



SPIDER開発を通して見たディジタル画像処理アルゴリズムの現状 (6)

—— 画像処理アルゴリズムの基本構造の分析 ——

田村秀行 坂上勝彦 横矢直和 (電子技術総合研究所)

1. はじめに

ここ数年、画像処理研究の中で、画像処理向きハードウェア^[1]やソフトウェア^[2]に関する研究の占める割合が急増している。メモリーやプロセッサの低廉化により、画像処理特有の並列演算を高速に実行するプロセッサやそのための専用言語の開発が盛んである。^{[3],[4]}

しかし、これらのプロセッサのほとんどは、理想的な2次元並列演算に徹するか、典型的な近傍演算や濃度階調変換等をいくつか具備しているにすぎない。複雑な画像処理・認識の過程をこれらのモジュールの組合せだけで表現するにはかなりの無理がある。その意味では、画像処理に必要な演算モジュールやデータ形式に関してはまだ十分な検討がなされていない。

我々は、代表的な画像処理アルゴリズムを収録したソフトウェア・パッケージSPIDER^[5]を世に問うた。同時に、主要な分野についての一連のサーベイも行なった。^[6]そこでのアルゴリズムの選取と解説は、手法の機能にのみ注目した。即ち、SPIDERはソフトウェアとして自由な発想に基づくアルゴリズムの集積であり、実行形式の観点からは全くの無作為抽出のサンプル集合である。従って、SPIDERを分析することによって、多様な画像処理アルゴリズムの基本構造を解明でき、体系化に有用な情報を得ることができよう。

実行形式面からの分析としては、これまでに鳥脇の画像処理アルゴリズム論^{[7],[8]}がある。我々のSPIDER分析は、これを踏襲し発展させるもので、特に高速プロセッサと専用言語設計の基礎を与えることを目的としている。

2. 引数の機能的分類

2.1 引数の分類意義

SPIDERでは、プログラムの移植性を高めるため、各FORTRANサブルーチンに対して、COMMON領域の使用を禁じ、情報の伝達はすべて引数を経由している。このため、各サブルーチンのCalling sequenceは、

```
CALL HGEQ1(IP,JP,ISX,ISY,HSTO,HSTN,NGRO,
           NGRN,JTBL)
CALL FFTA2(AR,AI,BR,BI,WR,WI,ISX,ISY,ISXF,
           ISYF,WST,WCT,IBR,ISMF,INV,JERR)
```

のように長くなり、使いにくさの一因ともなっている。

しかし、反面あらゆる情報がここに集約されているので、これを分析・分類することにより、プロセッサや言語設計上の要めとなる概念が得られよう。特に、

- (1) 機能的分類を行なうことにより、画像処理に固有なデータ型や、ハードウェア構成上の基本要素が判明する。
- (2) 各々のデータ型の表現形式や拘束条件を整理することにより、デフォルト設定やモジュール間の整合性検証が容易になる。等の利点がある。

2.2 画像の大きさや濃度レベル

SPIDERサブルーチンの引数が多い最大の原因は、汎用性を高めるため画像の大きさや濃度レベル数を可変とし、画像(及び関連する)配列を整合配列としたことである。

以下では、画像 $F = \{f_{ij}\}$ の大きさやレベル数について次の記号を用いるものとする。^{*}(F はいずれも省略できる)。

画像の大きさ(標準点数) = $(I_{sx}, I_{sy})_F$

ここで、 $I_{sx}(F)$: 水平方向の大きさ

$I_{sy}(F)$: 垂直方向の大きさ

濃度レベル数 = $N_{gr}(F)$

* 特に断らない限り、他の記号は文献[7]、[8]に従うものとする。

表1. 引数の分類結果

(a) 画像データ

濃淡画像(整数, 実数), 2値画像, 3値画像, 方向画像, 距離画像, ラベル画像, 半レシイ値画像, 確率ベクトル画像, 複素画像, その他.

(b) 中間形式

<1-D> 濃度ヒストグラム, 差分ヒストグラム, 断面, 投影, その他.

<2-D> 2次元ヒストグラム(Hough変換, 同時生起行列, ランレンジ行列), フーリエ変換像, パワースペクトル, パワー分布, フーリエ記述子, その他.

(c) 被演算子

幾何学的変換の係数, 階調変換テーブル, テンプレート, 重み係数, 点拡がり関数, 空間周波数フィルタ, その他.

(d) 演算領域の指定

窓領域, マスク領域,

(e) 要素属性表

輪郭点座標列, チェーン符号列, 曲率テーブル, 連結成分の特徴量(面積, 周囲長など), その他.

(f) スカラ・パラメータ

レシイ値, 繰り返し回数, 点の座標, テクスター特徴, 相関係数, その他.

(g) スイッチ

連結数指定, 断面をとる方向(X or Y)指定, ラプラシアンの種類指定, その他.

(h) 作業領域

反演演算において入力画像を保存するために必要な領域, ラベルの一覧表作成用, SIN/COSテーブル, その他.

(i) エラー・コード

実際, 入力画像から (I_{sx}, I_{sy}) , N_{gr} が決まると処理結果や中間結果の配列の大きさはこれに応じて決まることが多い. そこで,

←□F: 画像Fの大きさに依存する

←L F: 画像Fの濃度レベル数に依存する

←□F: 画像Fの大きさと濃度レベル数の両方に依存する.

という記法を用いる. (I_{sx}, I_{sy}) と N_{gr} は, 画像についての基本概念であり, これを媒介として他の引数を分類することにより, サブルーチン間の整合性や階層性を把握し易くなる.

2.3 機能的分類

既に, SPIDER仕様において判読性を高めるため, 引数名や入出力の別などが決められている(マニュアル^[9], pp. I-8 ~ I-9参照). これを一步進めて検討し, 最終的には表1の分類を得た. 以下に, 各項目の概念と意味を述べる(□内は関連主要サブルーチン名).

(a) 画像データ

入/出力, 正/負, 整/実数の別は問わず, 空間領域を標本化した2次元配列で, 処

理間で対応した画素の概念をもつものをいう. 従って, 同じ入力画像の処理結果でも, 境界抽出結果の線画[BDR41]やラベル画像[CLAB]は画像であるが, フーリエ・パワー分布像は原画像の画素との明確な対応がないので, 画像ではないと考える.

(b) 中間形式

画像データを処理して得られる(a)でない配列で, 画像特徴を保存したものをいう. 特徴量(スカラー)は, このデータをさらに変換処理して得られると考える. ヒストグラム(←L)[HIST1]や断面(←□)[CRSS]のように, 配列の大きさが入力画像データで決定されるものと, Hough変換面[ETDHI]のように別途パラメータ指定するものがある.

(c) (画像以外の)被演算子

画像演算の種類・機能により決定される被演算子で, (a), (b)以外の配列として表現されるものを考える. 配列の大きさは,

(i) 演算により決定され固定

(ii) 入力画像の (I_{sx}, I_{sy}) , N_{gr} に依存

(iii) ユーザの責任で指定

の3つに大別できる. 各々の代表例は, (i)

アフィン変換の係数[ACOE, AFIN1], (ii)濃度階調変換テーブル[HTBL1, GTRN1], (iii)マッピング用のテンプレート[SSCA1]である。

3×3近傍演算などは, Sobel [EGSB1], Kirsch [EGKS1]のように種類を決定することにより演算子としてプログラム内に埋め込まれるが, これを任意サイズに一般化する[FLWL1]と (iii)の級演算子として引数扱いになる。

(d) 演算領域の指定

入力画像の一部だけを処理するものとして, ウィンドウ処理とマスク処理(図1)を考えた。ウィンドウ(窓)は(c), マスクプレーンは(a)の一種とも考えられるが, 画像処理に特有の概念として別扱いにする。

(e) 要素属性表

画像内に含まれる要素(または成分)の属性を記憶する表で, その大きさは要素数で決定される。閉曲線の座標列[FECCL]のように各画素が要素となる場合や, 連結成分の面積[AREA1]のように領域が要素となる場合がある。いずれも, 画像データの内容に依存するため, 最大数を予測して大きめの表を確保する必要がある。扱いにくい対象である。

(f) スカラ・パラメータ

変数の値自体がアルゴリズムに重要な意味をもつ(いわゆる)パラメータである。しきい値[THDS1]やテクスチャー特徴量[COF]など, 種々のものがあるが 膨大な数になるので, 特にこれ以上は分類しない。

(g) スイッチ

スイッチなる引数(ISW, ITYPE他)によってサブルーチンの機能を適当に選択できる。その中には,

- (i) 別途 サブルーチン化するほどではないアルゴリズムの変型を選択するためのもの[AFIN1, BDFL1のNC],
- (ii) デフォルト・パラメータを何種類か選択するためのもの[EGLP, RXLN],
- (iii) 同じ入力データから算出する値(特徴量)の種類を選択するためのもの[COF, HGF], などがある。

ユーザの責任で選択すべきものであるが, 先行するサブルーチンに依存するものも多い。例えば, THDS2で計算したしきい値を使ってSLTH1でZ値化するときにはISW=1, 4に限定される。また連結数NCは首尾一貫させる必要のあることが多い。

(h) 作業領域

本質的には(a)~(e)に属するものでも, 入出力で意識する必要のないものを作業領域という。下位のサブルーチンでは意味のあるデータについて上位のサブルーチンではその内容に責任をもつ必要のないことがある。この場合, 上位のサブルーチンでは作業領域として扱われる[HGEQ1]。

(i) エラー・コード

マニュアル^[9]で述べた以上の議論は行ないない。

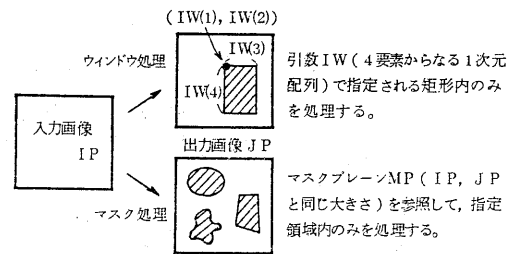


図1 ウィンドウ処理とマスク処理

3. データの表現形式

ここでは, 2. で分類した(a), (b), (c), (e)について, SPIDERサブルーチン群に現われる代表的なデータ型を整理する。これらはいずれも画像処理演算に特有な概念の表現であり, 専用言語における型宣言の候補となるものである。

例えば, 通常, 処理の対象となる原画像は, (時に実数演算を必要としない限り)非負の2次元整数配列として表現される。これに何らかの処理が施されると, 非負という制限がなくなる。一般的には, 逆にZ値化[SLTH1]とか領域分割LRMRG1といったアルゴリズムの種類によって, 出力のデータ形式は各々Z値画像とラベル画像に現

表2. 画像データの代表的な表現形式

画像データ	意味・内容	数値型	入力	出力
濃淡画像		整数 $[0, N_{gr}-1]$ 整数 実数	HIST1 ADDP1I ADDP1R	HGEQ1 ADDP1I ADDP1R
方向画像	gradient エッジ方向 k-曲率 $[k=1, 2, \dots]$ k-傾き $[k=1, 2, \dots]$	整数 $[0, 7]$ 実数 $(-\pi, \pi]$ 実数 $(-\pi, \pi]$ 実数 $(-\pi, \pi]$	ETOC1 — — —	EGRS2 EGPW2 CVK11 CVK11
距離画像	距離変換 $[N_c^{*2}=4, 8]$ 重み付き距離変換 $[N_c=4, 8]$ 輪郭点の重心からの距離 $[N_c=4, 8]$	整数 (≥ 0) " 実数 (≥ 0)	— — —	DIST DIST CVC1
ラベル画像	連結成分 $[N_c=4, 8]$ 領域分割	整数 $[0, N_r^{*3}]$ 整数 $[1, N_r^{*3}]$	CRCL1 AREA1	CLAB RMRGZ
24色画像	0: 背景, 1: 図形	2値 $\{0, 1\}$	CLAB	SLTH1
3値画像	0: 背景, 1: 図形(1), 2: 図形(2)	3値 $\{0, 1, 2\}$	—	BDFL1
半しきい値画像	0: 背景, 正整数: 図形	整数 $[0, N_{gr}-1]$	DIST	CLIP
ラベル確率画像	[ラベル番号]	確率値	RXLI	RXLN
複素画像	実部/虚部	実数	FFTAZ	FFTAZ

*1 入力画像の場合は予め決められていることが多い。階調変換処理の出力画像の場合は変換テーブルの型から決まる。 *2 連結性。 *3 データの成分数に依存。
*4 データの領域数に依存。

表3. 中間形式の代表的な表現形式

中間形式	意味・内容	数値型	配列の長さ	入力	出力
ヒストグラム	濃度 差分統計量[変位 (N_x, N_y)] Local extrema (水平/垂直) 一般[下限値, 区間幅, 区間数]	確率値 " " 度数(実数)	$N_{gr}(F)^{*1}$ $N_{gr}(F)$ $N_{gr}(F)$ $N \leftarrow \text{任意}$	THDS2 HGF LEXF NOHTIR	HIST1 DSTT1 LEXRZ STAS1
差分ヒストグラム	正差分/負差分	実数 (≥ 0)	$N_{gr}(F)$	—	DHST1
2-Dヒストグラム	Hough変換用 θ -P 同時生起行列[変位 (N_x, N_y)] ランレングス行列(4方向) 一般[下限値, 区間幅, 区間数]	重み付き度数 確率値 度数(整数) 度数(実数)	(N_θ, N_p) $(N_{gr}, N_{gr})_F$ $(N_{gr}, \max(x_{gr} , y_{gr}))_F$ (N_a, N_b)	— COF RLF —	ETDH1 COMX1 RUNL STAS2
断面	x方向/y方向	整数	$I_{xx}(F)/I_{yy}(F)$	—	CRSS
投影	"	"	"	—	PROJ1
フーリエ変換	実部/虚部	実数	$(I_{xx}, I_{yy})_F$	FFTAZ	FFTAZ
WH変換	Walsh/Hadamard ordering	"	"	WHT2	WHT2
Haar変換	"	"	"	HAR2	HAR2
離散的COS変換	"	"	"	DCF2	DCF2
Sinant変換	"	"	"	SLT2	SLT2
パワースペクトル	フーリエ変換 Walsh/Hadamard ordering	" "	" "	FLOG —	FPWC WWHP2
パワー分布	半径一再度 半径方向 再度方向	" " "	(N_a, N_b) N_a N_b	— — —	FPWP1 FPWP2 FPWP3
閉曲線のフーリエ展開	"	"	N_f	—	FECC
自己回帰モデルの重み係数	"	"	$(2M+1, 2N+1)^{*2}$	TXLE3	TXLE2

*1 入力濃淡画像。 *2 M, N は TXLE1 で決められる値。

表4. 被演算子の代表的な表現形式

被演算子	意味・内容	数値型	配列の大きさ	入力	出力
階調変換テーブル	等化/等高線/一般	整数 $[0, N_{grn}-1]$	$N_{gr}(F)$	GTRN1	HTBL1
テンプレート		整数	(I_{tx}, I_{ty})	SSCA1	——
重み係数		〃	(I_{wx}, I_{wy})	FLWL1	——
点拡がり関数		実数	(I_a, I_b)	CLSF	——
		〃	$(I_{sx}, I_{sy})_F$	WIEN	——
アフィン変換係数		〃	6	AFIN1	ACOE
2次元非線形変換係数		〃	12	NOLN1	——
空間周波数フィルタ		〃	$(I_{sx}, I_{sy})_F$	MULPIR	FQWS1

* 任意に設定可能。

表5. 要素属性表の代表的な表現形式

要素属性表	意味・内容	数値型	配列の大きさ	入力	出力
連結成分の特徴量	面積	整数 (>0)	$N_r(L)^{*1}$	RGED1	AREA1
	重心	〃	$(Z, N_r(L))$	CVC1	CGRV1
	周囲長 $[N_c=4, 8]$	〃	$N_r(L)$	RGED1	PRMT1
	サイズ $[N_c=4, 8]$	実数	〃	——	SIZE1
	円形度 $[N_c=4, 8]$	〃	〃	——	CRCL1
	モメント (重心/主軸回り) [次数]	〃	〃	——	MMNT1
	始点座標	整数 (>0)	$(Z, N_r(L))$	AREA2	STPT1
	外接長方形の座標	〃	$(4, N_r(L))$	AREA3	CQLT1
	座標列 (始点座標) $[N_c=4, 8]$	〃	(Z, N_p^{*2})	COCD	BDFL1
	〃 〃 〃	実数 (>0)	〃	FECC	——
輪郭点の特徴量	〃 (全図形) 〃	整数 (≥ 0)	〃	——	BDFL2
	〃 (ラベル指定) 〃	整数 (>0)	$(Z, M_{xp}^{*3}, N_r(L))^{*4}$	CVK3	BFLT1
	チェイン符号列 $[N_c=4, 8]$	整数 $[1, 8]$	$(M_{xo}, N_r(L))^{*4}$	CVK3	BFLT1
	重心からの距離値	実数 (≥ 0)	〃	——	CVC2
	k-曲率 $[k=1, 2, \dots]$	実数 $(-\pi, \pi]$	〃	——	CVK21
	k-傾き $[k=1, 2, \dots]$	〃	〃	——	CVK21

*1 入力ラベル画像の連結成分数, *2 テンタの輪郭点数に依存。
 *3 テンタの連結成分の最大輪郭点数に依存。
 *4 補助的な情報として右連結成分の輪郭点数を示す表が付属する。

定されることも多い。

分析の結果は、表2～表5に示した。右表では、各々のデータ型式が意味・内容、各配列要素のとり数値の型と範囲、配列の大きさと入力画像への依存性（表2については、すべて「□」なので省略した）、入力・出力としてこのデータ型式を必要とする代表的なサブルーチン名が記されている。

画像処理向きに列挙したこれらのデータ型には、この表には明確に述べられていない拘束条件や意味が含まれている。例えば、ラベル画像は非負の整数値2次元配列の一種で

あるが、(i)右連結成分内の全画素に同じ番号(ラベル)が割り当てられ、(ii)連結していない領域に同じ番号が付けられることはない、といった条件を暗黙のうちにもっている。また、ヒストグラムに関して、確率値を要素にもつもの（正規化ヒストグラム）では、全配列要素の値の和は1となるはずである。

2.でも再出たように、これらのデータ型の大部分は、扱う画像の (I_{sx}, I_{sy}) と N_{gr} に依存した大きさの配列である。表2～表5で、特にこの依存性を明記していないものは、ユーザの責任で決定すべきものである。

画像処理サブルーチン間の整合性については、こうしたデータ型とその大きさの制限を照合すればよい。データ搬送の事例については、6. で述べる。

表2～表5に示したデータ型式は、現在のSPIDERから代表的なものを列挙したにすぎないので、画像処理に必要なものを全て網羅した訳ではない。また、これらのデータ型式の中にも、階層性や相互関連がある。これを再整理するのは次の課題である。その際には、SPIDERでは取り扱わなかったデータ構造も含めて検討する必要がある。即ち、時系列画像やマルチスペクトル画像等の多次元画像、ピラミッドやquad-treeなどの階層的データ構造の表現形式も重要である。

4. 画像処理アルゴリズムの形態

従来の画像処理アルゴリズム論^{[10]~[12]}は、ほとんどが並列型の近傍演算を中心としたもので、その他の実行形式については、ほとんど議論がなされていない。ここでは、より一般的な立場から画像処理アルゴリズムの形態を眺め、特に逐次型について考察を試みる。

4.1 アルゴリズムの実行形式

(1) タイプ1：点演算 vs 近傍演算

文献[8]の定義に従う。(特に断らない限り、以下同様)。大局処理の定義は明確ではないが、フーリエ変換[FFTAZ]やアフィン変換[AFINI]のように、一定の近傍の範囲にとどまらない画素をアクセスするものを指す。

(2) タイプ2：並列型 vs 逐次型

概念的には、まず理想的な並列型があり、その他のものを逐次型とする。Hilditchの細線化法[THNG1]は、点削除の条件に並列型と逐次型を併用しているの、全体としては逐次型と呼ぶべきである。

(3) タイプ3：一様走査 vs 非一様走査

予め定められた順序に従って、ある特定の範囲内の全点を処理する方式を一様走査、入力画像や他の条件によって走査順序が変化するものを非一様走査と呼ぶ。

表6. 画像処理アルゴリズムの形態

タイプ1	タイプ2	タイプ3	代表例
点	並列	一様	単純な濃度階調変換[HGEQ1, LDG], 2値化[SLTH1, CLIP], 画像内演算[ADDP11, SUBP11]等多数。
	逐次	一様	同一濃度の細分化を行なうヒストグラム変換[HGEQ2, HGHYZ]
		非一様	SPIDERに該当例なし。
近傍	並列	一様	微分処理[EGLP, EGPW1], 平滑化[EGPR, ITEN1], 弛緩法[RXLN, RXEG]等多数
	逐次	一様	ヒステリシス・スムージング[HYSM1, HYSM2] 連結成分ラベル付け[CCLAB], 領域分割[RMRG1, RMRG2]
		非一様	ヒューストック・サーナ[EFMA] 画像の重ね合せ[SSCAZ, CORR2]

一様走査で最も一般的なのは、水平ラスタ走査であるが、予め定まっている、らせん線状等でもよい。ラスタと逆ラスタの切替えや、水平・垂直ラスタの切替えも、予め定まっている一様である。前述のウィンドウ処理やマスク処理も一様走査の一変形と考える。

並列演算は、本質的には走査の概念をもたないが、訂算機等での実行上は一様走査で実現される。結局、逐次型処理にのみ、一様と非一様の区別があり、これが文献[8](b)式の g_{ij} と g_{ij+1} を決定することになる。

{タイプ1}×{タイプ2}×{タイプ3}の組合せについて、SPIDERでの代表例を表6に示した。通常、点演算はすべて並列型と考えられていることが多いが、逐次型(一種の履歴処理)の点演算も存在することに注意されたい。

(4) 付加形式：反復演算と分割処理

各々、タイプ1～3の演算形式と独立に設定することができる。

4.2 実行上の諸性質

ここでは、走査順序、演算回数、処理結果等が前述の演算形式によって受ける影響、メモリ節約等について論じる。

(1) 始点・回帰依存性

非一様走査(i.e. 逐次型)では、処理の開始

点の選び方によって、実行上違いがあることが多い。これを始点依存、そうでないとき始点独立という。2値画像の境界追跡[BDFL1]は、走査順序(次に選ばれる点)は始点依存であるが、演算回数と処理結果は本質的に始点独立である。一オ、SSDA法ではすべてが始点依存である。

ラスタ型一様走査の始点と方向の選択は、入力画像を90°の倍数分回転したに等しい。(厳密には鏡像関係を含む)。従ってこの場合の諸性質の依存性を回転依存/回転独立と呼ぶ。並列処理の演算回数は常に回転独立であるが、近傍演算で条件が対称でない場合には処理結果が回転依存となる。非対称性の強いSRNK1や反復ごとに条件を変えるTHNG3がその例である。逐次処理ではむしろ回転依存の方が普通であるが、最大値/最小値フィルタ[FMAX, FMIN]や連結成分のラベルづけ[CLAB]では、処理結果は(ラベルの値を除いて)回転独立である。

(2) データ依存性

処理結果がデータ依存であるのはいくつでもない。走査順序は非一様処理なら明らかにデータ依存である。従って、データ依存性(文献[8], p.14)は演算回数について論じる。2.及び3.で述べた要素属性表に関連する処理は一様/非一様にかかわらずデータ依存である。その他では、一様走査の演算回数(処理時間)はデータ独立であることが多いが、文献[8]での指摘通り、収束条件をもつ反復演算はデータ依存となる。

(3) 画像配列の節約について

プログラム内の画像用配列や専用プロセッサが扱うフレームメモリを節約するために、入出力配列を兼用したいことがある。一般には、点演算では兼用可能、並列近傍演算では不可能である。逐次近傍演算は、むしろこの入出力配列兼用のため考案されたとも言えるが、前述のTHNG1のように一様走査途中に並列条件を含むと兼用不可能となる。

反復演算を含む場合については、マニュアル[9]の4.3節に詳述したので省略する。

4.3 逐次型処理の意義

並列処理、特に並列型近傍演算は画像処理の典型的演算であるため、高速プロセッサ設計もほとんどこれに照準を合せている。一オ、逐次型処理は、上記の検討でも分るように、種々複雑な性質を生む原因であり、現在の専用プロセッサではほとんど高速処理することができない。

しかし、ハードウェアの制約から離れて開発されたSPIDERでは、色々凝ったアイデアを駆使した逐次型処理を見つけ出すことができる。以下で、逐次型処理が採用される意味を考察する。

(a) 履歴の利用——逐次型処理の本質ともいうべき過去の処理結果を積極的に利用するもので、ヒステリシス平滑化[HGHY1]や濃度レベル細分化を行なう階調変換[GTNRN2]がこれに相当する。また、履歴自体に本質的意味はないが、処理過程の中間記憶(ヒストグラムなど)を利用して高速化を図っているもの[FMAX, MEDI]もある。

(b) 追跡・探索型——既に見つかっている点を利用して線追跡を助けるもの[BDFL1, EFMA](履歴利用の一種)や、計算時間の節約のために探索の候補を絞る目的のもの[SSCA1]。

(c) 成長・統合型——発想としては並列型であるが、インプリメントのし易さから逐次型を選んでいるもの[RMRG1, 2]。

(d) 消滅の防止——成長とは逆の発想で領域を削って行くとき、並列型では連結成分が消滅するので、これを防ぐ。縮退化[SRNK4]や細線化[THNG1]で利用されている。一オ、SRNK2やTHNG4は反復をサブサイクルに分けることにより、並列型を保持している。

(e) 要素依存性の解消——連結成分のラベルづけ[CLAB]は、並列型で実行すると成長型の演算となり成分の数に依存するため、2回の一様走査でこれを解決している。距離変換[DIST]では、並列型では距離の最大値だけの反復を要するのを、巧みな逐次処理により解決している。

(f) 画像配列の節約——4.2の(3)で触

れた使用法、上で述べたいくつかの方法は同時にこれらも達成している[FLWL2, FMAX, SRNK4].

5. 高速プロセッサとアルゴリズムの適合性

この章では、並列型アルゴリズムを中心に、画像処理用高速プロセッサとの適合性を検討する。

典型的な高速プロセッサとして、次の3つを考える。

- ① CLIP型——CLIP4^[13]のように画素に対応する要素プロセッサが同時に近傍演算を実行する完全並列型。
- ② PPP型——東芝開発のPPP^[14]のように、各画素に対する近傍演算を並列処理できる基本モジュールをいくつかもち、画面の一樣走査をパイプライン処理で高速化したもの。
- ③ AP型——いわゆるスーパーコンピュータやレイプロセッサのように、1次元ベクトル演算をパイプライン処理できるもの。

ただし、ここでいうCLIP、PPPは現存のプロセッサではなく、細部の点には理想化した代表名として使う。以下、SPIDERサブルーチン機能を代替えさせる立場から基本的問題を考察する。

5.1 一樣走査と必要モジュール

一般に、並列型近傍演算は、FORTRANプログラムの形式では、

```
DO 100 J=1,ISY
  DO 200 I=1,ISX
    :
    JP(I,J)= { IP(I,J) に関する }
              { 近傍処理 :  $\varphi_{ij}$  }
    :
  200 CONTINUE
100 CONTINUE
```

なる一回の一樣ラスタ走査で記述できる。プログラムでは、 $\{ \}$ の中にいかようにも複雑な演算を書くことができ、条件による分析も可能である。右辺にJPの要素を入れると、逐次型処理も実現できる。

CLIP型では、走査という概念がないため、この2重のDOループを意識することなく、 $\{ \}$ を要素プロセッサの単位演算機能に分解して

記述することになる。条件分枝は、通常マスクブレンを用いて行なう。プログラミング的には、PAX II^[15]に似た形式となる。

PPP型では、1つの基本モジュールの実行が1回の一樣走査に対応する。 $\{ \varphi_{ij} \}$ が複雑な場合、これを基本モジュールに分解するが、分解された処理結果の合成は2重DOループの外で計算することに等価である。3×3の積和演算やメディアンフィルタ等、有かなモジュールを具備すれば、プログラミング側からは大巾に軽減されるが、反面、用意されていない演算には全く対応能力がないので、必要なモジュールや扱えるデータ型式の選択には慎重を要する。

SPIDERプログラムは制約なく書かれているため、比較的単純なエッジ検出を例にとっても、tan⁻¹計算[EGSB2]、最大出力を与える方向の選択[EGPW4]、局所連結性の検定[EGRS3]等が含まれていて、CLIP型やPPP型への適合は容易でない。

また、CLIP型とPPP型は、完全並列と局所並列の差こそあれ、結局、画像内の全点を処理している。SPIDERプログラムでは、多数の近傍マスクを用いる並列処理[THNG3, SRNK3等]において、適当な条件判定で不必要なマスクの適用を回避している。複雑な近傍演算では、専用プロセッサで実行しても、基本演算の比率ほどは高速化できないことに注意する必要がある。

5.2 1次元分解可能性

AP型プロセッサでの高速処理は、上記の $\{ \varphi_{ij} \}$ がIについての近傍からなる時にのみ達成できる。即ち、変数Jについての因子を内側のDOループから排除することに相当し、これを1次元分解(文献[8], p.17参照)という。

SPIDERサブルーチンの分析により、次の検出結果が得られた。

- (a) 離散的フーリエ変換をはじめとする一連の2次元直交変換[FFTS2, WHT2, HAR2等]はいずれも原理的に1次元分解可能で、それを用いた1次元データ用のサブルーチンを用意している。
- (b) ある種の近傍演算[FMAX, FLWL2等]は、

1次元分解可能であり、これらをさらに逐次型処理化し、高速化を図っている。

(c) 1次元への多段分解を行なわない場合でも、 $m \times n$ 近傍を用いる演算では、 n 行分のバッファをもてば1次元処理に帰着できる。通常、AP型プロセッサの利用はこの方式を用いることが多い。

(d) ヒステリシス平滑化は本質的に2次元版の定義が適当でないため、便宜上、1次元処理へ分解している。しかし、前述のように回転依存である。また、メディアン・フィルタも、回転依存である1次元分解版を定義できるが、こちらのオは本質的な2次元版 [MEDI] が存在する。

6. 画像処理アルゴリズムの構造分析

SPIDER サブルーティンをいくつか組合せることによって、種々の画像処理手順を実現できる。また、いくつかのサブルーティンでは既に、内部に下位ルーティンをもって階層性をなしている。こうしたルーティン間では、2. 及び 3. で分類・整理したデータが授受されている。画像処理アルゴリズム構造を把握する一助として、図2の記法でサブルーティン間の相互関係を図示することを試みた。

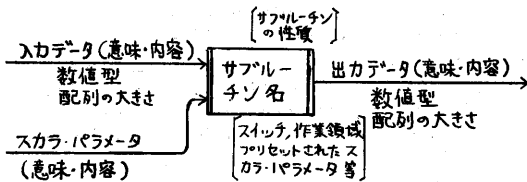


図2. 各モジュールの記法

また、配列要素の数値型や配列の大きさの依存関係と対応関係は、それぞれ $\rightarrow$, $\leftarrow$ で示した。

例1: ヒストグラム平坦化 HGEQ1

ヒストグラム平坦化を行なう HGEQ1 の構造を図3に示した。この例から分かる特長は次の通りである。

(a) 最高レベル HGEQ1 をいくつかの内部

ルーティンが階層的に支えている。

(b) 入力画像 F 、出力画像 $G \leftarrow \square$ 、指定出力レベル数 N_{grn} 以外は、各ルーティン間で自動的に必要な引数が授受されている。

(c) 良いモジュール性が保たれているので、HGEQ1L を HGHY1L に置換するだけで、ヒストグラム双曲線化 HGHY1 が実現できる。

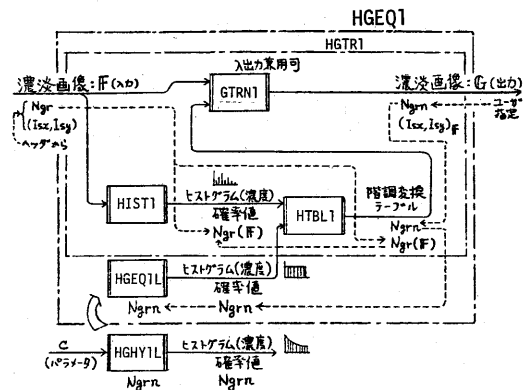


図3. HGEQ1 の構造

例2: SSDA法による画像のマッチングSSCA1

画像の高速マッチングを行なう SSDA 法の中、一定しきい値を用いるサブルーティン SSCA1 とその内部ルーティン SSCMO の構造を図4に示した。

一方、傾斜しきい値を用いる SSIA1 とその内部ルーティン SSIMO の関係もほとんど同じであるが、同図から分かるように、SSCMO の部分に SSIMO を代入することはできない。なぜならば、傾斜しきい値の決定に必要なパラメータ T_s, K, AL を与える端子が SSCA1 側になく、整合がとれないからである。

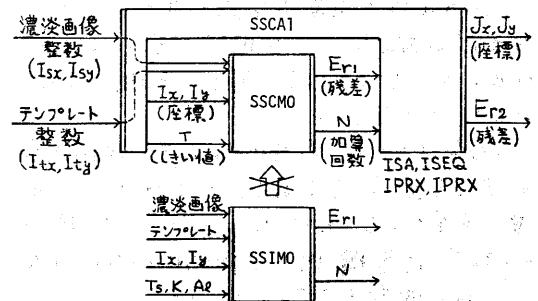


図4. SSCA1 の構造

例3: 粒子像計測の処理例

次の画像解析手順をSPIDERサブルーチンの結合で実現した例の構造を図5に示す。

- ① 背景と粒子物体からなる濃淡画像のしきい値を決める。
- ② 自動決定されたしきい値で原画像を2値化する。
- ③ 2値図形の連結成分をラベルづける。
- ④ 面積がIAREAより大きい成分のみを抽出し、各粒子の内形度を計測する。

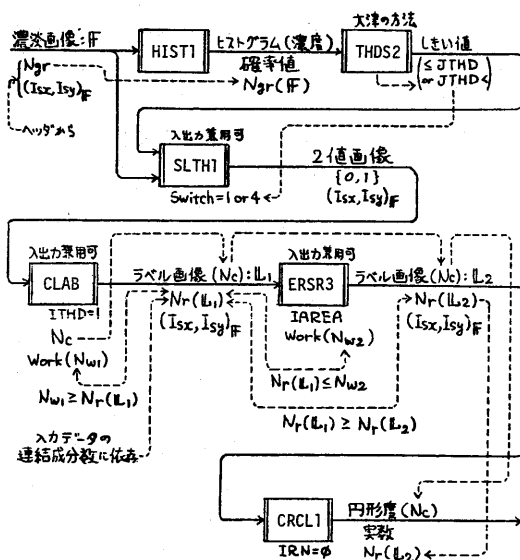


図5. 粒子像計測手順の構造

7. おすび

画像処理サブルーチン・パッケージSPIDERのアルゴリズム及びデータ構造を分析した。SPIDERはFORTRANプログラムの集合であるが、その下位レベルは専用高速プロセッサに、上位レベルは専用高級言語に置き換えて行くべきだと考える。この分析・検討は、これらプロセッサや言語設計の重要な基礎になるものと思われる。

本稿はその第一段階であり、まだ分析内容が消化不良の気味もある。3. 及び4. を再整理、発展させることにより、6. に示した構造分析が、計算機内で自動的に行なわれるシステムの開発を目指している。

謝辞

最後に、本研究に関して有益な御助言を頂いた国形処理研究室長・舟久保 登氏に感謝致します。

参考文献

- [1] 坂上, 不出版, "画像処理プロセッサの最近の動向", 情報会研報, CV18-4, May 1982.
- [2] 田村, 坂上, "画像処理ソフトウェアの動向 — 標準化と高級言語の開発", TV学会技報, LPA 59-3, Oct. 1982.
- [3] M.J.B. Duff & S. Levialdi (Eds.), *Language and Architectures for Image Processing*, Academic Press, 1981.
- [4] K. Preston, Jr. & L. Uhr, *Multicomputers and Image Processing: Algorithms and Programs*, Academic Press, 1982.
- [5] 田村, 坂根, 富田, 橋本, 金子, 坂上, "ホータブル画像処理ソフトウェア・パッケージSPIDERの開発", 情報会論議, Vol. 23, No. 3, pp. 321-328, May 1982.
- [6] 特集「画像処理アルゴリズム」, 電総研会報, Vol. 44, No. 7/8, 1980.
- [7] 鳥脇, 横井, "画像処理のアルゴリズム", 情報処理, Vol. 21, No. 6, pp. 613-619, July 1980.
- [8] 鳥脇, 横井, "実行形式面から見た画像処理アルゴリズム", 信学技報, PRL81-61, Dec. 1981.
- [9] 画像処理サブルーチン・パッケージSPIDER Users Manual, 国同システム開発(株), 1982.
- [10] M.J.B. Duff, "The elements of digital picture processing," in *Real Time/Parallel Computing - Image Analysis* (M. Onoe et al. Eds.) Plenum Press, pp. 1-9, 1981.
- [11] R.M. Haralick, "Some neighborhood operators," *ibid.*, pp. 11-35, 1981.
- [12] S. Levialdi et al., "Theoretical and practical considerations in the application of neighborhood logic to image processing," in *Proc. 4IJCPR*, pp. 139-145, 1978.
- [13] M.J.B. Duff, "CLIP4: A large scale integrated circuit array parallel processor," in *Proc. 3IJCPR*, pp. 728-733, 1976.
- [14] 木出他, "並列画像演算装置PPP", 東芝レビュー, vol. 34, no. 6, pp. 511-514, June 1979.
- [15] E.G. Johnston, "The PAX II picture processing system," in *Picture Processing and Psychopictorics* (B.S. Lipkin & A. Rosenfeld Eds.) Academic Press, pp. 427-512, 1970.

