r) = \begin{cases} p & \text{if } \max 2(T[p], T[q], T[r]) = T[p] \\ q & \text{if } \max 2(T[p], T[q], T[r]) = T[q] \\ r & \text{if } \max 2(T[p], T[q], T[r]) = T[r] \end{cases} \quad (14)$$

ハードウェア化の際には，これに基づくテーブルを用意することにより効率的な実現が可能である．

4 連結の連結成分のラベル付けを行うためには，図 15 のマスクを用いればよい．

(平成 11 年 12 月 27 日受付)

(平成 12 年 9 月 7 日採録)



鈴木 賢治 (正会員)

昭和 43 年生．平成 3 年名城大学理工学部電気電子工学科卒業．平成 5 年同大学大学院工学研究科修士課程修了．同年 (株) 日立メディコ技術研究所入社．平成 9 年愛知県立大学．現在，同大学情報科学部助手．医用画像処理・認識および学習型情報処理の研究に従事．電子情報通信学会，電気学会，日本神経回路学会，日本循環器学会，IEEE 各会員．



堀場 勇夫 (正会員)

昭和 23 年生．昭和 49 年名古屋大学工学部電気工学科卒業．同年 (株) 日立メディコ入社．昭 62 年名城大学理工学部電気電子工学科講師．平成 3 年同助教授．平成 10 年愛知県立大学情報科学部教授．現在に至る．生体情報の計測，医用画像処理，および交通系における画像処理応用に関する研究に従事．工学博士．電子情報通信学会，日本 ME 学会，日本放射線技術学会，日本超音波医学会，日本医用画像工学会，日本循環器学会等会員．



杉江 昇 (正会員)

昭和 7 年生．昭和 32 年名古屋大学工学部電気工学科卒業．同年通商産業省電子技術総合研究所入所．昭和 37~39 年カナダ・マギル大学客員研究員．昭和 45 年パイオニクス研究室長．昭和 53 年視覚情報研究室長．昭和 54 年名古屋大学大学院工学研究科情報工学専攻教授．昭和 60 年同大学工学部電気工学第二学科教授．平成 2 年同大学工学部情報工学科教授．平成 6 年名城大学理工学部電気電子工学科教授．平成 12 年同大学理工学部情報科学科教授．現在，情報科学科長．パイオニクス，医用工学，コンピュータビジョン，自然言語処理等の研究・教育に従事．工学博士．電子情報通信学会，計測自動制御学会，ロボット学会，エム・イー学会，映像情報メディア学会，日本神経回路学会，IEEE 等会員．