

同等ラベル解析に基づく1回走査ラベル付けアルゴリズム

何立風^{†1} 巢宇燕^{†2} 鈴木賢治^{†3}
中村剛士^{†4} 伊藤英則^{†4}

本論文では、同等ラベル解析に基づく1回走査ラベル付けアルゴリズムを提案する。提案アルゴリズムでは、唯一の走査で対象物画素のつながりからなる連ごとに暫定ラベル付けを行いながら、連のデータを記録し、同等ラベル解析を行う。走査後、同じ連結体に属する各連が同一最終ラベルを持つ。そして、再び画像を走査せず、記録した連のデータを用いて、各連に対して、その連にあるすべての画素にその連の最終ラベルを付ける。それにより、背景画素を1回だけ処理するとともに、暫定ラベル付けとラベルの書き換えという同等ラベル解析法にとって不可欠な操作は、対象物画素単位ではなく、連単位で実現できる。画像中の連の数は、対象物画素より一般的に少ないことから、提案手法では効率化を達成できる。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、提案アルゴリズムが特に平均連長の長い画像または対象物画素密度の低い画像に有効であることを確認した。

A Label-equivalence-based One-scan Labeling Algorithm

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

This paper presents a label-equivalence-based one-scan algorithm for labeling connected components in a binary image. Our algorithm scans an image only once. Moreover, we do not assign provisional labels to object pixels, but assign provisional labels to runs. After resolving label equivalences between runs, using the recorded run data, we assign each object pixel a final label. Experimental results demonstrated that our algorithm was highly efficient on images with long runs or small number of object pixels.

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つであり、コンピュータに対象物を認識させるうえで欠くことができない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、それらはラベル伝播に基づくアルゴリズムと同等ラベル解析に基づくアルゴリズムに大別できる。

ラベル伝播に基づくアルゴリズム^{2),9)}では、まず、画像からラベルを付けていない対象物画素を見つけ、その対象物画素に新しいラベルを付ける。そして、その後の処理で、その対象物画素と連結しているすべての対象物画素にそのラベルを付ける。これらの手法は画像における走査は不規則であるため、パイプライン処理、並列処理やハードウェア実現に向かない。

ほとんどの同等ラベル解析に基づくアルゴリズムでは、ラスト走査により画像中の各画素を2回以上走査する必要がある。第1走査では、各対象物画素に暫定ラベル付けを行う。そして、暫定ラベル間の同等関係の解析によるその後の走査で、各対象物画素の暫定ラベルを最終ラベルに書き換える。それらのアルゴリズムには、ラベル付けに必要な走査回数により、マルチ走査アルゴリズム⁴⁾、4回走査アルゴリズム¹⁰⁾、2回走査アルゴリズム^{3),5),6),12)}がある。

本論文は文献6)に提案された連に基づく2回走査（Run-based Two-Scan）アルゴリズム（以後RTSアルゴリズムと呼ぶ）を改良し、同等ラベル解析法に基づく1回走査アルゴリズムを提案する。

2. RTS アルゴリズムの概要

本論文では、対象物画素の値を1、背景画素の値を0としたサイズ $N \times M$ の2値画像

†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

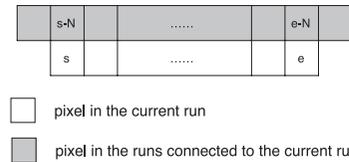


図1 注目連の8隣接区間

Fig. 1 Eight-connected range for the current run $r(s, e)$.

において, (x, y) にある画素の値を $p(y \times N + x)$ で表し, 画像の最外郭画素は背景画素とする. ただし, $0 \leq x \leq N - 1, 0 \leq y \leq M - 1$ である. また, i を画素の座標, n を連の通し番号, l を暫定ラベル番号とする.

画像中の同じ行にある $p(s)$ から $p(e)$ ($s \leq e$) までの連続した対象物画素のブロックを連と呼び, $r(s, e)$ で表す. 連のデータはラスタ走査中に容易に獲得できる. 処理中の連(注目連と呼ぶ)が $r(s, e)$ であるとき, 図1から分かるように, これまでのラスタ走査により発見した連のうち, $a \leq e - N + 1$ かつ $b \geq s - N - 1$ のすべての連 $r(a, b)$ が注目連 $r(s, e)$ と8隣接する(つまり, 少なくとも連 $r(a, b)$ の1つの画素が図1に示した8隣接区間にある).

RTS アルゴリズムでは, 走査中の各時点までに発見した同一連結成分に属する連の暫定ラベルを, その連結成分に対応する同等ラベル集合にまとめ, それらの暫定ラベル中の最小ラベルをその同等ラベル集合およびその集合中の各ラベルの代表ラベルとする. 便宜のため, $r_label[z]$ を暫定ラベル z の代表ラベル, $S(x)$ を代表ラベル x を持つ同等ラベル集合を表す. そうすると, 同等ラベル集合 $S(x)$ に属するすべてのラベル y において, $r_label[y] = x$ となる.

第1走査では, i を N から $N \times (M - 1)$ まで^{*1}, 画素 $p(i)$ をラスタ走査順に処理する. 新しい連(注目連)を発見したときは, そのデータを記録しておく. そして, 走査済みの連のうち, 注目連に8隣接している連が存在するかどうかをチェックする.

注目連に8隣接する連が存在しない場合, 注目連がこれまで発見した連のどれとも隣接していないことを意味する. そのため, 注目連に属するすべての画素に新しいラベル l を付け, l だけを要素とする同等ラベル集合を作り, その代表ラベルを l とする. つまり, $S(l) = \{l\}$, $r_label[l] = l$.

一方, 注目連に8隣接している連が存在する場合, これらの連が注目連と同一対象物に属す

ることが分かる. このとき, これらの連に対応する同等ラベル集合を1つに併合し, その中の最小ラベルを併合後の集合の代表ラベルとする. この併合処理は, たとえば, 2つの同等ラベル集合 $S(u)$ と $S(v)$ を併合する場合, $u = v$ のとき, 何もしない. $u < v$ であれば, u は $S(u)$ と $S(v)$ 中の最小ラベルであるため, $S(v)$ を $S(u)$ に併合し(つまり, $S(u) = S(u) \cup S(v)$), $S(v)$ に属するすべてのラベルに対して, そのラベルの代表ラベルを u に書き換える(つまり, すべての $x \in S(v)$ に対して, $r_label[x] = u$ にする). 一方, $u > v$ であれば, v は最小ラベルであるため, $S(u)$ を $S(v)$ に併合し(つまり, $S(v) = S(v) \cup S(u)$). $S(u)$ に属するすべてのラベルに対して, そのラベルの代表ラベルを v に書き換える(つまり, すべての $y \in S(u)$ に対して, $r_label[y] = v$ にする).

そして, 注目連にあるすべての画素にその代表ラベルを付ける. また, 注目連を $r(s, e)$ としたとき, 注目連を処理した後, $q \leq e - N$ が成り立つようなすべての連 $r(p, q)$ がその後の連結性検出に用いられることがないので, それらの連のデータを廃棄する.

第1走査後, 同一連結成分に属するすべての暫定ラベルが同一同等ラベル集合にまとめられ, 同一代表ラベル(最終ラベルとも呼ぶ)を持つことになる. そのため, 第2走査では, 各対象物画素に付与された暫定ラベルをその暫定ラベルの最終ラベルに書き換えれば, 各連結成分にユニークなラベルを付けることができる.

3. 提案アルゴリズムの概要

前述のように, RTS アルゴリズムでは, まず, 対象物画素ごとに暫定ラベルを付ける. そして, 暫定ラベル間の連結性解析後, 暫定ラベルを最終ラベルに書き換える. また, 連のデータは連結性の検出だけに使う. これに対して, 提案アルゴリズムでは, 同じ連結体に属する各連にあるすべての画素が必ず同一最終ラベル付けられることから, 連ごとに暫定ラベル付けを行う. そして, 連の連結関係を解析したあと, 連のデータを用いて, 対象物画素に最終ラベルを直接付ける. 提案の走査アルゴリズムを図2に示す. ただし, $run[y]$ を y 番目の連のラベルを表し, $resolve(p, q)$ は同等ラベル集合 $S(r_label[p])$ と $S(r_label[q])$ の併合処理を表す. また, j は, これまで処理済みの連から注目連と8隣接している連を探索するための変数である.

提案の走査アルゴリズムでは, まず, n, l, j の値はそれぞれ1に初期化する(1行目). そして, 画素 $p(i)$ を $i = N$ から $i < N \times (M - 1)$ まで(2行目), 画像をラスタ走査順に処理する(図3を参照).

新しい連 $r(s, e)$ を発見したときは, そのデータを $run_s[n] = s$ と $run_e[n] = e$ で記録

*1 画像の最外郭が背景画素であるため, 最上行と最下行を処理しなくてもよい.

```

1:  n = 1, l = 1, j = 1;
2:  for(i = N; i < N × (M - 1); i++)
3:    if(p(i))
4:      run_s[n] = i;
5:      i++;
6:      while(p(i))
7:        i++;
8:      end of while
9:      run_e[n] = i - 1;
10:     while(run_e[j] < run_s[n] - N - 1)
11:       j++;
12:     end of while
13:     if(run_e[j] ≤ run_e[n] - N)
14:       run[n] = run[j];
15:       j++;
16:       while(run_e[j] ≤ run_e[n] - N)
17:         resolve(run[j], run[n]);
18:         j++;
19:       end of while
20:       if(run_s[j] ≤ run_e[n] - N + 1)
21:         resolve(run[j], run[n]);
22:       end of if
23:     else if(run_s[j] ≤ run_e[n] - N + 1)
24:       run[n] = run[j];
25:     else
26:       run[n] = l;
27:       r_label[l] = l;
28:       l++;
29:     end of if
30:   end of for
31:   n++;
32: end of for

```

図2 提案の走査アルゴリズム
Fig. 2 The proposed scan algorithm.

する(3行目~9行目). 注目連 $r(s, e)$ において, $p(s - N - 1)$ より前に終了するすべての連(これらの連は注目連 $r(s, e)$ と8隣接しない)をパスし(10行目~12行目), 次の連, つまり, $p(s - N - 1)$ 以降に終了する最初の連 $r(a, b)$ ^{*1}は $p(e - N)$ 以前に終了するかどうかをチェックする(13行目).

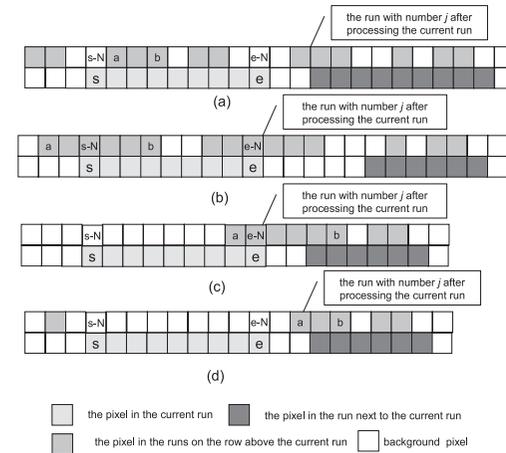


図3 注目連の処理例
Fig. 3 Examples for processing the current run $r(s, e)$.

連 $r(a, b)$ は $p(e - N)$ 以前に終了するとき(図3(a), (b)), 注目連はこの連と隣接しているため, 注目連にこの連のラベルを付ける(14行目). そして, そのあとの $p(e - N)$ 以前に終了する各連(これらの連は注目連と連結している)において(図3(a)), 注目連とその連に対応する同等ラベル集合の併合処理を1ずつ行う(16行目~19行目). さらに, 次の連(この連は $p(e - N + 1)$ 以降に終了する)は $p(e - N + 1)$ 以前から始まるかどうかをチェックする(20行目). そうであれば(図3(b)), この連も注目連と隣接しているため, 注目連とその連に対応するそれぞれの同等ラベル集合の併合処理を行う(21行目).

連 $r(a, b)$ は $p(e - N)$ 以降に終了するのであれば(図3(c), (d)), この連は $p(e - N + 1)$ 以前から始まるかどうかをチェックする(23行目). そうであれば(図3(c)), この連は注目連と隣接しているため, 注目連にこの連のラベルを付ける(24行目). そうでなければ(図3(d)), 1つ前の行に注目連と隣接する連がないから, 注目連 $r(s, e)$ に新しい暫定ラベル l を付け(26行目), その代表ラベルを $r_label[l] = l$ にセットし(27行目), l を1増やす(28行目). 最後に n を1増やして次の連の記録と処理に備える(31行目).

いずれの場合でも, 注目連 $r(s, e)$ を処理したあと, j の値は $p(e - N + 1)$ 以降に終了する最初の連に対応している(図3). 次の注目連を $r(s', e')$ (ただし, $s' > e + 1$) を処理する際, 番号は j より小さい任意連は $p(e - N)$ 以前に終了するので $r(s', e')$ と隣接すること

*1 注目連 $r(s, e)$ は $p(s - N - 1)$ 以降に終了する連である.

がないから、番号 j の連から連結性探索を行えばよい。

上記の走査で、注目連と連結しているすべての走査済みの連に付けられた暫定ラベルは同じ同等ラベル集合にまとめられ、同一代表ラベルを持つ。上記の走査が終了したとき、すべての連の始点と終点はそれぞれ $run_s[]$ と $run_e[]$ に記録され、同一連結成分に属するすべての連に付けられた暫定ラベルは同じ同等ラベル集合にまとめられている。その代表ラベルはその集合にあるすべての(連の)暫定ラベルの最終ラベルとなる。そして、連データを用いれば、次の操作で、画像を走査(第2走査)せず、各連にある画素にその連に対応する最終ラベルを直接付けることができる。

```

for(j = 1; j < n; j++)
  for(i = run_s[j]; i ≤ run_e[j]; i++)
    p(i) = r_label[run[j]];
  end of for
end of for
    
```

(1)

上述より、提案アルゴリズムでは、すべての背景画素が1回だけ処理されること、暫定ラベル付けとラベルの書き換えという同等ラベル解析法にとって不可欠な操作は、連単位で行われていることが明らかである。

4. 比較評価実験

提案アルゴリズムは、RTS アルゴリズム同様に実装することができる。ただし、RTS アルゴリズムでは、注目連 $r(s, e)$ を処理したあと、 $n \leq e - N$ が成り立つようなすべての連 $r(m, n)$ のデータを廃棄するから(つまり、 $p(e - N + 1)$ から $p(e)$ までの画素を含む連のデータだけを保存している)、連の始点と終点の保存をそれぞれ $(N/2 + 1)$ サイズの循環リストで実現できた。しかし、本アルゴリズムでは、画像にあるすべての連のデータを保存しなければならないので、連の始点と終点を保存するため、それぞれ $N \times M/4$ サイズの1次元配列が必要である^{*1}。

提案アルゴリズムの特性および性能を評価するため、最近提案された次の4つのアルゴリズムとの比較評価実験を行った。(1)文献9)で提案された Contour-Tracing (CT) アルゴリズム;(2)文献12)で提案された Scan plus Array-based Union-Find (SAUF) アル

*1 $N \times M$ サイズの画像において、最大の連数は $N \times M/4$ である。

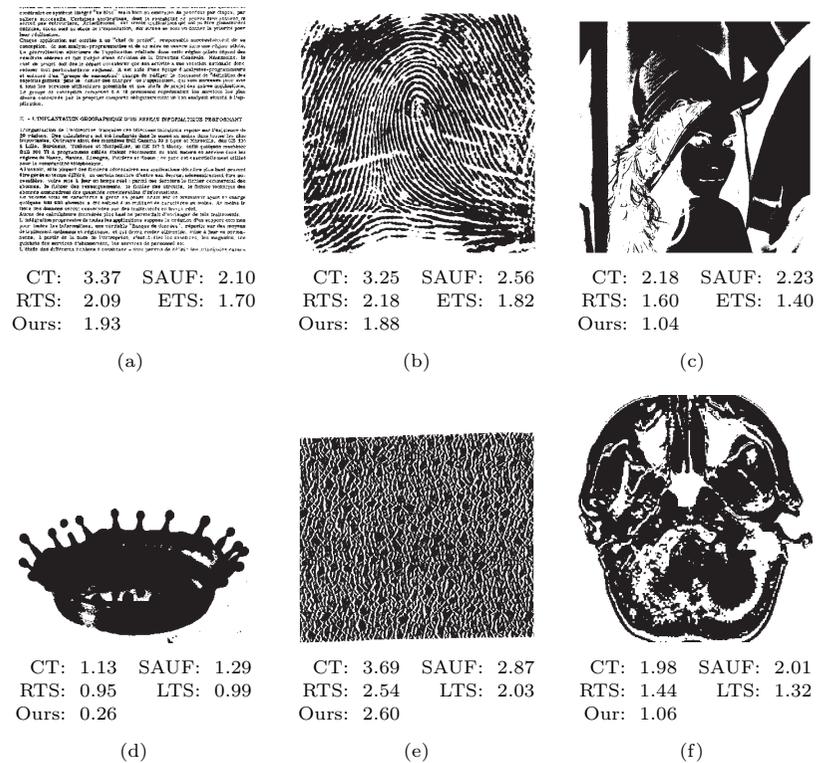


図4 6つの画像例における実行時間(ミリ秒):(a)テキスト画像;(b)指紋画像;(c) a portrait image;(d)スナップ画像;(e)テクスチャ画像;(f)医用画像

Fig. 4 Execution time [msec] of labeling algorithms for six representative images: (a) a text image; (b) a fingerprint image; (c) a portrait image; (d) a snapshot image; (e) a texture image; (f) a medical image.

ゴリズム;(3)RTS アルゴリズム;(4)文献5)で提案された Efficient Two-Scan (ETS) アルゴリズム。

比較評価実験は、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実行時間を用いて行った。実験に用い

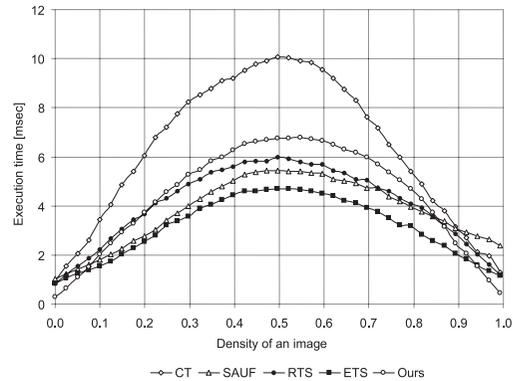


図 5 画像の対象物画素密度に対する実行時間の特性

Fig. 5 Execution time versus the density of object pixels in an image.

た CT アルゴリズムのプログラムは著者のホームページ^{*1}よりダウンロードしたものであり、他のアルゴリズムは著者らから提供されたものである。また、すべてのプログラムは C 言語で作成したものであり、それらの実行ファイルは上述コンピュータの上で同じコンパイラによるコンパイルしたものである。なお、実行時間はシングルコアを用いた 5,000 回の実行の平均値である。

実験には次の 4 種類の画像を用いた。(1) ノイズ画像: 512×512 サイズの画像に、1,000 階調までの白色一様ノイズを加え、閾値を 0 から 1,000 まで 25 ごとに変化させて 2 値化した合計 41 種類の 2 値画像。このような画像は自然画像に比べ、複雑な幾何学形状と複雑なラベル連結性を持つため、ラベル付けアルゴリズムをより厳格に評価するのに適している。(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⁷⁾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^{*2}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。

*1 <http://dar.iis.sinica.deu.tw/Download>*2 <http://www1.cs.columbia.edu/CAVE/software/curet/index.php>

表 1 各アルゴリズムが最速であった画像数 (画像種類別)

Table 1 Numbers of images for which the algorithm was the fastest.

Image type	Total	CT	SAUF	RTS	ETS	Ours
natural	50	0	0	0	13	37
medical	25	0	0	0	1	24
textural	7	0	0	1	2	5

表 2 自然画像、医用画像、テクスチャ画像における実験結果 (ミリ秒)

Table 2 Comparison of various execution times [msec] with natural images, medical images, and textural images.

Image type		#CC	CT	SAUF	RTS	ETS	Ours
natural	max.	2,660	4.28	3.23	2.90	2.33	2.65
	mean	847	2.34	2.07	1.71	1.49	1.27
	min.	19	1.13	1.29	0.95	0.95	0.26
texture	max.	1,525	3.69	2.87	2.54	2.03	2.60
	mean	281	2.66	2.57	1.67	1.56	1.58
	min.	20	1.58	2.37	1.17	1.14	0.57
medical	max.	372	2.59	2.28	1.71	1.47	1.54
	mean	83	1.92	1.89	1.37	1.23	0.99
	min.	1	1.52	1.54	1.19	0.93	0.77

#CC, number of connected components

ノイズ画像を用いた場合の画像の濃度に対する各アルゴリズムの実行時間特性を図 5 に示す。

自然画像、医用画像およびテクスチャ画像を用いて比較を行った。各アルゴリズムが最も速かった画像の枚数を画像種類別に表 1 に示す。また、各画像種類において各アルゴリズムの最大、平均、最小の実行時間を表 2 に示す。

最後に、6 つの画像例とそれらにおける各アルゴリズムの実行時間を図 4 に示す。

5. 考 察

注目対象物画素の近傍にある処理済みの対象物画素に付けられた暫定ラベルの状況により、注目対象物画素に暫定ラベル付けを行う同等ラベル解析に基づくラベル付けアルゴリズムに対して、RTS アルゴリズムでは、連にあるすべての画素に同じ暫定ラベルを付ける。そして、提案アルゴリズムでは、連ごとに暫定ラベル付けを行う。図 6 はある連結成分におけるそれぞれの方法による暫定ラベル付けの結果を示す。提案アルゴリズムでは、平均連長が長い画像において、暫定ラベル付けの処理の手間が他のアルゴリズムより少なくなるこ

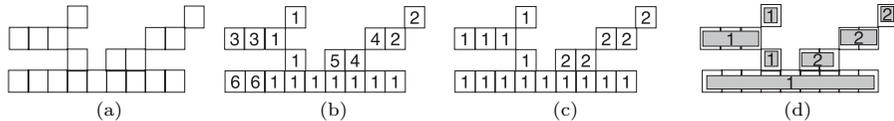


図 6 第 1 走査後の暫定ラベル付け結果：(a) 対象連結成分；(b) 従来の同等ラベル解析に基づくアルゴリズム（RTS アルゴリズム以外）；(c) RTS アルゴリズム；(d) 提案アルゴリズム

Fig. 6 Labeling results after the first scan by various algorithms: (a) an original connected component; (b) by all conventional label-equivalence-based algorithms except the RTS algorithm; (c) by the RTS algorithm; and (d) by our algorithm.

とが明らかである。

また、提案アルゴリズムでは、同等ラベル解析後に暫定ラベルを最終ラベルに書き換える処理は画像を走査せず、連のデータを用いて各連の画素に最終ラベルを直接付けることから、背景画素を 1 回しか処理しない。そのため、画像を走査してすべての画素に対して書き換えを行うアルゴリズムに対して、対象物画素密度が低い（つまり、背景画素が多い）画像において、提案アルゴリズムはほかのアルゴリズムより効率的である。

一方、連長が短くなると、画素単位と連単位による暫定ラベル付けの処理時間の差異は小さくなる。また、連の数が多くなると、背景画素の数が少なくなり、背景画素を 1 回だけ処理するメリットが小さくなるとともに、連データの記録や連データにより連にある画素へのラベル付けの手間が増える。そのため、平均連長の短い連が多数に存在する画像においては、提案アルゴリズムの利点が十分発揮されないことが多くなる。

以上の分析は、4 章に示した実験結果と一致する。図 5 により、対象物画素密度の非常に低い画像と密度の非常に高い（そのとき、平均連長が長い）画像において、提案アルゴリズムの効率は最も良い。しかし、それ以外の画像では、提案アルゴリズムの効率が LTS アルゴリズムなどより悪くなる。実画像である図 4(a), (b), (e) のような複雑な連結成分を多数持つ画像においては、提案アルゴリズムの効率は LTS アルゴリズムに及ばない。しかし、図 4(c), (d), (f) のような比較的単純な連結成分を持つ画像において、提案アルゴリズムは最速である。

ただし、表 2 より、82 枚の各種類の実画像中、本アルゴリズムは 66 枚において、最速であることに留意されたい。

最後に、提案アルゴリズムの RTS アルゴリズムに対する改良点は 2 つある。1 つは第 1 走査で、従来画素ごとに行っていた暫定ラベル付け処理を、連ごとに行うことである。もう 1 つは、暫定ラベルを最終ラベルへの書き換えは対象物画素だけに対して行う。それぞれの改

表 3 処理 1 と処理 2 において提案アルゴリズムが速かった画像の枚数
Table 3 Numbers of images for which our algorithm was faster.

Image type	Total	処理 1	処理 2	全処理
natural	50	38	50	50
medical	25	16	25	25
textural	7	6	7	6

良効果を考察するために、自然画像、医用画像およびテクスチャ画像の各種類の画像に対して、第 1 走査（処理 1 と呼ぶ）および対象物画素に最終ラベルを付与する操作*1（処理 2 と呼ぶ）において、提案アルゴリズムが速かった画像の枚数を表 3 に示す。表 3 から分かるように、すべての画像において、提案手法の処理 2 は RTS の処理 2 より効率的である。また、ほとんどの画像において、提案手法の処理 1 も RTS の処理 1 より効率的である。

連単位で連結性を解析するアルゴリズムはいくつか提案されている^{13),14),16)}。これらのアルゴリズムでは、画素で表現する画像において、ラベル付けは 2 回走査が必要であり、4 つのステップからなる。(1) 第 1 走査で、画像から連のデータを取得する；(2) 第 2 走査で、各連に暫定ラベル付けを行う；(3) 同等ラベル解析を行い、各連の最終ラベルを計算する；(4) 連データとその関連の最終ラベルデータを用いて画像の対象物画素にラベル付けを行う。そのとき、同等ラベル解析において、文献 13), 14) のアルゴリズムでは、 $P \times P$ サイズの 2 次元テーブルを用いて行う。ただし、 P は暫定ラベルの数であり、 $N \times M$ の画像において、最大 $N \times M/4$ である。つまり、このテーブルのサイズのオーダーは $O(N^2 \times M^2)$ である。一方、文献 16) のアルゴリズムでは、注目連を処理する際、連結性を解決するために、これまで処理済みのすべての連を探索しなければならない。

これに対して、提案アルゴリズムでは、2 つのステップからなる。(1) 唯一の走査中で連のデータの記録、暫定ラベル付けおよび同等ラベル解析を行う；(2) 走査終了後、連のデータを用いて対象物画素にラベル付けを行う。また、連結性の解決には、5 つの $N \times M/4$ サイズの 1 次元配列だけで*2、注目連を処理する際、連結性を解決するために、1 つ前の注目連の連結性処理で探索した最後の走査済みの連から探索すればよい。

また、背景画素を 1 回だけ処理するラベル付けアルゴリズムも提案されている^{15),17)}。ただし、文献 15) のアルゴリズムでは、4 分木で表現されている特殊な画像形式しか適応でき

*1 この操作は、RTS アルゴリズムにおいて、第 2 走査であり、提案アルゴリズムにおいて、式 (1) の処理である。

*2 提案アルゴリズムの実装には、RTS アルゴリズムの実装に必要な 3 つの $N \times M/4$ サイズの 1 次元配列⁶⁾ に加え、連の始点と終点を記録する 2 つの $N \times M/4$ サイズの 1 次元配列が必要である。

ないものであり、文献 17) のアルゴリズムはラベル伝播に基づくアルゴリズムである。

6. おわりに

本論文では、同等ラベル解析に基づく1回走査ラベル付けアルゴリズムを提案した。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、平均連長の長い画像または対象物画素密度の低い画像において、本アルゴリズムが最も効率的であることを示した。

本論文では、画像の最外郭の画素は背景画素であることを前提としているが、最外郭の画素は背景画素に限らない場合、連のデータに行の情報に加えて工夫すれば、提案アルゴリズムも適応できる。

今後の研究課題として、提案アルゴリズムの計算量の解析や連長圧縮画像への応用などがあげられる。

謝辞 本研究の一部は、豊秋奨学会の研究助成を受けている。また、貴重なご指摘とコメントをいただいた査読者に感謝する。

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 (1988).
- 2) Hu, Q., Qian, G. and Nowinski, W.L.: Fast connected-component labeling in three-dimensional binary images based on iterative recursion, *Computer Vision and Image Understanding*, Vol.99, pp.414–434 (2005).
- 3) Rosenfeld, A. and Pfalts, J.L.: Sequential operations in digital picture processing, *J. ACM*, Vol.13, No.4, pp.471–494 (1966).
- 4) Haralick, R.M.: Some neighborhood operations, *Real Time/Parallel Computing Image Analysis*, pp.11–35, Plenum Press, New York (1981).
- 5) 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).
- 6) He, L., Chao, Y. and Suzuki, K.: A run-based two-scan labeling algorithm, *IEEE Trans. Image Processing*, Vol.17, No.5, pp.749–756 (2008).
- 7) Otsu, N.: A threshold selection method from gray-level histograms, *IEEE Trans. Systems Man and Cybernetics*, Vol.9, pp.62–66 (1979).
- 8) Rosenfeld, A. and Kak, A.C.: *Digital Picture Processing*, Vol.2, 2nd edition, Academic Press, San Diego, CA (1982).
- 9) 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, pp.206–220 (2004).
- 10) Suzuki, K., Horiba, I. and Sugie, N.: Linear-time connected-component labeling

based on sequential local operations, *Computer Vision and Image Understanding*, Vol.89, pp.1–23 (2003).

- 11) Wang, K.B., Chia, T.L. and Chen, Z.: Parallel Execution of a Connected Component Labeling Operation on a Linear Array Architecture, *J. Information Science and Engineering*, Vol.19, pp.353–370 (2003).
- 12) Wu, K., Otoo, E. and Suzuki, K.: Optimizing two-pass connected-component labeling algorithms, *Pattern Analysis & Applications*, doi:10.1016/j.patcog.2008.10.013 (2008).
- 13) Shapiro, L.G. and Stockman, G.C.: *Computer Vision*, Section 3.4, p.75 (2002). <http://www.cse.msu.edu/~stockman/Book/2002/Chapters/ch3.pdf>
- 14) Haralick, R.M. and Shapiro, L.G.: *Computer and Robot Vision*, Volume I, pp.28–48, Addison-Wesley (1992).
- 15) Samet, H.: Connected Component Labeling Using Quadrees, *J. ACM*, Vol.28, No.3, pp.487–501 (1981).
- 16) Appiah, K., Hunter, A., Dickinson, P. and Owens, J.: A Run-Length Based Connected Component Algorithm for FPGA Implementation, *Proc. International Conference on Field-Programmable Technology*, pp.177–184 (2008).
- 17) 井村誠孝：ラベリングクラス。
<http://chihara.naist.jp/people/STAFF/imura/products/labeling/labeling.pdf>

(平成 20 年 8 月 12 日受付)

(平成 21 年 3 月 6 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 年名古屋工業大学大学院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。画像処理、定理証明、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。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准教授。博士(学術)。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・シカゴ大学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年同大学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工学研究科博士後期課程修了。博士(工学)。同年同大学助手。2003年同大学大学院工学研究科情報工学専攻助教授。2007年同准教授。現在に至る。感性情報処理、ソフトコンピューティング、CG、CVの研究に従事。電子情報通信学会、日本知能情報ファジィ学会、感性工学会、IEEE、ACM各会員。



伊藤 英則(正会員)

1974年名古屋大学大学院工学研究科博士課程満了。工学博士号取得。同年NTT入社、横須賀研究所勤務。1985年(財)新世代コンピュータ技術開発機構出向。1989年より名古屋工業大学教授。感性情報、自動推論、マルチメディア、マルチエージェント等に興味を持つ。電子情報通信学会、人工知能学会、ファジィ学会各会員。