

解説



画像処理のアルゴリズム†

鳥脇 純一郎†† 横井 茂樹†††

1. ま え が き

デジタル画像処理の分野において、多種多様な処理方法の開発はいまなお活発であるが、それと共に、最近では、過去の成果の評価、体系化、蓄積、そして効率の利用へ向けての努力も急速に進展している。このような時期において、従来の画像処理アルゴリズムを種々の観点から整理しておくことの意義は大きい。ところで、画像処理手法に関しては既にいくつかの専門書や解説があるが<sup>1)~7)</sup>、それらは、アルゴリズムの機能に重点をおく解説である。そこで、本文では、これらとはやや異なった観点、すなわち、処理の実行形式に重点をおいて、画像処理アルゴリズムの性質を調べてみよう。なお、本文では、デジタル画像処理を対象とし、したがって、処理は電算機(専用デジタル処理装置を含む)で実行されるものとする。また、画像処理アルゴリズムは、漠然と、画像処理の手順を記述したもの、という程度にとらえておく。もちろん、その内容は(有限個の変数パラメータの値を除いて)厳密に規定されており、想定される処理対象画像(入力画像)のすべてに対して、有限時間内に処理が終了しなくてはならない。

2. 画像処理アルゴリズムの性質

2.1 アルゴリズムの階層構造

画像処理アルゴリズム(以下単に(画像)処理手順とも言う)の各々は、より単純な手順の組に分解でき、一種の階層的構成を持つとみなされる(図-1)。この階層の上位レベルは、下位のものよりも処理内容がより高度で手順も複雑になっているが、各階層を厳密に規定する方法はない。本文では、比較的まとまった単位

となる手順を適当に選んで扱うが、それらは、必ずしも同一の階層とみなされるわけではない。しかし、以下に述べるような考察を適用するには、それに適した階層に分解して考える必要はある。例えば、テクスチャ解析のあるものは、《局所統計量フィルタ—差分型フィルタ—しきい値処理—細線化》と分解して考えられる。但し、残念ながら処理手順の分解と合成に関する組織的検討はまだ非常に少ない<sup>8)</sup>。

2.2 画像処理機能

画像処理アルゴリズムの機能は、それに対する入力情報と出力情報の形態から、次のように分類される。なお、本文で扱う画像はすべて標準化画像とする。すなわち、画像は2次元の各格子点に濃度値が与えられたものと考えればよい。厳密な定義は2.4で与える。(以下→の左が入力情報、右が出力情報を示す)

(1) 画像→画像: 2.3 参照。

(2) 画像→ベクトル, 記号列等の表: 画像からの特徴量の計測、線図形からの方向コード列への変換<sup>9)</sup>、画像からグラフによる記述の生成<sup>10)~12)</sup>等。

(3) ベクトル, または、記号列→画像: (2)の逆過程としての画像生成。画像処理の研究や画像出力、表示装置の活用のための道具として不可欠であるが、与えられた画像の解析のためには用いなくとも済む。

以下、本文では(1)のみを扱う。(2)~(3)については前記文献1)~4), 9), 10)等やパターン認識、グラフィックス関係の専門誌を参照されたい<sup>11)</sup>。個別

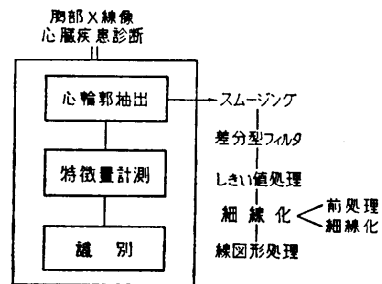


図-1 画像処理アルゴリズムの階層構造

† Algorithms for Image Processing by Jun-ichiro TORIWAKI (Department of Information and Computer Science, Toyohashi University of Technology) and Shigeki YOKOI (Department of Electronics Engineering, Faculty of Engineering, Mie University).

†† 豊橋技術科学大学情報工学系

††† 三重大学工学部電子工学科

処理機能の分類表が文献 12) にみられる。

2.3 処理対象の性質

画像→画像の変換機能をもつ処理手順は、その入出力画像の性質で分類される。一般に、画像は、濃度値とその分布状態が担う濃度情報と図形輪郭線や等高線の形が担う形状情報をもつ。両者ともその厳密な定義や相互間の関係については今後の研究に待たねばならず、ここでは直感的な記述にとどめる。まず、濃度情報に注目するとき、2値画像と濃淡画像に分類され、他方の形状情報に注目するとき、背景(または“地”)と図形の部分に分かれるものと、そうでないもの(全面画像とよんでおく)に分けられ、さらに、前者は点状図形、線状図形、塊状図形に細分される。そして、画像処理手順はこれらの相互間の変換を実行する。表-1、および図-2にその例を示す。もちろん、このような分類はかなり便宜的なもので厳密な定義はない。

2.4 画像演算とその実行形式

第*i*行*j*列標本点(画素)の濃度値が  $f_{ij}$  である画像を、 $F = \{f_{ij}\}$  で表わす。画像→画像の変換は、デジタル画像全体の集合  $S$  の適当な部分集合  $S_1, S_2$  をそれぞれ定義域、値域とする写像  $O: S_1 \rightarrow S_2$  として定式化される。この  $O$  を画像演算(または単に演算子)と呼ぶ。画像演算の代数的性質を用いて画像処理手順の理論的解析が可能となるが、その詳細は他の文献<sup>9), 13)</sup>にゆずり、ここでは、画像演算の実行形式に注目する。以下、ある処理に対する入力画像を  $F = \{f_{ij}\}$ 、出力画像を  $G = \{g_{ij}\}$  で表わす。

画像演算の最も一般的な形は、次のようにならう。

$$g_{ij} = \varphi_{ij}(F), \forall (i, j) \in S \quad (1)$$

但し、 $\varphi_{ij}$  は適当な多変数関数\*、 $S$  は標本点番号全体に対する整数対の集合である。たとえば、ある種の観測系による処理は次の形にかかれる<sup>2)</sup>。

$$g_{ij} = \sum_{(p, q) \in S} h(p, q; i, j) f_{pq}, \forall (i, j) \in S \quad (2)$$

以下に、実際によく用いられる演算子の種類を示す。

(1) 並列型と逐次型 次の形の演算を並列型演算子(処理)(parallel operation)という。

$$g_{ij} = \varphi_{ij}(f_{11j}, f_{12j}, \dots, f_{injn}), \forall (i, j) \in S,$$

\* 式(1)の定義はすべての画素  $(i, j)$  に関して出力  $g_{ij}$  を求める操作を意味する。したがって、 $(i, j) \in S$  なるすべての  $(i, j)$  に対する  $\varphi_{ij}$  の集合  $\{\varphi_{ij}; (i, j) \in S\}$  によって1つの画像演算子が定義される。以下この種の表現はすべて同様の意味である。

\*\* 出力  $G$  の定義としては右辺に  $g_{ij}$  を含む形も有り得るが<sup>2)</sup>、このときの  $G$  の計算は別のアルゴリズムによらねばならない。

\*\*\* この点で、逐次型処理はこの順序で画素の出力を決定する逐次決定過程とみなすこともできる。しかし、決定の段数と2次元画面上の近傍関係は必ずしも一致しない点に注意を要する。

表-1 画像→画像の変換機能をもつ処理の例

入力 出力	処 理 の 例
濃淡→濃淡	フィルタリング、フーリエ変換
濃淡→2値	しきい値処理、特徴点抽出、要素マーキング
2値→濃淡	距離変換、連結成分ラベル付け
2値→2値	細線化、図形融合
点状→点状	特徴点抽出、フィルタリング
点状→線状	外部スケルトン
点状→塊状	スケルトンからの図形復元
線状→点状	特徴点抽出
線状→線状	線図形処理、線図形距離変換
線状→塊状	スケルトンからの図形復元
塊状→点状	スケルトン抽出
塊状→線状	細線化、境界点抽出

$$(i_k, j_k) \in \mathcal{N}((i, j)), \forall k (1 \leq k \leq N) \quad (3)$$

但し、 $\varphi$  は任意の  $N$  変数関数、 $N$  は任意の整数、 $\mathcal{N}((i, j))$  は要素  $(i, j)$  の近傍で、次の式で定義される。

$$\mathcal{N}((i, j)) = \{(i-p, j-q); (p, q) \in S'\} \quad (4)$$

$S'$  は  $S$  の適当な部分集合。

また、次の演算を逐次型演算子(処理)(sequential operator)という<sup>20)</sup>。

$$g_{ij} = \varphi_{ij}(g_{11j}, g_{12j}, \dots, g_{imjm}, f_{im+1, jm+1}, \dots, f_{in-1, jn-1}, f_{ijn}), \forall (i, j) \in S$$

$$(i_k, j_k) \in \mathcal{N}((i, j)), \forall k (1 \leq k \leq N) \quad (5)$$

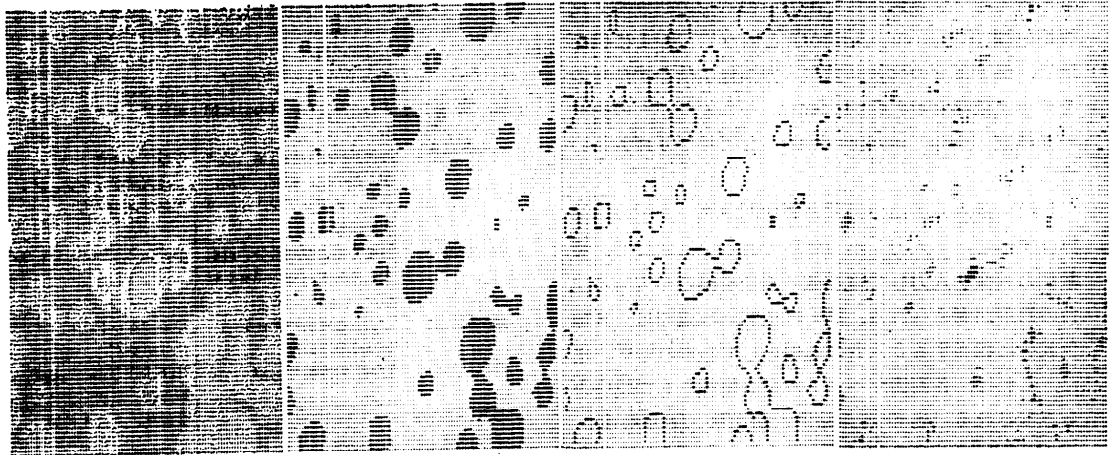
但し、 $\varphi$  は任意の  $N$  変数関数、 $M, N (1 \leq M \leq N)$  は任意の整数。また、右辺には  $g_{ij}$  は含まないとする\*\*。

並列型の特徴は、①出力  $g_{ij}$  は入力  $F$  の値のみから計算される、②異なる  $(i, j)$  における出力  $g_{ij}$  の値は、互いに独立に、したがって、原理的には同時並列的に計算できる等の点にあり、逐次型のそれは、①出力  $g_{ij}$  が入力  $F$  と出力  $G$  の両方を用いて計算される、②異なる  $(i, j)$  に対する出力  $g_{ij}$  は同時並列的には計算できず、一定の手順で1個ずつ計算される\*\*\*等の点にある。多くの処理が、このいずれでも表現できるが、それを実行するハードウェア、ソフトウェアの構成や実現の容易さ、実行時間等はかなり異なる。

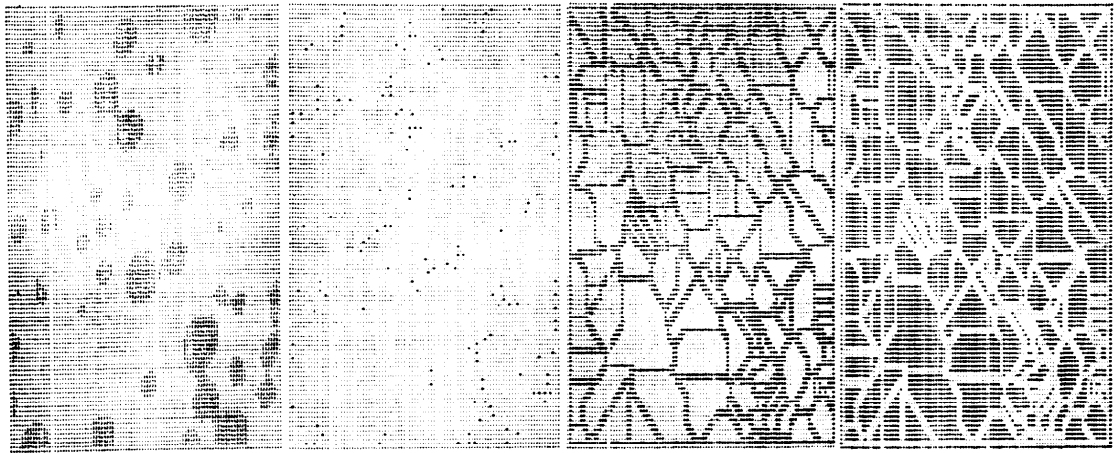
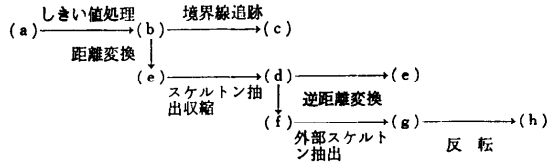
逐次型において、処理順序が入力画像  $F$  に依存して定まるものがある。その中で、適当な条件を満たす出発点から、入力と既に得られた出力の値を調べながら、必要な要素を順に進んで処理を進めるといった形のもの、特に、逐次追跡型とよぶ。

これらの処理の例を表-2, 3, 4 に示す。

(2) 位置不変と位置可変 式(1), (3), (5)



(a) 全面画像 (b) 2値塊状図形 (c) 2値線状図形 (d) 濃淡点状図形



(e) 濃淡点状図形 (f) 2値点状図形 (g) 2値線状図形 (h) 2値塊状図形

図-2 1つの入力画像より導かれる種々の画像の例

における関数  $\varphi_{ij}$  が位置  $(i, j)$  に依存しない処理を位置不変(position invariant), その他のものを位置可変(position variant) という。

(3) 局所処理と大局的処理 式(3), (5)における近傍  $\mathcal{N}((i, j))$  の大きさ(その中の要素数)が画面全体に比べて十分小さい場合を局所処理(演算)(local operator), そうでない場合を大局的処理(global operator)という。明らかに, 小さい近傍を用いる処

理(局所処理)では,  $(i, j)$  における出力の決定に大局的情報は寄与しない。

(4) 分割処理可能性 画面をいくつかの部分に分割し, 各部分ごとに処理をしても, 全画面に対する処理と同一の結果が(定義域内のすべての入力画像に対して)得られるとき, この処理は分割処理可能という。小規模装置での処理には重要な性質である。

(5) 反復型処理 同一種類の演算をある条件が満

たされるまで反復適用して出力を得る形の処理を、反復型処理 (iterative operator) という<sup>7)</sup>。表-4 に例を示す。終了までの反復回数が入力画像に依存するものをデータ依存 (data dependent), そうでないものをデータ独立 (data independent) なアルゴリズムという<sup>7)</sup>。

(6) 多入力多出力処理と画像間演算 上記の諸演算はすべて、1入力1出力処理であるのに対して、複数個の画像を入力して複数個の画像を得る多入力多出力の処理が考えられる。例えば色彩画像処理は、3原色成分に相当する3入力の処理として扱われる<sup>14)</sup>。また、これは、各画素がベクトル量の情報をもつ場合ともみなされる (場の効果法<sup>15)</sup>, 緩和法 (relaxation method)<sup>16)</sup>等)。2入力の場合は画像2項演算として定式化される<sup>13)</sup>。他の例として、動画画像解析<sup>17)</sup>, 3次元画像処理<sup>18)</sup>がある。

(7) その他特殊なデータ構造 (たとえばピラミッド構造) には、それ特有の処理形式があるが詳細は略す<sup>4), 26)</sup>。

3. 並列型局所処理

並列型局所処理 (parallel local operator-PLO) は、実際の画像解析に何らかの形で必ず用いられる重要な処理手順の類である。それは、各画素の濃度値を近傍画素の情報に基づいて替換する処理ともみなされることからセル論理演算 (cellular logic operator-CLO) とも呼ばれる<sup>7)</sup>。主なものを、非反復型と反復型に分けて表-2, 3 に示す。

非反復型 PLO は、雑音除去, 画像強調, 輪郭検出等, 実際の画像解析に不可欠な機能をもつが, アルゴリズムの構造はおおむね単純で, その定義から容易に理解できる。これに対して, 反復型処理は, 複雑な構造をもつため, その構造がやや把握し難く, それだけに, アルゴリズムとして面白いものが多い。以下に, 2, 3の例を紹介しよう。

[アルゴリズムA1] 入力画像  $F = \{f_{ij}\} (f_{ij} \geq 0, \forall (i, j))$  から, 次の式で画像系列  $G^{(k)} = \{g_{ij}^{(k)}\}, k = 0, 1, 2, \dots$  を求める。

$$G^{(0)} = M \cdot F, M \text{ は十分大きい正定数,}$$

$$g_{ij}^{(k+1)} = \begin{cases} \min\{g_{ij}^{(k)}, g_{i,j-1}^{(k)} + f_{ij}, g_{i-1,j}^{(k)} + f_{ij}, g_{i+1,j}^{(k)} + f_{ij}, g_{i,j+1}^{(k)} + f_{ij}\}, \\ f_{ij} > 0 \text{ のとき,} \\ 0, \text{ その他.} \end{cases}$$

ある  $k$  の値  $K$  に対して,  $G^{(K+1)} = G^{(K)}$  ならば終了。  
 $G^{(K)} = G (= \text{出力画像})$  とする\*\*。

表-2 並列型局所処理の例 (非反復型)

局所処理の内容 ( $\{f_{pqr}\} = (i, j)$ の近傍の濃度値の集合)	例
加重和 $\sum a_{pqr} \cdot f_{pqr}, (a_{pqr})$ は定数	線形フィルタ
局所統計量 近傍濃度値の統計量 (最大値, 最小値, 範囲, 平均, 分散, メディアン, K-S 統計量等々)	各種フィルタ (最大値フィルタ, メディアンフィルタ等)
濃度値差分 濃度分布の空間微分を近似する差分式 (特に 3x3 近傍)	gradient, Laplacian 等
類似度 $\{f_{pqr}\}$ と局所パターン $\{w_{pqr}\}$ の類似度	局所パターンマッチング
非線形処理 $\{f_{pqr}\}$ の非線形関数	Hueckel 演算, 連結数, 曲率係数, 高次曲面あてはめ 2値化, 0と正值等
しきい値処理	
平行移動 $f_{i-p, j-q}$	
マーキング 指定条件を満たす要素のみ 1その他は0	境界点抽出
ラベル付け 要素分類記号の割当て	連結成分のラベル付け, 要素分類

表-3 並列型局所処理の例 (反復型)

アルゴリズム	反復される基本処理
距離変換	最小値フィルタ
逆距離変換	最大値フィルタ
細線化	消去条件式計算, 正值要素消去
収縮 (shrinking)	連結数計算と 1, 0 の反転
連結成分ラベル付け	最大値フィルタ, または最小値フィルタ
図形融合	最小値フィルタおよび最大値フィルタ
伝播	最大値フィルタ
場の効果法, Relaxation	ベクトルの更新

表-4 逐次型局所処理の例

- (1) 逐次型\*: 距離変換, 逆距離変換, 一線重み線形フィルタ, 最大(小)値フィルタ, 中央値フィルタ, 細線化, 図形収縮, 領域生成
- (2) 逐次追跡型: 境界線追跡, 尾根線追跡, 等高線追跡

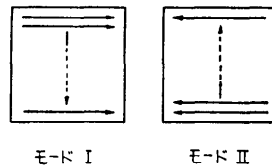


図-3 両面の走査方法

\* これらは, 全く等価な並列型アルゴリズムが存在するもの (距離変換等), 全く等価ではないがほぼ同機能をもつ並列型アルゴリズムが存在するもの (細線化等), および並列型アルゴリズムが明確には知られていないもの (領域生成等) の3種類に分けられる。

\*\* 画像  $F = \{f_{ij}\}, G = \{g_{ij}\}$ , 定数  $M$  に対して,  $M \cdot F$  は  $(M \cdot f_{ij})$  (全要素の濃度値を  $M$  倍したもの),  $F = G$  は,  $f_{ij} = g_{ij}, \forall (i, j) \in S$  ( $F$  と  $G$  の同じ位置の画素の濃度値は常に等しい) を意味する。

これは、重みつき距離変換 (4 近傍型) (Grey weighted distance transformation) の並列型アルゴリズムである<sup>8),19)</sup>。入力が 2 値画像のとき出力  $G$  の値  $g_{ij}$  は、図形境界から画素  $(i, j)$  までの最短距離値に等しい。図形の計数、スケルトン化、形状解析の有力な手段となる。データ依存手順である。

[アルゴリズム A 2] 入力 2 値画像  $F = \{f_{ij}\}$  から次の式で画像系列  $G^{(k)} = \{g_{ij}^{(k)}\}$ ,  $k=0, 1, 2, \dots$  を求める。

$$G^{(0)} = F$$

- ①  $k=0, 1, 2, \dots, M-1$  に対して,  

$$g_{i,j}^{(k+1)} = \begin{cases} 1, & g_{i,j-1}^{(k)} \wedge g_{i-1,j}^{(k)} \wedge g_{i,j+1}^{(k)} \wedge g_{i+1,j}^{(k)} \\ & = 1 \text{ の時,} \\ 0, & \text{その他} \end{cases}$$
- ②  $k=M, M+1, \dots, 2M-1$  に対して,  

$$g_{i,j}^{(k+1)} = \begin{cases} 1, & g_{i,j-1}^{(k)} \vee g_{i-1,j}^{(k)} \vee g_{i,j+1}^{(k)} \vee g_{i+1,j}^{(k)} \\ & = 1 \text{ の時,} \\ 0, & \text{その他} \end{cases}$$

但し、 $\wedge$  は論理積、 $\vee$  は論理和、 $M$  はあらかじめ定める正整数であり、 $G^{(2M)}$  を出力  $G$  とする。

これは図形融合 (第 1 種) (fusion) とよばれる処理で、出力  $G$  では、入力  $F$  の図形中で幅  $2M$  以下の部分が消去される<sup>8),20)</sup>。2 値画像中の雑音除去や図形整形に有効である。データ独立な手順である。

[アルゴリズム A 3]<sup>21)</sup> 入力 2 値画像  $F = \{f_{ij}\}$  から、次の式で画像系列  $G^{(k)} = \{g_{ij}^{(k)}\}$ ,  $k=0, 1, 2, \dots$  を計算する。

$$G^{(0)} = F$$

$$g_{i,j}^{(k+1)} = \alpha[\alpha(g_{i,j-1}^{(k)} + g_{ij}^{(k)} + g_{i+1,j}^{(k)} - 1) + \alpha(g_{ij}^{(k)} + g_{i+1,j-1}^{(k)} - 1)],$$

$$\text{但し, } \alpha(x) = \begin{cases} 0, & x \leq 0 \text{ の時} \\ 1, & x > 0 \text{ の時} \end{cases}$$

ある  $k$  の値  $K$  において、 $g_{ij}^{(K)} = 0, \forall (i, j) \in S$  となれば終了。

これは図形収縮 (shrinking) の一例である。入力  $F$  内の値 1 の画素の連結成分 (図形) は、この反復のどこかで必ず 1 点に収縮し、次の反復で消去される。したがって、孤立点の消滅個数から入力画像中の図形の個数を計測できる。データ依存手順である。

ある PLO  $O_1$  の出力に他の PLO  $O_2$  を適用するという処理を 1 個の画像演算とみると、それは再び PLO になり、そこに用いられる近傍は  $O_1, O_2$  に用

\* 2つのアルゴリズム (画像演算) が、入力として想定される (定義域の) すべての画像に対して同一の出力を与えるとき、両者は等価であるという。

いられた近傍よりも大きい。したがって、PLO の反復適用 (直列合成<sup>13)</sup>) によってある程度大局的な処理を実現できる。局所処理は、ソフトウェア、ハードウェアの両面において大局的処理よりも実現し易いことが多いから、この性質は応用上有用である。距離変換、場の効果法、緩和法、等もこの一例とみなせる。

#### 4. 逐次型局所処理

2.4 で述べたように、逐次型の本質的特徴は、画素  $(i, j)$  における出力  $g_{ij}$  の計算に、既に計算の終わっている出力値も用いるという点にある。したがって、各画素の出力値の計算順序 (画面走査方式—以下走査モードとよぶ) が重要な意味をもつ。図-3 に代表的な 2 種の走査モードを示す。次に、逐次型アルゴリズムの 2, 3 の例を示す。

[アルゴリズム A 4] 入力画像  $F = \{f_{ij}\} (f_{ij} \geq 0, \forall (i, j))$  から、次の式で画像系列  $G^{(k)} = \{g_{ij}^{(k)}\}$ ,  $G^{(k)}, k=0, 1, 2, \dots$  を求める。

$$G^{(0)} = M \cdot F, (M \text{ は十分大きい正定数})$$

$$\textcircled{1} \quad g_{ij}^{(k+1)} = \min\{g_{ij}^{(k)}, g_{i-1,j}^{(k+1)} + f_{ij}, g_{i,j-1}^{(k+1)} + f_{ij}\}, \forall (i, j) \in S.$$

但し、 $(i, j)$  については走査モード I の順序で計算する。

$$\textcircled{2} \quad g_{ij}^{(k+1)} = \min\{g_{ij}^{(k)}, g_{i+1,j}^{(k+1)} + f_{ij}, g_{i,j+1}^{(k+1)} + f_{ij}\}, \forall (i, j) \in S.$$

但し、 $(i, j)$  については走査モード II の順序で計算する。

ある  $k$  の値  $K$  で  $G^{(K)} = G^{(K)}$ , または、 $G^{(K)} = G^{(K-1)}$  ならば終了。 $G^{(K)}$  (または  $G^{(K-1)}$ ) =  $G$  を出力とする。

これは重みつき距離変換の逐次型アルゴリズムであり、A 1 と完全に等価である\*。

[アルゴリズム A 5]<sup>22)</sup> 入力 2 値画像  $F = \{f_{ij}\}$  から、次の手順で画像系列  $G^{(k)} = \{g_{ij}^{(k)}\}$ ,  $k=0, 1, 2, \dots$  を求める。

$$G^{(0)} = F,$$

$$g_{ij}^{(k+1)} = \begin{cases} 0, & g_{ij}^{(k)} = 1, \text{ かつ, } \mathcal{N}_c(\mathcal{A}_c) = 1 \text{ のとき,} \\ g_{ij}^{(k)}, & \text{その他,} \end{cases}$$

但し、 $\mathcal{N}_c(\mathcal{A})$  は、 $\mathcal{A}$  を用いて計算される  $(i, j)$  の連結数<sup>22)</sup>,

$$\mathcal{A} = \{g_{i-1,j-1}^{(k+1)}, g_{i-1,j}^{(k+1)}, g_{i-1,j+1}^{(k+1)}, g_{i,j-1}^{(k+1)}, g_{i,j+1}^{(k)}, g_{i+1,j-1}^{(k)}, g_{i+1,j}^{(k)}, g_{i+1,j+1}^{(k)}\}$$

但し、 $(i, j)$  の計算順序は走査モード I による。ある

$k$  の値  $K$  において  $G^{(K+1)}=G^{(K)}$  のとき終了。

$G^{(K)}=G$ =出力画像とする。

これは図形収縮の逐次型手順の1つであり、出力  $G$  では、 $F$  内にある単連結図形はすべて1点に収縮し、多重連結図形はすべて線図形となっている。但し、 $A_3$  とは等価ではない。一般に、並列型と逐次型の互いに対応するアルゴリズムが知られているか否かに関しては種々の場合がある(表-4)。任意の PLO と等価な逐次型手順の存在に関しては文献 20) に論じられている。しかし、筆者の考えでは、これは電算機1語の容量の制約と関連づけて考察しないと意義は少ないように思われる。この点を考慮した厳密な考察はまだない。

逐次型手順の利点は、電算機における実行に際して、処理時間の短縮が可能な点にある。例えば、2値画像の距離変換において、並列型(A1)の反復回数は入力  $F$  中の図形の最大幅に比例するが、逐次型(A4)では常に2回でよい。一様重み線形フィルタ、中央値フィルタ等の高速化の例もある。但し、すべての逐次型手順がこのような利点を持つわけではない<sup>25)</sup>。

## 5. むすび

本文では、画像処理アルゴリズムを特徴づける性質を、特にその実行形式に重点を置いて整理した後、実際に最も広く用いられる並列型および逐次型の局所処理についてやや詳しく説明した。従来の画像処理手順の解説は主として個別処理機能別に論じられており、本文のような観点から述べたものは比較的少ない。しかし、今後の画像処理手法の体系化や優れた処理システムの実現のためには、このような観点からの整理も有用であろう。但し、こういう方向からの画像処理手順の研究はまだそれほど多くはなく、本文で述べた事項にも未解決の部分が少なくない。むしろ、読者の方向に、ここから今後の興味ある研究課題を見い出していただくことを期待したい。最後に、紙数の制約のため、多数の重要な話題を割愛し、また参考文献もごく少数に制限せざるを得なかったことをお断りしておく。

## 謝辞

有益な御助言を頂いた名古屋大学本多波雄教授、福村晃夫教授に感謝する。

## 参考文献

- 1) Rosenfeld, A.: Picture Processing by Computer, Academic Press, New York (1969).
- 2) Rosenfeld, A. and Kac, A.C.: Digital Picture Processing, Academic Press, New York (1976).
- 3) Pratt, W. K.: Digital Image Processing, John-Wiley (1978).
- 4) Pavlidis, T.: Structural Pattern Recognition, Springer, New York (1977).
- 5) 長尾 真: 画像処理のためのソフトウェア, 電子通信学会誌, Vol. 58, No. 1, pp. 13-20 (1975).
- 6) 伊藤貴康: 画像処理のソフトウェア, 電子通信学会誌, Vol. 59, No. 11, pp. 1208-1215 (1976).
- 7) Preston, K., Jr., Duff, M. J. B., Levialdi, S., Norgren, P. E. and Toriwaki, J.: Basics of Cellular Logic with Some Applications in Medical Image Processing, Proc. IEEE, Vol. 67, No. 5, pp. 826-856 (May 1979).
- 8) 横井, 鳥脇, 福村: 濃淡図形処理における図形融合, 距離変換, およびスケルトンの諸性質, 電子通信学会論文誌, J 61-D, 9, pp. 613-620 (1978).
- 9) Fu, K. S. (ed.): Syntactic Pattern Recognition, Applications, Springer-Verlag, New York (1976).
- 10) Pavlidis, T.: A Review of Algorithms for Shape Analysis, Computer Graphics and Image Processing, Vol. 7, No. 2, pp. 243-258 (1978).
- 11) 例えば, IEEE Trans. (Computer, Systems, Man and Cybernetics, Pattern Analysis and Machine Intelligence 等), Computer Graphics and Image Processing.
- 12) 鳥脇, 福村: 画像処理サブルーチンライブラリ SLIP について——機能一覧, 情報処理学会コンピュータビジョン研究会資料, コンピュータビジョン 1-2 (1979).
- 13) 横井, 鳥脇, 福村: 標準化図形の演算系の代数的構造とその応用, 電子通信学会論文誌, 60-D, 6, pp. 430-410 (1977).
- 14) Ito, T.: Towards Color Picture Processing, Computer Graphics and Image Processing, 2, pp. 347-354 (1972).
- 15) 森, 森, 山本: 場の効果法による特徴抽出, 電子通信学会論文誌, 57D, 5, pp. 308-315 (1974).
- 16) Rosenfeld, A.: Relaxation Methods in Image Processing and Analysis, Proc. 4th IJCP, pp. 186-211 (1978).
- 17) Martin, W. N. and Aggarwal, J. K.: Dynamic Scene Analysis, Computer Graphics and Image Processing, 3, pp. 356-374 (1978).
- 18) Arcelli, C. and Levialdi, S.: Parallel Shrink-

\* 2.2(2)の画像→記号列, ベクトルの処理では, モード1走査で全面を1回走査し, 主記憶に高々2行分を置くのみで実行できる処理(線形処理, onepass アルゴリズム)は実用上の意義が大きい<sup>27)</sup>.

- ing in Three Dimensions, Computer Graphics and Image Processing, 1, pp. 21-30 (1972).
- 19) Levi, G. and Montanari, U.: Grey Weighted Skeleton, Information and Control, Vol. 17, No. 1, pp. 62-91 (1970).
- 20) Rosenfeld, A. and Pfaltz, J. L.: Sequential Operations in Digital Picture Processing, J. ACM, Vol. 13, No. 4, pp. 471-494 (1966).
- 21) Levialdi, S.: On Shrinking Binary Picture Patterns, Commun. ACM, Vol. 15, No. 1, pp. 7-11 (1972).
- 22) 横井, 鳥脇, 福村: 標準化された2値図形のトポロジカルな性質について, 電子通信学会論文誌, 56D, 11, pp. 662-669 (1973).
- 23) 鳥脇, 横井, 福村: 電子計算機による画像処理におけるフィルタリングのための高速アルゴリズム, 情報処理, Vol. 17, No. 3, pp. 215-221 (1976).
- 24) Narendra, P. M.: A Separable Median Filter for Image Noise Smoothing, Proc. IEEE Computer Soc. Conf. on PRIP, pp. 137-141 (1978).
- 25) 間瀬, 鳥脇, 福村, 横井: 可変近傍系列を用いた一般距離変換における逐次型アルゴリズムについて, 電子通信学会パターン認識と学習研究会資料, PRL 79-40 (1979).
- 26) Shapiro, L.: Data Structures for Picture Processing: A Survey, Computer Graphics and Image Processing, 11, pp. 162-184 (1979).
- 27) Agrawala, A.K. and Kulkarni, A.V.: Sequential Approach to the Extraction of Shape Features, Computer Graphics and Image Processing, 6, pp. 538-557 (1977).

(昭和55年1月30日受付)