

境界追跡型ラベリングボード

石山豊 *1 船岡千洋 *1 久保文雄 *1 富田文明 *2 高橋裕信 *3

*1 スタンレー電気技術研究所 *2 電子技術総合研究所 *3 三洋電機筑波研究所

あらまし 本報告は、既に発表しているラスタ走査と領域の境界追跡を組み合わせてラベル付けを行なうアルゴリズムのハードウェア化に関するものである。

試作したボードは、境界追跡を利用しているため、初期ラベルを記憶する作業用メモリを必要としない。従って、回路がコンパクトであり、最大ラベル数に制限なくラベル付けができる。また、ラベル付けと同時に穴の数の抽出や、周囲長によるラベル付け制御の機能も付加することができた。汎用 IC のみで構成された本ボードでは、 512×512 画素の実画像に対して、処理時間約50msecという実験結果を得た。

Labeling Board Based on Boundary Tracking

Yutaka ISHIYAMA *1 Chihiro FUNAOKA *1 Fumio KUBO *1

Fumiaki TOMITA *2 Hironobu TAKAHASHI *3

*1 Stanley Electric Co., Ltd. R&D
Midori-ku, Yokohama Kanagawa 227, Japan

*2 Electrotechnical Laboratory
Umezono, Tsukuba Ibaraki 305, Japan

*3 Tsukuba Research Center, Sanyo Electric Co., Ltd.
Koyadai, Tsukuba Ibaraki 305, Japan

Abstract This paper describes newly developed labeling board based on algorithm of which concept has been already presented as a combination of raster scanning and boundary tracking.

The developed board employs the boundary tracking method and does not require a working memory for initial labels, and thus it can process labels in 512×512 images without restricting the maximum numbers of connected components. It can also count holes within components at the same time of labeling, and is capable to set the minimum or maximum boundary length of the region to be labeled. The processing time is 50 msec for 512×512 real images.

1. はじめに

ラベリング処理は、同じ領域に属する全ての画素に同じラベル（番号）を割り当てる処理であり、これにより各領域を区別して、特徴量（面積、重心など）を求めることができる。従って、画像処理の中でも使用頻度の高い処理のひとつであるが、処理時間がかかるという問題があり、高速化が強く望まれている。

ラベリング処理のハードウェア化については、今までにいくつかの報告(3)-(6)がされている。これらはラスタ走査のみにより、ラベル付けを行なう方法である。従って、最大ラベル数に影響する初期ラベルを記憶するための作業用メモリを必要とし回路規模が大きくなる。

本報告は、既に提案しているラスタ走査と領域の境界追跡を組み合わせてラベル付けを行なうアルゴリズム(1)-(2)のハードウェア化に関するものである。

今回試作したボードは、以下の特長を持つ。

- (1) 1回のラスタ走査で連結領域に通し番号のラベルを付ける。
- (2) 画像サイズ512×512の画像全てに対して、最大ラベル数に制限なくラベル付けする。
- (3) 実画像に対して、実用的な処理時間（約50msec）でラベル付けができる。
- (4) 画像内の穴の合計がラベル付けと同時にわかる。
- (5) 領域の境界を追跡しながらラベル付けを行なうため周囲長が求まり、それによりラベル付けを制御できる。（ノイズ除去）
- (6) 初期ラベルを記憶しておくための作業用メモリを必要としないので、回路がコンパクトである。
- (7) 専用プロセッサを使用せずに、汎用ICのみで構成されている。

以下に、基本アルゴリズムの概要を述べ、試作したボードの回路構成と実験結果を示す。

2. 境界追跡型ラベル付けアルゴリズム

図1(a)に示す2値画像に対して、4連結

```

0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 R R R R R R R R R R
0 0 R R R R R R R R R R R
0 R R R R R R R R R R R R
0 R R R R 0 0 0 0 R R R R
0 R R R R 0 R R R 0 R R R R
0 R R R R 0 R R R 0 R R R R
0 R R R R R R R R R R R R
0 R R R R R R R R R R R R
0 0 0 0 0 0 0 0 0 0 0
    
```

(a)

```

      N R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R
    
```

(b)

```

      N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R
    
```

(c)

```

      N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R R R R R
N R R R R R
    
```

(d)

```

      N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N N N N N
N N N N N N
    
```

(e)

図1 処理過程

でラベリング処理を行なうことを考える。領域はR、領域以外は0とする。はじめにラベルの最大値(N)を0に初期化し、画像のラスタ走査を開始する。そして、走査点が領域(R)になった時、図3に示す近傍画素のデータに基づいて次の3通りに分類して処理を行なう。

(a) 外周境界追跡

図3(a)に示す走査点(アンダーライン)はラベル付けされていない領域の最初の点で、領域の外周境界線上の点である。そこで $N = N + 1$ としラベル値を更新し、走査点を境界追跡の始点として外周境界を時計回りに追跡する。このとき領域の境界点は、表1の条件(方向コードは図2参照)によりラベル付けされる。その結果図1(b)のように、少なくとも左側にエッジ(0)のある外周境界線上の点にはラベル(N)が付くことになる。そして、元のラスタ走査に戻る。

(b) ラスタ走査

図3(b)に示す走査点は領域(N)内の点であり、走査点にラベル(N)を付ける。その結果を図1(c)に示す。

(c) 内周境界追跡

図3(c)に示す走査点は領域(N)内の穴に隣接する内周境界線上の点である。そこで、走査点を境界追跡の始点として内周境界を反時計回りに追跡する。このとき、領域の境界点は表1の条件によりラベル付けされる。その結果、図1(d)のように、左隣にエッジ(0)のある内周境界線上の点にだけラベル(N)が付くことになる。そして、元のラスタ走査に戻る。

以上の処理を画像の右下まで繰り返すと、1回のラスタ走査で画像内の各領域に1からN(領域数)の通し番号を付けることができる。

この処理は、境界追跡によりラン(同じ領域に属する連続点)の左端に確定したラベルを付けた後、ラスタ走査で横方向のランにラベル付けしているのである。表1に示すように境界追跡中にラベルを付けない理由は、常にラベルを付けるようにすると、図4のように幅が1の領域部分があると、外周境界追跡時にラベルが付くために穴を見つけることができず、誤ったラベルを付ける場合があるためである。

8連結の場合も条件を変えることによって、同様にラベル付けできる。

	3			5	6	7
2	*	0		4	*	0
	1			3	2	1

(4連結) (8連結)

図2 方向コード

0		N
0 <u>R</u>	<u>NR</u>	0 <u>R</u>
(a)	(b)	(c)
	(4連結)	
0		N
0 <u>R</u>	<u>NR</u>	0 <u>R</u>
(d)	(e)	(f)
	(8連結)	

図3 ラベル付け開始条件

		出力方向			
		0	1	2	3
入力方向	0				
	1	×	×	○	
	2	○	×	○	
	3	○	○	○	

(4連結)

		出力方向							
		0	1	2	3	4	5	6	7
入力方向	0						-		
	1	×	×	×	×	○	-	×	
	2	-	×	×	×	○	○	-	
	3	-	×	×	×	○	○	○	
	4	○	-	-	×	○	○	○	
	5	○	○	-	×	○	○	○	
	6	○	○	○	-	-	○	○	○
7	○	○	○	○	-	○	○	○	

(8連結)

表1 ラベル付け条件

- : ラベルを付ける
- × : ラベルを付けない
- 空 : どちらでも良い
- : 存在しない

```

R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R
R R R R R R R R R R

```

図4 幅1の部分がある領域

3. 回路構成

本ボードは、ラベリング処理を行なう際に、周囲長によるラベル付けの制御と、画像内の穴の数の抽出を行なうことを付加した。前者は、領域の境界追跡の際に周囲長を調べ、予め設定しておいた周囲長しきい値より大きい領域（または小さい領域）にだけラベルを付けることができる。後者は、図3(c, f)の条件を満たした時の回数を数えることにより求まる。ボードの回路構成を図5に示し、4連結の場合について個々の機能を説明する。

○対象画像メモリ

対象画像を格納しておくための2値画像メモリで、ラベル付けを行なう対象領域を値1とし、それ以外を値0とする。

○ラベル画像メモリ

処理結果のラベル画像を格納するための多値画像メモリで、領域以外は値0が、各領域は通し番号1～Nのラベル値が格納される。

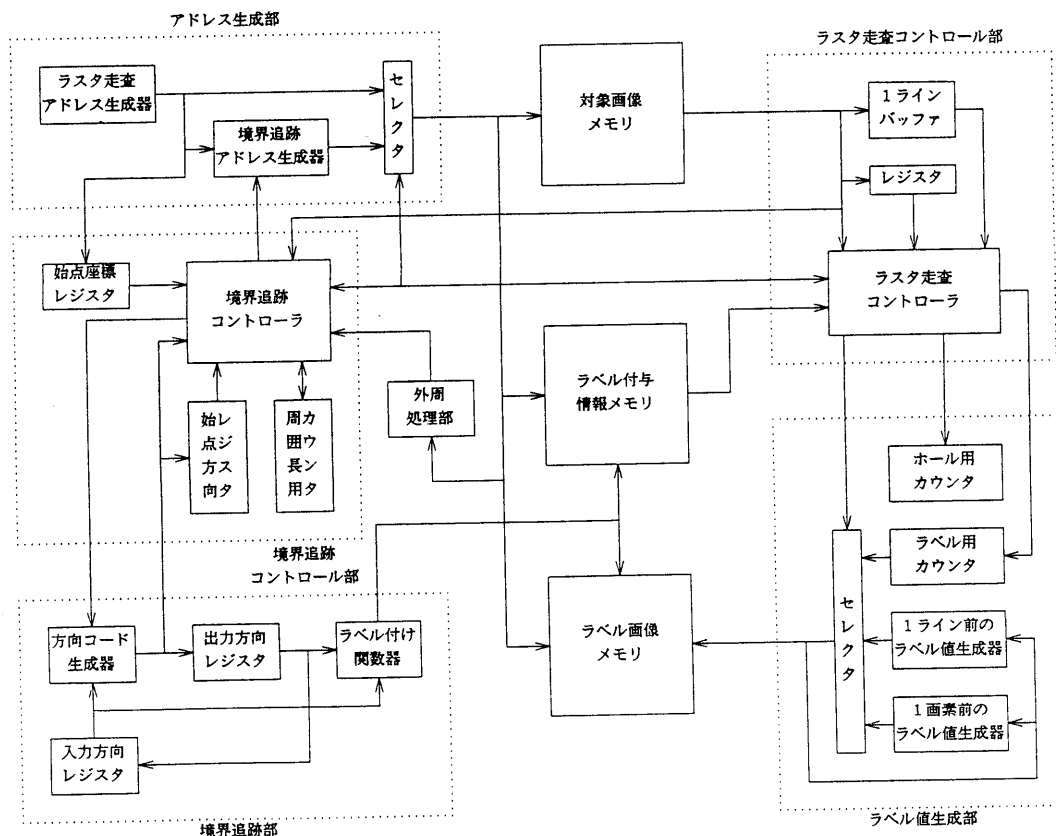


図5 回路構成

○ラベル付与情報メモリ

走査点がすでに領域の境界追跡により、ラベル付けされているかどうかを記憶しておくための2値画像メモリで、未処理を値0とし、処理済を値1とする。

○ラスタ走査コントロール部

対象画像メモリをラスタ走査し、ラベル付与情報メモリを参照しながら、ラベル値生成部と境界追跡コントロール部を制御する。

○ラベル値生成部

ラスタ走査コントロール部からの指示により、ラベル用カウンタ、ホール用カウンタを制御し、付与するラベル値を選択する。

○境界追跡コントロール部

ラスタ走査コントロール部からの指示により、始点から再び始点にもどるまで、領域の境界を周囲長用カウンタによってカウントしながら追跡する。

○境界追跡部

方向コード生成器により、追跡された境界点から次の境界点を選択する。その際に、ラベル付け関数器はラベル付けを制御する。

○アドレス生成部

ラスタ走査時のアドレスと境界追跡時のアドレスを生成する。ラスタ走査の時はラスタ走査アドレスが生成され、境界追跡の時は基準点（追跡された境界点）を中心としてそれぞれの方向のアドレスを生成する。

○外周処理部

走査点が外周になった時、画像の外1ライン分を値0と仮想して例外処理を行なう。

処理の手順を以下に示す。

(1) 初期設定

初期設定としてラベル用カウンタ、ホール用カウンタをクリアし、対象画像メモリに対象画像を、ラベル画像メモリとラベル付与情報メモ

りに値0をそれぞれ格納する。また、周囲長しきい値条件を設定する。

(2) ラスタ走査

ラスタ走査コントローラによりラスタ走査を開始して、対象画像メモリとラベル付与情報メモリを参照する。対象画像メモリの値が1（領域）で、ラベル付与情報メモリの値が0（未処理）の時、ラスタ走査を一旦停止する。そして図3に示した近傍の画素値により3通りに分類する。（a）の場合、（3）により領域の境界追跡を開始する。（b）の場合、ラベル値生成部はラベル画像メモリに記憶されている1つ前のラベル値（N）を生成し、ラベル画像メモリに書き込む。そして再びラスタ走査を行なう。（c）の場合、ラベル値生成部はラベル画像メモリに記憶されている1ライン前のラベル値（N）を生成し、ホール用カウンタをインクリメントする。そして（3）により領域の境界追跡を開始する。走査点が画像の右下に達した時、処理は終了し、結果として周囲長しきい値条件を満足したラベル画像とラベル数、ホール数が同時に得られる。

(3) 境界追跡制御

境界追跡コントロール部は始点の位置と追跡方向をそれぞれ始点位置レジスタと始点方向レジスタに記憶し、境界追跡を開始する。位置と追跡方向の両方の値が始点位置レジスタと始点方向レジスタの値と一致するまで（4）により境界追跡を繰り返す。図3（a）の場合、追跡の際にステップ数を周囲長用カウンタでカウントする。領域の境界を1周し、周囲長しきい値条件を満足していたら、ラベル値生成部はラベル用カウンタをインクリメントし、再び境界を1周して、そのラベル値をラベル画像メモリに書き込む。また、図3の条件（c）の場合、領域の境界を1周して、（2）で生成されたラベル値をラベル画像メモリに書き込む。追跡が終了したら（2）に戻り再びラスタ走査を行なう。

(4) 境界追跡

対象画像メモリの基準点から近傍画素を表2に示す優先順位に従って参照し、最初に値1（領

域)となる方向を出力方向レジスタに記憶する。入力方向レジスタの値と出力方向レジスタの値が表1の条件を満足した時、ラベル値生成部で生成されたラベル値をラベル画像メモリに、値1(処理済)をラベル付与情報メモリにそれぞれ書き込む。そして出力方向レジスタの値を次の境界の入力方向として追跡を繰り返す。また、全ての方向の近傍画素が値0(領域以外)である時は、孤立点であり、周囲長しきい値条件を満足していたら、ラベル値をラベル画像メモリに書き込む。

4. 性能評価と実験結果

試作したボードの概要を以下に示す。

4-1 基本仕様

画像サイズ 512 × 512
 入力画像 2値画像
 ラベル画像 多値画像(17bit)
 連結性 4連結/8連結
 最大ラベル数 131, 072
 (画像サイズ512 × 512における最大値)
 最大ホール数 130, 050(同上)

4-2 回路規模

ボード外形寸法(写真1参照)
 233 [mm] × 160 [mm] × 20 [mm]
 (VMEbusダブル・ハイ・ボード
 1スロット幅)
 部品数 汎用IC 140個程度
 基本マシンサイクル 125 nsec

4-3 処理時間

処理時間は対象画像の複雑さにより変化するが、概算は次式で表すことができる。

$$\begin{aligned} & \text{(画像サイズ)} \\ & + 3 \times \text{領域の全画素数} \\ & + 2A \times \text{外周の境界追跡のステップ数} \\ & + A \times \text{内周の境界追跡のステップ数} \times M \end{aligned} \quad \text{--- (1)}$$

Aは近傍画素の平均参照回数で、4連結の場合は2、8連結の場合は4に近似できる。また、Mは基本マシンサイクルを示している。

4-4 実験結果

サンプル画像として粒子画像(サンプル1:図6)、文字画像(サンプル2:図7)、線画像(サンプル3:図8)と、式(1)から本方式でいちばん処理時間がかかると考えられる線幅、線間ともに1の渦巻状のテストパターン(サンプル4:図9は原画を4倍に拡大)について実験を行なった。全ての領域にラベル付けた時の結果を表3に示す。また、サンプル1に対し、小領域を除去(周囲長しきい値:50)してラベル付けた時の結果を表4に示す。表3の結果から実画像に対しては、50 msec程度で処理することがわかる。また、表4の結果から小領域(ノイズ)を除去したラベル画像が得られ、画像内の粒子の数がわかる。そして、これにより処理時間が減少することも示している。

入力方向	参照方向の優先順位
0	3 → 0 → 1 → 2
1	0 → 1 → 2 → 3
2	1 → 2 → 3 → 0
3	2 → 3 → 0 → 1

(4連結)

入力方向	参照方向の優先順位
0	5 → 6 → 7 → 0 → 1 → 2 → 3 → 4
1	6 → 7 → 0 → 1 → 2 → 3 → 4 → 5
2	7 → 0 → 1 → 2 → 3 → 4 → 5 → 6
3	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7
4	1 → 2 → 3 → 4 → 5 → 6 → 7 → 0
5	2 → 3 → 4 → 5 → 6 → 7 → 0 → 1
6	3 → 4 → 5 → 6 → 7 → 0 → 1 → 2
7	4 → 5 → 6 → 7 → 0 → 1 → 2 → 3

(8連結)

表2 参照方向の優先順位

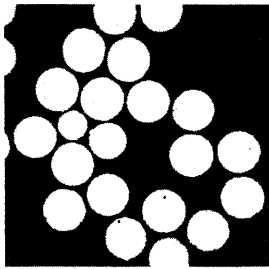


図6 サンプル1

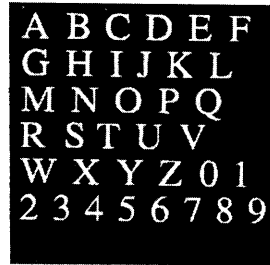


図7 サンプル2

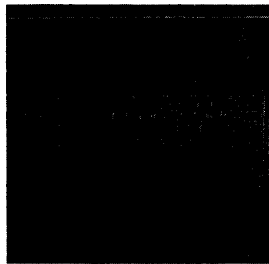


図8 サンプル3

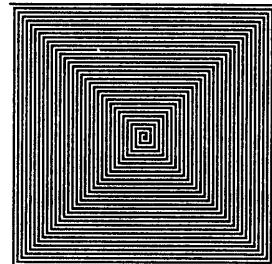


図9 サンプル4

対象画像	連結数	ラベル数	ホール数	処理時間 [m s e c]
サンプル1	4	42	16	59 (770)
	8	29	24	60 (800)
サンプル2	4	36	14	46 (730)
	8	36	14	49 (780)
サンプル3	4	2080	0	52 (830)
	8	631	52	60 (850)
サンプル4	4	1	0	258 (3460)
	8	1	0	389 (5050)

処理時間の () 内のデータはSUN3-260(4MIPS)によるソフトウェアでの処理時間である。

表3 処理時間

対象画像	連結数	ラベル数	ホール数	処理時間 [m s e c]
サンプル1	4	24	16	58
	8	24	24	59

表4 処理時間 (周囲長50未満除去)

5. まとめ

領域の境界を追跡しながらラベル付けを行なうアルゴリズムをハードウェア化する方法を示し、その試作実験を行なった。

本アルゴリズムは、初期ラベルを記憶しておくための作業用メモリを必要としないので、回路規模が小さくなりコンパクトにまとまった。さらに、画像サイズ512×512の画像に対して、最大ラベル数に制限なく実用的な処理時間でラベル付けができる。また、領域の境界を追跡しながらラベル付けするため、周囲長によりラベル付けを制御でき、これにより微小ノイズを除去したラベル画像が得られる。なお、本ボードを使用するには、画像バス制御とVMEインターフェイス用のボードを必要とする。

謝辞

本ボードの開発にあたり、御協力頂いたスタンレー電気(株)技術研究所の関係者各位に感謝致します。

参考文献

- (1) H.Takahashi, F.Tomita: "Fast Region Labeling with Boundary Tracking" ICIP'89, pp.369-373 (1989)
- (2) 高橋, 富田: 境界追跡を用いたラベル付けアルゴリズム, 情処学会35回全国大会, 4J-4(1987)
- (3) 太田, 古明地, 後藤, 吉田: ビデオ・レートラベリングプロセッサ, 情処学会CV研, 54-1(1988)
- (4) 後藤, 太田, 吉田, 白井: 連結領域の高速ラベル付けアルゴリズム, 信学会論文誌, Vol.J72-D, No.2, pp.247-255(1989)
- (5) 榭井, 那須, 志水: 内容照合型LUTを用いたラベリング方式, 信学会, PRU 87-79(1987)
- (6) 藤原, 磯貝: 高能率ラベリングプロセッサのための一手法, 情処学会CV研, 56-1 (1988)

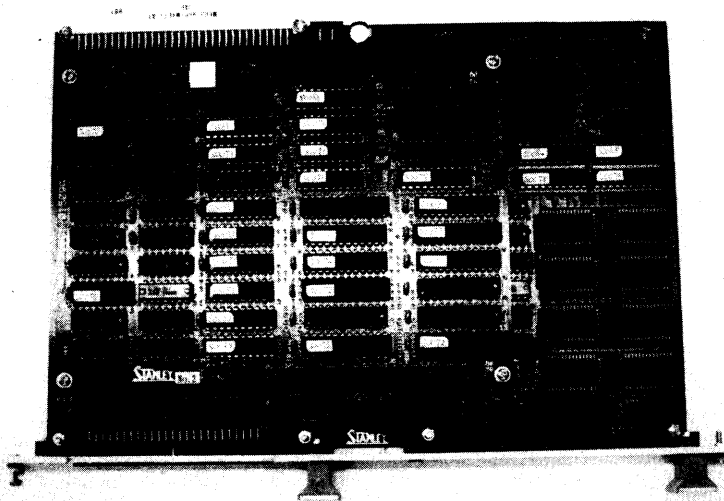


写真1