

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

何 立風<sup>†a)</sup>                      巢 宇燕<sup>††</sup>  
 鈴木 賢治<sup>†††</sup>                      中村 剛士<sup>††††</sup>  
 伊藤 英則<sup>††††</sup>

A Strategy for Efficiency Improvement of the First-Scan in Raster-Scan-Based Labeling Algorithms  
 Lifeng HE<sup>†a)</sup>, Yuyan CHAO<sup>††</sup>, Kenji SUZUKI<sup>†††</sup>, Tsuyoshi NAKAMURA<sup>††††</sup>, and Hidenori ITOH<sup>††††</sup>, *Nonmembers*

<sup>†</sup> 愛知県立大学情報科学研究科, 愛知県  
 Graduate School of Information Science and Technology,  
 Aichi Prefectural University, Aichi-ken, 480-1198 Japan  
<sup>††</sup> 名古屋産業大学環境情報マネジメント研究科, 尾張旭市  
 Graduate School of Environment Management,  
 Nagoya Sangyo University, Owariasahi-shi, 488-8711 Japan  
<sup>†††</sup> シカゴ大学放射線医学研究科, 米国  
 Department of Radiology, The University of Chicago, USA  
<sup>††††</sup> 名古屋工業大学大学院工学研究科, 名古屋市  
 Graduate School of Engineering, Nagoya Institute of Technology,  
 Nagoya-shi, 466-8555 Japan  
 a) E-mail: helifeng@ist.aichi-pu.ac.jp

あらまし ラスタ走査型ラベル付けアルゴリズムでは、第一走査において、対象物画素への暫定的なラベル付けと暫定ラベルの連結性の関連付け処理を行う。従来手法では、注目対象物画素と8連結している四つの走査済みの画素からなるマスクを用いてすべての対象物画素を同一手順で処理する。これに対して、本論文では、注目対象物画素が背景画素の後にあるのか対象物画素の後にあるのかということが計算せず分かることに注目し、それぞれの場合に分けて処理を行う方法を提案する。これにより、注目対象物画素の前の画素をマスクから除外できる。つまり、走査用のマスクは残りの三つの画素だけになる。その結果、対象物画素が存在しない画像を除き、すべての画像において、提案手法は従来手法より効率である。様々な性質の画像を用いた従来手法との比較評価実験により、提案手法の有効性を確認した。

キーワード ラベル付け, ラスタ走査, 第一走査, パターン認識, 連結成分, 2 値画像

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、形状や大きさ（面積）などの対象物の特徴を知るために不可欠であり、最も基本的かつ重要な画像処理の一つである [1]。これまでに多くのラ

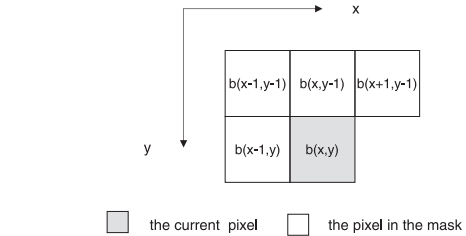


図 1 8 連結用走査マスク  
 Fig. 1 Mask for the first scan.

ベル付けアルゴリズムが提案されており、ラスタ走査型 [2] ~ [4] はその主流<sup>注1)</sup>である。特に、指紋や人物の高速照合システム、航空画像からの目標物追跡システム、材料の実時間欠損検出システム、医用画像診断システムなどの動画像処理システムでは、少しでも効率的なラベル付けアルゴリズムの開発が求められている。なお、文献 [3] で提案された手法は、これまでに提案された最速のラスタ走査型アルゴリズム [4] より、平均 8 倍速くなっており、現在も最速のラベル付けアルゴリズムであるため、その手法の更なる改良はラベル付け問題に重要な意義がある。

ラスタ走査型ラベル付けアルゴリズムでは、少なくとも 2 回画像を走査することによりラベル付けを行う。第一走査において、従来手法では、すべての対象物画素を同一手順で処理する。これに対して、本論文では、注目対象物画素が背景画素の後にあるのか、対象物画素の後にあるのかにより場合分けして処理する方法を提案する。

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、画像の輪郭の画素は背景画素とする。

従来手法では、第一走査において、図 1 に示す注目画素と 8 連結している走査済みの四つの画素からなるマスクを用いてラスタ走査の順序に走査しながら、注目対象物画素を処理していく。そのとき、マスク中には図 2 に示す 16 ケースをとり得る。

従来手法の中、文献 [3] に提案された手法を除き、す

(注1): 多くのリアルタイム画像処理システムでは、画像の読み出し書込みがラスタ走査の順に行われているため、ラスタ走査型アルゴリズムはそれらとの整合性が良く、リアルタイム処理、ハードウェア実装、並列処理やパイプライン処理に向いている [4]。

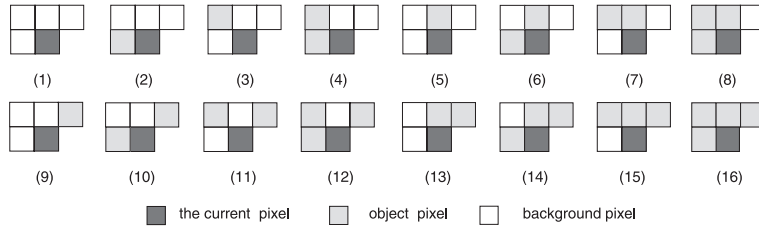


図 2 マスク中に取り得る 16 ケース

Fig. 2 Sixteen possible cases for the current object pixel.

```

m = 1;
for (y = 1; y < M - 1; y++)
  for (x = 1; x < N - 1; x++)
    if (v(x, y) > 0)
      if (v(x, y - 1) > 0)
        v(x, y) = v(x, y - 1);
      else if (v(x - 1, y) > 0)
        v(x, y) = v(x - 1, y);
      if (v(x + 1, y - 1) > 0)
        resolve(v(x + 1, y - 1), v(x - 1, y));
      end if
    else if (v(x - 1, y - 1) > 0)
      v(x, y) = v(x - 1, y - 1);
      if (v(x + 1, y - 1) > 0)
        resolve(v(x - 1, y - 1), v(x + 1, y - 1));
      end if
    else if (v(x + 1, y - 1) > 0)
      v(x, y) = v(x + 1, y - 1);
    else
      v(x, y) = m;
      m = m + 1;
    end if
  end if
end for
end for

```

図 3 第一走査の従来手法

Fig. 3 A conventional first-scan algorithm.

すべての提案手法では、注目対象物画素を処理する際、マスクにある四つの画素をすべてチェックしなければならない。つまり、各対象物画素を処理するため、マスクにある画素をチェックする回数は 4 回である。

文献 [3] に提案された手法では、図 3 に示した手順でマスクにある画素をチェックしながら処理する。ただし、 $resolve(u, v)$  は、暫定ラベル  $u$  と  $v$  間の連結関係を解決する演算であり、 $m$  は暫定ラベルの値を表す。

以上の処理により、マスクにある画素は対象物画素であるかどうかのチェックする回数は、図 2 に示したケース (5), (6), (7), (8), (13), (14), (15) 及び (16) のとき 1 回、ケース (2), (4), (10) 及び (12) のとき

```

%% Process1
if (v(x, y - 1) > 0)
  v(x, y) = v(x, y - 1);
else if (v(x - 1, y - 1) > 0)
  v(x, y) = v(x - 1, y - 1);
  if (v(x + 1, y - 1) > 0)
    resolve(v(x - 1, y - 1), v(x + 1, y - 1));
  end if
else if (v(x + 1, y - 1) > 0)
  v(x, y) = v(x + 1, y - 1);
else
  v(x, y) = m;
  m = m + 1;
end if

```

図 4 処理手順 1

Fig. 4 Process1.

3 回、ケース (1), (3), (9) 及び (10) のとき 4 回となる。その平均回数は  $(1 \times 8 + 3 \times 4 + 4 \times 4) / 16 = 2.25$  回となり、他の従来手法の 4 回より、43.8%に減少した。便宜のため、これから、文献 [3] に提案された手法を比較対象とし、単に従来手法と呼ぶ。

しかしながら、上述の手順でも、すべての対象物画素を同一手順で処理することは明らかである。マスクにある画素は対象物画素であるかどうかのチェックする回数は最大で 4 回のみである。

本論文では、注目対象物画素背景画素の後にあるのか対象物画素の後にあるのかということは、処理中に計算せず自然に分かることに注目し、それぞれの場合に分けて処理を行う。そのとき、注目画素の直前画素、つまり、 $b(x - 1, y)$  を再びチェックしなくてもよい。そのため、 $b(x - 1, y)$  をマスクから除外でき、マスクは残りの三つの画素だけになる。

注目画素  $b(x, y)$  の直前画素  $b(x - 1, y)$  が背景画素であるとき、図 2 に示しているケース (1), (3), (5), (7), (9), (11), (13) 及び (15) の 8 ケースがある。文献 [3] に示した分析方法で分析すれば、図 4 に示した手順 (Process1 と呼ぶ) で処理できる。

```

Process2
v(x, y) = v(x - 1, y);
if (v(x, y - 1) = 0 and v(x + 1, y - 1) > 0)
    resolve(v(x, y), v(x + 1, y - 1));
end if

```

図 5 処理手順 2  
Fig. 5 Process2.

```

m = 1;
for (y = 1; y < M - 1; y++)
    for (x = 1; x < N - 1; x++)
        if (v(x, y) > 0)
            Process1;
            x++;
            while (v(x, y) > 0)
                Process2;
                x++;
            end while
        end if
    end for
end for

```

図 6 第一走査の提案手順  
Fig. 6 The proposed first-scan procedure.

一方、注目画素  $b(x, y)$  の直前画素  $b(x - 1, y)$  が対象物画素であるとき、図 2 に示しているケース (2), (4), (6), (8), (10), (12), (14) 及び (16) の 8 ケースがある。図 5 に示した手順 (*Process2* と呼ぶ) で処理できる。

そして、第一走査の手順は図 6 に示したように改良できる。

つまり、背景画素の次に発見した対象物画素を *Process1* で処理する。そして、その対象物画素の後に続くすべての対象物画素を *Process2* で処理する。

### 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 D 930 3.0 GHz + 3.0 GHz CPUs, 2 GB Memory, Mandriva Linux OS) 上での CPU 実行時間を用いた比較評価実験を行った。なお、実行時間は第一走査のみの時間 (連結性の検出と解決の時間を含む) で、5,000 回の実行の平均値である。

実験には次の 5 種類の画像を用いた。(1) ノイズ画像: 五つのサイズ (32×32, 64×64, 128×128, 256×256 と 512×512 画素) の画像に、1,000 階調までの白色一様ノイズを加え、しきい値を 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 (各サイズごとには 41 種類)。(2) 自然画像: SIDBA (Standard Image Data Base) 及び USC (University of Southern California) の標準画像データベースから

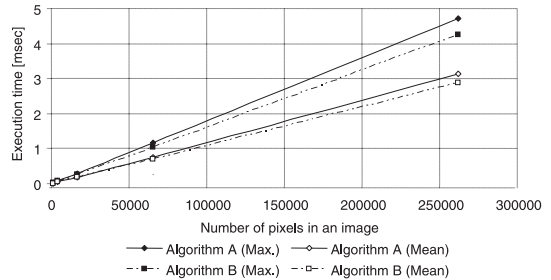


図 7 画像サイズに対する実行時間の線形性

Fig. 7 Linearity of the execution time versus image size.

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 [ms] 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.	2.25	2.18
	mean	1.40	1.32
	min.	0.92	0.87
medical	max.	1.45	1.31
	mean	1.21	1.09
	min.	0.92	0.90
textural	max.	1.92	1.80
	mean	1.47	1.38
	min.	1.10	1.05
artificial	max.	1.67	1.53
	mean	0.83	0.80
	min.	0.29	0.26

風景画像、航空画像、人物画像、静物画像、指紋画像、スナップ画像及びテキスト画像を含む 50 種類の自然画像を Otsu のしきい値選択法 [5] 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 Database<sup>(注2)</sup>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、蜂巢状パターンを含む人工画像 [4]。

図 7 に、ノイズ画像における画像サイズに対する従来手法 (Algorithm A) と提案手法 (Algorithm B) の実行時間特性を示す。提案手法も線形特性をもつことが確認できる。

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。すべての種類の画像に

(注2): <http://www1.cs.columbia.edu/CAVE/software/curet/index.php>

表 2 各ケースにおいて画素のチェック回数  
Table 2 The times for checking neighbour pixels in each case.

ケース	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
従来手法	4	3	4	3	1	1	1	1	4	3	4	3	1	1	1	1
提案手法	3	2	3	2	1	1	1	1	3	2	3	2	1	1	1	1

において、提案手法が高速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、図 2 に示した 16 ケースが同確率で現れるとする。

注目画素が背景画素であるとき、提案手法も従来手法と同様、何にも処理しないから、両手法の効率と同じである。

注目画素が対象物画素であるとき、従来手法では、2. での議論より、8 連結している走査済みの画素は対象物画素であるかどうかをチェックする回数は、図 2 に示したケース (5), (6), (7), (8), (13), (14), (15) 及び (16) のとき 1 回、ケース (2), (4), (10) 及び (12) のとき 3 回、ケース (1), (3), (9) 及び (10) のとき 4 回となる。

これに対し、提案手法では、注目対象物画素が背景画素後にある場合、*Process1* により、図 2 に示したケース (5), (7), (13) 及び (15) のとき 1 回、ケース (1), (3), (9) 及び (11) のとき 3 回となる。一方、注目対象物画素が対象物画素後にある場合、*Process2* により、ケース (6), (8), (14) 及び (16) のとき 1 回、ケース (2), (4), (10) 及び (12) のとき 2 回となる。

以上の分析より、図 2 に示すマスク中の 16 ケースにおいて、注目対象物画素を処理するとき、従来手法と提案手法の 8 連結している走査済みの画素をチェックするそれぞれの回数は、表 2 にまとまる。

表 2 により、第一走査において、対象物画素を処理する際、8 連結している画素をチェックする平均回数において、従来手法では、 $(1 \times 8 + 3 \times 4 + 4 \times 4) / 16 = 2.25$  回であり、提案手法では、 $(1 \times 8 + 2 \times 4 + 3 \times 4) / 16 = 1.75$  回である。つまり、その平均回数において、提案手法は従来手法より約 23% 減少した。また、すべてのケースにおいて、提案手法の隣接画素のチェック回数は従来手法以下であることが分かる。

一方、提案手法では、注目対象物画素が、背景画素の後にあるのか、対象物画素の後にあるのかを判定しなければならないが、図 6 に示した提案手順から容易に分かるように、この条件判定は計算せず処理中に自

然に実現しているため、計算コストはかからない。

また、第一走査における上述処理（マスクにある画素のチェック）以外の操作は、各ケースにおいて、提案手法は従来手法と同様に行うので、それに関連する効率も同じである。

以上の議論をまとめると、第一走査において、背景画素に対する処理の効率は、提案手法が従来手法と同様である。任意の対象物画素に対する処理の効率において、提案手法は従来手法の以上である。対象物画素が存在する画像には、図 2 に示すケース (1) が少なくとも 1 回現れるため、提案手法の効率は従来手法よりよい。言い換えれば、対象物画素が存在しない（すべての画素が背景画素である）画像を除き、すべての画像において、提案手法が有効である。

更に、表 2 より、ケース (5), (6), (7), (8), (13), (14), (15) と (16) の場合、提案手法の効率は従来手法の効率同様であり、その以外の場合、提案手法が効率である。そのため、垂直線分が多い画像に対して、提案手法の効率化の効果は低いが、水平線分が多い画像に対して、提案手法の効率化効果は高い。例えば、1 行間隔の垂直線分の画像において、提案手法と従来手法の実行時間はそれぞれ 0.56 ミリ秒と 0.59 ミリ秒であり、提案手法の実行時間は従来手法のより 5.1% 減少したのに対して、1 行間隔の水平線分の画像において、それぞれの実行時間は 0.63 ミリ秒と 0.74 ミリ秒であり、提案手法の実行時間は従来手法のより 14.9% 減少した。

なお、表 1 に示すように、自然画像、医用画像、テクスチャ画像と人工画像において、提案手法の第一走査の平均時間は従来手法より、平均それぞれ 5.7%, 9.9%, 6.1%, 7.8% 減少した。また、実験に用いたすべての画像において有効であることも確認できた。

#### 5. む す び

本論文では、ラスト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を提案した。提案手法は対象物画素が存在しない画像以外のすべて画像に有効である。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。

本手法はほとんどのラスタ走査型ラベル付けアルゴリズムに組み込むことができ、それらのアルゴリズムの効率化につながると考えられる。

謝辞 本研究の一部は、豊秋奨学会の研究助成による。

#### 文 献

- [1] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 3rd ed., pp.40–43, Prentice Hall, 2007.
  - [2] 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,” 信学論 (D-II), vol.J72-D-II, no.2, pp.247–255, Feb. 1989.
  - [3] 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,” 信学論 (D), vol.J91-D, no.4, pp.1016–1024, April 2008.
  - [4] K. Suzuki, I. Horiba, and N. Sugie, “Linear-time connected-component labeling based on sequential local operations,” Computer Vision and Image Understanding, vol.89, no.1, pp.1–23, Jan. 2003.
  - [5] N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Syst. Man Cybern., vol.9, no.1, pp.62–66, Jan. 1979.  
(平成20年10月14日受付, 21年1月7日再受付)
-