コンピュータビジョン　6 4 − 1 2
(1990. 1. 26)

セル構造による線図形の骨格線抽出方法

鈴 木 俊 博　　森　俊 二
リコー 中央研究所

あらまし

線図形の骨格線を抽出する方法として細線化や芯線化が知られているが、いずれの方法も観測単位の設定が十分でないためにノイズに弱く、分岐や交差付近の正しい骨格線を得ることが困難であった。本稿では、セルと呼ぶ観測単位を導入した新しい骨格線抽出方法を提案する。まず、推定線幅に従って画像を小矩形(メッシュ)に等分割する。つぎにメッシュを統合してセルを求める。1つのセルは丁度1つの黒連結成分をもち、線端･連結･屈曲･分岐･交差のいずれかを表すと見なすことができるため、各セルパターンを認識して対応する骨格線(部分グラフ)を求めることで全体の骨格線が求められる。実験により分岐や交差付近の骨格線をかなり正しく抽出できることと、つぶれやかすれをある程度吸収できることを確認した。

Skeleton Extraction Method Based on the Cell Structure

Toshihiro Suzuki　Shunji Mori
Research and Deveropment Center, RICOH Co.,LTD.

Abstract

In this paper, we propose a skelton extraction method based on the cell structure.For the purpose of the skelton extraction, many thinning algorithms and many vectorizing algorithms have been proposed. However, they cannot extract a skelton precisely particulally at a corner, a branch, and a cross, because their observation units are not suitable for the purpose. We propose the cell as the observation unit. At first, image data are devided into small meshes. Next, meshes are integrated into cells. A cell has only one black region which is seen to be one of the following: an end, a concatenation, a corner, a branch, and a cross. Thus we extract a skelton of the image by recognizing each cell pattern and making partial skeltons each of which represents the cell pattern.

1. はじめに

近年、情報処理技術や電気通信技術の発達に伴い、機械を用いた事務作業の効率化が重要となっている。なかでも、文字･図面･表･グラフなどの線図形に対する所望の処理を、計算機で効率よく行うことへの要求は非常に大きい。例えば、文字や図面の認識に加え、線図形の編集、蓄積･検索、伝送などがそれである。こうした要求に応えるためには、イメージデータを計算機で処理しやすい形に抽象化することが必要となる。

線図形を抽象化するうえで最も汎用的な方法は骨格線を抽出することである。骨格線を抽出するための従来方法は、細線化と芯線化の二つに大別できる[1]。

細線化[2][3][14]は、連結性を損なわないように境界画素を少しづつ消去していく方法である。この方法には、どのような線幅や解像度であっても適用できるという利点がある反面、処理量が多いうえ、骨格線の歪みやヒゲの出現といった欠点がある。(図1(a)参照)

一方の芯線化[4][5]は、線の境界をベクトル化したのち、向かい合うベクトルペアの間に芯線(骨格線)を作るという方法である。この方法には、処理量が比較的少ないという利点がある反面、ベクトルペアを決定するためのパラメータを線幅や解像度に適応して自動設定することは困難であるうえ、分岐･交差や短いストロークに対する骨格線が得られないという欠点がある。(図1(b)参照)

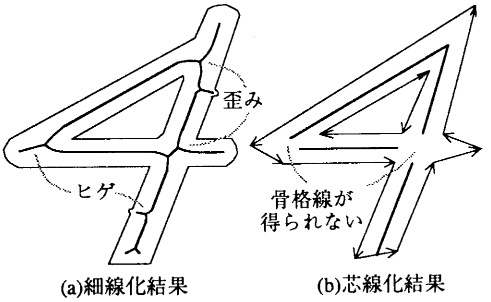


図1　骨格線抽出のための従来技術

こうした欠点の理由として、幅をもつ線を捉えるための単位として画素近傍やベクトルでは不十分なこと、及びその影響を十分に吸収できるような上位プロセスをもたないことが挙げられる。

いずれにしても、骨格線を正確に抽出するためには従来とは異なるアプローチが望まれる。本稿では、セルと呼ぶ観測単位を導入した新しい骨格線抽出方法を提案する。

以下、2章ではセル構造の基本的な意味を述べ、3章では具体的な骨格線抽出方法について詳述する。4章では文字を対象とした実験について述べ、5章ではこれらをまとめて結論とする。

2. セル構造の基本的な意味

パターン情報処理の分野で非常に重要な概念の一つとして、等価情報源[6]と呼ばれるものがある。これを簡単に言えば、情報源の観測の方法は一意ではなく、観測単位の設定の仕方によって情報源アルファベットは異なるという概念である。線図形、多値画像、カラー画像、音声を問わず、パターン情報を処理する場合には、処理目的に応じた観測単位の設定が必要である。

観測単位の設定と表現の方法に関する理論は存在しないが、観測される情報源シンボルの種類が少なく、組み合せが容易であることが望ましいのは言うまでもない。一般的には、観測単位を大きく設定するほど局所的でない観測が可能となる一方、情報源シンボルの種類が増し、その組み合せは複雑になると言える。

線図形の骨格線を抽出するアプローチを、等価情報源という観点から考えてみよう。画素(または画素近傍)を観測単位としたものとして前章で述べた細線化があるが、これが局所的であるが故に欠点をもつことに関しては先述の通りである。

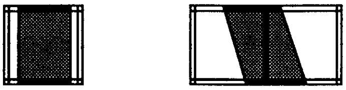
ランを観測単位としたものとして、Pavlidisらによる方法[9][10]があるが、その処理目的は文字認識のための一つの特徴量を抽出することにあり、グラフ表現の精密さという点では不十分である。

ある程度の大きさの面を観測単位としたものに、Moriらの方法[11]がある。これは射影に基づきイメージデータを適応的に非固定サイズの矩形に分割する方法であるが、図面認識における切り出しという意味合いが強く、実際には骨格線は抽出されない。

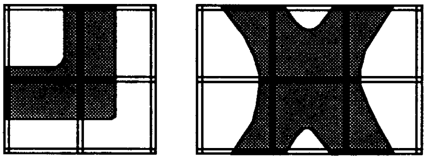
幅のある線を正しく捉えるという観点からは、面を観測単位とすることが望ましいにもかかわらずそれが困難であった原因は、情報源シンボル数の爆発を抑えるための拘束条件を見い出せなかったためだと言える。

本稿では、固定サイズ(線幅よりやや大きめ)の小矩形を適応的に統合してセルを求めることで、この問題の解決を図った。セルは、図2(a)~(d)に示すように黒領域を一つだけもち、広い意味での情報源シンボルは、線端･連結･屈曲･分岐･交差などのわずか数種類となる。また、セル境界の黒ランであるリンクが丁度二つのセルに共有されることを利用してリンクの中点にノードを作れば、線図形のグラフを求めるという問題は、セルパターンを認識してノードを正しく接続するという比較的簡単な問題に置き換えて考えることができる。簡単な例で言うならば、ノードが2個のときは直接つなぐかノードを追加してそこで曲がるようにすべきかを認識する、ノードが3個のときはY型かT型かV型かを認識する、といった問題として考えることができる。

セル構造の概念図を図3(a)に示す。このように、イメージデータを、セルの特徴(構造的、統計的)とセル間の関係とに分けて記述し、近傍セル間での協調処理を行うことで、局所的な柔軟性(すなわち、つぶれ/かすれへの対処や必要に



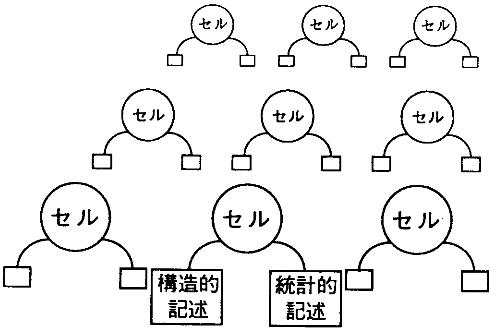
(a)メッシュ数1の例　(b)メッシュ数2の例



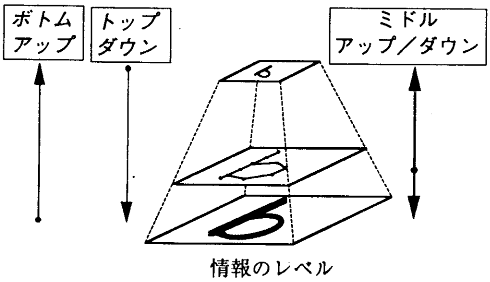
(c)メッシュ数4の例　(d)メッシュ数6の例

図2　セルの例

応じた厳密な特徴抽出)と、大局的に見た正確性との効率のよい両立が期待できる。同図(b)に示すように、従来方法が、ボトムアップアプローチとトップダウンアプローチのいずれか一方(またはその併用)であったのに対し、セル構造を用いたアプローチは、ミドル·アップ/ダウンアプローチであると言える。



(a)セルの特徴とセル間の関係の記述



情報のレベル

((b)ミドル・アップ／ダウンアプローチ

図3　セル構造の意味

3.セル構造を導入した骨格線抽出処理

この章では、2章で述べたセル構造を導入して骨格線を抽出する方法について詳述する。

3.1骨格線抽出処理の概要

骨格線抽出処理の流れを図4に示す。各用語の意味は以下の通りである。標本化走査線とは、イメージデータを観測する走査線(縦、横方向)を一定間隔で網目状に標本化したものである。ランとは、各標本化走査線上の黒画素の連結成分である。メッシュとは、標本化走査線によって囲まれる同一サイズの小矩形領域である。セルとは、角点に黒画素が現れないように一つ以上のメッシュを統合した領域であり、内部に黒領域を一つだけもつ。リンクとは、セルの境界上のランであり、丁度二つのセルに共有される。

図4の流れを簡単に説明すると、ステップ1で求めたメッシュサイズに従ってイメージデータを等分割し、ステップ2でメッシュを統合してセルを求め、ステップ3で、セル毎にセルパターンを認識して対応する部分骨格線(部分グラフ)を求め、ステップ4でセル特徴を大局化することによりグラフの修正を行う、という流れになる。

次節以降では、ステップ1から3についての具体的なアルゴリズムを詳述する。ステップ4は4.2節で述べる。

3.2前処理

この節では、図4のステップ1について詳述する。対象は、線図形であれば二値画像でも多値画像でもよい。但し以下の議論で黒画素といった場合には、多値画像であればしきい値以上の画素を意味する。

3.2.1境界点傾き方向の揺らぎ平滑化処理

まず、境界点となる黒画素を求めたのち、線幅を推定するためにその傾き方向の揺らぎを平滑化する。平滑化の対象は画像平面上における傾き方向のみであり、一般に傾きの大きさと呼ばれる濃淡軸次元に関しては言及しない。

傾き方向の揺らぎを平滑化する方法は現在のところ提案されていない。画像データ全体に対する平滑化処理を流用することは可能であるが、処理量が多いわりに傾き方向の揺らぎの平滑化としての効果は保証されない。

ここでは、簡単な処理で効果の高い平滑化方法を提案する。この処理は、任意の境界点を追跡開始点とし、境界点を追跡しながら行う。境界点列の構造として、追跡開始点と追跡終了点とが隣り合うサイクリックな構造を想定しているが、簡単のため一次元で表記することとし、剰余を用いた記述は行わない。傾き方向の表記についても同様とする。まず、p_i,ang(p_i),dang(p_i)を次のように定める。

p_i:　開始点からi番目の境界点

ang(p_i):　p_iの画像平面上での傾き方向の角度表現

dang(p_i):　先頭の境界点のang(p_i)と後ろの境界点のang(p_{i-1})との差分。角度の差分であるため2通りの解が考えられるが、スキャナの解像度が十分に高ければ、全てのp_iに対して式(3.1)が成立すると仮定しても問題はな

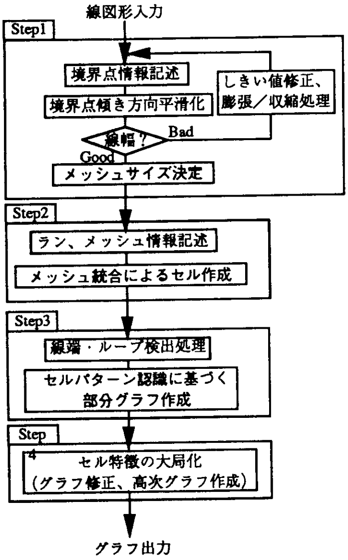


図4　骨格線抽出処理の流れ

いので、絶対値の小さい方とする。(図5参照)

$$-\pi < dang(pi) < \pi \quad (3.1)$$

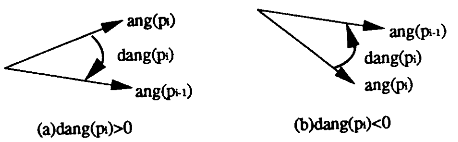


図5　dang(pi)の説明

傾き方向の連続性に関し、p_iとp_{i-1}, p_{i+1}との関係について考える。これら3点が理想的な直線または曲線の境界上にあるとき、式(3.2)にが成立する。非平滑化対象点の例を図6に示す。

$$(dang(p_i) \leqq 0 \cap dang(p_{i+1}) \leqq 0) \cup (dang(p_i) \geqq 0 \cap dang(p_{i+1}) \geqq 0) \quad (3.2)$$

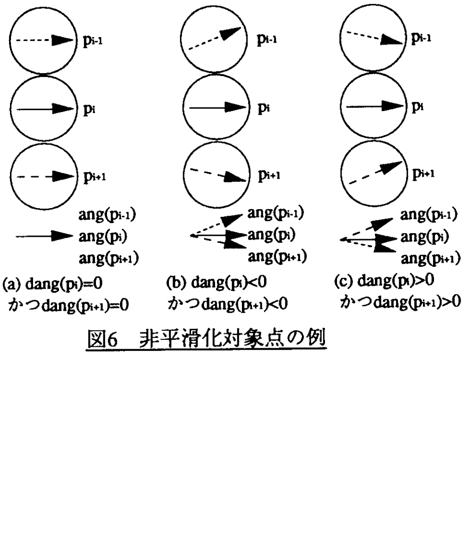


図6　非平滑化対象点の例

一方、式(3.2)が成立しない場合にはang(p_i)の修正が必要であると見なし、p_iを平滑化対象点とする。平滑化対象点の例を図7に示す。

平滑化対象点列PS_jとは、境界点p_iのうち平滑化対象点とされたものの連なりのj番目を意味し、式(3.3)で表される。(図8参照)

$$PS_j = \{p_{j_s}, p_{j_s}+1, \ldots, p_{j_e} \mid p_k(k=j_s, \ldots j_e)\text{は平滑化対象点}\} \quad (3.3)$$

PS_jを、その近傍での凹凸に従って次のように平滑化する。

(1)dang(p_{j_s}-1)とdang(p_{j_e}+2)の符号が異ならない場合

この場合、PS_jは凹または凸の高々一つを表現すればよい。PS_jに含まれる境界点の傾き方向が取るべき範囲R(PS_j)は、ang(p_{j_s}-1)とang(p_{j_e}+1)との間である。但しこれも2通りあるので、PS_j近傍の凹凸が保たれる方とする。凹凸に注意して、求めた傾き範囲R(PS_j)を(PS_jに含まれる点の数+1)個に等分割するように、傾き方向ang(p_k) ($k=j_s, \ldots j_e$)を修正することで平滑化を行う。

なお、R(PS_j)の大きさが十分小さい場合には直線と見なし、ang(p_k) ($k=j_s, \ldots j_e$)を全てR(PS_j)を2分割する方向としてもよい。

(2)dang(p_{j_s}-1)とdang(p_{j_e}+2)の符号が異なる場合

これが起こることは稀であるが、この場合はPS_jは凹凸を少なくとも一つづつ表現しなければならないため、完全な平滑化は困難である。そこで、ang(p_k) ($k=j_s, \ldots j_e$)に対し、式(3.4)の平滑化を行う。これは、ang(p_{i-1})とang(p_{i+1})との間を2分割することを意味する。

$$ang(p_k) = ang(p_{k-1}) + (dang(p_k) + dang(p_{k+1}))/2 \quad (3.4)$$

以上述べてきた処理により境界点の傾き方向を平滑化することができる。なお、処理量を軽減する場合には平滑化対象点列を求めずに、式(3.4)の平滑化のみとすればよく、平滑化の効果を高める場合には、何度か繰り返せばよい。ここで述べた平滑化処理は、処理対象となるのが全画素ではなく境界点のみであることにより処理量が少なく、平滑化対象点列がなくなるように平滑化しているため、平滑化の効果が高いという特徴をもつ。

3.2.2 線幅の推定に基づくメッシュサイズ決定

続いてメッシュサイズを決めるため、境界点の傾き方向を利用して線幅を推定する。ここで、以下の議論のために対向境界点について述べる。

対向境界点

図9(a)に示すように、注目境界点p_iからp_iの傾き方向ang(p_i)に黒画素を追跡し、最初に現れる境界点qをp_iに対する対向境界点と呼ぶ。但し、同図(b)に示すように境界点追跡を8連結で行った場合は境界点に出会う前に非境界点(白画素)に出会うことがあり、このときはこの白画素に隣接する境界点から適当なものを一つ選ぶ。

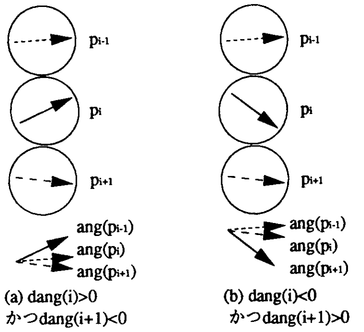


図7 平滑化対象点の例

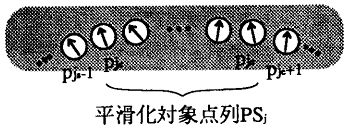


図8 平滑化対象点列

全ての境界点から対向境界点までの距離のヒストグラムを求め、求めたヒストグラムのピークを与えるところを推定線幅wとする。

ここで、推定線幅が大きすぎる(小さすぎる)場合は、つぶれ(かすれ)の可能性があるので、二値化しきい値の修正や、画像全体に膨張/収縮処理を施すことで対処する。

メッシュの一辺の長さTは推定線幅wよりやや大きくする。目安として推定線幅wの1.5倍程度とする。従って、メッシュが線分の内部にすっぽりと入ることは殆どない。

上述の処理は、ランレングスの統計から線幅を推定する方法に比べ、処理量は多くなるが、斜め方向の線が多い場合でも正しく線幅を推定できるため、メッシュサイズを適切に設定することができる。

3.3メッシュ統合によるセル作成処理

メッシュサイズが決定されたならば、イメージデータをメッシュに分割し、ランとメッシュに関する情報を記述する。メッシュの記述は、メッシュ内に黒領域が存在する場合のみ行えばよいが、領域数が2以上の場合は、領域の数だけメッシュを多重化する。

続いて、メッシュののりしろの役目をもつランに着目

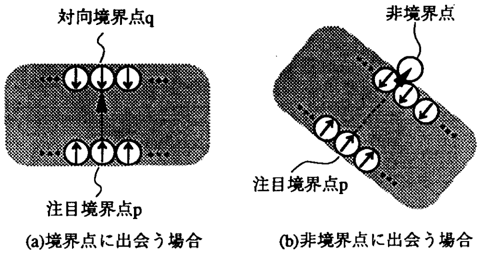


図9 対向境界点の説明

してメッシュを統合し、セルを求める。ランがセル境界としてふさわしいかどうか、すなわち、ランrが線分を横切っていると見なせるかどうかを評価する論理関数eval(r)を、次のように定める。

eval(r)=TRUE if

$len(r) \leqq T_1 \cap ((|ang_1(r) - ang_2(r)| \simeq \pi) \cup$

$(ang_1(r)$と$ang_2(r)$がrに対して同じ側を向かない$))$ (3.5)

eval(r)=FALSE if

$len(r) > T_1 \cup (\neg(|ang_1(r) - ang_2(r)| \simeq \pi) \cap$

$(ang_1(r)$と$ang_2(r)$がrに対して同じ側を向く$))$ (3.6)

ここに、len(r)はランrの長さ、$ang_1(r), ang_2(r)$はランrの両端の境界点の傾き方向、T_1はメッシュの一辺の長さTを最小値として変動するしきい値である。

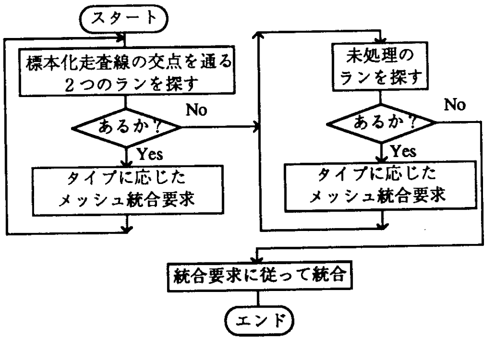


図10 メッシュ統合処理の流れ

メッシュ統合処理の流れを図10に示す。まず、標本化走査線の交点画素、すなわちメッシュの角点が黒画素であるものを取り出し、この黒画素を共有する横･縦方向の二つのラン$r_x \cdot r_y$の組合せのタイプを求める。ランの組み合せは次の三つのタイプに分けられる。

(a) $eval(r_x) \cap (\neg eval(r_y) \cup (len(r_x) \leqq len(r_y)))$

(b) $eval(r_y) \cap (\neg eval(r_x) \cup (len(r_x) > len(r_y)))$

(c) (a), (b)以外

各タイプ(a)から(c)に対応して、図11(a)から(c)の統合要求を出す。

次に、まだ調べられていないラン(二つのメッシュに共有されているラン)を取り出し、ラン評価関数が偽であればこのランを共有する二つのメッシュを統合する要求を出す。(図11(d),(e)参照)

最後に、全ての統合要求を満たす統合を行ってセルを求めるとともに、セル境界として残ったラン(リンク)に対し、その中点の座標をもつノードを作成する。

なお、次のような好ましくないランに対しては、殆どの場合にラン評価関数が偽となるため、セル境界として残ることはない。

- 線分の方向に一致する長いラン
- 線分境界のギザギザ状のノイズによってできるラン
- 分岐内部を横切るラン

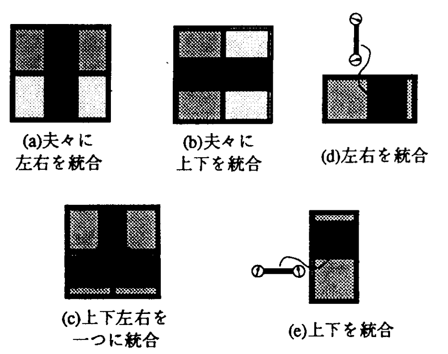


図11　メッシュ統合要求のタイプ

3.4セルパターン認識に基づく部分グラフ作成処理

この節では、セル毎に部分グラフを求める処理(図3のステップ3)について詳述する。

3.4.1線端/ループ検出に基づくセル分割処理

3.3節の処理で求められたセルは、内部に短いストロークやループ(孔領域)をもつ可能性があるため、これを検出してセルを分割することが必要となる。(図12参照)

(1)線端検出処理

まず、線端候補検出処理について説明する。はじめに、各境界点のラベルをクリアし、カウンタに初期値を設定する。続いて、境界点列の開始点からラベリングを始め、注目境界点のラベルをその時点でのカウンタの値で更新する。このとき対向境界点のラベルが注目境界点のラベルと等しければ、線端を過ぎて折り返したものと見なせるので、対向境界点から注目境界点までを線端候補とし、次の候補に備えてカウンタの値を更新する。

上記の線端候補検出処理を、追跡方向を時計回り/反時計回りとしたそれぞれの場合について行い、いずれの追跡方向でも線端候補となった境界点列を線端として検出する。

この処理にはしきい値は不要であるが、処理量を軽減する場合には適当なしきい値を定めたうえでラベリングを一方向のみとし、「線端候補が見つかった時点で、注目境界点の傾き方向と対向境界点の傾き方向との差分の大きさがπに十分近ければ線端としてカウンタの値を更新する」とすることもできる。

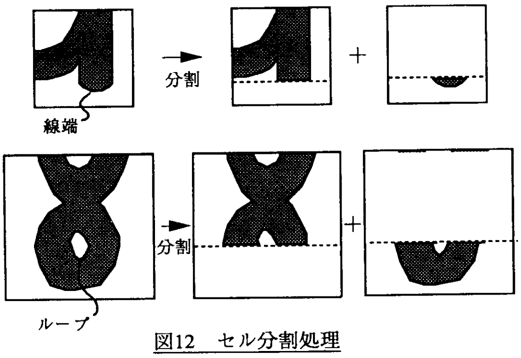


図12　セル分割処理

(2)ループ検出処理

セルにつながるリンクの数よりもセル内の境界点列の数が多ければ、孔領域があることが分かる。そこで、セル境界に現れない境界点列をループと判定する。

線端やループが検出されたならばセルを分割し、新たなセル境界に関するノードを作成する。

3.4.2ノード数に応じた部分グラフ作成処理

あらかじめ、ノードと境界点列を互いに参照可能にしておく。3.4.1で述べた処理によってセル内に孔領域も突起も存在しないことが保証されるため、図14に示すようにノード数に応じて設定した屈曲や分岐のモデルを用いて、セルパターン認識をトップダウン的に行うことができる。

それぞれの処理の説明に先立ち、凹度と最遠点について説明する。凹度は境界点列の凹みの程度を表す特徴量であり、値が正で大きい程凹みが大きく、値が負で絶対値が大きいほどでっぱりが大きいことを表す。

凹度の求め方(図13参照)

境界点列B_iに対する凹度$\kappa(B_i)$は、B_iの始点と終点をそれぞれ$p_s(B_i)$,$p_e(B_i)$とし、$p_s(B_i)$と$p_e(B_i)$を通る直線から最も遠い点(これを最遠点と呼ぶ)を$p_{far}(B_i)$とするときの、ベクトル$p_s(B_i)p_{far}(B_i)$とベクトル$p_{far}(B_i)p_e(B_i)$とのなす角度として求める。

この他にも、境界点の傾き方向が境界点列の方向にほぼ直交することを利用し、境界点列iに含まれる境界点の傾き方向の範囲の大きさに凹凸を考慮した符号をつけて凹度とすることも考えられる。

続いて、ノード数に応じた処理内容を説明する。(図14参照)

(1)ノード数が0の場合

セルに属する黒連結成分の重心座標をもつノードを一つ追加するが、アークは作らない。

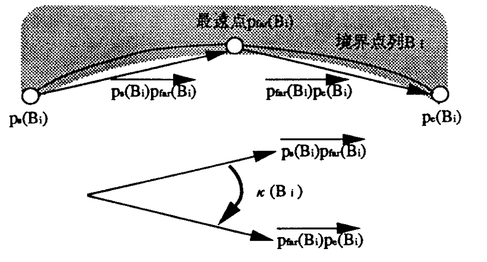


図13　凹度と最遠点

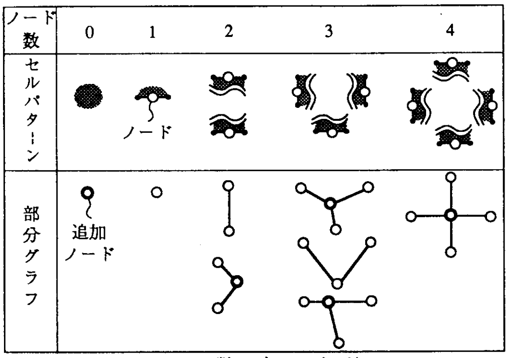


図14　ノード数に応じた処理

(2)ノード数が1の場合

端点を表すノードは既にあるので、何もしない。

(3)ノード数が2の場合

2つのノードをそれぞれn_1,n_2、2つの境界点列をb_1,b_2とする。途中に連結点が必要かどうかを判断するため、b_1とb_2に対する凹度を求める。求めた凹度の絶対値が十分大きい場合、すなわち、しきい値Th_1に対して、式(3.7)が成立するときは途中に連結点が必要であると見なし、2つの最遠点$p_{far}(b1)$,$p_{far}(b2)$を結んだ線分の中点の座標をもつ連結点ノードを追加して、このノードとn_1,n_2それぞれとの接続を表すアークを作る。

$$\max(|\kappa(b_1)|,|\kappa(b_2)|) > Th_1 \qquad (3.7)$$

式(3.7)が成立しないときは連結点は不要と見なし、n_1とn_2との直結を表すアークを作る。

(4)ノード数が3の場合

3つのノードをそれぞれn_1,n_2,n_3とし、3つの境界点列を凹度の小さな順にb_1,b_2,b_3とする。$\kappa(b_1)$と$\kappa(b_2)$とから、セルパターンが表す論理的な接続関係が、(a)Y型か、(b)V型か、(c)T型かを認識し部分グラフを求める。

(a) $\kappa(b_1) > Th_2$ (Th_2はしきい値)のとき

最も凹んでいない境界点列が十分に凹んでいることから、Y型の分岐と考えられる。3直線が1点で交わる保証はないので、3つの境界点列の最遠点の重心の座標を求め、これを座標としてもつ分岐点ノードを追加し、このノードと3つのノードn_1,n_2,n_3それぞれとの接続を表すアークを作る。

(b) (a)でなく、$\kappa(b_2) \leqq Th_3$ (Th_3はしきい値)のとき

2番目に凹でない境界点列がほとんど凹でないことから、V型の分岐と考えられる。そこで、b1とb2につながるノードと他の2つのノードとの接続を表す2つのアークを作る。

(c) (a)でも(b)でもないとき

Y型でもV型でもないので、T型の分岐と見なす。まず、b_1につながる2つのノードを結ぶ直線L_1を求める。次に、残りのノード(これをn_0とする)を通る線分の分岐付近の様子を表す直線L_2を求める。L_2の求め方は次の通りである(図15参照)。n_0につながる境界点列はb_2とb_3であることに注意する。b_3に属する境界点pの対向境界点p_{pair}がb_2に属し、かつp_{pair}の対向境界点が再びb_3に属するようなpのうちで、n_0から最も遠いものを求める。このときのpとp_{pair}の中点を通り、pとp_{pair}の傾き方向を2分する方向を傾きとする直線を求めて、これをL_2とする。

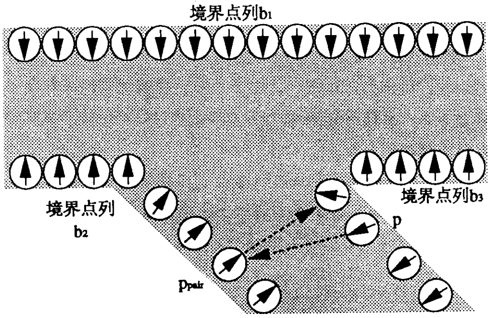


図15　分岐付近の直線の求め方

最後に、L_1とL_2との交点を求め、これを座標としてもつ分岐点ノードを追加し、分岐点ノードと3つのノードn_1, n_2, n_3それぞれとの接続を表す3つのアークを作る。

(5)ノード数が4の場合

内部にX型の交差があると見なせるので、それぞれ2つずつのノードを通りX型に交差するような2つの直線を求める。これら2直線の交点を求め、これを座標としてもつ交差ノードを追加して、4つのノードそれぞれとの接続を表すアークを作る。

(6)ノード数が5以上の場合

このようなセルが出現することは極めて稀であり、セルパターンの認識が困難であるとして、全て境界点列の最遠点の重心を求め、これを座標としてもつ不確定点ノードを追加して、各ノードそれぞれとの接続を表すアークを作る。

なお、上述の(3)から(6)の処理において、方向の信頼性が低くなりがちな短いアークの生成を避けるために次のようにする。追加されるノードをn_{new}とし、n_{new}の座標に最も近い座標をもつ既存のノードをn_{old}とするとn_{new}とn_{old}との距離が十分近ければn_{new}を作らずに、n_{new}の座標をn_{old}に移し、n_{new}に関するアークがn_{old}と接続するよう変更する。

4. 実験結果と課題

4.1 実験結果

我々が提案する骨格線抽出方法の有効性を調べるために手書き数字を対象とした実験を行った。処理内容は、図4のステップ1から3でありステップ4の大局化処理は含まない。

実験結果を図16(1)~(3)に示す。それぞれの画像の品質は、(1)良質、(2)つぶれ、(3)かすれ である。

大局化処理が含まれていないにも拘わらず、観測単位を

大きく設定したことによって、細線化に比べてヒゲが出現することもなく、分岐や交差の付近でも骨格線がかなり正しく抽出されていることが判る。

また、つぶれやかすれに対しては、前処理レベルで動的に膨張/収縮処理が施されているため、細線化との差異が顕著に現れていると言える。

4.2今後の課題

本手法は、セルパターン認識をトップダウン的に行っているため例外には弱い。すなわち、線端が検出されない場合やノード数が5以上になった場合の対処が十分ではないという問題がある。

しかし、今回はセルの特徴として構造的な特徴のみしか用いておらず、統計的な特徴と大局化処理(セル間の協調処理)によって対処できる問題であると考えられる。

5. おわりに

セル構造を導入した新しい骨格線抽出方法について述べた。幅のある線を捉えるのに十分な大きさをもつ観測単位であるセルを導入したことより、従来方法に比べて、屈曲·分岐·交差の骨格線を正しく得ることができた。また、つぶれやかすれに関しては、前処理レベルでかなりの対処が可能であることを示した。

今後は、統計的なセル特徴を用いた大局化処理を行うことで、例外に対する強さと大局的な正確さとを追究していく予定である。

参考文献

[1]中嶋、安居院、吉井、今井:「地図·図面入力の技術動向」、信学技報、IE88-5(1988)

[2]田村:「図形の細線化についての比較研究」、情処学会IP研資1、pp.1-12(1975)

[3]田村:「細線化についての諸考察」、信学技報、PRL75-66(1975)

[4]岡田:「輪郭追跡による二値図形の細線化法の検討」、信学技報、IE78-74(1978)

[5]大沢、坂内:「多次元データ構造を用いた図面処理ー図面のベクトル化ー、信学論(D)、No.4、pp.845-852(1985)

[6]坂井:「情報基礎学ー通信と処理の基礎工学」、コロナ社(1982)

[7]坂井:「情報基礎学詳説」、コロナ社(1983)

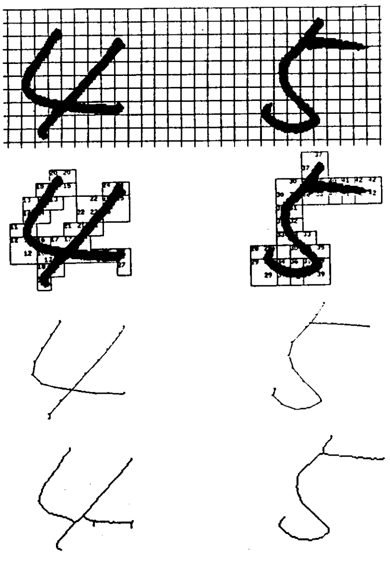
[8]坂井:「情報基礎学演習編」、コロナ社(1986)

[9]T Pavlidis: 「A Vectorizer and Feature Extractor for Document Recognition 」、CVGIP 、Vol.35 、pp.111-127(1986)

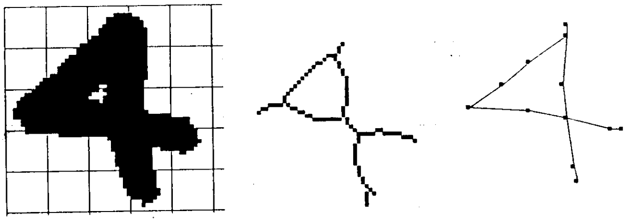
[10]S Kahan, T Pavlidis: 「On the Recognition of Printed Characters of Any Font and Size」、IEEE Trans on PAMI、Vol.PAMI-9、No.2(1987)

[11]S. Mori, M. Maeda, H. Kinbara, T.Kunieda, S. Deguchi:「A Handwritten Mechanical Drawing Recognition System」、IEEE MIV-89、pp.270-275(1989)

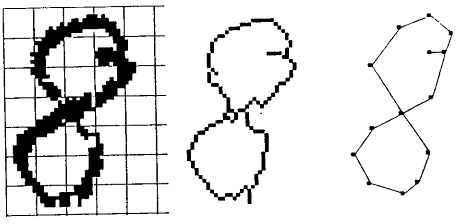
[12]森:「画像認識の基礎[I]」、オーム社(1986)



(a)良質画像
(上から原画像、セル、抽出された骨格線、細線化結果)



(b)つぶれ画像
(左から原画像、細線化結果、抽出された骨格線)



(c)かすれ画像
(左から原画像、細線化結果、抽出された骨格線)

図16 実験結果