

ラスタ走査型ラベル付けアルゴリズムにおける新しい第1走査手法

何 立 風^{†1} 巢 宇 燕^{†2}
鈴 木 賢 治^{†3} 中 村 剛 士^{†4}

従来のラスタ走査型ラベル付けアルゴリズムの第1走査では、画像を1行ずつ走査し、画素ごとに処理する。注目画素に対して、走査済みの4つの隣接画素からなるマスクを用いて、暫定ラベル付けと連結性処理を行う。本論文では、画像を2行ずつ走査し、走査済みの5つの隣接画素からなるマスクを用いて画素を2つずつ処理する手法を提案する。それによって暫定ラベル付けと連結関係検出のための画素のチェック回数を減少させ、ラベル付けの効率化を実現した。また実験によって、様々な性質の画像を用いた従来手法との比較評価を行い、提案手法の有効性を示した。

A New First-scan Strategy for Raster-scan-based Labeling Algorithms

LIFENG HE,^{†1} YUYAN CHAO,^{†2} KENJI SUZUKI^{†3}
and TSUYOSHI NAKAMURA^{†4}

This paper proposes a new first-scan method for label-equivalence-based labeling algorithms. During the first scan, conventional label-equivalence-based labeling algorithms process image lines one by one, using the mask consisting of four processed 8-connectivity neighbor pixels. In comparison, our proposed method processes image lines two by two, using the mask consisting of five processed 8-connectivity neighbor pixels. By our method, the number of times for checking pixels for assigning provisional labels and finding label equivalences will decrease, thus, the efficiency of labeling can be improved. Experimental results demonstrated that our method is effective for label-equivalence-based labeling algorithms.

1. ま え が き

パターン認識やコンピュータビジョンにおいて、2値画像中の対象物（連結成分）に別々のラベルを付けるラベル付けは、最も基本的かつ重要な処理の1つである¹⁾。

これまでに多くのラベル付けアルゴリズムが提案されており、それらはラスタ走査型^{2),4),7)}と輪郭追跡型^{3),6)}に大別できる。輪郭追跡型アルゴリズムは画像の走査が不規則となるため、並列処理やハードウェア実現に向かないうえ、効率上もラスタ走査型アルゴリズム⁴⁾に及ばない。そのため現在では、ラスタ走査型が主流になっている。

従来のラスタ走査型アルゴリズムの第1走査では、画像を1行ずつ走査し、画素ごとを処理する。注目画素が背景画素であるとき、何もしない。注目画素が対象物画素であるとき、図1に示すマスクにある画素、つまり、注目画素の左、左上、上および右上にある走査済みの4つの隣接画素をチェックし、注目画素の暫定ラベル付けおよび注目画素とマスク内の対象物画素間の連結性処理を行う。

しかしながら、注目画素を処理するとき、注目画素と注目画素の左下、下および右下の画素の連結性も容易に分かる。そのため、次の行の走査で、それらの画素間の連結関係を調べ

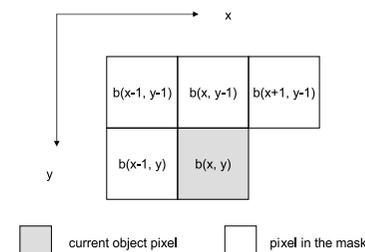


図1 8連結用走査マスク

Fig.1 Mask for the eight-connected connectivity.

†1 愛知県立大学大学院情報科学研究科
Graduate School of Information Science and Technology, Aichi Prefectural University
†2 名古屋産業大学大学院環境情報マネジメント研究科
Graduate School of Environment Management, Nagoya Sangyo University
†3 シカゴ大学大学院放射線医学研究科
Department of Radiology, The University of Chicago
†4 名古屋工業大学大学院工学研究科
Graduate School of Information Science, Nagoya Institute of Technology

て処理する方法より、注目画素を処理するとき一緒に処理しておいたほうが効率的であると思われる。

本論文では、画像を 2 行ずつ走査し、画素を 2 つずつ処理するアルゴリズムを提案する。それによって、暫定ラベル付けと連結関係の検出を効率的に処理できる。ここでは、様々な性質のテスト画像を用いて従来の最速手法⁴⁾との比較評価実験により、提案手法の有効性を示す。

2. 提案手法の概要

$N \times M$ 画素の 2 値画像中の対象物画素の値を 1、背景画素の値を 0 とする。 (x, y) ($0 \leq x \leq N - 1, 0 \leq y \leq M - 1$)にある画素を $b(x, y)$ 、その画素の値を $v(x, y)$ で表す。また、画像輪郭上の画素は背景画素とし、画素間の 8 連結性のみ考える。

本論文では、図 2 に示した新しいマスクを用いて、画像を 2 行間隔で走査し、2 つの注目画素 b (注目画素 1 と呼ぶ) と b_0 (注目画素 2 と呼ぶ) を同時に処理する手法を提案する。ここでは、 $b, b_0, b_1 \sim b_5$ がそれぞれ画素 $b(x, y), b(x, y + 1), b(x - 1, y + 1), b(x - 1, y), b(x - 1, y - 1), b(x, y - 1)$ と $b(x + 1, y - 1)$ を表す。

文献 4) により、ラスタ走査型アルゴリズムの第 1 走査において、注目画素は背景画素であるとき、何もしなくてもよい。注目画素が対象物画素であるとき、図 1 に示したマスクに対象物画素があるかどうかを調べる。マスクに対象物画素が存在しない場合、注目対象画素に新たな暫定ラベルを付ける。それ以外の場合、注目対象画素にマスクにある任意の暫定ラベルを付けてもよい。つまり、走査中に任意の時点で、処理済みの領域にある各連結成分

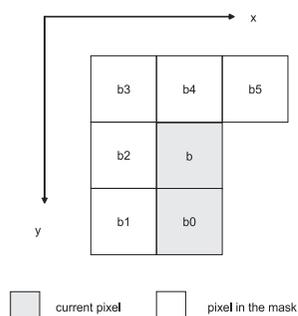


図 2 本手法に用いられる 8 連結用走査マスク
Fig. 2 Mask used in our proposed method.

の対象物画素に付与されたすべての暫定ラベルは同一であると見なすことができる。また、マスクに元々は独立な 2 つの画素^{*1}は注目画素を通して連結になった場合のみ、両画素と関連する連結体間の連結関係の解決を考慮する必要がある。一方、文献 5) により、注目対象物画素を処理するとき、注目画素が背景画素の後にあるか対象物画素の後にあるかに分けて処理するのが効率である。提案手法も同様の手法を用いて処理を行う。

まず、注目画素 1 の処理について考える。注目画素 1 が背景画素の後にある対象物画素である場合、図 3 に示す 16 ケースがある。ケース (1) では、マスクに注目画素と 8 連結する処理済みの対象物画素がないため、注目画素 1 に新たなラベルを付与する。それ以外の場合、注目画素 1 に、注目画素と 8 連結する処理済みの任意の対象物画素のラベルを付与できる。一方、元々独立していた対象物画素が注目画素の存在で連結になったのは、ケース (4) (画素 b_1 と画素 b_3)、ケース (6) (画素 b_1 と画素 b_4)、ケース (8) (画素 b_1 と画素 b_4, b_3 の連結体)、ケース (10) (画素 b_1 と画素 b_5)、ケース (11) (画素 b_3 と画素 b_5)、ケース (12) (画素 b_1 と画素 b_3 および画素 b_5)、ケース (14) (画素 b_1 と画素 b_4, b_5 の連結体)、ケース (15) (画素 b_1 と画素 b_3, b_4, b_5 の連結体) である。文献 4) に示したカルノー図を用いた分析手法で、図 4 に示す $v(x, y)$ を処理する手続き $process1(x, y)$ が得られる。ただし、 $resolve(a, b)$ は暫定ラベル a と b の連結性を解決する操作を表し、 m は暫定ラベル変

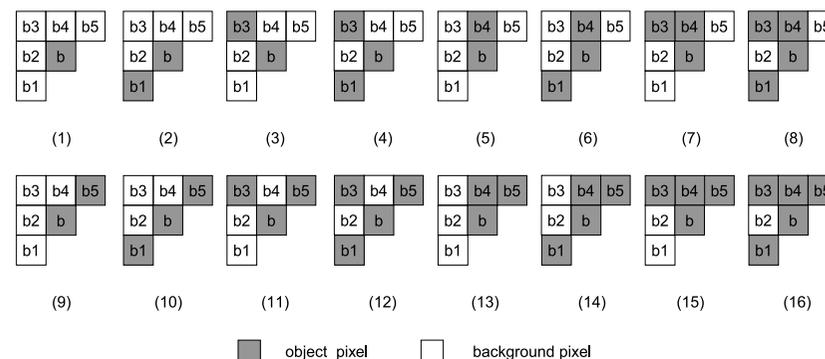


図 3 注目対象物画素が背景画素後にあるとき、マスク中にとりうる 16 のケース

Fig. 3 Sixteen possible cases for the current object pixel following a background pixel in the mask.

*1 本論文では、マスクに連結している対象物画素のかたまりを連結体と呼ぶ。連結体の各画素に付与された暫定ラベルは同一である。

```

1:  if (v(x - 1, y + 1) > 0)
2:      v(x, y) = v(x - 1, y + 1);
3:      if (v(x, y - 1) > 0)
4:          resolve(v(x - 1, y + 1), v(x, y - 1));
5:      else
6:          if (v(x - 1, y - 1) > 0)
7:              resolve(v(x - 1, y + 1), v(x - 1, y - 1));
8:          end if
9:          if (v(x + 1, y - 1) > 0)
10:             resolve(v(x - 1, y + 1), v(x + 1, y - 1));
11:          end if
12:         end if
13:     else if (v(x, y - 1) > 0)
14:         v(x, y) = v(x, y - 1);
15:     else if (v(x - 1, y - 1) > 0)
16:         v(x, y) = v(x - 1, y - 1);
17:         if (v(x + 1, y - 1) > 0)
18:             resolve(v(x - 1, y - 1), v(x + 1, y - 1));
19:         else if (v(x + 1, y - 1) > 0)
20:             v(x, y) = v(x + 1, y - 1);
21:         else
22:             v(x, y) = m;
23:             m = m + 1;
24:         end if

```

図 4 処理手続き $process1(x, y)$
Fig. 4 $process1(x, y)$.

数 (初期値は 1) である。

$process1(x, y)$ はまず、画素 $b1$ が対象物画素であるかどうかをチェックする (1 行目)。 $b1$ が対象物画素であれば、注目画素 1 に $b1$ の暫定ラベルを付与する (2 行目)。そして、 $b4$ が対象物画素であれば (ケース (6), (8), (14) および (16)), 画素 $b1$ と画素 $b4$ 間の連結性を解決して (4 行目) 終了する。画素 $b4$ が背景画素であるとき、画素 $b3$ が対象物画素であれば (ケース (4) と (12)), 画素 $b1$ と画素 $b3$ 間の連結性を解決し (7 行目)、画素 $b5$ が対象物画素であれば (ケース (10) とケース (12)), 画素 $b1$ と画素 $b5$ 間の連結性を解決して (10 行目) 終了する。また、画素 $b3, b4$ および $b5$ はともに背景画素であるとき (ケース (2)), 連結性の考慮は必要なく、そのまま終了する。

一方、 $b1$ が背景画素であれば、画素 $b4$ が対象物画素であるかどうかをチェックする (3 行目)。 $b4$ は対象物画素であれば (ケース (5), (7), (13) および (15)), 注目画素 1 に $b4$

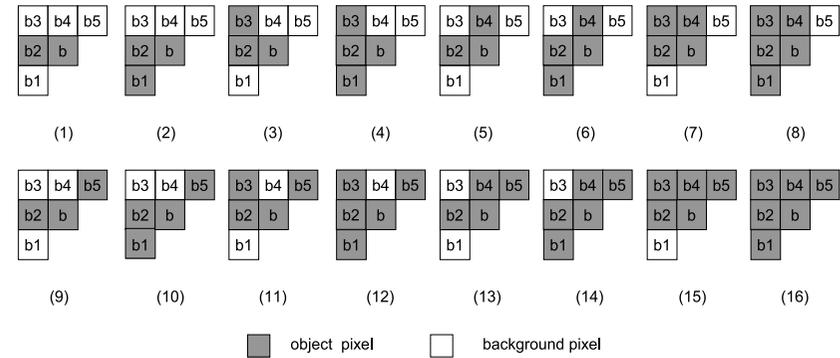


図 5 注目対象物画素が対象物画素後にあるときの 16 のケース
Fig. 5 Sixteen possible cases for the current object pixel following a background pixel in the mask.

の暫定ラベルを付与して (14 行目) 終了する。 $b4$ は背景画素であれば、画素 $b3$ が対象物画素であるかどうかをチェックする (15 行目)。 $b3$ は対象物画素であれば (ケース (3) とケース (11)), 注目画素 1 に $b3$ の暫定ラベルを付与し (16 行目)、 $b5$ は対象物画素である場合 (ケース (11)), 画素 $b3$ と画素 $b5$ 間の連結性を解決して (18 行目) 終了する。画素 $b3$ も背景画素であれば、画素 $b5$ が対象物画素であるかどうかをチェックする (19 行目)。 $b5$ は対象物画素である場合 (ケース (9)), 注目画素 1 に $b5$ の暫定ラベルを付与して (20 行目) 終了する。

画素 $b1, b3, b4$ および $b5$ はともに背景画素であれば (ケース (1)), マスクに対象物画素が存在しない。そのとき、注目画素 1 に新たな暫定ラベルを付与し (22 行目)、暫定ラベル変数 m の値を 1 増やして (23 行目) 終了する。

注目画素 1 は対象物画素の後にある (つまり、画素 $b2$ は対象物画素である) 対象物画素である場合、図 5 に示す 16 ケースがある。すべてのケースにおいて、注目画素 $b1$ に対象物画素 $b2$ に付与された暫定ラベルを付与すればよい。また、 $b4$ は背景画素であるかつ画素 $b5$ が対象物画素であるとき (つまり、ケース (9), (10), (11) および (12)) のみ、画素 $b2$ と $b5$ の連結性を解決する必要がある。この場合の注目画素 1 の処理手続き $process2(x, y)$ を図 6 に示す。

次に、注目画素 2 が対象物画素であるときの処理について議論する。このとき、注目画素 2 の存在で、マスクに元々独立な対象物画素が連結するようになったことがないから、連結性を考えなくてもよい。また、画素 $b0$ の暫定ラベル付けは、その画素と隣接する 4 つの画

```

1:  $v(x, y) = v(x - 1, y);$ 
2: if  $(v(x, y - 1) < 0 \text{ and } v(x, y + 1) > 0)$ 
3:    $resolve(v(x - 1, y), v(x, y + 1));$ 
4: end if
    
```

図 6 処理手順 2
Fig. 6 $process2(x, y)$.

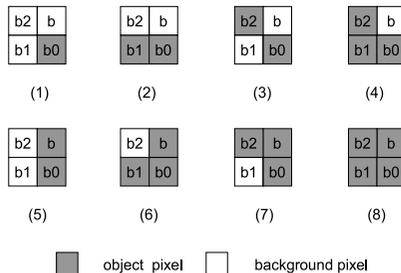


図 7 注目対象物画素 2 が対象物画素であるときの 8 ケース

Fig. 7 Eight possible cases for the current object pixel following a background pixel in the mask.

素の状態を考えればよい。そのとき、図 7 に示す 8 ケースがある。注目画素 1 が対象物画素であるとき（ケース (5), (6), (7) および (8)）、注目画素 2 に注目画素 1 の暫定ラベルを付与するだけで済む。注目画素 1 が対象物画素の後にある背景画素、つまり、画素 b_2 が対象物画素であれば（ケース (3) と (4)）、注目画素 2 に画素 b_2 の暫定ラベルを付与すればよい。上記以外の場合（つまり、ケース (1) と (2)）、画素 b_1 が対象物画素であるとき（ケース (2)）、注目画素 2 に画素 b_1 の暫定ラベルを付与する。一方、注目画素 2 の前にある画素が背景画素であれば（ケース (1)）、注目画素 2 に新たな暫定ラベルを付与する。

3. 比較評価実験

提案手法の性能を評価するために、ワークステーション (Intel Pentium Duo 930 3.0 GHz + 3.0 GHz CPUs, 2 GB Memory, Mandriva Linux OS) 上で従来手法と提案手法を C 言語で同じコーディング方法と同じコンパイラコマンドを用いて実装し、両手法の CPU 実行時間を用いた比較評価実験を行った。なお、提案手法は第 1 走査における効率化手法であるため^{*1}、実行時間は第 1 走査のみの時間とし、シングルコアを用いた 5,000 回の実行の平均値である。

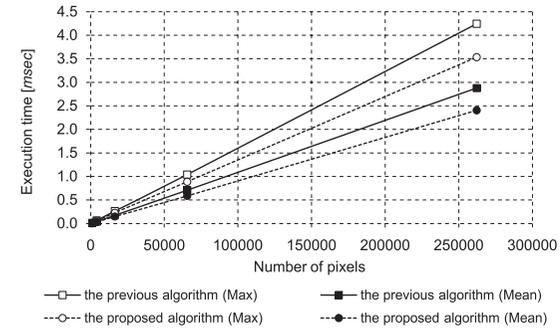


図 8 画像サイズに対する実行時間の線形性
Fig. 8 Linearity of the execution time versus image size.

実験には次の 5 種類の画像を用いた。(1) ノイズ画像：5 つのサイズ (32 × 32, 64 × 64, 128 × 128, 256 × 256 と 512 × 512 画素) の画像に、1,000 階調までの白色雑音を加え、閾値を 0 から 1,000 まで 25 ごとに变化させて 2 値化した合計 205 種類の 2 値画像 (各サイズごとに 41 種類)。(2) 自然画像：SIDBA (Standard Image Data Base) および USC (University of Southern California) の標準画像データベースから風景画像、航空画像、人物画像、静物画像、指紋画像、スナップ画像およびテキスト画像を含む 50 種類の自然画像を Otsu の閾値選択法⁸⁾により 2 値化した画像。(3) 医用画像：シカゴ大学医用画像データベースから 25 種類の医用画像を Otsu 法により 2 値化した画像。(4) Columbia-Utrecht Reflectance and Texture Database⁹⁾ から 7 種のテクスチャ画像を Otsu 法により 2 値化した画像。(5) テストパターン画像：ラベル付けアルゴリズムを評価するため作成した、階段状、渦状、のこぎり状、チェッカーボード状、蜂巣状パターンを含む人工画像⁷⁾。

文献 5) で提案されたラベル付け手法はこれまでの最速手法であるから、この手法 (従来手法と呼ぶ) を今回の比較対象とした。

図 8 に、ノイズ画像における画像サイズに対する両手法の実行時間特性を示す。提案手法も線形特性を持つことが確認できる。また、図 9 は、512 × 512 のノイズ画像における対象物画素の数に対する両手法の実行時間特性を示す。

自然画像、医用画像、テクスチャ画像と人工画像を用いた実験結果を表 1 に示す。すべ

*1 提案手法の第 2 走査は従来手法の第 2 走査とまったく同じである。

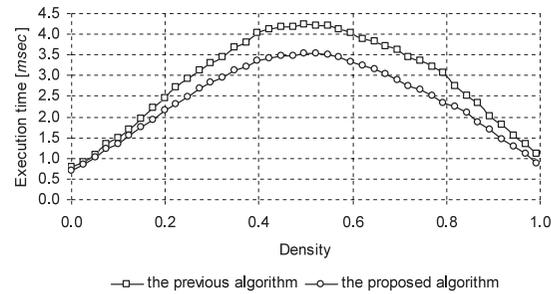


図 9 対象物画素の数に対する実行時間の特性

Fig. 9 The execution time versus the number of object pixels in an image.

表 1 自然画像, 医用画像, テクスチャ画像, テストパターン画像における実験結果 (ミリ秒)

Table 1 Comparison of various execution times [msec] on natural images, medical images, textural images, and special-shape artificial images.

画像種類		従来手法	提案手法
natural	max.	2.18	2.04
	mean	1.32	1.28
	min.	0.87	0.84
medical	max.	1.31	1.27
	mean	1.09	1.03
	min.	0.90	0.87
textural	max.	1.80	1.68
	mean	1.38	1.31
	min.	1.05	0.97
artificial	max.	1.53	1.32
	mean	0.83	0.79
	min.	0.26	0.24

ての種類の画像において, 提案手法が有効であることが分かる.

4. 考 察

本章は提案手法の効率化の要因について分析する. 便宜上, マスク中の各パターンが同確率での現れるものとする.

ラスタ走査型ラベル付けアルゴリズムでは, 第 1 走査で対象物画素を処理するとき, 走査マスクにある画素をチェックしながら暫定ラベル付けと連結性検出を行う. 従来手法の中にマスクにある画素のチェックの平均回数が最も少ないのは 1.75 回である⁵⁾.

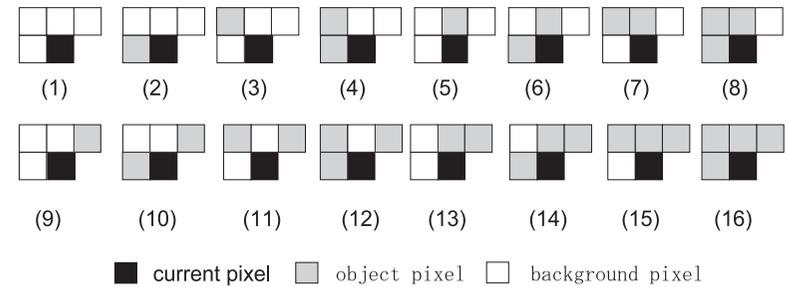


図 10 注目対象が対象物画素であるとき, 従来手法のマスクにおける 16 ケース

Fig. 10 Sixteen possible cases for the current object pixel in convention labeling raster-scan algorithms.

提案手法における対象物画素を処理するとき, 暫定ラベル付けと連結性検出のため隣接する画素をチェックする必要な回数を分析する.

まず, 注目画素 1 が対象物画素であるときの平均チェック回数を計算する. 注目画素 1 が背景画素の後にあるとき, $process1(x, y)$ より, 図 3 に示したケース (1), (2), (3), (4), (9), (10), (11) および (12) のとき 4 回, ケース (5), (6), (7), (8), (13), (14), (15) および (16) のとき 2 回となる. 注目画素 1 が対象物画素の後にあるとき, $process2(x, y)$ より, 図 5 に示したケース (1), (2), (3), (4), (9), (10), (11) および (12) のとき 2 回, ケース (5), (6), (7), (8), (13), (14), (15) および (16) のとき 1 回となる. その平均回数は $(4 \times 8 + 2 \times 8 + 2 \times 8 + 1 \times 8) / 32 = 2.25$ 回となる.

次は, 注目画素 2 が対象物画素であるときの平均チェック回数を計算する. 図に示した第 1 走査の処理手続きより, 注目画素 1 が対象物画素であるとき, つまり, 図 7 に示したケース (5), (6), (7) および (8) のとき 0 回, 注目画素 1 が対象物画素の後にある背景画素であるとき, つまり, 図 7 に示したケース (3) および (4) のときも 0 回, それ以外のとき, つまり, 図 7 に示したケース (1) および (2) のとき 1 回となる. その平均回数は $(0 \times 4 + 0 \times 2 + 1 \times 2) / 8 = 0.25$ 回となる.

注目画素 1 と注目画素 2 が対象物画素である確率は同じとすれば, 提案手法の対象物画素を処理するため, 隣接する画素をチェックする必要な平均回数は $(2.25 + 0.25) / 2 = 1.25$ 回となり, 従来法中最良の 1.75 回よりさらに約 28.6% 減少した.

一方, 1 つの対象物画素を処理するため, 連結性を解決する操作 $resolve$ を呼ぶ平均回数について考える. 従来法では, 対象物画素を処理するとき図 10 に示す 16 ケースがある.

表 2 ノイズ画像における対象物画素を処理するための主な演算の平均回数
Table 2 Average times of main operations for object pixels on noise images.

	従来手法	提案手法
比較演算	1.978	1.317
代入演算	3.325	3.362
<i>resolve</i> の呼び出し	0.129	0.135

そのなか, ケース (10), (11) および (12) の場合, 操作 *resolve* が 1 回必要となる. その他のケースは必要がない. その平均回数は $3/16 \approx 0.19$ 回となる.

提案手法の場合, 注目対象物画素 1 において, 図 3 に示す 16 ケースのなか, ケース (4), (6), (8), (10), (11), (14) および (15) のとき, 操作 *resolve* が 1 回, ケース (12) のとき, 操作 *resolve* が 2 回必要となる. また, 図 5 に示す 16 ケースのなか, ケース (9), (10), (11) および (12) のとき, 操作 *resolve* が 1 回必要となる. その他のケースは必要がない. その平均回数は $(1 \times 7 + 2 \times 1 + 0 \times 9 + 1 \times 4 + 0 \times 12) / 32 = 13/32$ 回となる. 注目対象物画素 2 において, *process2* により, 連結性解決が必要ないので, 操作 *resolve* を呼ぶ回数は 0 となる.

注目画素 1 と注目画素 2 が対象物画素である確率は同じとすれば, 提案手法の対象物画素を処理するため, 操作 *resolve* を呼ぶ平均回数は $(13/32 + 0) / 2 \approx 0.20$ 回となり, 従来法の 0.19 回とほとんど変わらない.

以上の分析を検証するために, 3 章に述べたすべての 512×512 のノイズ画像を用いて第 1 走査における対象物画素を処理するために周りの画素をチェックする際の比較演算の平均回数, 代入演算の平均回数と操作 *resolve* の呼び出し平均回数について実験を行った. 実験結果を表 2 に示す.

表 2 の結果は上記の分析とほぼ一致する. 代入演算と *resolve* の呼び出しの平均回数において両手法は同じぐらいだが, 比較演算の平均回数においては, 提案手法が従来手法より著しく減少した.

5. む す び

本論文では, 画像を 2 行ずつ走査し, 画素を 2 つずつ処理するラスタ走査型ラベル付けアルゴリズムにおける新しい第 1 走査手法を提案した. それによって, 暫定ラベル付けと連結関係の検出を効率的に処理できた. 様々な性質の画像を用いた比較評価実験により, 提案手法の有効性を示した. 将来の課題としては, 提案手法の計算量分析やメモリのキャッシュ

ヒットとの関係の解析などがあげられる.

謝辞 本研究の一部は, 栢森情報振興財団, 人工知能振興財団および日比科学技術財団の研究助成を受けている.

参 考 文 献

- 1) 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, 東京 (1988).
- 2) Shirai, Y.: Labeling connected regions, *Three-Dimensional Computer Vision*, pp.86-89, Springer-Verlag, New York (1987).
- 3) Chang, F., Chen, C.J. and Lu, C.J.: A linear-time component-labeling algorithm using contour tracing technique, *Computer Vision and Image Understanding*, Vol.93, No.2, pp.206-220 (2004).
- 4) 何 立風, 巢 宇燕, 鈴木賢治, 中村剛士, 伊藤英則: 高速 2 回走査ラベル付けアルゴリズム, 電子情報通信学会論文誌, Vol.J91-D, No.4, pp.1016-1024 (2008).
- 5) 何 立風, 巢 宇燕, 鈴木賢治, 中村剛士, 伊藤英則: ラスタ走査型ラベル付けアルゴリズムにおける第一走査の効率化手法, 電子情報通信学会論文誌, Vol.J92-D, No.6, pp.951-955 (2009).
- 6) Shima, Y., Murakami, T., Koga, M., Yashiro, H. and Fujisawa, H.: A high-speed algorithm for propagation-type labeling based on block sorting of runs in binary images, *Proc. 10th International Conference on Pattern Recognition*, pp.655-658 (1990).
- 7) Suzuki, K., Horiba, I. and Sugie, N.: Linear-time connected-component labeling based on sequential local operations, *Computer Vision and Image Understanding*, Vol.89, No.1, pp.1-23 (2003).
- 8) Otsu, N.: A threshold selection method from gray-level histograms, *IEEE Trans. Systems Man and Cybernetics*, Vol.9, No.1, pp.62-66 (1979).
- 9) <http://www1.cs.columbia.edu/CAVE/software/curet/index.php>

(平成 22 年 9 月 8 日受付)

(平成 23 年 1 月 14 日採録)



何 立風 (正会員)

1982年中国・西北軽工業学院卒業。同年同大学助手。1987年同大学講師。1997年名古屋工業大学工学研究科博士後期課程修了。博士(工学)。現在、愛知県立大学大学院情報科学研究科准教授。陝西科学技術大学客員教授。知的画像処理, コンピュータビジョン, 医用画像処理, 自動推論, 知識データベース, マルチエージェント等に興味を持つ。



巢 宇燕

2000年名古屋大学大学院人間情報文化工学研究科博士後期課程修了。博士(学術)。2000年4月から2002年9月まで名古屋工業大学研究員。現在、名古屋産業大学大学院環境マネジメント研究科教授。陝西科学技術大学客員教授。画像処理, 定理証明および図面理解, CAD等に興味を持つ。



鈴木 賢治 (正会員)

1991年名城大学理工学部電気電子工学科卒業。1993年同大学大学院修士課程修了。同年(株)日立メディコ技術研究所入社。1997年愛知県立大学情報科学部助手。2001~2002年米国・シカゴ大放射線医科学研究科カートロスマン放射線像研究所客員研究員。2002年同研究員。2003年同研究講師, 2004年同研究助教授。2006年同大学放射線医科学研究科助教授。2007年同大学医用物理学研究科助教授兼任。2007年同大がん研究センター助教授兼任。現在に至る。博士(工学)(名古屋大学)。コンピュータ支援診断, 医用画像処理・認識および機械学習の研究に従事。IEEE (Senior Member), AAPM, 情報処理学会等各会員。2002年 Paul C. Hodges 賞, 2003年 RSNA Certificate of Merit 賞, 2004年 RSNA Research Trainee 賞, 2005年 CRF Young Investigator 賞, 2006年 SPIE Honorable Mention Poster 賞, RSNA Certificate of Merit 賞等受賞。



中村 剛士

1998年名古屋工業大学大学院博士後期課程修了。1998年同大学助手。2003年同大学大学院工学研究科助教授。2007年准教授。現在に至る。CG, CV, HRI, ソフトコンピューティング等に関する研究に従事。ACM, IEEE, 電子情報通信学会, 日本人工知能学会, 日本知能情報ファジィ学会, 芸術科学会各会員。博士(工学)。