輪郭線情報を利用したラベリング法

宮崎　敬

One Labeling Method Using Contour Lines

Takashi MIYAZAKI

Labeling algorithm is an image processing that marks individual connected regions to distinguish or separate. That is fundamental and important, especially in visiual recognition. One of them is the raster scanned labeling. This algorithm, however, takes a long time to mark respective regions consecutive numbers, as, for example, in the case of spreading branches or bending intricately. In this paper, first of all, we have investigated the problem, and then proved that a number of labels, called the evanescent label, were made for such regions and it was necessary to unify them from their connections. So we found the idea that we could decrese them by surrounding with the contour lines. And then this paper proposes the labeling algorithm using contour lines. Finally, simulation results comparison with the raster scanned algorithm are described.

1. はじめに

画像処理分野の中の視覚認識関係において，画像中に存在する複数の領域を分離する処理方法の一つとしてラベリング法がある．例えば，細胞の顕微鏡写真中における細胞の個数を調べるときや，スキャナー等をとうして取り込まれた図面中における連結領域の数を調べるとき等に利用される重要な処理である．

これまでのラベリングの処理方法を大別するとランダム走査形とラスタ走査形の二つに分けられる[1]．ランダム走査形は，ラベリングの対象領域をランダム走査により見つけ，ラベルの伝搬を繰り返しながらラベリングを行う方法である．ハードウェア化するに際して，この画像をランダムにアクセスする難しさがあり，実用化には不向きな方法である．一方ラスタ走査形は，単純に画像をラスタ走査し画像をアクセスするだけでよいので，ハードウェア化が容易である．この方法は次のような手順でラベリングを行うものである．初めに，設定した2×3のウィンドウを画像に対してラスタ走査しながら連結領域を見つけ，ラベル付けの対象画素であればウィンドウ内の他の画素の分析を行い仮ラベル付けをする．次にこの仮ラベル同士の接続関係を基にして仮ラベルの統一処理を行う[2]．しかし，この方法は仮ラベルの統一のために何度もラスタ走査を行わなければならないので多大な処理時間が必要であるという問題点を持っていた．

* 電気工学科　講師
原稿受付　平成3年9月30日

本論文では，初めにこの問題点を解決するために，従来のラスタ走査形ラベリング法を分析し検討する．次に輪郭線情報を利用して仮ラベルの発生を低減したラスタ走査形ラベリング法を提案する．最後に，基準パターンを用意し，従来のラスタ走査形ラベリング法と本手法の処理速度を比較し評価を行う．

2. ラスタ走査形ラベリング法

2—1. アルゴリズム

従来のラスタ走査形ラベリング法の処理概要を図1に示す．この方法は，画像をラスタ走査し，処理対象となる画素が見つかった時，2×3のウィンドウをその画素の近傍に設定し，その近傍画素の情報によりラベリングを行うものである．この処理は次に上げる仮ラベル付けとラベルの統一に大きく二つに分けられる．

（I） 仮ラベル付け

まず画像データ（2値画像）を入力する．次に画面の左上をラスタ走査開始点とし，ラスタ走査をしていく．そして画素が1の箇所を捜し，図2に示す4つの隣接画素の値を配列に記憶する．この配列の記憶をもとにラベルを付けていく．配列の4つの値の和が0であった場合，つまり4つの隣接点が全て0であった場合，新しいラベルを付ける．次に配列の4つの値の和が0でなかった場合，つまり4つの隣接点のどこかにラベルが付いていた場合，そのなかで一番古いラベルを付ける．ラベルが一種類だけだったら，そのラベルを付ける．以上の処理を全ての1の画素について行い，ラスタ走査を終了する．

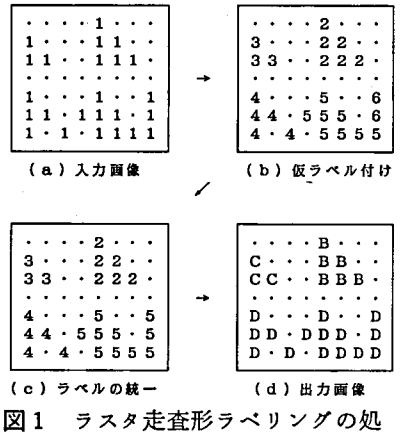


図1 ラスタ走査形ラベリングの処理手順

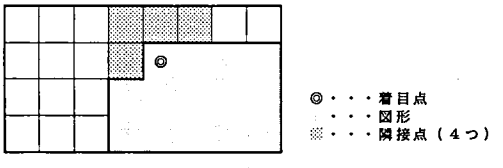


図2 仮ラベル付け用ウィンドウ

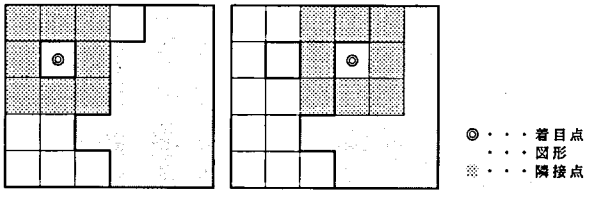


図3 ラベル統一用ウィンドウ

（II） ラベルの統一

仮ラベル付けをしたあと，連結領域の状態をみると2通りある．一つはラベリングが完成した状態になる場合で，連結領域が単純な形状の場合の処理結果である．もう一つは，一つの連結領域に複数の仮ラベルがついている状態で，連結領域が複雑である場合の処理結果である．一般に，ラベルの統一は後者の場合に行うことが多く，方法は仮ラベル付けの処理と基本的に同じである．

再びラスタ走査を開始しラベルの付いている画素を捜す．その時，今度は8つの隣接している画素のラベルの値と，着目している画素を配列に記憶する．着目している画素の配列の値を除いた，配列の8つの値の和が0であったら孤立点とみなし0でなかったらラベルを付け換える．方法は配列の値のなかで一番小さい値，つまり一番古いラベルを選んで付け換える．この処理をラベルの付いている全ての画素について行う．全ての連結領域のラベルが統一されるまでラスタ走査を繰り返す．

ラベルを統一した画面をみると，ラベルの値が飛んでいる場合がある．この対策としては，ラベルを統一する時に不用になったラベルを記憶しておくことにより連続したラベルに付け換えることができる．

2—2. 問題点と対策

ラスタ走査形ではラスタ走査の方向によって処理時間に違いが生じる．一般にラスタ走査の開始点を左上一点に固定するよりも，画像の対角線上の隅点を交互に開始点としたり，あるいは四隅の各点を順次選んで開始点とした方が処理時間が少ない．つまり，ラスタ走査の方向が一定であるとラベルの伝搬によるラベルの付け換えが少しずつしか行われないためである．特に連結領域の形状として枝分かれが複雑か多数ある場合に著しい．

このことから，各連結領域を最初に一つのラベルで囲んでおくと，つまり，輪郭線をとるとラスタ走査の方向に依存せず，また，凸部には同じラベルが付いているために仮ラベルの数を減少させることが可能となる．

仮に，連結領域の中に穴がある場合には，領域の中にも別の輪郭線のラベルが生じることになる．しかし，2回目のラスタ走査で，仮ラベル同士での結合点が必ず発生するので，これを記憶しておき最後に接続関係を調べることにより統一することが可能である．

3. 本 手 法

輪郭線追跡処理は，図形境界がつくる閉曲線を，全ての成分について抽出する処理であり，主に塊状図形の輪郭形状を解析するのに重要な役割を果たす．また，この処理で得られる輪郭線は，元の図形を画像データのまま記憶するよりもはるかに少ない容量で記憶できることから，データ圧縮技術の一つとしても利用される．

また，輪郭線を抽出する方法には，図形境界の画素かどうかを判断しながらラスタ走査し

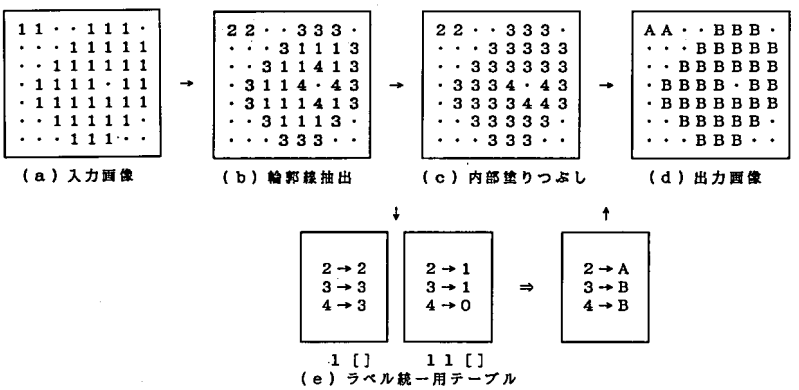


図4　本手法の処理手順

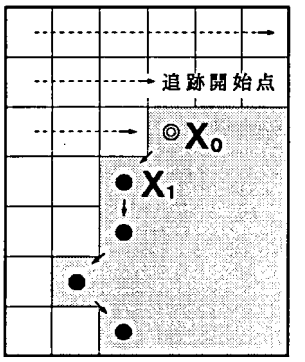
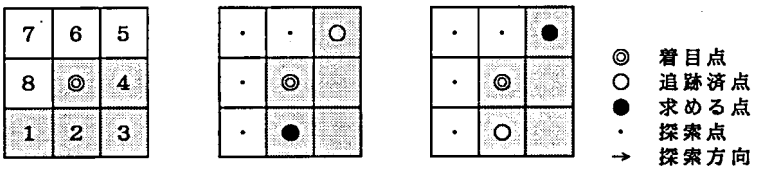


図5 追跡開始点を探すためのラスタ走査



（a）初期探索 （b）反時計回り探索 （c）時計回り探索

図6 輪郭線の探索順序

ていく方法があるが，今回はラベル付けを行うのに有効な，境界画素を追跡する方法を使用している．

本手法は，図4に示すように図形の輪郭をとるための輪郭線抽出部分，輪郭線内部を塗りつぶす部分，およびラベルを統一する部分（テーブル解析部分）に分けられる．以下の各部の詳細を説明する．

（I） 輪郭線抽出

最初に輪郭線追跡のための追跡開始点を見つけ，続いて順に連結領域の輪郭を追跡し，追跡した点には追跡済みマークを付けながら処理を進め，一巡した時点で一本の輪郭線とする．この手順を繰り返し用いることで，画像中すべての連結領域の輪郭線を抽出する（図4参照）．以下にアルゴリズムを示す（図5参照）．

① 画面左上画素を，追跡開始点を探すためのラスタ走査初期点とする．

② ラスタ走査を継続し，まだ追跡済みマークの付いていない1の画素を求め，この1の画素（境界点）に追跡済みのマークを付け，追跡開始点 X_0 とする．このとき，上記の1の画素が画像中になければ，アルゴリズムを終了．

③ 開始点 X_0 を中央にみる8近傍を図6（a）とするとき，番号1の要素から反時計回りに（内側輪郭線のときは番号4から時計回りに）調べ，最初の1の画素（図6（a）では番号1の要素）を X_1 とし，次の境界点とする．このとき1～8までで一つも1の画素がなければ，この点は意味のない雑音要素（孤立点）と見なして再び②へ戻る．

④ X_1 にマーク付けを行い，X_1 を中央にみる8近傍を図6（b）とするとき，図のよ

うに矢印の順に X_2 とし，次の境界点（図では●の位置）とする．

⑤ X_1 から X_2 を手順④を繰り返し，次々に隣接境界点を求める．

このとき，$X_{n+1}=X_1$，$X_n=X_0$ となったならば，X_0，X_1，…，X_nが求める一本の輪郭線となる．

⑥ 別の輪郭線を求めるため再び②へと戻る．

なお，輪郭線のラベルの値は，⑤が終わった時点で1を加算し，次の輪郭線とは区別するようにする．

（II） 輪郭線内部塗りつぶし

この部分では，輪郭線内部の画素に輪郭線のラベルと同じ値を付けていく作業を行う．以下にアルゴリズムを示す（図4（c）参照）．

① 画面左上画素からラスタ走査を行う．

② ラスタ走査を継続．このとき，画像中になにもなければアルゴリズムを終了．

③ 0以外の値をもった画素をみつける．1の画素だったら④へ，その他の値だったら⑤へ行く．

④ 左隣の画素と同じ値を付ける（図面は輪郭線で囲まれているので必ず最初は左隣が輪郭線となる）．再び②へ戻る．

⑤ その画素の値をB，その画素の左隣の画素の値をAとすると，B＞AかつA≠0であれば，その画素は内側輪郭線ということになるので，内側輪郭線を外側輪郭線の値と同じにするため，このBの値をもつ画素をすべてAの値に変換する．これを $\ell[\ ]$ という配列に，$\ell[B]=A$ という形で記憶しておく．また，$\ell\ell[B]=0$，$\ell\ell[\ell[B]]=1$（$=\ell\ell[A]=1$）という形で，Bの値のラベルは存在せず，Aの値のラベルは存在するということを $\ell\ell[\ ]$ という配列に記憶しておく．

（III） テーブル解析

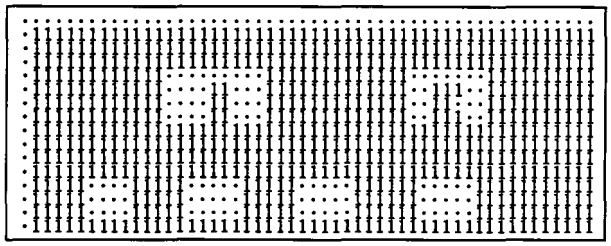
（II）の⑤で，存在しなくなったラベルの値があるので，小さい順に連続したラベルを付けていく作業をする．$\ell\ell[\ell[x]]$ の値が1であったら，xという値のラベルが存在するという事なので，$\ell\ell[\ell[x]]=s$ とする．xの値を，（I）の輪郭線抽出部分で求めた図形の数mまで変化させて繰り返すが，$\ell\ell[\ell[x]]=1$ であるたびにxの値を1ずつ加算していくことで，順序よくラベルが並べ換えられる（図4（d）参照）．

4. 測 定 結 果

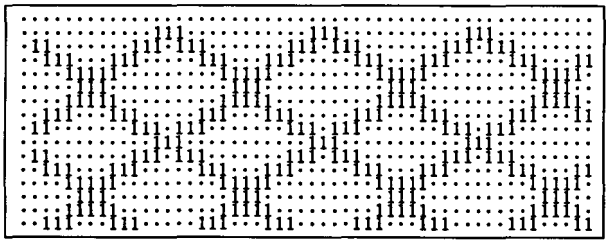
4—1. 測定方法

測定には本手法の他にはラスタ走査型を比較のために使用した．また，測定に用いたデータは，サイズが80×80画素以内の一つの連結領域からなるものを8種類使用した（図7参照）．

また，図形の大きさ，個数および複雑度の三要素を変えたデータを使用した．図形の大きさとの関係を調べるためには，サイズが12×12～77×77画素の正方形を使用した．図形の個数との関係を調べるためには，サイズが4×4画素の正方形を3個から200個まで変化させたものを使用した．図形の複雑度との関係を調べるためには，図形の枝分かれ数を5個から50個まで変化させたものを使用した（図8参照）．

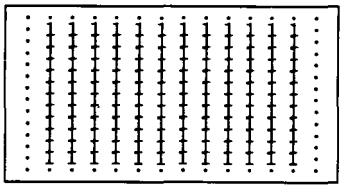


（a）データ番号3

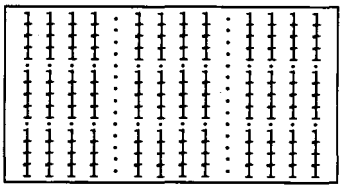


（b）データ番号8

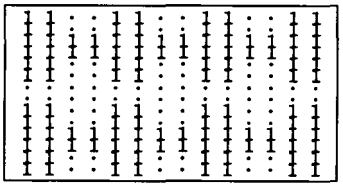
図7 測定用データI（部分）



（a）大きさ用データ



（b）個数用データ



（c）複雑度用データ

図8 測定用データII（部分）

表1 処理時間（sec）

DATA	1	2	3	4	5	6	7	8	平均
従来の方法	3.47	7.47	4.72	5.53	3.78	8.12	4.17	39.85	9.64
本手法	2.87	2.93	2.80	2.93	2.92	2.93	2.88	2.98	2.91

4—2. 測定結果と考察

従来の方法と輪郭線情報による方法の処理速度を比較すると，表1のようになる．従来の方法によると，処理時間はデータにより大きな違いを生じている．データ8については，他のデータに比べて時間がかかっているが，これは，他のデータの図形に比べて1つの連結領域が複雑に回り込んでいる図形であるためと考えられる．しかし，本手法では，データによる処理時間の差がほとんどみられない．

このように一つの連結領域が複雑に回り込んでいるような図形で処理の比較をする．従来の方法では，ある程度までラスタ走査を行い画素を調べていくまでは，一つの連結領域とは判断できず，また，そのラベルを統一するのに何度もラスタ走査を行うので，その結果が処理時間の差となってしまう．

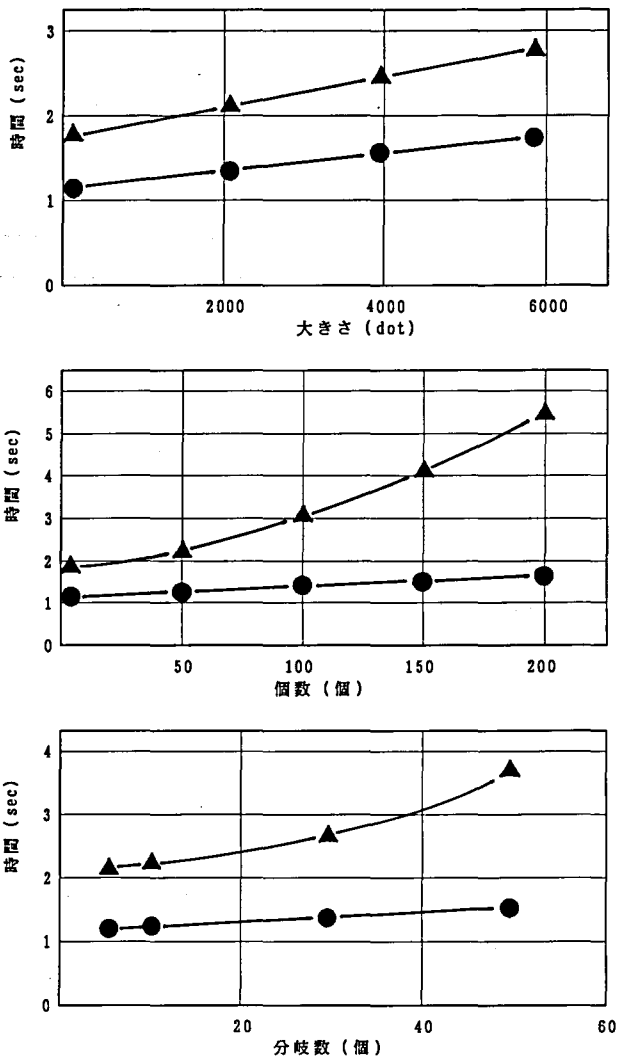
本手法では，最初に輪郭線をとるために複雑な連結領域に対しても一つの連結領域と判断できるので，ラスタ走査は輪郭線をとるために1回と，内部を塗りつぶすための1回の，2

回で処理は終わってしまうため，処理時間が短い．

次に一つの連結領域内に穴があいているような図形を比較する．従来の方法では，1回のラスタ走査で1つのラベルに統一されるが，本手法では，外側輪郭線て内部輪郭線では異なる値のラベルを付けてしまうので，このラベルを統一するために，輪郭線内部を塗りつぶす部分でテーブル（配列）に記憶させておく手間がかかる．しかし，これはラスタ走査に関係なく，また，後で値を置き換えるという簡単な作業であるので，従来の方法と比べた場合処理時間ではほとんど差はない．

また，図形の大きさ，個数および複雑度の三要素と処理時間の結果を図9に示す．

大きさとの関係では，両手法とも大きさに比例する特性を示している．時間差も大きくないのは使用したデータが単純な正方形であるため，ラスタ走査形においても仮ラベルの発生が生じないからであると考えられる．



▲：従来の手法　●：本手法

図9　測定結果

個数との関係においては，ラスタ走査形がデータの個数が75個を越えるあたりから急激に処理時間が増加するのに対して，本手法は個数に比例はするがその増加の割合はわずかな増加量にとどまっている．

図形の複雑度との関係では，ラスタ走査形が緩やかな2次増加量を示す．また，本手法は比例関係にはあるが，その増加量はわずかである．

5. むすび

本論文では，輪郭線情報を利用し処理速度の向上が得られた方法について述べた．ラベリング処理の問題点は，仮ラベル付けの方法とラベルの隣接関係の記憶との二つであると思われる．

前者については，従来のラスタ走査形ラベリング法は，処理対象の図形の形状が複雑にな

るほど仮ラベルの発生が増えるために，その処理時間も増加するのが問題であった．この仮ラベルの発生が図形の複雑度，特に凸部に関係していることが判明したので，この凸部で発生する仮ラベルを図形の輪郭線を求めることにより減らすことができることに着目して本手法を開発した．

従来の方法ではラスタ走査を行い，個々の画素に対して局所的に処理するので図形全体としての連結情報が把握できなかった．しかし，本手法では，輪郭線をとることで，個々の連結領域のおおよその大きさや情報連結領域の概数を得ることができるために，図形全体としての連結情報が把握でき，連結領域レベルの大局的な処理が可能となる．

以上のような理由から，ラスタ走査形ラベリングよりも本手法が仮ラベルの発生が少なく処理速度も速い結果になっている．

また，隣接関係の記憶問題については，間接的に隣接しているラベルをどのように記憶しておくかという点があげられる．今後は輪郭線抽出法とテーブル解析法を吟味し，より高速なラベリングができるようにする．また，カラー画像への応用も検討していく予定である．

謝　辞

本研究を行うにあたり，ご協力頂いた本校卒業生の丸山貴裕君（現在松下通信工業）と松沢孝昭君（現在アルプス電気）に感謝いたします．

参 考 文 献

(1) A. Rosenfeld：“Digital Picture Processing”. Academic Press, New York (1976).
(2) 長谷川，興水，中山，横井：“画像処理の基本技法”，技術評論社（1986）．