

# 図形の細線化についての比較研究

A SURVEY AND COMPARATIVE STUDY OF LINE THINNING SCHEMES

田村 秀行

HIDEYUKI TAMURA

電子技術総合研究所

ELECTROTECHNICAL LABORATORY

## §1. はじめに

今回から、情報処理学会イメージ・プロセッシング研究会が開かれることになった。それだけこの分野の研究人口も増え、その応用分野が社会的にも期待されているということだろう。確かにデジタル画像処理や画像パターン認識関係の論文の数も年々増加しているおり、様々な手法や成果が発表されている。他方、計算機によるパターン認識は期待された割に実用化に到っているのはほんの僅かである。

一口に画像処理といっても、その対象は実に多様であり、ましてや認識レベルに近いところでは、その目的によって処理手順もかなり異なる。決定的な方法が見い出せばいいが、既存の技術を整理してその長所、短所、適用限界などを明確にすることが、この分野の着実な発展につながると思われる。デジタル画像の場合、その標本化や量子化による誤差とオペレータのデジタル近似のバリエーションによって、処理結果に大きな影響を与えている。数多く発表されている方法の中には、こういった僅かな違いのものも多い。どうも自らの手法のオリジナリティを強調する方が（ホントは大して目新しくなくても）論文になりやすいせいか、既存の方法の有効性が十分に検討されていない傾向がある。特に、smoothing, thinning, thresholding, edge detection等の基礎的な前処理の方法はかなり研究されており、これらと有効に用いるだけでかなりの画像処理が行はえる対象もある。

このような観点から、細線化の諸方法の比較検討を行なった。これは以前ある画像を処理している際、細線化の必要性に直面してどの方法を使ったらいいのか戸惑ったからである。（結局比較は間に合わず、自分で考えたものを使った）本報告はこの経験に基づくものである。少しでも、これから同じ問題に取り組む方への参考になればと考えている。

## §2. 細線化と骨格化の意義

細線化 (Thinning) について考える前に、骨格化 (Skeletonization) について触れる必要がある。これらは手法的にも共通点が多く、用語的にもしばしば混同されている。

### 2.1 骨格化の概念と問題点

骨格化は図形の global な形の記述法として Blum[1] によって提唱されたものである。色々なモデルが考えられ改良されてきたが、要点は図形の境界より一せいに同じ速さの波を伝播させ、その波頭のぶつかり合う点 (quench point)\* の集合 (軌跡) を求める操作である。このような点の集合を skelton、この操作を

skelton transformation 又は skeltonization と呼ぶ。またこれが図形の対称軸と行るところから、それぞれ medial axis (MA), medial axis transformation (MAT)とも呼ばれる。

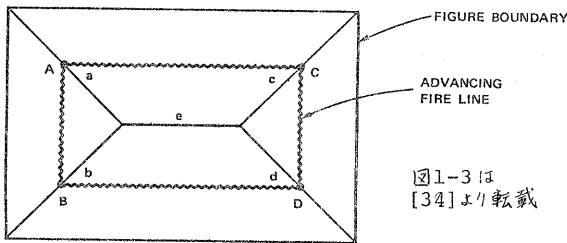


図1-3は [34]より転載

図1. 矩形の skelton

[\*] 波の伝播モデルの他に、図形を草原とみられてその周囲から火を放つという説明もよく用いられる。この場合、火のぶつかる奥で火が消えると考えるところから、quench 奥と呼ばれる。また MAT は prairie fire transformation, glass fire transformation とも呼ばれる。矩形に対する例を図1に示す。

図形のトポロジカルな特徴をよく表現しており、quench 奥とこの奥の発生までの伝播時間（又は境界からの距離）との対 (skelton pair) からもとの図形が完全に再生できることにある。即ち、skelton は図形のすべての形状の情報を保存しながらデータの reduction をしている。これは境界による表現に比べて、中や分岐の情報をよく残している奥で、図形の符号化・記述の方法として注目された。

実際にMAを求めるには、図形の境界からの最短距離を計算して、境界の二箇所以上から等距離にある奥と quench 奥とするので、MAT は距離 (metric) の選び方に依存する。図形をアプログ平面で考える場合はユークリッド距離を用いれば良いが、デジタル図形に対しては距離の取り方と quench 奥の選び方によって、次のような問題が生じた。[31]

- a) Discrete MAT (DMAT) は原図形の再生の保証がない。
- b) 原図形の連結性がMAにおいて保証されない。
- c) 原図形の回転と rescale に対して不変ではない。

代表的なDMATについての比較を表1に示す。これらの問題奥はその目的によって不都合ではないこともある。図形の輪郭が滑らかでなく凹凸が多い場合には、MAは小さな枝（いわゆる‘ヒゲ’）が多く発生する。これは原図形の再生と必要としない場合（パターン認識の前処理として用いられる場合にはほとんど必要はない）には、形状の記述としてはこのような枝をカットした方が、人間の視覚的にも自然である。この対策として、比較的ゆるやかな角からの skelton、即ち quench 奥の伝播速度の小さな部分を無視する方法がある。図3はこの例である [36]。デジタル図形でこれを実現するには、やはり角度の測り方に問題がある ([21], [21])。また濃淡図形の二値化の閾値の選び方によって、境界が劇的に変化する図形に対して、grey-weighted skelton [6] の提案がある。

	a) 再現性	b) 連結性の保証	c) 回転の影響
Rosenfeld & Pfaltz [3]	可	無	大
Philbrick [2]	不可	無	大
Hilditch [12]	不可	有	大
Montanari [4]	可	無	小
Arceili et al. [21]	不可	無 (ほぼ連結)	大
Levi & Montanari [6]	不可	無	小

表1. Discrete Medial Axis Transformation の特性

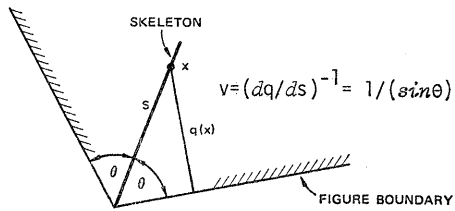


図2. Propagation verocity:v

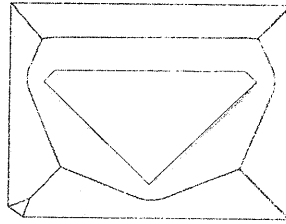


図3(a). Skelton

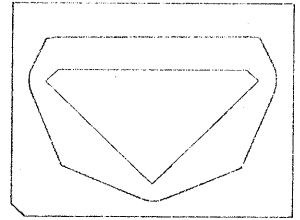


図3(b).  $v < 1.5$

## 2.2 文字認識における細線化処理

MATは、MAがstick figure になる性質により、線中をなくす — 細線化の方法としても注目された。既に文字以外の対象について線の細めの例[36]-[40]もいくつか有り、特に染色体について用いられたが、数々の問題が文字認識の分野の研究と共に明らかにされてきた。これは文字自体が線要素から組み立てられているものであり、線中には意味はないからである。従って、smoothing → thresholding → thinningといった一連の前処理によって線の交点とトポロジカルな特徴と抽出する方法が用いられた。DMATがこのような細線化に適用されたが以下のような莫が問題となった。

- i) 線の端や交差部においてMAにY形の分岐が生じる。(以後Y現象と呼ぶ)
- ii) MAでは線中が必ずしも1にならない。
- iii) 周辺ノイズの影響により多くのヒゲが発生する。
- iv) MAにおいて線のとぎれが目立つ。

ii), iv)は2.1で述べたようにDMATの欠点であり、iii)は対象となる図形による。しかし、i)のY現象については理想的なMATにも生じる結果で、quench莫という距離概念で定義されるもので表現する上での限界である。しかしi)-iv)の問題は、線中を細めて線の追跡や交点の目的には余りにも不都合であった。MATの概念を用いない文字認識の前処理としての方法もいくつか試みられた([7]-[10])が、上記の問題莫に対する満足な解答ではなかったため、細線化は役に立たないといった感じを与えた時期があった。

文字認識における前処理の方法としての価値は、あくまで後の識別の方法に深い関係をもつ。その後、この分野では相関法や領域法のような細線化と必要としは方向に進んだケースも多い。しかし、現在なお細線化と不可欠な前処理として採用している例(一次元マッチング法等)も多い。

## 2.3 細線化の意義と利用価値

Y現象は距離概念によって生じるもので、線の細めという目的には不都合なものである。線とはそもそも巾のないものであり、細線化という言葉自体妙なものである。しかし、我々が実際に扱う線は巾がなければ描くことも眺めることも出来ないもので、現実の線図形で線中は存在する。この不要な情報である線巾を1にするのが細線化である。2.1で述べたように、骨格化は形状の記述のため全境界からの距離を考えている。一方、細線化には巾方向の距離だけを考えれば良いので、線の端から長さ方向に波を発生させる必要はないのである。Y現象は長さ方向にも距離を考えたための結果である。

この意味から、細線化(thinning)と骨格化(skeltonization)と区別して考えるべきである。\* 骨格化のモデルに類して、細線化は巾の方向にだけ波を発生さ

せて quench 臭と考えるモデルで定義できる。ここから、細線化は骨格化の特殊形と考えることができる。Y 現象は一取形のまま MAT を使った結果であり、しかも不完全な DMAT を用いたため i)-iv) の問題臭が生じたのである。これを限界として細線化という概念を低く評価するのは誤りである。

細線化の効用は、(1) データ量の削減、(2) グラフとしての取り扱い易さ、(3) 追跡 (tracking) の容易さ、(4) elongation の尺度 等にある。特に線巾 1 とすることにより、(3) の効用は大きく、長さの測定や追跡による分岐の削除などに利用される。本報告の目的は、デジタル図形の基礎的処理としての細線化の代表的な方法を比較し、利用目的によって適宜使い分けられるようにそれぞれの長所短所を明確にすることである。

[\*] Stefanelli & Rosenfeld [19] によって用語の使い分けが提案されている。距離変換を行なうものを skelton, 周辺臭の連続的削除によるものを medial line としている。[12], [18], [22] などでは skeltonization という言葉を用いているがこれは確かに thinning の方法であり、距離変換は用いてない。しかし、距離変換で細線化と骨格化と分岐のほどであろうか？両者の違いは端臭の定義の仕方の違いである。もし、巾方向にだけ距離変換が可能なら (局所演算では難しいが) やはり細線化というべきである。周辺臭除去の順序と端臭処理の工夫によって explicit には距離変換を行わずに Y 現象を防いでいる細線化法が多い。現象回避のものを細線化法と呼ぶべきであろう。さて、骨格化の結果は skelton であり、細線化の結果は何と呼ぶべきか？ medial line の直訳「中心線」を著者もしばらく用いたが、中心位置に基いた方法もあるのだから、以後の比較の話の中ではどうもピッタリじゃない。細線 (thinned line) というべきとらざるうか？ 相英相方ともこれは余り用いられていない行である。仕方がないので「心線」(core line) を用いることにする。

### §3. 細線化法における検討事項

#### 3.1 細線化法のセマンティックス

Hilditch [12] は心線の持つべき性質を次のようにあげている。

1. 線巾 (Thickness) --- 心線の線巾が 1 となること。
2. 線的位置 (Position) --- 心線はもとの線の中心に位置すること。
3. 連結性 (Connectivity) --- 心線において原パターンの連結性を保存すること。
4. 安定性 (Stability) --- 細線化過程で心線の長さが縮退し続けられないこと。

この中で最も重要視されるのは連結性である。ほとんどの細線化法では、周辺から順次臭と除去して行き、連結性に影響を与えない臭をチェックして保存している。線巾についても削り過ぎないようにチェックされるが、問題となるのは線巾 2 の場合である。線の両方向から同時に削ると線が消滅するので、処理をいくつかのサブサイクルに分けて片側ずつ削るか、中チェックの条件を非対称にすることで対処している。ただしこの偶奇性による非対称が線の中心位置に影響を与えることがある。サブサイクルの順序はこれを考慮して工夫する必要がある。細線化アルゴリズムの停止条件には、残った臭がすべて保存臭としてチェック済みにするか、図形がこれ以上変化しないと、というチェックを用いる。

この 4 条件を細線化と考える上で不可欠な第一級条件とするなら、第二級条件として次の 3 臭と心線の品質評価に用いる。

5. ノイズに対する強さ (noise insensibility) --- 境界上の小さな凹凸によるヒゲと心線に付加しないこと。
  6. 等方性 (isotropy) --- 原パターンの回転によって心線が影響を受けられないこと。
  7. 線交差部の無歪 (linear fidelity) --- 線の交差において心線が歪まないこと。
- 周辺ノイズによるヒゲの対策は、やるやかは角を持つ頂臭をまず削ることである。これは、除去臭チェックのオペレータに埋め込まれる場合が多い。サブサイク

ルの順序と除去条件の非対称がまた、異方性と交差部の歪みの原因となる。特に正方格子上で表現される図形に対しては、水平・垂直方向と両対角方向との異方性は本質的欠点であるが、サブサイクルの順序の工夫と対角方向にも境界点と除去することによりかなり緩和される。交差部の歪みについては、+形交点と除去することによりかなり緩和される。交差部の歪みについては、+形交点と除去することによりかなり緩和される。交差部の歪みについては、+形交点と除去することによりかなり緩和される。交差部の歪みについては、+形交点と除去することによりかなり緩和される。

### 3.2 細線化のシンタックス

前項で示した心線に対する要求をみたすべく、様々な細線化アルゴリズムが考えられているが、これらとインプリメントする上での要求をあげる。ほとんどが計算機プログラムに、OCRなどでは並列演算のハードウェアに組み込まれるが、図形は正方格子上の点として与えられる。三角格子 (hexagonal array) や六角格子 (triangular array) 上の図形に対する細線化も定義されている([15]-[17])がこのような図形を扱うことは稀であり、ここでは比較の対象としない。

さて、2.3で述べた細線化のモデルの実現は、線の中方向に両側から点と削って行くことであるが、前述のように局所操作では線の方向が決定できない。また対象となる図形に色々な方向の線が含まれている場合は結局、すべての境界点と端点と扱わざるを得ない。そこで、境界点のうちまず除去候補点を指定し、この中でY現象防止、連結性、偶奇性等の視差から除去すべきでない点と「一時保存」又は「永久保存」点としその他の点と図形から除去する。この操作を適当な停止条件まで繰り返すといった手続きが考えられる。

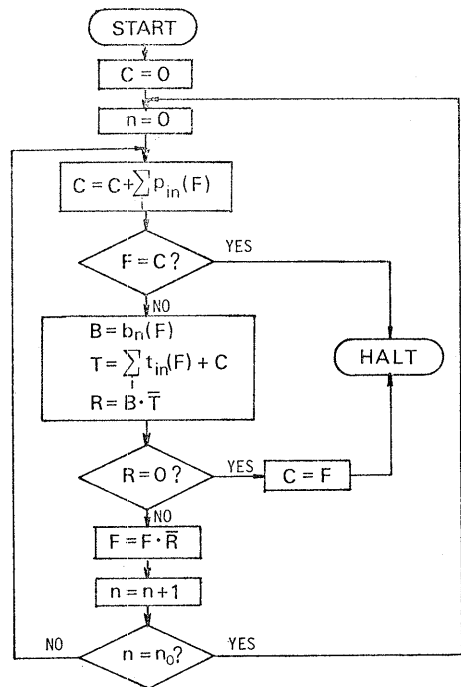
これを細線化アルゴリズムの標準モデルと考え、図4にフローチャートで示した。このモデルにそって細線化の手順とそのインプリメンテーションについて考える。

(1) 処理方式

並列処理と逐次処理があり、一般に、前者はハードウェア化した場合に後者は計算機での演算に向いている。

逐次処理での細線化はその処理点のシーケンスによって、境界線追跡型 (boundary following) とラスタースキャン型に分類できる。前者は別に境界のプログラムの用意と、開始点の指示の必要があり面倒である。後者はそのスキャン方向による非対称が、前述の条件2, 6, 7の不都合の原因となる。細線化では計算機でも擬似並列処理とすることが多い。

並列処理では偶奇性による線中の



F: Figure n<sub>0</sub>: number of subcycles  
 C: Core line {b<sub>n</sub>}: boundary function  
 B: Boundary point {p<sub>in</sub>}: permanent function  
 R: Removal point {t<sub>in</sub>}: temporary function  
 T: Temporary memory

図4. STANDARD MODEL OF CONTOUR EROSION

問題があり、境界線除去とサブサイクルに分けるか、非対称な条件を用いるかである。サブサイクルは上下左右に分ける 4-cycle 型と、それを二つずつ組み合わせた 2-cycle 型がよく用いられる。標準モデルはこの方式と中心に作られている。

2) 除去防止条件

$t_{in}$  と  $p_{in}$  のいずれか一方もしくは両方用いても良い。この条件の中に、Y 現象防止、連結性保存、偶奇性対策、周辺ノイズ除去等がすべて含まれるが、そのほとんどは  $3 \times 3$  のマスクによる近傍線処理で達成される。 $3 \times 3$  のマスクは最もシンプルでよく研究されており、セル・オートマトンで実現する場合や、連結性のチェックにも都合が良い。しかしその局所性のため交差部の歪みなどに対しては無力である。より大きな  $4 \times 4, 5 \times 5$  などを用いて線巾処理(こうするとサブサイクルに分ける必要がない)が行なわれることもあるが、あまり条件を複雑にするとソフトでは計算時間がかかり、ハードではコスト高になる。

(3) 演算時間

ハードの場合、心線を得るまでのステップ数  $n$  に比例する。一般に、4-cycle より 2-cycle が、さらにサブサイクルに分けられない場合は、一度に多くの境界線を処理する(即ち  $n_0 \rightarrow$  小,  $b_n(F)$  の 1 要素  $\rightarrow$  多)ので、少ないステップ数で済む。ソフトの場合には、 $n \rightarrow$  大,  $i \rightarrow$  大, 各  $t_{in}, p_{in}$  で参照する近傍線の数  $\rightarrow$  大に付るにつれ演算時間がかかる。

(4) メモリ量

ソフトの場合、逐次処理では並列処理に比べて 2次元配列が 1 つ少なく済む。並列処理もソフトの場合、2 つ以上の 2次元配列を適当な数に割りつけることにより 1 つの配列(大きな数を入れる場合)で表現できる[3]が、条件処理や計算は面倒になる。ハードと同様に考えると、図 4 において不可欠なのは  $F, C, R$  の 3 つである。さらに  $t_{in}$  と用いなければ  $R$  と、 $p_{in}$  と用いなければ  $C$  と省略できる。

(5) 停止条件

図 4 に見るように 2ヶ所アルゴリズム停止のための判定がある。上からそれぞれ C-halt, R-halt と呼ぼう。フローチャートは一般化のために複雑になっているが、 $F, R, t_{in}$  による実現には R-halt と、 $F, C, p_{in}$  による実現には C-halt ということになる。但しそれぞれ弱線がある。前者では  $t_{in}$  チェックした線が次のステップで削られて、 $R=0$  になるまで心線が縮退することがあるので R-halt だけでは安定性を満足しない場合がある。Rutovitz [11] での例を後で示す。一方、Stefanelli & Rosenfeld [19] では後者を用いているが、図 5 の中心の 5 線は永遠に  $p_{in}$  で得られない。従って、 $R=0$  となっても  $F \neq C$  であるので、このアルゴリズムは永久に停止しない。(停止しなければアルゴリズムとは云えない!!)

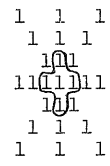


図 5. C-halt の不適例

このため Tamura [27] は、 $F, C, \{p_{in}\}$  に  $R$  を加えて R-halt を採用している。両停止条件が同じ意味を持つ場合が多いが、上のケースで余分にメモリを要しても R-halt を用いねばならない。

細線化方法	処理方式	停止条件	補助フレーン	3x3以外の近傍点を用いるもの	備考		
Sherman [7]	sequential	boundary following	R-halt	R	予め線巾測定		
Fraser [14]				R			
O'Callaghan [20]				R			
Hilditch [12]				R			
Deutsch [9]		raster scanning	線巾検査	?			
Yokoi [26]	parallel	fixed condition	R-halt	C, B(2)	対称性重視、線巾2 マスク多数、高速 〔13〕の条件を反転		
Rutovitz [11]				R		○	
Deutsch [13]			反復回数指定	0		○	
Tojo [23]		2-cycle	R-halt	R			
Yamamoto [26]		2- or 4-cycle	C-halt	C			
Deutsch [17]		4-cycle	R-halt	R, C			
Stefanelli [19]			反復回数指定	T(2)			
Tamura [27]		parallel + sequential	4-cycle + searching	C-halt		C+ D(8)	○
Saraga [8]							
Triendl [18]							
Murthy [22]					平滑化を含む・連結性不適		
					端点、交点より線を引く		

表 2. 細線化アルゴリズムの分類

#### §4. 細線化法の比較・検討

##### 4.1 細線化アルゴリズムの分類

現在までに発表されている主要な細線化法を、§3で考えた事項をもとに表2に分類した。公表されてなくて有効な方法があるかも知れないが、大体の傾向がつかめると思う。ただしこれらは、正方格子上で表現された2値図形についてのものである。最近、濃淡図形を直接扱う方法〔12〕,〔28〕-〔30〕もかなり発表されているが今回は対象外とした。既にYokoi〔29〕によって少し比較が行われているし、どのような場合に濃淡で扱うべきか(後で少し触れるが)まだ明確ではないからである。

一見して分からない項目については、代表的な方法について実際のデータに対して実験を試みた。この種の試みは過去にDeutsch〔33〕とO'Callaghan〔20〕によって行われている。前者は文字認識の前処理としての細線化の4つの方法を比較している。特にsmoothingと組み合わせて使って評価している。この4つの方法は全く異なった発想のものだが、このうちのRutovitz〔11〕が先に述べたモデルの基礎となるものである。Deutsch自身、当時最も良いとした自らの方法〔9〕よりも、後年Rutovitzと改良した方法〔13〕から更に一般化〔16〕〔17〕を行ってある事は注目に値する。後者は、代表的な3つの方法を取り上げて自らの方法と比較しているが、対称性に一般性を欠き、議論も深くはなされていない。今回の実験は、Deutschが志向した系列の方法を取り上げ、O'Callaghanの議論をより一般的に行なったものである。

##### 4.2 比較実験結果

###### 4.2.1 実験対象

比較対象としたのは表3に示す4種(バリエーションを含むと11種)の細線化法である。さらに連結性と考える基本は、交差数(crossing number)と連結数(connecting number)の2つに大別できる。

細線化の対象としては、(1)写真画像に微分、エッジ強調、閾値処理を施した線図形、(2)手書文字、(3)参考文献で使われている図形等、171例の2値図形を用いた。定性的実験には更に、数種の理想パターン及びそれにノイズ添加・回転したものと追加した。これらはすべて72×72の2次元配列として、TOSBAC 5600で計算機処理を行なった。

【\*】Deutsch[13]と並んでStefanelli & Rosenfeld[19]もししばしば引用され、O'Callaghan [20]の比較の対象にも入っている。しかし、このアルゴリズムは先に触れた停止条件の不備に加えて、 $\{P_{in}\}$ の非対称・連結性の証明の誤り等、計5ヶ所も不備な点がある。筆者らが用いていたアルゴリズム [27] はこれらの誤りを修正したものに对应し、かつ明確に4-連結と8-連結の使い分けができるよう  $\{P_{in}\}$ が簡単になっている。この5ヶ所の誤りとその改善については、名大の横井氏とO'Callaghan[35]によって確認されている。従って、ここでは彼らのアルゴリズムは比較の対象としなかった。

#### 4.2.2 定量的実験

上述171例の図形についての計算機処理を表3に示す。

以後の議論では簡単のため各アルゴリズムをRHA(2) (Rutovitz-Hilditch's Algorithm)のように記す。

細線化終了までの反復回数は、並列アルゴリズムをハード化した場合の処理速度に対応する。1回の操作での除去候補点の数は、

$$AA:DA(1):TA(1):TA(5)=1;\frac{3}{4}:\frac{1}{2}:\frac{1}{4}$$

$$RHA(1)=RHA(2)>DA(1)=DA(2), \quad TA(1)=TA(2)=TA(3)$$

であるから、ほぼこの逆比が反復回数となる。実際にはhalt checkのための1反復を加える必要があり、また $n=n_0$ のサイクル終了前に細線化完了となることがあるので、線巾があまりない時にはRHA, DAとTA, YAの間にはそれほど差はない。RHA(1) > RHA(2), DA(1) > DA(2)に注目しよう。RHA(1), DA(1)は対称性と欠いているので、線と片側から削ることになり、反復回数は増す。特にDA(1)は著しい。逐次処理の場合には画面の走査数が対応するが特に意味はない。線の中方向と真走査方向が一致している部分だけ速くなる(それだけ中心位置が影響を受ける)はずだが、RHA(3)は線の縮退が著しく、YAはサイクル終了まで停止しない(反復回数は必ず偶数)ためむしろ遅くなった。

計算時間はまさに計算機での平均実行時間である。逐次処理が並列処理より遙かに速いのは、FORTRANでプログラムして1要素をIF文でスキップしたのに対し、並列アルゴリズムはPAXを用いて擬似並列演算を行ない、すべての要素を近傍点処理したためである。並列アルゴリズムもFORTRANで同様に処理すれば大した差はない。

PAXを用いた目的は、並列処理の1画面の処理時間を求めるためである。保存条件が多いほど、参照すべき近傍点数が多い(即ち条件が複雑な)ほど、計算時間を要するのだから、この単位処理時間はプログラムの面倒さとハードウェア構成の複雑さの尺度と考えられる。この尺度は、プログラミング言語やその技巧に依存するので絶対的な評価関数ではないが、ある程度の傾向は把握できる。細線化の場合、対称性を重視して並列アルゴリズムを使うことが多いが、ハード構成は複雑でも高速性を重視するならRHA, DAと、シンプルさを求めるならTAということになる。計算機での並列処理には大差ないが、対称性にこだわらなければYAがシンプルでかつ速い。



比較事項	細線化方法 (A) Rutovitz-Hildith			(B) Deutsch		(C) Tamura**				(D) Yokoi	
	(1)	(2)*	(3)	(1)	(2)*	(1)	(2)	(3)	(5)	(4)	(8)
処理方式	parallel		sequential	parallel		parallel				sequential	
平均反復回数	5.0	4.6	6.1	7.4	5.7	12.2	11.1	12.1	6.9	7.4	
平均計算時間(sec)	4.5	4.1	2.0	7.0	5.4	2.9	6.1	8.9	8.0	1.2	
単位処理時間(sec/frame)	0.90		0.33	0.95		0.24 (0.22)***	0.55 (0.49)	0.74 (0.36)	1.16 (0.58)	0.18	
使用言語	PAX		FORTRAN	PAX		PAX				FORTRAN	
連結度の測定	crossing number			crossing number		connecting number				connecting number	
連結性	不完全8連結			不完全8連結		4連結	8連結	不完全8連結	4連結	8連結	
線巾	△	△	△	⊙	⊙	⊙	⊙	⊙	△	⊙	⊙
中心位置	⊙	⊙	△	×	⊙	⊙	⊙	⊙	△	⊙	⊙
安定性	×	×	△	○	○	△	△	△	△	⊙	⊙
等方性	△	○	×	×	○	⊙	⊙	⊙	○	△	△
ノイズに対する強さ	△	○	○	○	○	⊙	⊙	⊙	⊙	×	×
交差部の歪み(+)	△	△	△	△	⊙	○	○	○	○	○	△
" (T)	△	△	△	×	⊙	⊙	⊙	○	⊙	○	○
" (L)	○	○	×	×	⊙	○	○	○	○	×	×
	[11]		[12],[19]	[13]	[16][17]	[27]				[26]	

\* (1)の条件を180°反転させて交互に適用する方法。  
 \*\* [27]に述べられている方法の番号。  
 \*\*\* 周辺ノイズを除去しない場合。

Cf. Rosenfeld & Pfaltz[3](PAX)--- 1.7 sec  
 Arcelli et al. [21] (PAX) ---3.6 sec, 3.6 steps

表 3. 比較実験結果

### 4.2.3 定性的実験

心線の品質について、3.1に述べた項目別に評価を下し、表3に示した。心線として好ましいものから順に、⊙、○、△、×としたが、相対的で幾分主観的評価であって明確な基準はない。

線巾については、連結性と追跡の良からいってすべてである。ただし、4-連結が対角方向に階段に続く箇所は、Deutsch[13]の指摘通り線巾2に見えてしまう(図5(a)(d)(e)(f)(h))。心線として視覚的には、8-連結でL形及びT形交差部のみ4-連結を残すのが良いとされている。DAはこの点最も優れている。しかし、線図形として心線に完全な4-連結が要求される場合には、TA(1)、TA(2) YA(4)は最適線巾とせねばならない。この点RHA、TA(5)は中途半端と言える。

交差数xは4-連結と8-連結の混在を許した場合、連結数 $N_d$ はいずれかに限定した場合の連結度の測定\*に用いられ、 $x=2$ 、 $N_c=1$ の良は除去しても連結成分数に変化しないことが証明されている。([16],[26]) 3x3の近傍処理で連結性保存と考えるのに、この2つの概念は究極のものである。並列処理の場合にサブサイクルに分ける工夫とすれば連結性は完全である。TA(5)では4-連結は保証されない(図6(a)-(f)は文献[19]中のFig.についての細線化である。Stefanelli & Rosenfeldの方法では4-cycleでも4-連結と8-連結が混在していることに注目されたい。

この他の心線の品質としては、RHA(1)、DA(1)の中心位置のずれと方向特性が目立つ。(図6(a)(b)) 保存条件の非対称のためである。これを180°反転させたRHA(2)、DA(2)は良い結果を得ている。(図6(c)) また走査方向への依存性はYA(4)にも現れている。(図6(f))

線の縮退はRHAにおいて著しい。(図6(g)) これは対角方向の端点保存が不適当だからで、DAはこれもまた改善している。TAでは周辺ノイズ対策として、線巾分だけの縮退を許している。(図6(h)) DA、YAの結果からも判断して、周辺

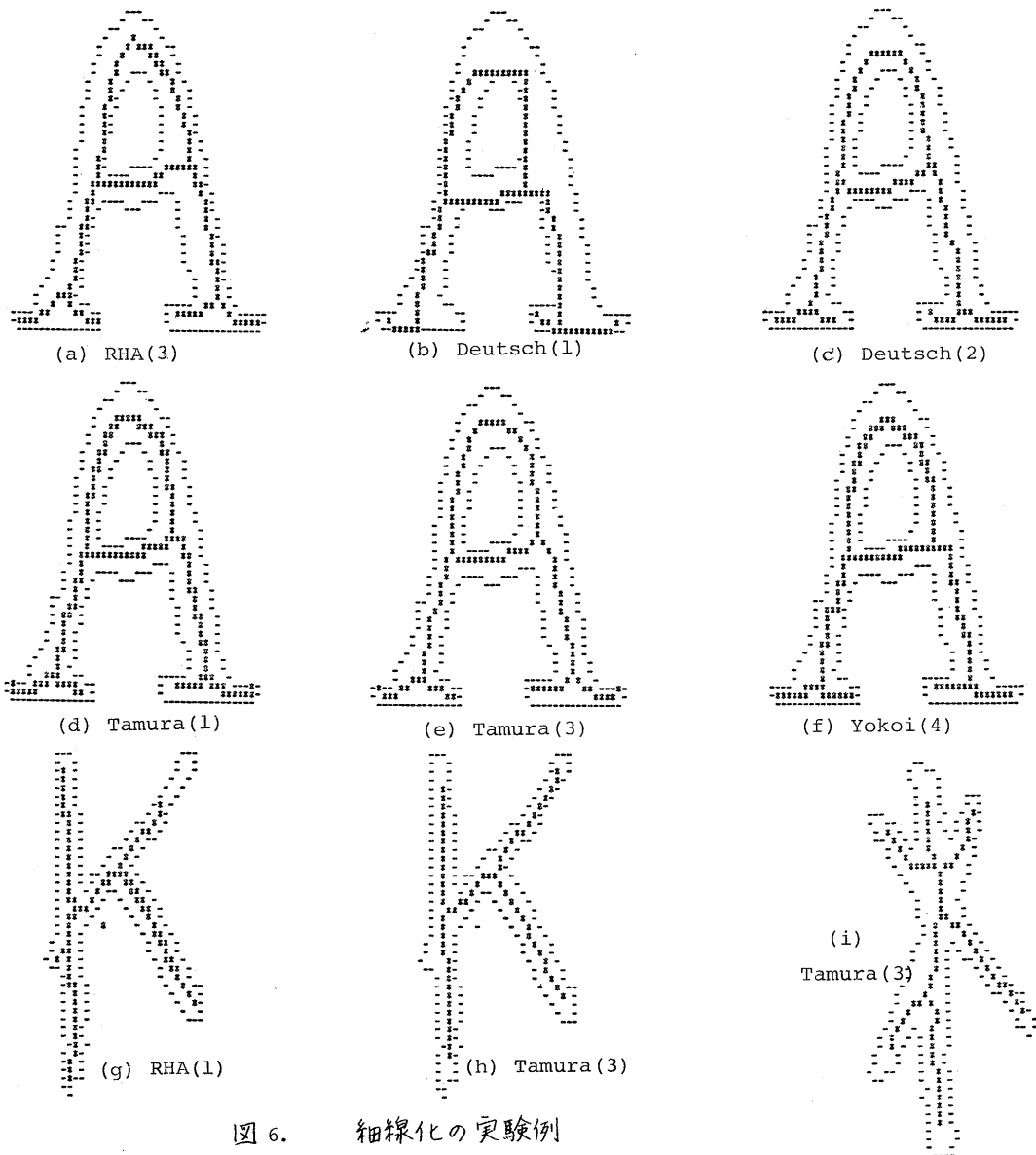


図 6. 細線化の実験例

ノイズ除去と線の縮退は trade off である。

交差部の歪みの評価は理想パターンに対するものである。RHA, DA(1) は本質に歪む性質と持っている。その他の方法でも、線中の不均一な場合や複雑な交差に対しては、忠実な交差と抽出しない。図6(i)のような中広い交差に対しては、3x3の局所演算の欠点と露呈している。

[\*] 交差数と連結数はそれぞれ次のように定義される。([11], [20])

交差数 (crossing number) : 
$$x = \sum_{k=1}^8 |r_k - r_{k+1}|$$

連結数 (connecting number) : 
$$N_c^{[4]} = \sum_{k \in S_1} (r_k - r_k r_{k+1} r_{k+2})$$

$r_4$	$r_3$	$r_2$
$r_5$	$r_0$	$r_1$
$r_6$	$r_7$	$r_8$

$$N_C^{[8]} = \sum_{k \in S_1} (\bar{r}_k - \bar{r}_k \bar{r}_{k+1} \bar{r}_{k+2}), \quad \text{ここで } S_1 = \{1, 3, 5, 7\}$$

### 4.3 総合評価と考察

細線化は単純な発想の処理であり、過去の各方法も似たり寄ったりのものである。画期的な方法など出て来そうにもない。それだけに、わずかな違いで結果が大きく違うというデジタル図形処理の難しさの典型例である。

今回の比較から総合的に言うなら、4-連結には TA(2) (TA(1) よりソフトもハードも作り易い)、8-連結には TA(3) だが単純に心線が得たいなら DA(2) を推薦する。さすが先人 Deutsch が Rutovitz-Hilditch と改良しただけあって良く出来ている。完全 8-連結を得るには [17] にあるように 8-消去可能点を除けばよい。(後処理に頼るのは邪道だと O'Callaghan [35] は言うが……) 逆に TA(1), TA(2) から得た 4-連結心線に、DA(2) を使うか TA 自身と少しアレンジして使うと、L 形、T 形交点が滑らかなものが得られる。経験的に云うと、すべて DA(2) を使うより線として最も滑らかなものが得られ、TA(1)+DA(2) は計算時間も最も短い。

本稿において心線の着眼点と各方式の欠点を述べたから、各自目的と対策に応じて使い分けるか、上のようにアレンジを考えると良い。近傍点処理に頼るなら連結性保存の点除去方式は DA, TA が限界であろう。3.1 の第 1 級条件はほぼ完全に満足しているし、第 2 級条件でも問題となるのは交差点の歪みだけである。この解決策には、何らかの形で線方向の情報を得ることである。これと点除去の保存条件に理め込むのは難しい。むしろ後処理で心線を整形する方が実現しやすい。最近発表された Murthy & Udapa [22] はこの例である。この方法はまだ線方向が 8 方向に限定される欠点があるが、端点と交点と方向を記憶して searching によって心線を整形する後処理は、今後も工夫してみる価値はある。

後処理よりも、前処理とのマッチングは以前から重要性が述べられている。

Smoothing と threshold selection の充実により、周辺ノイズはかなり改善される。またこれらにより大抵の濃淡図形は処理できると考えている。濃淡情報を直接細線化に使う必然性はあるのか疑問であるし、その著しい計算は全体のパフォーマンスを損う感がある。細線化に過剰期待したり、従って複雑で目新しい(ように見える)方法を編み出すよりも、細線化の限界を知って前処理・後処理とも調和を考えるべきだと思われる。

### §5. おわりに

初めに述べた趣旨からの試みなので、少しでもお役に立てば幸いである。紙数と時間的制約から不満足なものとなった。近い将来、加筆修正する予定である。恐らく、浅学のゆえ思い違いもあろうし、既にもっと深く検討された方も居られるだろう。是非、御批判、御提言頂きたいと考えている。

謝辞 本稿の基礎をなす各方法の理解と検討事項は、名古屋大学横井茂樹氏と CSIRO の O'Callaghan に負うところが大きい。特に横井氏からはプログラムの一部も提供して頂いた。京都大学長尾教授、名古屋大学鳥脇純一郎助教授から適切な御指摘と、文字認識については日立中研安田道夫氏、東芝総研森健一氏並びに渡辺貞一氏、電総研森俊二室長にお話を賜った。手書き文字データ使用については図形処理研究室の山本和彦氏、山田博三氏の協力を得た。これらの方々に深く感謝します。

## REFERENCES

### Schemes

- [1] H.Blum:"A transformation for extracting new descriptors of shapes". In *Models for the Perception of Speech and Visual Form*, W.Wathen-Dunn, Ed., MIT Press, pp.362-380 (1967).
- [2] O.Philbrick:"Shape description with the medial axis transformation". In *Pictorial Pattern Recognition*, G.C.Cheng et al. Eds., Thompson, pp.395-407 (1968).
- [3] A.Rosenfeld & J.L.Pfaltz:"Sequential operations in digital picture processing". *J.ACM* 13, 4, pp.471-494 (Oct. 1966).
- [4] U.Montanari:"A method for obtaining skeltons using a quasi-Euclidean distance". *J.ACM* 15, 4, pp.600-624 (Oct. 1968).
- [5] U.Montanari:"Continuous skeltons from digitized images". *J.ACM* 16, 4, pp.534-549 (Oct. 1969).
- [6] G.Levi & U.Montanari:"Grei-weighted skelton" *Inf. & Cont.*17, pp.62-91 (August 1970).
- [7] H.Sherman:"A quasi-topological method for the recognition of the line patrnrs". Proc. Int. Conf. on Information Processing, Paris, UNESCO, pp.232-238 (1959).
- [8] P.Saraga & D.J.Woollons:"The design of operators for pattern recognition". IEE-NPL Conf. Pattern Recognition, PP.106-116 (July 1969).
- [9] E.S.Deutsch:"Computer smilation of a character-recognition machine, part 1". *The Post Office Electrical Engineers J.* 60, pp.39-44 (April 1967).
- [10] E.S.Deutsch:"Preprocessing for character recognition". IEE-NPL Conf. on Pattern Recognition, pp.179-190 (July 1967).
- [11] D.Rutovitz:"Pattern recognition". *J. Royal Statist. Soc.* 129, Series A, PP.504-530 (1966).
- [12] C.J.Hilditch:"Linear skeltons from square cupboards". In *Machine Intelligence IV*, B.Meltzer & D.Michie, Eds., University Press, Edinburgh, PP.403-420 (1969).
- [13] E.S.Deutsch:"Comments on a line thinning scheme". *Comput. J.* 12, p.4]2 (Nov. 1969).
- [14] J.G.Fraser:"Further comments on a line thinning scheme". *Comput. J.* 13, 2, pp.221-222 (May 1970).
- [15] M.J.E.Golay:"Hexagonal parallel pattern transformations". *IEEE Trans on C-18*, 8, PP.733-740 (August 1969).
- [16] E.S.Deutsch:"Towards isotropic image reduction". IFIP Congress '71, North Holland Pub. Co., pp.161-172 (1971).
- [17] E.S.Deutsch:"Thinning algorithms on rectangular, hexagonal, and triangular arrays". *C.ACM* 15, 9, pp.827-837 (Sept. 1972).
- [18] E.E.Triendl:"Skeltonization of noisy handdrawn symbols using parallel operations". *Pattern Recognition* 2, PP.215-226 (Sept 1970).
- [19] R.Stefanelli & A.Rosenfeld:"Some parallel thinning algorithm for digital pictures". *J.ACM* 18, 2, pp.255-264 (April 1971).
- [20] J.F.O'Callaghan & J.Loveday:"Quantative measurement of soil cracking patterns". *Pattern Recognition* 5, 2, pp.255-264 (April 1971).
- [21] C.Arceilli, L.Cordella & S.Levialdi:"A grassfire transformation for binary digital pictures". Proc. 2nd Int. Joint Conf. on Pattern Recognition, pp.152-154 (1974).
- [22] I.S.N.Murthy & Udapa:"A search algorithm for skeltonization of thick patterns". *Computer Graphics & Image Processing* 3, 3, pp.247-259 (Sept. 1974).
- [23] 標上:"高度並列演算装置による図形の記述(I)". *電試彙* 31, 8, pp.930-946 (昭42-08).
- [24] 標上, 山口, 青山:"同上 (VI)". *ibid.* 33, 5, pp.479-505 (昭44-05).
- [25] 山本 地:"手書き数字認識論理の設計". *信学論* (C) 53(C), 10, pp.691-698 (昭45-10).
- [26] 横井, 島協, 福村:"標本化された二値図形のトポロジカル性質について *ibid.* 56-(D), 11, pp.662-669 (昭48-11).
- [27] 田村, 森:"二値図形の並列細線化アルリズムと連結性". 昭49 信全大 No.1539, p.1390.
- [28] 鹿野, 島協, 福村:"濃淡図形と標図形に変換する一方法 —wave propagation methodについて—". *信学論* (D) 55-(D), 10, pp.668-675 (昭47-10).
- [29] 横井, 島協, 福村:"濃淡図形の細線化の一方法". *信学会 PRL 研資* 73-69 (昭48-11).
- [30] 伊藤, 佐藤, 源田:"濃淡標図形の細線化の一方法". 昭49 信全大, No.1623, p.1474.

### Commentaries

- [31] J.C.Mott-Smith:"Medial axis transformation" in *Picture Processing and Psychopictorics*, B.S. Lipkin & A.Rosenfeld Eds., Academic Press, pp.267-278 (1970).
- [32] D.Rutovitz:"Data structure for operations on digital images". in *Pictorial Pattern Recognition*, G.C.Cheng et al., Eds., Thompson Book Co., pp.105-133 (1968).
- [33] E.S.Deutsch:"On some preprocessing techniques for character recognition". In *Computer Processing in Communications*, J.Fox, Ed., Polytechnic Press, pp.221-234 (1970).
- [34] R.O.Duda & P.E.Hart:*Pattern Classification and Scene Analysis*. Wiley (1973).
- [35] J.F.O'Callaghan: Private letters to the author (Sept. & Nov. 1974).

### Applications

- [36] R.A.Kirsch et al.:"Experiments in processing pictorial information with a digital computer". Proc. ECC, pp.221-229 (1957).
- [37] N.F.Izzo & W.Coles:"Blood cell scanner identifies rare cells". *Electronics* 35, pp.173-183 (April 27, 1962)
- [38] B.H.McCormivk:"The Illinois pattern recognition computer-Illiic III". *IRE Trans. on EC-12*, 5, pp.791-813 (Dec. 1963).
- [39] R.Narasimhan:"Labelling schemata and syntactic description of pictures". *Inf. & Cont.* 7, pp. 151-179 (1964).
- [40] J.L.Pfaltz & A.Rosenfeld:"Computer representation of planar regions by their skeltons". *C.ACM* 10, 2, pp.119-122,125 (Feb.1967).