

三次元 2 値画像における高速ラベル付けアルゴリズム

何 立風[†] 巢 宇燕^{††} 鈴木 賢治^{†††} 中村 剛士^{††††}
伊藤 英則^{††††}

A High-Speed Labeling Algorithm for Three-Dimensional Binary Images

Lifeng HE[†], Yuyan CHAO^{††}, Kenji SUZUKI^{†††}, Tsuyoshi NAKAMURA^{††††},
and Hidenori ITOH^{††††}

あらまし 本論文では、同等ラベル解析に基づく三次元 2 値画像における連結成分の高速ラベル付けアルゴリズムを提案する。同等ラベル解析に基づくラベル付けアルゴリズムでは、注目対象物画素を処理する際、注目対象物画素に暫定ラベル付け、及び同等ラベル検出のためにマスクの画素をチェックしなければならない。このとき、画素チェックの順番は画素のチェック総回数に大きく影響し、アルゴリズムの効率を左右する要因となる。本論文で我々は、効率的な画素チェックの順番を提案し、それによって効率的な暫定ラベル付けと同等ラベル検出を実現した。また、従来手法との比較評価実験により、提案手法の有効性を示した。

キーワード ラベル付け, パターン認識, 三次元 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 値画像中の独立な対象物（連結成分）に別々のラベルを付けるラベル付けは、対象物の形状やサイズ（面積または体積）などの特徴を知るために欠くことができない最も基本的かつ重要な処理の一つである [1]。

二次元 2 値画像においては、様々なラベル付けアルゴリズムが提案され、それらはラベル伝搬法 [2], [3] と同等ラベル解析法 [4] ~ [6] に大別できる。

ラベル伝搬法は、まず、ラベル付けされていない対象物画素を画像中から探す^(注1)。このような対象物画素を見つけたら、その画素に新しいラベルを付ける。また、この対象物画素と連結しているすべての対象物画素に対し、同じラベルを付ける。ラベル伝搬法は、

このような手順でラベル付けを実現する。

一方、同等ラベル解析法^(注2)では、まず、1 回のラスト走査で画像中の対象物画素に暫定ラベルを付けながら、同じ連結成分に付けられた暫定ラベルを同等ラベルとして記録する。次に、同等ラベル解析を行い、すべての同等ラベルを一つの代表ラベルに対応させる。更に、各対象物画素に付けられた暫定ラベルをその暫定ラベルの代表ラベルに書き換える。同等ラベル解析法は、このような手順でラベル付けを実現する。

コンピュータ技術及び計測技術の発展により、三次元画像は医療分野をはじめ、様々な分野に 응용が広がっている [7]。二次元 2 値画像におけるラベル付けアルゴリズムは原則三次元 2 値画像にも応用できるが、三次元画像の場合、画素間の連結性は二次元 2 値画像よりも大分複雑になり、データ量も膨大になるため、更なる工夫が必要になる。三次元 2 値画像のラベル付けアルゴリズムにおいて、文献 [8] で提案された二次元 2 値画像における同等ラベル解析法に基づく 2 回走査ラベル付けアルゴリズムの直接的拡張である手法 [9]

[†] 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

^{††} 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

^{†††} シカゴ大学放射線医学研究科, 米国
Department of Radiology, The University of Chicago, USA

^{††††} 名古屋工業大学工学研究科, 名古屋市
Graduate School of Engineering, Nagoya Institute of Technology,
Nagoya-shi, 466-8555 Japan

(注1): 通常、ラスト走査によって、ラベル付けされていない対象物画素を探索する。

(注2): 同等ラベル解析法には、マルチ走査, 4 回走査及び 2 回走査アルゴリズムがある。2 回走査アルゴリズムは最も効率的である [5], [11] から、本論文では、2 回走査アルゴリズムだけを対象として議論する。

(従来手法 1 と呼ぶ) と、反復処理と再帰反復を用いたラベル伝搬法に基づくアルゴリズム [10] (従来手法 2 と呼ぶ) が提案されている。従来手法 2 の筆者らは、提案アルゴリズムが同等ラベル解析に基づく手法より効率的であることを主張している。

本論文では、文献 [5] で提案された 2 値画像における同等ラベル解析法に基づくラベル付けアルゴリズム (He ラベル付けアルゴリズムと呼ぶ) を三次元 2 値画像におけるラベル付けに拡張する。

2. He ラベル付けアルゴリズム

He ラベル付けアルゴリズムでは、次のように 2 回走査でラベル付けを行う。第一走査において、注目画素が背景画素であるときは何もしない。注目画素が対象物画素であるとき、図 1 に示したマスクにおいて、対象物画素があるかどうかを $p_3 \rightarrow p_1 \rightarrow p_2 \rightarrow p_4$ の順番でチェックする。対象物画素が見つかったとき、その対象物画素の暫定ラベルを注目対象物画素に付ける。マスクに対象物画素が存在しない場合、注目対象物画素に新たな暫定ラベルを付ける。一方、マスクにおいて、もともとは独立な二つの対象物画素が注目対象物画素の存在によって連結となった場合、その二つの画素に付けられた暫定ラベルは同一連結成分に属する同等ラベルであることが分かる。この場合、それら同等ラベルの統合を行う。

同等ラベルの統合には、同等ラベル集合と代表ラベルテーブルを用いる。

同等ラベル集合において、最小のラベルはその集合及びその集合中のすべての暫定ラベルの代表ラベルとする。暫定ラベル a の代表ラベルは、代表ラベルテーブル $r_table[]$ を用いて $r_table[a]$ で記録する。また、代表ラベル h である同等ラベル集合を $S(h)$ で表す。そうすることで、 $\forall l \in S(h), r_table[l] = h$ が成り立つ。

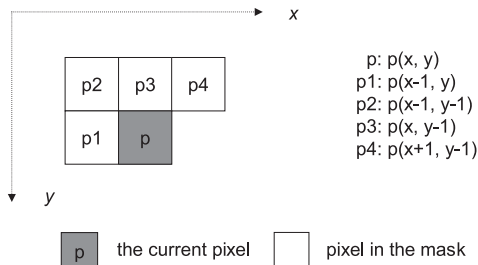


図 1 二次元 2 値画像における 8 連結走査マスク
Fig. 1 The mask for 2-dimensional binary images with 8-connectivity.

対象物画素に新たな暫定ラベル m を付けるとき、 $S(m) = \{m\}$, $r_table[m] = m$ とする。二つの同等ラベル p と q の統合を行うとき、 p と q の代表ラベルはそれぞれ u と v (つまり、 $p \in S(u)$, $q \in S(v)$, $r_table[p] = u$, $r_table[q] = v$) とすると、 $u = v$ であれば、暫定ラベル p と q は同じ同等ラベル集合に属していることが分かる。そのため、何もしない。 $u < v$ であれば、同等ラベル集合 $S(v)$ を $S(u)$ に合併するとともに、 $S(v)$ に属するすべての暫定ラベルの代表ラベルを u に改める。つまり、 $S(u) = S(u) \cup S(v)$, $\forall x \in S(v), r_table[x] = u$ 。一方、 $u > v$ であれば、同等ラベル集合 $S(u)$ を $S(v)$ に合併し、 $S(u)$ に属するすべての暫定ラベルの代表ラベルを v に改める。

走査中、注目画素のマスクに異なる暫定ラベル (つまり、同等ラベル) を発見するたびに、同等ラベルの統合を行うので、任意の時点において、走査済みの領域の各連結成分に付けられた暫定ラベルは同一同等ラベル集合にまとめられ、同一代表ラベルをもつ。そのため、第 1 走査が終了する時点、同じ連結成分に付けられたすべての暫定ラベルは同一同等ラベル集合に統合されており、同一代表ラベルをもつ。更に、第 2 走査で、各対象物画素の暫定ラベルをその代表ラベルに書き換えれば、ラベル付けが完了する。

3. 三次元 2 値画像におけるラベル付けの提案手法

本章では、He ラベル付けアルゴリズムを三次元 2 値画像のラベル付けへと拡張する。三次元画像の場合、対象物画素間の連結性について、6 連結、18 連結と 26 連結 [9], [10] があるが、本論文では、26 連結だけを考える。この場合、 $N \times M \times W$ サイズの三次元 2 値画像において、注目画素 $v(x, y, z)$ ($0 \leq x \leq N-1$, $0 \leq y \leq M-1$, $0 \leq z \leq W-1$) のマスクは図 2 に示す 13 画素 (つまり、走査済みの画素中、注目画素と 26 隣接しているすべての画素) から構成される。

三次元 2 値画像におけるラベル付けは、基本的に二次元と同様に次のような暫定ラベル付けと同等ラベルの検出及び統合によって行うことができる。注目画素が背景画素であれば、何もしない。注目画素が対象物画素であるとき、マスクに対象物画素がなければ、注目画素に新しいラベルを付ける。一方、マスクに対象物画素があれば、その中の任意のラベルを注目対象物画素に付けることができる。また、注目対象物画素の存在によって初めて連結となった対象物画素に付けら

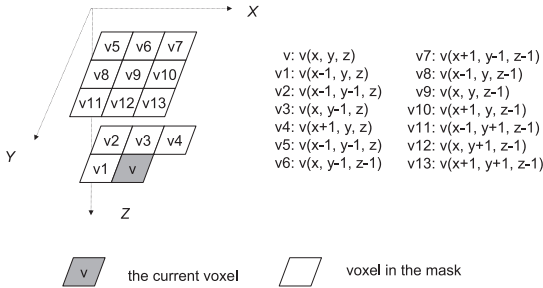


図 2 三次元 2 値画像における 26 連結走査マスク
 Fig. 2 The mask for 3-dimensional binary images with 26-connectivity.

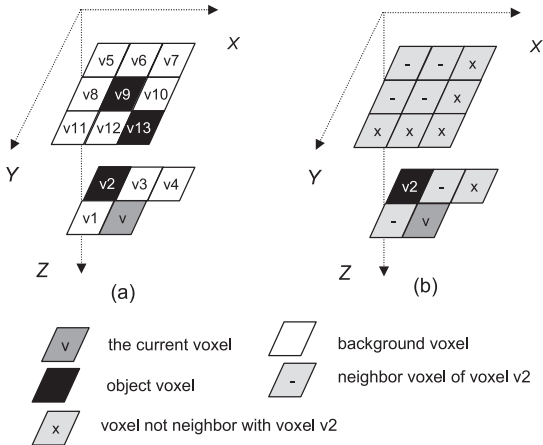


図 3 マスクの状況の 1 例
 Fig. 3 An example of mask.

れた暫定ラベルが同等ラベルであり、それらの統合を行えばよい。

マスクにある任意の暫定ラベルを注目対象物画素に付けることができるため、マスクにある各画素が対象物画素である確率が同じであれば、画素チェックの順番は暫定ラベル付けの効率に影響しない。しかし、マスクの同等ラベルの統合は、画素チェックの順番によって、処理の手間が大きくと変化する。

例として、図 3 (a) に示したケースを考える。 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{13}$ の順番 (順番 1 と呼ぶ) で画素をチェックするとき、まず、 v_1 をチェックする。 v_1 は背景画素であるため、次に、 v_2 をチェックする。 v_2 は対象物画素であるから、注目対象物画素に v_2 のラベルを付ける。また、マスクに v_2 と隣接していない画素 (つまり、 $v_4, v_7, v_{10}, v_{11}, v_{12}, v_{13}$) (図 3 (b)) が対象物画素であるとき、 v_2 の暫定ラベルとそれらの画素の暫定ラベルの統合を考えればよい。 $v_4, v_7, v_{10}, v_{11}, v_{12}$ は背景画素であるため、何もしなくてもよい。

v_{13} は対象物画素であるが、注目対象物画素の存在によって v_2 と初めて連結となる必要条件是、 v_9 が背景画素であることである。しかし、 v_9 は対象物画素であるため、画素 v_2 と画素 v_{13} が連結していることは対象物画素 v_9 を処理するときに分かっている。つまり、画素 v_2 の暫定ラベルと画素 v_{13} の暫定ラベルの統合は注目対象物画素を処理する前に既に完了している。そのため、何もしなくてよい。

一方、 v_9 からチェックすれば (順番 2 と呼ぶ)、 v_9 は対象物画素であるから、注目対象画素に v_9 の暫定ラベルを付ける。 v_9 はマスクの他のすべての画素と隣接しているので、 v_9 の暫定ラベルとマスクの他の対象物画素 (存在すれば) の暫定ラベルとの統合は注目対象物画素を処理する前に既に完了していることが分かる。そのため、それ以上の処理は何もない。

順番 1 では、注目対象物画素を処理するため、マスクの画素を 9 個チェックしなければならないが、順番 2 の場合、画素を 1 個だけチェックで済む。よって、注目画素が対象物画素であるとき、マスクの画素チェックの順番は処理の効率を左右する要因となる。

二次元画像の場合、マスクには四つの画素がある。その四つの画素は対象物画素であるか背景画像であるかによって、 $2^4 = 16$ パターンがある。文献 [5] では、各パターンにおいて同等ラベルの統合が必要となるパターンのカルノー図を用いて同等ラベルの統合をできるだけ避けるように画素チェックの順番を決定した。しかし、三次元画像の場合、マスクに 13 個の画素があるから、パターン数は $2^{13} = 8192$ 通りもあるため、カルノー図による解析が事実上困難である。また、マスクの 13 個の画素チェックの順番の総数は 13 の階乗、62 億通り以上もあるから、すべての順番を実装して比較することも不可能に近い。

走査中の任意の時点において、走査済み領域の各連結成分に付けられた暫定ラベルは同一同等ラベル集合に統合され、同一代表ラベルをもつことから、各連結成分に付けられた暫定ラベルを同一とみなすことができる。よって、マスクにおいて、互いに隣接している対象物画素に付けられたすべての暫定ラベルは既に同一同等ラベル集合に統合されており、それら暫定ラベルの統合を考えなくてもよいことが分かる。そのため、マスク中の各画素は対象物画素である確率が同じとすれば、隣接画素数の多い対象物画素において、チェックしなければならない画素 (つまり、隣接していない画素) 数が少ないため、隣接画素数の多い順でマスク中の画素

チェックの順番を決めれば効率的であると考えられる。

図 2 により、マスクにおける各画素の隣接画素数を表 1 に示す。

表 1 により、マスク中の画素チェックの順番は $v_9 \rightarrow v_3, v_6 \rightarrow v_1, v_8 \rightarrow v_{10} \rightarrow v_2, v_5, v_{12} \rightarrow v_4, v_7 \rightarrow v_{11} \rightarrow v_{13}$ となる。

v_9 が対象物画素であるとき、 v_9 はマスクのすべての画素と隣接していることから、マスクにおいて、すべて暫定ラベルの統合は完了している。そのため、注目対象物画素に v_9 のラベルを付けるだけでよい。

v_9 は背景画素、 v_3 が対象物画素であるとき、まず、注目対象物画素に v_3 のラベルを付ける。

v_3 と隣接していないのは、 v_{11}, v_{12} と v_{13} の 3 画素だけである (図 4(a))。そのため、 v_3 の暫定ラベルとこの三つの画素の暫定ラベルの統合だけを考えればよい。

このときも、三つの画素中、隣接画素数の多い画素からチェックする。画素 v_{12} の隣接画素数は 2、 v_{11} と v_{13} の隣接画素数は 1 であるから、 v_{12} からチェックを行う。 v_{12} が対象物画素であり、かつ、 v_1, v_8 と v_{10} が背景画素であるとき (図 4(b)) のみ、注目対象物画素の存在で v_3 が v_{12} と連結していることが初めて分かる。このとき、 v_3 の暫定ラベルと v_{12} の暫定ラベルの統合が必要となる。また、 v_{12} は v_{11} と v_{10} に隣接しているので、これらの画素の暫定ラベルの統合が

必要であれば完了している。そのため、 v_{11} と v_{10} はチェックしなくてもよい。

一方、 v_{12} が背景画素であれば、 v_{11} は対象物画素であり、かつ、 v_1 と v_8 が背景画素であるとき (図 4(c))、画素 v_3 と画素 v_{12} が連結していることが注目対象物画素の存在によって初めて分かる。このとき、 v_3 の暫定ラベルと v_{12} の暫定ラベルを統合する必要がある。また、 v_{13} が対象物画素であり、かつ、 v_{10} が背景画素であるとき (図 4(d))、 v_3 の暫定ラベルと v_{13} の暫定ラベルの統合が必要である。

他のケースについても同様に分析して処理できる。例えば、 $v_9, v_3, v_6, v_1, v_8, v_{10}, v_2, v_5, v_{12}, v_4$ が背景画素、 v_7 が対象物画素であるとき、マスクの状態を図 5 のように示すことができる。このとき、まず、注目対象物画素に v_7 のラベルを付け、 v_{11} が対象

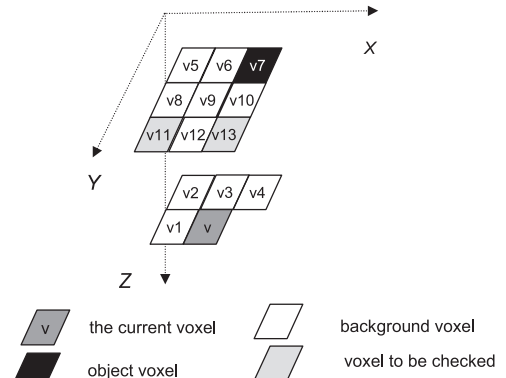


図 5 $v_9, v_3, v_6, v_1, v_8, v_{10}, v_2, v_5, v_{12}, v_4$ は背景画素、 v_7 は対象物画素であるときのマスク
Fig. 5 The mask when $v_9, v_3, v_6, v_1, v_8, v_{10}, v_2, v_5, v_{12}, v_4$ are background pixels, and v_7 is an object pixel.

表 1 マスクにおいて各画素の隣接画素数

Table 1 The number of neighbor pixels of a voxel in mask.

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
8	6	9	5	6	9	5	8	12	7	4	6	3

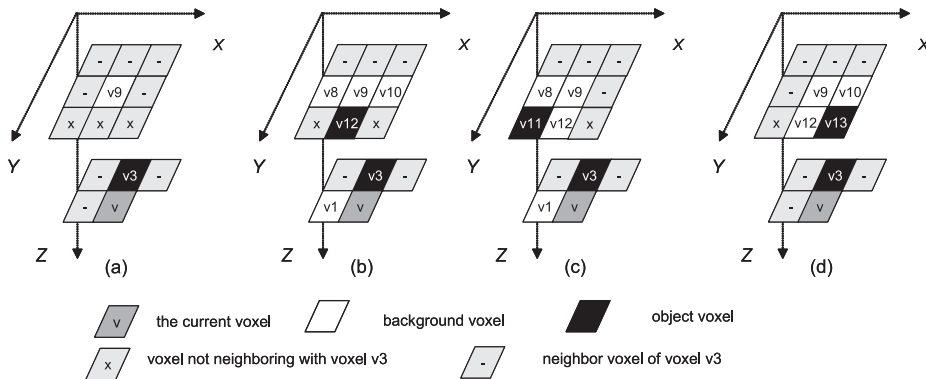


図 4 v_9 は背景画素、 v_3 は対象物画素であるときの処理
Fig. 4 The process when v_9 is a background pixel and v_3 is an object pixel.

$l_i (i = 1, 2, \dots, 13)$: v_i に付与された暫定ラベルである。ただし, v_i が背景画素であるとき, l_i の値は 0 である。

$resolve(u, v)$: 暫定ラベル u に属する同等ラベル集合と暫定ラベル v が属する同等ラベル集合の併合処理

Label: 暫定ラベルの通し番号

$l(x, y, z)$: $v(x, y, z)$ に付与される暫定ラベル

```

Label = 0;
if ( $l_9 > 0$ )
  |  $l(x, y, z) = l_9$ ;
else if ( $l_3 > 0$ )
  |  $l(x, y, z) = l_3$ ;
  | if ( $l_{12} > 0 \ \& \ l_8 == 0 \ \& \ l_{10} == 0$ )
  | |  $resolve(l_3, l_{12})$ ;
  | else
  | | if ( $l_{11} > 0 \ \& \ l_8 == 0$ )
  | | |  $resolve(l_3, l_{11})$ ;
  | | end of if
  | | if ( $l_{13} > 0 \ \& \ l_{10} == 0$ )
  | | |  $resolve(l_3, l_{13})$ ;
  | | end of if
  | end of if
else if ( $l_6 > 0$ )
  |  $l(x, y, z) = l_6$ ;
  | if ( $l_{12} > 0 \ \& \ l_8 == 0 \ \& \ l_{10} == 0$ )
  | |  $resolve(l_6, l_{12})$ ;
  | else
  | | if ( $l_{11} > 0 \ \& \ l_8 == 0$ )
  | | |  $resolve(l_6, l_{11})$ ;
  | | end of if
  | | if ( $l_{13} > 0 \ \& \ l_{10} == 0$ )
  | | |  $resolve(l_6, l_{13})$ ;
  | | end of if
  | end of if
else if ( $l_1 > 0$ )
  |  $l(x, y, z) = l_1$ ;
  | if ( $l_{10} > 0 \ \& \ l_{12} == 0$ )
  | |  $resolve(l_1, l_{10})$ ;
  | else if ( $l_7 > 0$ )
  | |  $resolve(l_1, l_7)$ ;
  | | if ( $l_{13} > 0$ )
  | | |  $resolve(l_1, l_{13})$ ;
  | | end of if
  | else if ( $l_4 > 0$ )
  | |  $resolve(l_1, l_4)$ ;
  | | if ( $l_{13} > 0$ )
  | | |  $resolve(l_1, l_{13})$ ;
  | | end of if
  | else if ( $l_{13} > 0 \ \& \ l_{12} == 0$ )
  | |  $resolve(l_1, l_{13})$ ;
  | end of if

```

```

else if ( $l_8 > 0$ )
  |  $l(x, y, z) = l_8$ ;
  | if ( $l_{10} > 0 \ \& \ l_{12} == 0$ )
  | |  $resolve(l_8, l_{10})$ ;
  | else if ( $l_7 > 0$ )
  | |  $resolve(l_8, l_7)$ ;
  | | if ( $l_{13} > 0$ )
  | | |  $resolve(l_8, l_{13})$ ;
  | | end of if
  | else if ( $l_4 > 0$ )
  | |  $resolve(l_8, l_4)$ ;
  | | if ( $l_{13} > 0$ )
  | | |  $resolve(l_8, l_{13})$ ;
  | | end of if
  | else if ( $l_{13} > 0 \ \& \ l_{12} == 0$ )
  | |  $resolve(l_8, l_{13})$ ;
  | end of if
else if ( $l_{10} > 0$ )
  |  $l(x, y, z) = l_{10}$ ;
  | if ( $l_{11} > 0 \ \& \ l_{12} == 0$ )
  | |  $resolve(l_{10}, l_{11})$ ;
  | end of if
  | if ( $l_5 > 0$ )
  | |  $resolve(l_{10}, l_5)$ ;
  | else if ( $l_2 > 0$ )
  | |  $resolve(l_{10}, l_2)$ ;
  | end of if
else if ( $l_2 > 0$ )
  |  $l(x, y, z) = l_{12}$ ;
  | if ( $l_4 > 0$ )
  | |  $resolve(l_{12}, l_4)$ ;
  | else if ( $l_7 > 0$ )
  | |  $resolve(l_{12}, l_7)$ ;
  | end of if
  | if ( $l_2 > 0$ )
  | |  $resolve(l_{12}, l_2)$ ;
  | else if ( $l_5 > 0$ )
  | |  $resolve(l_{12}, l_5)$ ;
  | end of if
else if ( $l_5 > 0$ )
  |  $l(x, y, z) = l_5$ ;
  | if ( $l_4 > 0$ )
  | |  $resolve(l_5, l_4)$ ;
  | else if ( $l_7 > 0$ )
  | |  $resolve(l_5, l_7)$ ;
  | end of if
  | if ( $l_{11} > 0$ )
  | |  $resolve(l_5, l_{11})$ ;
  | end of if
  | if ( $l_{13} > 0$ )
  | |  $resolve(l_5, l_{13})$ ;
  | end of if

```

```

else if ( $l_2 > 0$ )
  |  $l(x, y, z) = l_2$ ;
  | if ( $l_4 > 0$ )
  | |  $resolve(l_2, l_4)$ ;
  | else if ( $l_7 > 0$ )
  | |  $resolve(l_2, l_7)$ ;
  | end of if
  | if ( $l_{11} > 0$ )
  | |  $resolve(l_2, l_{11})$ ;
  | end of if
  | if ( $l_{13} > 0$ )
  | |  $resolve(l_2, l_{13})$ ;
  | end of if
else if ( $l_4 > 0$ )
  |  $l(x, y, z) = l_4$ ;
  | if ( $l_{11} > 0$ )
  | |  $resolve(l_4, l_{11})$ ;
  | end of if
  | if ( $l_{13} > 0$ )
  | |  $resolve(l_4, l_{13})$ ;
  | end of if
else if ( $l_7 > 0$ )
  |  $l(x, y, z) = l_7$ ;
  | if ( $l_{11} > 0$ )
  | |  $resolve(l_7, l_{11})$ ;
  | end of if
  | if ( $l_{13} > 0$ )
  | |  $resolve(l_7, l_{13})$ ;
  | end of if
else if ( $l_{11} > 0$ )
  |  $l(x, y, z) = l_{11}$ ;
  | if ( $l_{13} > 0$ )
  | |  $resolve(l_{11}, l_{13})$ ;
  | end of if
else if ( $l_{13} > 0$ )
  |  $l(x, y, z) = l_{13}$ ;
else
  | Label = Label + 1;
  |  $l(x, y, z) = Label$ ;
end of if

```

図 6 提案アルゴリズムの擬似コード

Fig. 6 The pseudo codes of our proposed algorithm.

物画素であれば、 v_7 の暫定ラベルと v_{11} の暫定ラベルの統合を行う。また、 v_{13} が対象物画素であれば、 v_7 の暫定ラベルと v_{13} の暫定ラベルの統合を行う。

暫定ラベル間の連結性解決は、同等ラベル集合と代表ラベルテーブルを用いて二次元 2 値画像の場合と同様に行うことができる。

提案アルゴリズムにおいて対象物画素 $v(x, y, z)$ を処理する擬似コードを図 6 に示す。

4. 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 3.0GHz + 3.0GHz CPUs, 2 GByte Memory, Mandriva Linux OS) 上で、CPU シングルコアを用いた従来手法 1 及び従来手法 2^{注3)}との比較評価実験を行っ

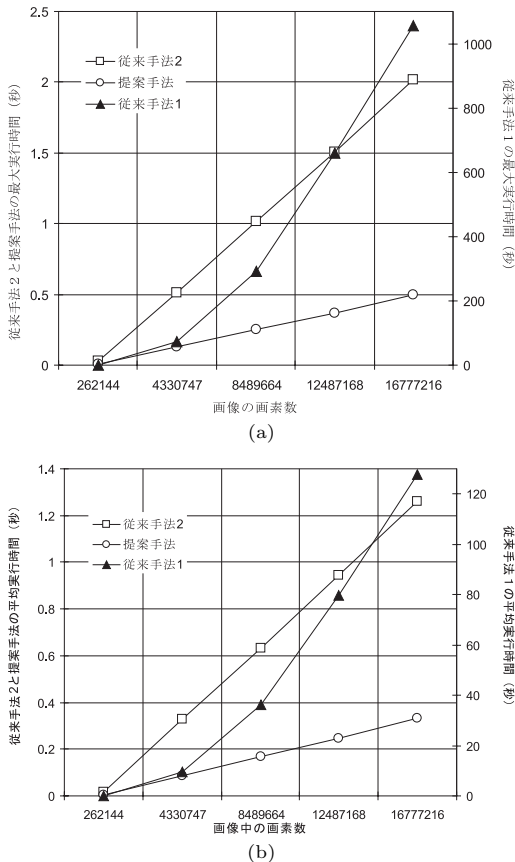


図 7 画像の画素数に対する実行時間 (a) 最大実行時間, (b) 平均実行時間

Fig. 7 The execution times versus the number of pixels in an image: (a) the maximum execution time; and (b) the average execution time.

た。なお、実験に用いた従来手法 1 のプログラムは MIST (Media Integration Standard Toolkit, <http://mist.suenaga.cse.nagoya-u.ac.jp/trac/>) からダウンロードしたものであり、従来手法 2 と提案手法のプログラムは C 言語で実装したものである。三つのプログラムとも GNU C compiler version 4.2.3 のコンパイラによってコンパイルした。

五つのサイズ $64 \times 64 \times 64$, $163 \times 163 \times 163$, $204 \times 204 \times 204$, $232 \times 232 \times 232$ と $256 \times 256 \times 256$ のノイズ画像を用いて実験を行った。各サイズの画像において、1,000 階調までの白色雑音を加え、しきい値を 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 値画像を用意した。このようなノイズ画像は複雑な連結成分をもっていることから、ラベル付けアルゴリズムの厳格な評価に適切と考える。各サイズにおいて従来手法と提案手法の最大及び平均実行時間をそれぞれ図 7 (a) と図 7 (b) に示す。

$256 \times 256 \times 256$ サイズの各画像を用いて、各手法の画像密度に対する実行時間を図 8 に示す。

次に、従来手法 2 を提案した文献 [10] と同様、Montreal Neurological Institute のホームページ (<http://www.bic.mni.mcgill.ca/brainweb>) からダウンロードしたグレースケール (輝度値は 0 から 255

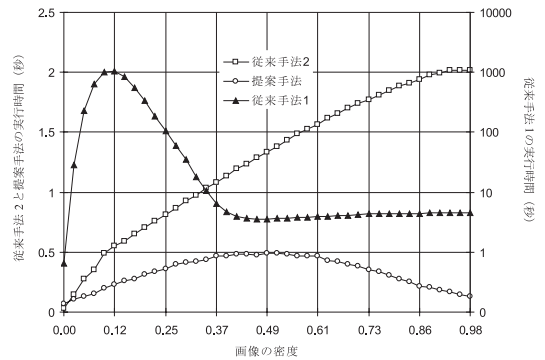


図 8 画像密度に対する実行時間

Fig. 8 The execution time versus density of an image.

(注 3): 文献 [10] では、反復処理と再帰処理 (未処理の対象物画素の局所領域を反復処理でラベル付けを行い、局所領域の外郭の画素に対して再帰処理を行う) を用いた三次元 2 値画像におけるラベル付け手法を二つ提案した。その一つは、再帰と反復処理を用いて画像に直接ラベル付けを行う。もう一つは、文献 [2] で提案された輪郭追跡法の拡張 (反復処理と再帰処理で輪郭追跡を行う) である。文献 [10] で示した実験結果により、2 番目の手法の方が性能が良い。また、局所領域のサイズが小さければ小さいほど処理速度が向上するが、必要な記憶領域も増える。本論文は、最小の局所領域サイズ (つまり、1) の 2 番目の手法を用いて実験を行った。

まで) $181 \times 217 \times 181$ サイズの頭部の仮想 3D MR 画像 (ノイズなし, 照度一様) を, 補間によりサイズ $362 \times 434 \times 362$ に拡張し, しきい値を 120 から 5 まで (画像密度は 2.5% から 24.1% まで変化する) 5 ごとに減らして 2 値化した合計 24 種類の 2 値画像を用いて実験を行った. 図 9 は, 実験に用いる仮想 3D MR 頭部の 1 枚スライス画像としきい値 45 に対応する 2 値画像を図 9 に示す. 各手法のしきい値に対する実行時間を表 2 に示す.

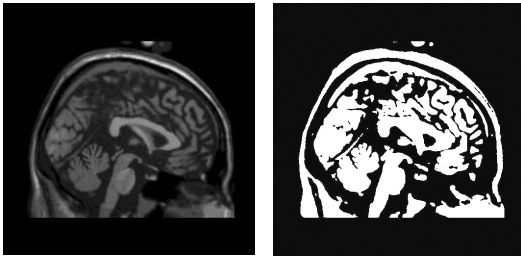


図 9 実験に用いる仮想 3D MR 頭部の 1 枚スライス画像としきい値 45 に対応する 2 値画像

Fig.9 A slice of the 3D MR image used in our test and the binary image corresponding with threshold 45.

表 2 3D MR 頭部画像におけるしきい値に対する各手法の実行時間 (秒)

Table 2 The execution times [s] of different algorithms on 3D MR images.

しきい値	密度	連結成分数	従来手法 1	従来手法 2	提案手法
120	2.41	2343	12.717	0.363	0.181
115	2.94	1769	15.340	0.400	0.190
110	3.46	1430	17.962	0.438	0.197
105	4.01	1404	20.442	0.477	0.214
100	4.59	1321	22.119	0.517	0.215
95	5.22	1197	24.399	0.565	0.211
90	5.90	1058	27.636	0.613	0.213
85	6.58	911	25.444	0.656	0.209
80	7.22	737	23.455	0.698	0.215
75	7.81	609	21.461	0.734	0.213
70	8.33	536	20.583	0.762	0.217
65	8.80	384	18.096	0.786	0.215
60	9.23	286	16.397	0.812	0.217
55	9.68	226	14.499	0.839	0.210
50	10.15	201	12.527	0.867	0.217
45	10.68	182	12.693	0.902	0.213
40	11.26	270	12.065	0.935	0.208
35	11.88	919	19.363	0.963	0.215
30	12.60	3672	30.231	1.004	0.202
25	13.63	9793	50.310	1.077	0.201
20	15.27	9405	49.916	1.219	0.207
15	17.72	649	19.057	1.385	0.219
10	20.65	4	4.535	1.543	0.221
5	23.10	1	0.969	1.640	0.197

5. 考 察

従来手法 1 は, 文献 [8] で提案された二次元 2 値画像における同等ラベル解析法に基づく 2 回走査ラベル付けアルゴリズムを三次元 2 値画像に対応するようそのまま拡張したものである. この手法では, 一次元代表ラベルテーブル $T()$ だけを用いて同等ラベルの統合を行う. このとき, 暫定ラベル l の代表ラベルを $T(l)$ で表す. 注目対象物画素を処理する際, マスクに対象物画素が存在しない場合, 注目対象物画素に新たな暫定ラベル $Label$ (初期値は 1 である) を付与し, $Label$ を 1 増やし, $T(Label) = Label$ の処理を行う. マスクに対象物画素が存在する場合, それらに付与した暫定ラベル中の最小ラベル $h(1)$ を注目対象物画素に付与する. そのため, マスクにある 13 個の画素をすべてチェックしなければならない上, 最小ラベルの計算も必要である. また, マスクにある異なる同等暫定ラベル $h(1), \dots, h(n)$ の統合は, 次のように行う.

```

for(i = 1; i < Label; i++)
    for(j = 2; j <= n; j++)
        if(T(i) == h(j))
            T(i) = h(1);
        end of if
    end of for
end of for
    
```

すなわち, 注目対象物画素が属する連結成分に付けられたすべての暫定ラベル (同等ラベル) の代表ラベルを, それらの最小ラベルに書き換える. そのとき, これまでに処理済みの画像領域に対して付与したすべての暫定ラベル (その数は $Label$ である) をマスクにある各暫定ラベルと比較しなければならない. マスクにおいて, 異なる暫定ラベルの数が m であるとき, $Label \times m$ 回の処理が必要である. $N \times M \times W$ サイズの画像において, $Label$ の最大値は $N \times M \times W / 27$ であるため, このアルゴリズムの最大計算量は $\mathcal{O}(N^2 \times M^2 \times W^2)$ であると考えられる. そのため, 複雑連結成分をもつ画像 (暫定ラベル数が多い画像) において, 実行時間は極めて長くなる.

これに対して, 提案手法ではマスクの状態によってチェックする画素数が変わる一方^(注4), 最小ラベルの

(注4): $256 \times 256 \times 256$ サイズの 41 枚のノイズ画像を用いて実験した結果, 対象物画素を処理するとき, 提案手法がマスクの画素をチェックする平均回数は 3.86 であった.

計算は必要ない．更に，マスクにおいて，異なる暫定ラベルに対する統合は，それら暫定ラベルが属する同等ラベル集合にある暫定ラベルのみと関係する．

各サイズのノイズ画像における実験結果（図 7）においては，従来手法 1 の実行時間は画像のサイズに対し 2 乗倍の関係をもつことがうかがえる（従来手法 1 の画像画素数 N_p （百万）に対する最大実行時間 T_m （秒）の関係は $T_m = 3.26 \times N_p^2 + 10.17 \times N_p - 14.71$ で，平均実行時間 T_a （秒）の関係は $T_a = 0.38 \times N_p^2 + 1.42 \times N_p - 1.70$ でそれぞれ近似できる）のに対して，提案手法の実行時間は画像のサイズと比例している．また，図 8 より，従来手法 1 の実行時間は暫定ラベル数が大きいときに最大となっていることが分かる．なお， $256 \times 256 \times 256$ サイズのノイズ画像においては，提案手法は従来手法 1 に比べ平均 380 倍以上速い．

仮想 3D MR 頭部画像における実験結果（表 2）より，提案手法はすべての画像において実行時間があまり変動しないが，従来手法 1 は連結成分の数が多い画像において実行時間が長くなる．なお，すべての画像において，提案手法は従来手法 1 より最低約 5 倍，平均 98 倍，最大 250 倍以上速いことが分かる．

従来手法 2 では，ある対象物画素に付与されたラベルを，隣接する対象物画素に伝搬するために，その画素の 26 個隣接画素をすべてチェックしなければならない．そのため，実行時間は画像内の対象物画素の数とほぼ比例する．図 8 から，従来手法の実行時間は画像密度の増加に従ってほぼ線形増加していることが確認できる．

それに対して提案手法では，処理中の対象物画素に対して，マスクの状態によってチェックしなければならない隣接画素の数が変わる上，同等ラベルの統合に関係する処理（同等ラベルの合併と代表ラベルテーブルの更新）も連結成分の複雑性により変わる．提案手法では，連結成分が複雑であればあるほど，その処理時間が長くなる．密度の低いノイズ画像においては，連結している対象物画素が少ないので，同等ラベルの統合はあまり発生しないため実行時間が短い．一方，密度の高いノイズ画像においては，各対象物画素のマスクにおいて， v_9 の画素も対象物画素であることが多いので，同等ラベルの統合もあまり起こらない．そのため，実行時間も短い^(注5)．したがって，提案手法の最大実行時間はノイズ画像の密度 0.5 前後で発生すると推測できる．

以上の分析は，画像密度に対する実験結果（図 8）

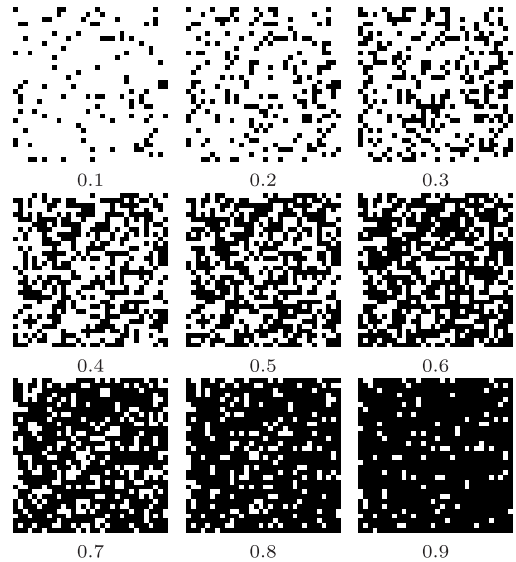


図 10 各種密度のノイズ画像

Fig. 10 Noise images with various densities.

と一致する．図 10 は密度が異なる 9 枚のノイズ画像の断面図の一部を示す．図 10 から分かるように，画像密度が 0.1 から 0.5 までに増加していく過程で，連結成分は単純なものから複雑なものになっていく．また，それに対応するように，提案手法の実行時間も増加している．しかし，密度が 0.5 以上になると，密度の増加に従って，連結成分の複雑性は逆に下がり，提案手法の実行時間も減少している．提案手法では，密度 0.5 前後で最大実行時間を取る．このように，提案手法の実行時間は連結成分の複雑性に関連すると考えられる．

提案手法は，すべての画素を 2 回処理しなければならないことから，画像密度の極めて低い画像においては，従来手法 2（処理時間は対象物画素数とほぼ比例する）の方が効率的であるが，画像密度が 0.02 以上になると，提案手法の方が効率的である．特に，画像の密度が 1 に近いと，提案手法は従来手法 2 より 15 倍以上速くなる．

仮想 3D MR 頭部画像において，表 2 から分かるように，提案手法の実行時間は各しきい値においてほとんど変わらないが，従来手法 2 の実行時間は，しきい値の減少（密度の増加）に従って増加する．なお，従

(注5): ただし，このとき，画素 v_9 のチェックや対象物画素に暫定ラベル付けを行う必要があるから，密度の低い画像より全体の処理時間が長くなる．

来手法 2 に比べ最小で 2 倍, 平均で 4 倍, 最大で 8 倍以上速いことが分かる。

6. む す び

本論文では, 同等ラベル解析に基づく三次元 2 値画像における新しいラベル付けアルゴリズムを提案した。また, 比較評価実験により, 提案手法が従来手法より高速であることを示した。

更に, 文献 [11] で提案された He アルゴリズムに対する改良 (注目対象物画素の一つ前の画素が対象物画素か背景画素かによって処理を分けて行う) を提案手法に取り込めば, 更なる高速化が期待できる。

将来の課題として, 提案手法の計算量の分析や, 処理手順が最も効率的であるかどうかの検証, より効率的な処理手順の提案などが考えられる。

謝辞 本研究の一部は, 人工知能研究振興財団の助成による。

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] F. Chang, C.J. Chen, and C.J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," CVIU, vol.93, no.2, pp.206-220, 2004.
- [3] Y. Shima, T. Murakami, M. Koga, H. Yashiro, and H. Fujisawa, "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.
- [4] Y. Shirai, "Labeling connected regions," in Three-Dimensional Computer Vision, pp.86-89, Springer-Verlag, New York, 1987.
- [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," 信学論 (D), vol.J91-D, no.4, pp.1016-1024, April 2008.
- [6] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," CVIU, vol.89, no.1, pp.1-23, 2003.
- [7] 3 次元画像コンファレンス. <http://www.3d-conf.org/>
- [8] A. Rosenfeld and A. Kak, Digital picture processing, vol.2, Academic Press, San Diego, CA, 1976.
- [9] 米倉達広, 横井茂樹, 鳥脇純一郎, 福村晃夫, "3 次元デジタル空間における図形の連結性とオイラー数," 信学論 (D), vol.J65-D, no.1, pp.80-87, Jan. 1982.
- [10] Q. Hu, G. Qian, and W. Nowinski, "Fast connected-component labeling in three-dimensional binary images based on iterative and recursion," CVIU, vol.99, pp.414-434, 2005.
- [11] 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," 信学論 (D), vol.J92-D, no.6, pp.951-955,

June 2009.

(平成 21 年 5 月 22 日受付, 7 月 28 日再受付)



何 立風

1997 名古屋工業大学工学研究科博士後期課程了。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, 医用画像処理等に興味をもつ。中国陝西科学技術大学客員教授。



巢 宇燕

2000 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了。博士 (学術)。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。画像処理, 人工知能, CAD などに興味をもつ。中国陝西科学技術大学客員教授。



鈴木 賢治

平 3 名城大・理工・電気電子卒。平 5 同大大学院修士課程了。同年 (株)日立メディコ技術研究所入社。平 9 愛知県立大学情報科学部助手。平 13-14 米国・シカゴ大放射線医学研究科カートロスマン放射線像研究所客員研究員。平 14 同研究員。平 15 同研究講師, 平 16 同研究助教授。平 18 同大放射線医学研究科助教授。平 19 同大医用物理学研究科助教授兼任。平 19 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・認識及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。



中村 剛士 (正員)

1998 名古屋工業大学工学研究科博士後期課程了。博士 (工学)。現在, 同大学情報工学科准教授。CG, 感性情報, ソフトコンピューティングなどに興味をもつ。ACM, IEEE, 日本知能情報ファジィ学会, 感性工学会各会員。



伊藤 英則 (正員)

1974 名古屋大学大学院工学研究科博士課程満了。工博。現在, 名古屋工業大学教授。感性情報, 画像処理などに興味をもつ。人工知能学会, ファジィ学会各会員。