

# 図形特徴量計算のための1パスビデオレートアルゴリズム

依田晴夫

日立製作所 中央研究所

ラスタ走査で入力される画像信号をビデオレートのまま実時間処理し、個々の図形の形状特徴量を計測するハードウェア向き計算手法を提案する。この手法は、①1本のラスタ上の図形要素の連結状況を属性記号列で表現する手法、②任意のラスタの属性記号列を逐次的に自動生成する手法、③属性記号列を参照して各図形ごとの特徴量を累積計算する手法、とから成る。本手法によれば、どのように錯綜して配置された複雑な図形群に対しても、正確かつ完全な結果が得られる。

## An Algorithm for Determining Shape Descriptors based on Video-rate One-pass Image Processing Techniques

Haruo YODA

Central Research Laboratory, Hitachi Ltd.,  
1-280, Higashi-Koigakubo, Kokubunji-shi,  
Tokyo 185, JAPAN

A new one-pass, raster-scan-type algorithm for determining shape descriptors of figures is proposed for video-rate, real-time image processing. The algorithm assigns an attribute symbol to each figure intercept in every scan-line of the image for representing figure connectivity. Shape descriptors are calculated based on the assigned symbols and are output at every instant the scanning of each figure is completed. This algorithm can perfectly handle any intricate arrangements of complicated figures, and can minimize the capacity of buffer memory needed for calculation, thus facilitating the realization of a high-speed image processor for effective feature extraction.

## 1. 緒 言

1枚の画像中に多数の図形が含まれている時、その1つ1つの図形の形状的特徴量（例えば、面積、周長、投影長など、以下形状特徴量と記述）を、ラスタ走査画像からビデオレートを保持したまま、1パス方式で逐次計測していく実時間画像処理手法を開発した。従来、このような実時間計算手法としては、特殊な巡回型フィルタを用いて図形を右下側に膨張変形させ、図形の特徴量をその変形に沿って右下端の画素に累積させていく手法<sup>1,2</sup>が知られている。しかし、この手法には、変形によって2つ以上の図形が接触してしまうと、それを誤って1個の図形とみなしてしまう欠点があった。また、図形の変形を必要としない手法としては、実時間処理によって生成されたコマンド列をFIFO型のバッファメモリに蓄積し、そのコマンド列を後段のマイクロプロセッサによって処理する手法がある<sup>2,3</sup>。しかし、図形の部分特徴量を合成する複雑な処理をソフトウェアで行うために、図形の数が多くなったとき、真に実時間性が保証されるかは明確でなかった。筆者は、先に、図形変形を最小にして計測の信頼度を向上させる実用的な実時間1パス手法を開発し報告した<sup>3,4,5</sup>が、この手法であっても誤りを完全に防ぐことは出来なかった。今回報告する手法は、図形がどのように入り組んでいても、ビデオレートで正確に特徴量を計算できる。

本論文では、まず、処理中のラスタ上の各交差点（ラスタと図形とが交差している部分）が、そのラスタ以前の図形によってどの様に連結されているかを、各々の交差点に1つの属性を割り当てることによって表現する手法を提案する。これにより、各ラスタ上の交差点の連結状態は、交差点と同数の属性記号からなる一次元配列（以後、単に属性記号列という）で表現される。次に、この属性記号列を、ラスタ走査に従って逐次的に自動生成する手法を述べる。最後に、この属性記号列の生成過程を参照しながら、図形の特徴量を図形ごとに累積していく手法を述べる。

## 2. 交差点の連結状態記述法

1本のラスタ走査線上の交差点の連結状態を表現するために、図1のように、各交差点に4種の属性（L, M, R, S）の1つを割り当てる。すなわち、同一図形を構成する交差点が2つ以上の場合には、その1番左（ラスタ走査の始点方向）のものに“L”、中間のものに“M”、一番右のものに“R”を割り当て、交差点が1つしかない場合には“L”と“R”の縮退形の“S”を割り当てる。この記述法によれば、各ラスタ上の交差点の連結状況は唯一の属性記号列で表わせ、また逆に属性記号列から交差点の連結状況も一意に知ることができる。

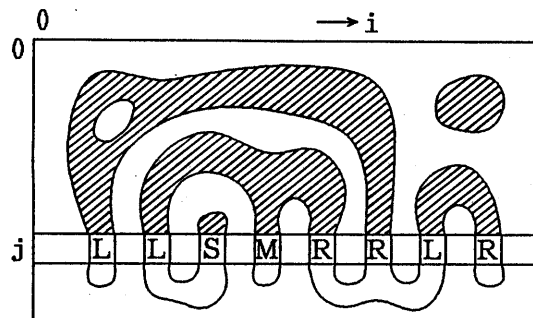


図1. 図形交差点連結状態の記述例

この記述法の最も大きな利点は、属性記号列でスタックのプッシュ/ポップを制御すれば、容易に図形単位の処理を行うことが出来ることである。すなわち、交差点の属性をラスタ走査順に調べてゆき、①交差点が“L”ならば、それは新しい図形の始まりを意味するので、スタックをプッシュしてその図形の“値”を初期化し、②“M”ならば、スタックのトップレベルにあるその図形の値を更新し、③“R”ならば、処理中の図形がこの交差点で終了するのでその図形の値を外部に出力するとともに、スタックをポップする。また、④“S”ならば、“L”と“R”の処理を一度に実行する。このようにすれば、1つの図形の処理の合間にたまたま他の図形が入り子で割り込んだとしても、図形の値はスタックにプッシュされて保存されるので、図形単位の処理を誤ることはない。

しかしながら、一般に第 $j$ ラスタの交差点の処理を行うとき、第 $j$ ラスタの属性記号列は未確定であって、それをスタック制御に用いることは出来ない。そのため、本手法では、第 $j$ ラスタの処理のために、既に求まっている第 $j-1$ ラスタ上の属性記号列を用いる。第 $j$ ラスタ上の図形の値を、第 $j-1$ ラスタの属性記号列でスタック管理できる理由は、第 $j$ ラスタ上で同一の図形を構成する交差点群が、（新たに発生する孤立交差点“S”を除くと）必ず第 $j-1$ ラスタ上の同一図形の交差点群に連結して生成される性質があるからである。

## 3. 属性記号列の自動生成

3. 1. 自動生成のための制御信号 ラスタ走査形式で属性記号列を自動生成するためのタイミング信号は、入力中の画素の値  $f(i, j)$  とちょうどその1ラスタ前の隣接画素の値  $f(i, j-1)$  とから作られる数:  $q(i, j) = 2^i \cdot f(i, j-1) + 2^j \cdot f(i, j)$  を入力信号とする順序回路によって生成される。図2はその状態遷移図を示したもので、丸は内部状態、丸から出る矢印はその根元の数値が  $q$  の値として入力されたときに遷移する方向を示す。この順序回路は、 $S_0$  を初期状態として、入力  $q(i, j)$  の時間系列に従い6個の内部状態間を次々と遷移し、その遷移時に矢印上の箱内を示す制御信号を発生する。図3は発生される制御信号の様子を、対応する図形パターンの上で示したものである。図3では表現を簡単にするために、第  $j$  ラスタ、第  $j-1$  ラスタ上の交差部だけを矩形で示した。

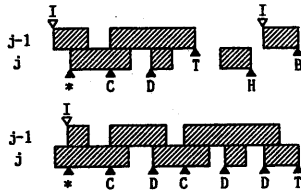


図3. 入力図形に従って生成された制御信号

発生する制御信号は I (Initiation), \* (continue), C (Connection), D (Division), T (Termination), B (Bottom), H (Head) の7種類である。これらの制御信号は、図3からもわかるように、I は最初に第  $j-1$  の交差部が発生した時点、\* は交差部の継続が検知された時点、C は新しい交差部との連結が生じた時点、D は分岐が生じた時点、T は連結交差部群が終了した時点、B は第  $j-1$  ラスタの交差部が継続すること無く終端した時点、H は第  $j$  ラスタに新しい図形の交差部が生成した時点をそれぞれ示す。この各々の時点において、後述の対応処理が実行される。

なお、本手法においては4連結で定義された図形を対象としたが、図2に示した順序回路を少し修正すれば、容易に8連結の図形を対象とするように変更することが出来る。

3. 2. 属性記号自動生成の課題 属性記号の生成は、まず、図形単位パラメータ (前述の図形ごとの“値”の1種で生成処理の制御に使うもの) として交差部数カウンタ ( $I_c$ ) を設けて図形ごとの交差部数を計数し、最初の交差部に L を割当て、それ以降の交差部に M を割当てる。また、別の図形単位パラメータとして最終交差部ポイント ( $R_p$ ) を設けて一番最後に書き込んだ属性の書き込み番地 (第  $j$  ラスタ上の交差部番号) を記憶するようにしておき、第  $j-1$  ラスタ上の R, S 交差部の処理が終了した時点 (T) で、その図形の1番遅く割当てた第  $j$  ラスタ上の交差部の属性を R ( $I_c = 1$  の時は S) に書き替える。このようにすれば、常に図形毎の最左端交差部が L に、また最右端交差部が R に割り当てられることになる。

この割り当て処理を行う上で最も問題となるのは、交差部に連結が生じた時の処理方法である。連結時の処理は、

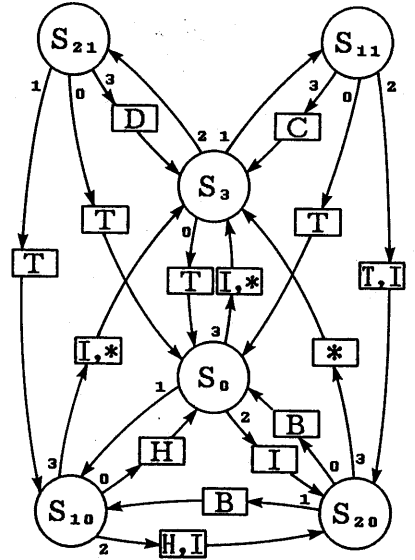
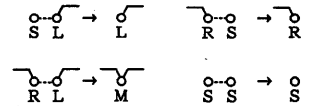


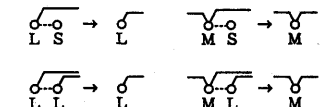
図2. 制御信号生成のための状態遷移図



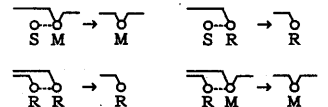
(a) Connection of same figure intercepts



(b) Connection of adjacent figures



(c) Left-side connection of stacked figures



(d) right-side connection of stacked figures

図4. 図形の連結形式

連結される交差部の属性に依存して多数の場合がありえて、単純でない。連結の状況は図4のように属性の組合せによって4種類16通りが存在する。ここでは、その各々について次のような処理方法を開発した。

(a) は同一図形を構成する交差部同士の連結である。この場合には、連結の発生時点(C)で図(a)のように第j-1ラスタ上の2つの交差部の属性を1つ変換し、あたかもその2つの交差部がもともと1つの交差部であったかのように扱う。これにより、連結は無かったものとなり、矛盾のない処理が実行できる。

(b) はラスタ上で横に並んだ2つの図形の交差部の連結である。この場合にも図(b)のように2つの交差部の属性を1つの属性に変換するだけで良い。このようにすると、2つの図形がもともと1つの図形であったことになり、矛盾のない処理が実行できる。

(c) は、入り子になった2つの図形がその左部において連結した場合である。この場合、連結する交差部の属性がLとSあるいはMとSならば、Sは孤立した交差部なので属性ラベルを各々単にL、Mに替え、Sが無かったものとして扱うだけで良い。LとL、MとLの場合には連結交差部の変更とともに、連結される図形のもう一端の交差部の属性Rの変更も必要になる。そこで、第5図①～③に示すように、縮退数 $R_n$ を初期値=0の新しい図形単位パラメータとして持たせ、

(i) LとL(またはMとL)の連結が生じたことを検知すると、読み込まれた新しいLをMと解釈し、L(またはM)に変換すると共に $R_n$ を+1だけインクリメントする、(ii)そして、図5④～⑥のように、R属性が読み込まれた時に $R_n$ が正の数であったならば、 $R_n$ を1だけデクリメントするとともに読み込まれたRを強制的にMと解釈するようにする。このようにすれば、連結される内側図形のL交差部はR交差部と必ず1対1で対応しているの、何重に左部重なり形連結があったとしても、もともと1つの図形があったかのように扱われ、矛盾なく処理されることになる。

(d) は、入り子になった2つの図形がその右部において連結している場合である。右部連結の場合には、左部連結でもある場合があり得るが、前述の左部連結処理を実施すれば、内側図形のR交差部がM交差部に書き換えられるので、処理の上でそのようなことは起り得ない。また、連結交差部の属性がLとSあるいはMとSならば、左部連結の場合と同様に、Sを無視するだけで良い。したがって、解決すべき状態は、RとR、RとMの形の右部重なり型連結であって、しかも $R_n=0$ 、すなわち左部連結のない場合のみである。本手法では、図6に示すように、各図形のL交差部の番地(第jラスタ上の交差部番号)を新たな図形単位パ

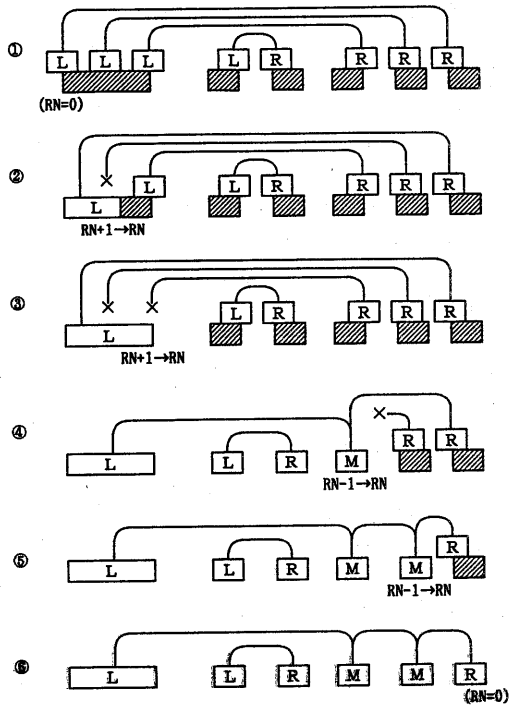


図5. 左部重なり型連結の処理例

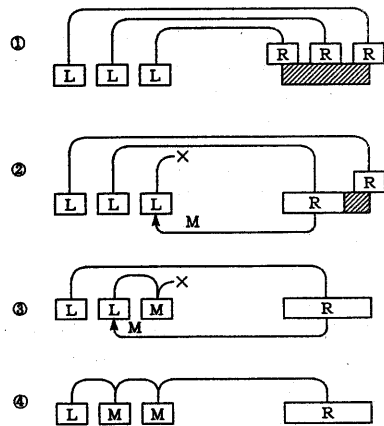


図6. 右部重なり型連結の処理例

ラメータ(Lp)として記憶しておき、右部重なり形連結が検知されたら、連結されている内側の図形のR交差部を除去すると共に、Lpを用いて内側図形のL交差部をM属性に変更するようにする。ただし、特殊な場合として外側図形のIcが“0”の場合には、内側図形のL交差部が連結図形の先頭交差部になってしまうので、LPを用いた交差部属性の書き替えは防止する。このようにすれば、何重に右部重なり形連結があったとしても、矛盾無く処理できることになる。

このようにすれば、どんなに複雑な連結が生じて、矛盾なく連結処理を実行することができる。これに必要な図形単位パラメータは、Ic, Rp, Rn, Lpの4種である。

3. 3. 属性記号の自動生成手順 前述の基本的な考え方に従えば、具体的な生成手順を構成することができる。その生成手順は順序回路から発生される前述の7種の制御信号に同期して実行される。表1はその詳細な手順を示したものである。表1は、各制御信号(CNT1)

毎に条件部と実行部を示したものである。ただし、ハードウェアでの実現を考慮して、前述の図形単位パラメータのスタックは、トップレベル値を記憶するレジスタ(WR1)と次レベル以降の値を記憶するスタック(STK-P)とに分割して構成されている。また、パラメータIcの2以上の値は2に固定している。表1において、WR2は現在属性を保持する内部レジスタ、F#1は1ラスタ前の属性記号列を記憶する入力側のメモリ、F#2は注目ラスタの属性記号列を記憶する出力側のメモリ、n<sub>1</sub>, n<sub>2</sub>はそれぞれそのアドレスポインタである。この表の処理は、概略、次の通りである。

(1) Iの処理：第j-1ラスタの新しい交差部の属性(以下、入力属性という)がSまたはLならば、スタックをプッシュし、図形単位パラメータ(Ic, Rn, Lp, Rp)を初期化する。現在属性は、前述の左部重なり型連結の場合を除き、入力属性をそのまま設定する。

(2) \*の処理：現在図形の図形単位パラメータを1交差部分だけ更新するとともに、新しい第jラスタ上の交差部の属性をIcパラメータによって決定し、出力する。

(3) Dの処理：図形単位パラメータを1交差部分更新するとともに、新しい第jラスタ上の交差部の属性をMとして出力する。

(4) Cの処理：現在属性と入力属性とから連結形式を識別し、①同一図形内連結、②横連結、③左部重なり形連結、④右部重なり形連結、のそれぞれについて、前述の連結処理を行う。現在属性は、連結によって新しい属性に変更される。右部重なり型連結の場合には、スタックのトップレベル

表1. 属性記号の自動生成処理

CNT1	条件部				実行部									
	WR2	F#1 (n <sub>1</sub> )	WR1	STK-P	STK	WR1	F#2				n <sub>1</sub>	n <sub>2</sub>		
	Rn	Ic	Rn	Ic	WR2	-CNT	Rn	Ic	Lp	Rp	ADR	DATA		
I	L M R R S	>0 0			L M M R S	push	0	0					**1	
*		>0 0						2 1		n <sub>2</sub>	n <sub>2</sub>	n <sub>2</sub>	M L	**1 **1
D								2		n <sub>2</sub>	n <sub>2</sub>	M	**1	**1
C	L	R	>0		L		**1						**1	
	L	R	0		S								**1	
	L	M			L								**1	
	M	R	>0		M		**1						**1	
	M	R	0		R								**1	
	M	M			M								**1	
	S	L			L								**1	
	R	S			R								**1	
	R	L			L		**1						**1	
	R	M			M		**1						**1	
S	S	M			M	pop	*S	2	*S	*S	Lp	M	**1	
	S	M			M	pop	*S	1	*S	*S	Lp	M	**1	
	S	R	>0		M	pop	*S-1	2	*S	*S	Lp	M	**1	
	S	R	0		M	pop	*S-1	1	*S	*S	Lp	M	**1	
	S	R	>0		R	pop	*S	2	*S	*S	Lp	M	**1	
	S	R	0		R	pop	*S	1	*S	*S	Lp	M	**1	
	S	R	>0		R	pop	*S	2	*S	*S	Lp	M	**1	
	S	R	0		R	pop	*S	2	*S	*S	Lp	M	**1	
	S	R	>0		M	pop	*S	2	*S	*S	Lp	M	**1	
	S	M	0		M	pop	*S	2	*S	*S			**1	
T	S		>1			pop	*S	*S	*S	*S	Rp	R		
	S		1			pop	*S	*S	*S	*S	Rp	S		
	L													
R	M		>1			pop	*S	*S	*S	*S	Rp	R		
	M		1			pop	*S	*S	*S	*S	Rp	S		
	R		1			pop	*S	*S	*S	*S				
B	S					pop	*S	*S	*S	*S				(出力)
	L													
	M													
R	R		>1			pop	*S	*S	*S	*S	Rp	R		
	R		1			pop	*S	*S	*S	*S	Rp	S		
	R		0			pop	*S	*S	*S	*S				(出力)
H										n <sub>2</sub>	S		**1	

注：\*は自分自身の値を、\*SはスタックSTK-Pのトップレベルのパラメータを意味する。

と次レベルの図形単位パラメータを1つの図形単位パラメータに合成し、1回ポップしたスタックのトップレベルに格納する。

(5) Tの処理：現在属性がSまたはRならば、最後に出力した交差部が注目する図形の最後の交差部となるので、最終交差部の出力属性をRpパラメータを用いてSまたはRに書き換えるとともに、スタックをポップして次の図形のパラメータを呼び出す。

(6) Bの処理：Tの場合と同様に、現在属性がSまたはRならば、その図形に関する最終交差部の出力属性をSまたはRに書き換えるとともに、スタックをポップする。ただし、その図形のIcパラメータが“0”であるならば、その図形の全ての交差部が終了したことになるので、出力属性の書き換えは行なわず、後述の図形特徴量の計算結果の出力時点とする。

(7) Hの処理：S属性をその新しい交差部の属性として単に出力する。現在属性、スタックは、更新しない。

以上の処理を、1画素走査の時間内に実行すれば、完全にビデオレートで属性記号列が生成できる。このとき、TとI、Iと\*、HとIの制御信号が同時点で発生することがあるが、別々にその順に発生したと同等の処理を1回の処理で実行させる。これは回路上の工夫で容易に実現可能であり、問題が生じることはない。

#### 4. 形状特徴量の自動計算法

各ラスタの交差部の属性記号列は、前述の方法により、1本前のラスタの属性記号列と画像データを参照することにより、順次生成することが可能である。この生成過程において、各交差部の各図形に関するかわり方が全て判定できるので、各交差部単位で計算される部分特徴量を各図形ごとに合成し累積していけば、個々の図形の特徴量を自動的に計算することが出来る。ここでは、この特徴量計算手法の考え方について述べる。

(1) 1つの交差部の特徴量の計算 特徴量計算の基本となる個々の交差部の特徴量は、独立した演算回路によって計算する。例えば、特徴量が面積の場合には、カウンタを用意して交差部の始点でゼロクリアし、そのまま“1”画素の数を数数するようにする。また、特徴量が図形の最小X座標の場合には、交差部の始点でX座標iをセットする。最大X座標の場合には、画素値が“1”ならば常にそのX座標iをセットするようにする。交差部の特徴量は、このようにしてその交差部の画像データの入力が完了する時点までに計算され、次の交差部が開始するまでのあいだ、保持されるようにする。

(2) 1ラスタ走査終了時点での形状特徴量の記憶 処理中のラスタに交差している図形の特徴量は、そのラスタの走査終了時点までにその図形の属性記号L(またはS)の交差部の付属情報として記憶する。すなわち、図形の終端記号であるR(またはS)属性を書き込む時に、その時点までに累積された形状特徴量を、パラメータLpをアドレスとしてメモリに書き込む。

(3) 1ラスタ走査中の形状特徴量の保持 処理中の形状特徴量は図形単位の値なので、前述の図形単位パラメータと同様に、スタックで管理する。その制御方法は図形単位パラメータの場合と全く同様である。1本のラスタ上で特定の図形に関する計算が完了したら、その時点でその特徴量をその図形の“L”または“S”交差部の付属情報として記憶する。

(4) 形状特徴量の計算 形状特徴量の計算は、(1)で計算された各交差部の特徴量Qと(2)に述べた第j-1ラスタまでの形状特徴量とを入力とし、(3)に述べたスタックのトップレベルをワークエリアとして実行され、各ラスタの終わりで累積された形状特徴量を再びLまたはSの付属情報として記憶する。図形の終了時点は第jラスタ上にその図形の交差部が全く存在しない(すなわち、Ic=0)ことから検知できるので、その時点で、累積された特徴量が外部に出力される。これらの計算処理は全て前述の7種の制御信号に応じた1ステップの処理で実行可能であり、完全にビデオレート1パス型の形状特徴量計算回路が実現できる。

#### 5. シミュレーション結果

本論文で提案した実時間1パス手法を、計算機上でシミュレーションした結果の一例を図7に示す。図において、@が“1”の画素を示し、#が形状特徴量の出力時点を示す。右端の文字列は、そのラスタの属性記号列である。

また、図の下には#の時点で出力された各種の形状特徴量が、出力順にリストの形で表示されている。特徴量は、リストの最初の要素から順番に、面積、最大X座標、最小X座標、最大Y座標、最小Y座標である。このような一連のシミュレーション結果から、ここで提案した手法が誤りなく動作し、かつ図形がどのように複雑に入り組んでいても、正しくその特徴量が計測されることを確認した。

## 6. 結 言

ラスタ走査で入力される画像をビデオレートで保持したまま実時間処理し、個々の図形の特徴量を自動計測することのできる、ハードウェア向きの新しい計算手法を開発した。本方式に従えば、テレビカメラやリニアセンサアレイなどからえられる映像信号を直接連続的に処理することができ、画像メモリを持つ必要がない。従来から、画像を画像に変換する前処理回路では、パイプライン処理によりこのようなことが可能なので、前処理回路の後段に本処理回路を直結すれば、極めて高機能の超高速認識装置を容易に構成できる。画像認識技術を実用装置に結び付けるためには、処理速度、処理機能の面で人間の能力を凌駕することが重要であり、このような本手法の高速性と多様な特徴量の計算能力は、そのための1つの突破口になるものと考えている。

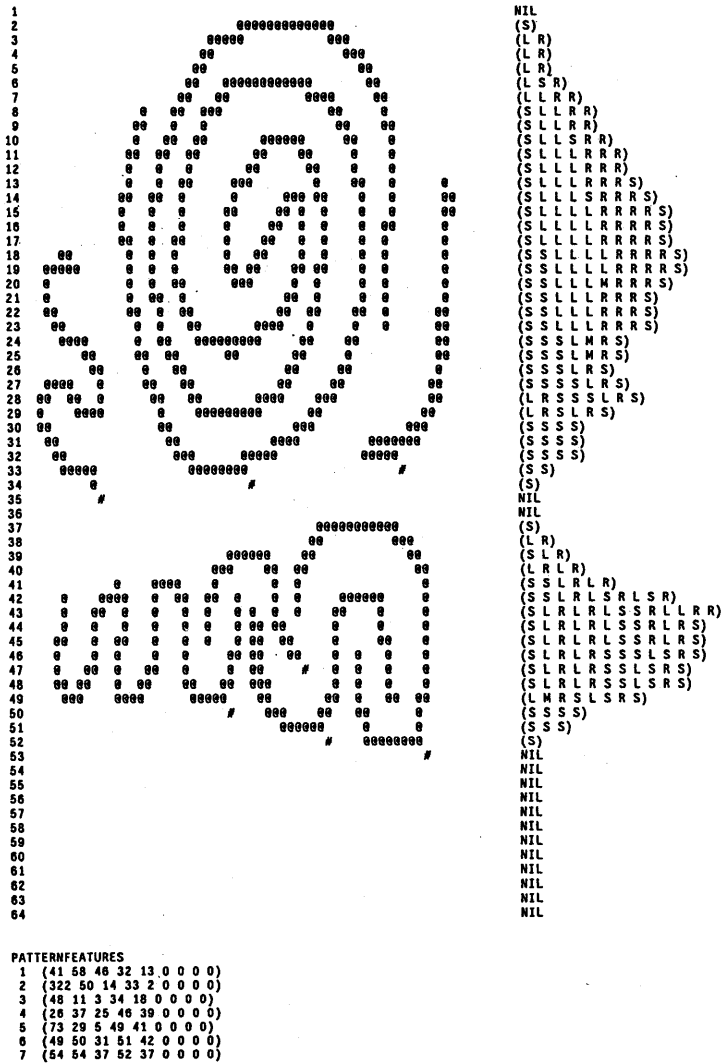


図7. 特徴量計算処理のシミュレーション結果

### 【引用文献】

- (1) “映像分析用計算装置”，特許公報，特公昭48-3447
- (2) Ullmann, “Video-rate Digital Image Analysis Equipment”, Pattern Recognition, Vol.14, Nos.1-6, pp.305-318(1981)
- (3) 依田,井内,江尻,“粒子画像解析のためのビデオレート1パス形手法”,昭和62年電子情報通信学会情報・システム部門全国大会予稿 128,(1987)
- (4) Yoda,H.,Ohuchi,Y.,Taniguchi,Y.,Ejiri,M.,“An Automatic Wafer Inspection System Using Pipelined Image Processing Techniques”, IEEE Trans. Pattern Anal. & Machine Intell., Vol.PAMI-10,No.1,pp.4-16,(Jan.,1988)