図形の分離と特徴抽出†

藤村 貞夫*・嵯峨山 茂樹**

Segmentation and Feature Extraction of Patterns

Sadao FUJIMURA* and Shigeki SAGAYAMA**

A method for segmentation of quantized patterns from a standpoint of connectivity is proposed. A pattern consisting of black points connected to each other is separated as a unit. These points are labeled during a scan of a picture. By an algorithm shown in this paper the same label is given to the points that join together. After one scan of the picture, a pattern is separated from the others by making use of the labels given to the points. The use of this method makes it also possible to count patterns. The number of labels used indicates how many patterns are contained in the picture. An improved rather noise-free method is also proposed, in which a pattern is regarded as composed of segments instead of points. The method enables us to pick up information while scanning for the segmentation, which is necessary to extract several kinds of features from each pattern with ease. These are global features, such as a circumscribed rectangle, center of gravity and axes of symmetry, and partial ones, such as corner points, loops and end points. The information is available for discriminating patterns or for adapting the segmentation algorithm to objects.

1. はじめに

文字・図形などの識別や処理を行なう場合、文字や図形があらかじめ分離されているか、分離のための指標（枠など）が与えられていれば、それにしたがって識別・処理すべき対象を逐次選んで作業を進めることができる。しかし、このような場合はむしろ稀で、

一般には文字なり図形なりが分離されないまま提示されることが多い。文章中に書かれた文字や生物学的データ（染色体、赤血球などの顕微鏡写真など）あるいは航空写真などがその例として挙げられよう。

このようなものの識別・処理を行なう際、多くの場合、まず全体を構成する各要素に分ける必要が生じる。この分離に関しては、これまでに、対象の形が既知として重なった要素を分離する方法^{1)~3)}や、各図形の縁を追跡する方法⁴⁾などが提案されている。

ここでは、互いに重なっていない数字やアルファベット大文字のような図形の分離を念頭におき、連結性(connectivity)にのみ着目して、つまり平面内で1つながりのものを1個のものとして分離する簡単な方法を提案する^{5)~7)}。この方法は他の方法に比して走査・処理が単純で、速度が速くなるという特長がある。ただ、平面内の個々の図形に異なる記号を付して区別を行なうため、かなり大きい記憶容量を要する。

ところで、図形を分離するのは、識別などのための前処理ということのほか、画面に含まれている図形の数を数えるという目的もありうる⁸⁾。もちろん、後者の目的は図形を分離する手続きが見出せば容易に達成できる。

一方、前者においては、分離後改めて分離された図形について正規化なり特徴抽出なりを始めることが考えられるが、これらの手続きと分離のそれとは走査その他で重複する部分のあることが予想される。したがって分離の際に、少し余分の手間をかければ得られる正規化や特徴抽出のための手がかりは同時に求めておくほうが、重複を避ける意味で有利であろう。

また、分離自身実際は対象の識別と密接な関係にあり、対象がどんなものであるかがわかれば分離も容易になるという面がある。さらに、漢字のように偏とつくりを合わせて1つの文字とするような場合には上の連結性だけでは不十分で、より高次の構造的な特徴と

† 第14回自動制御連合講演会で発表(昭46・11)

* 電気通信大学 調布市調布ヶ丘 1-5-1

** 東京大学宇宙航空研究所 東京都目黒区駒場 4-6-1

* University of Electro-Communications, Chofu

** Institute of Space and Aeronautical Science, University of Tokyo, Meguro-ku, Tokyo

(Received March 31, 1973)

もいべき性質を知ることが要求され、こういった目的のためには、分離と同時にこの性質の手がかりを得ることが必要となろう。

これらのことを考慮して、ここでは分離の手続きをわずかに拡張して得られる特徴を求める方法についても考察する。

2. 図形の分離

2.1 原理

ここで考察の対象とする図形は網目の上に黒白 (1 か 0) で表現されたものとし、分離のための基準としては、すでに述べたように、連続性のみを用いることとする。点 (i, j) が黒のとき、この点の周囲 8 つの網目 $(i-1, j-1), (i, j-1), (i+1, j-1), (i-1, j), (i+1, j), (i-1, j+1), (i, j+1), (i+1, j+1)$ のいずれかにある黒の点をこの点とつながっているとみなす。

この条件の下で、たとえば、Fig. 1 のような図形をどのようにして分離すればよいかを考える。ただし、上の条件により、Fig. 1 右上の図形は周囲の枠と中央の塊が別のものとして分離されることを注意しておこう。

ここで述べる方法は、1 つにつながった図形に同じ色を塗って色分けし、この色にしたがって分離を行なうものといえる。具体的な手順を述べる前に、どのようなことを行なうかを簡単な比喻を用いて示そう。そのために、画面の白の部分を陸地に黒の部分を池にたとえる。そうすると、ここで提案する方法は以下のように述べるができる。画面全体を左から右、上から下へと走査して、新しい池に出会うごとに、その地点に各池ごとに異なる色のインクを 1 滴落とす。落とされたインクは拡散によって池全体に拡がり池全体を同一色にしてしまうであろう (Fig. 2 (a) 参照)。ただし、走査の途中では Fig. 2 (b) のような池では同

じ池の片側を新しい池の出現と誤まって違った色のインクを落とすことになる。しかし、これは両方のインクが拡散してきて混合し始める時点 (P) で同じ色にすべきことがわかるので、両方の色を等しいとみなすことを記憶したのち、どちらかの色に統一して拡散が進行するようにしておけばよい。

こうして、画面を走査し終えた時点で、たとえば赤とそれに等しい色に染まった部分だけを取り出してくれば、ある 1 つながりの図形を他から分離して取出すことができ、等しい色を除いてインクを合計何色用いたかを求めれば池の個数も知ることができる。

2.2 走査

ある点 X につながっている点を見出すために、Fig. 3 に示した「窓」を用いて画面を走査する。また、上の色分けに対応して、図形の黒点に順次番号づけを行なう。画面上の白黒をそれぞれ 0, 1 で表わすと、 X が 0 の間は窓を 1 つずつ右にずらし、右端で復帰改行をして同じ操作を繰返す。この走査において、Fig. 4 の状態、つまり

$$X=1, A+B+C+D=0 \quad (1)$$

の成立するときを新しい図形の出現とみなして、 X に新しい番号をつける。最初に (1) が成立したとき $X=1^{*)}$ とし、2, 3 度目にはそれぞれ $X=2, X=3$ とする。1 か所に番号がつけばそれに連なっている点にも同じ番号をつけるわけで、そのときには

$$X=1, A+B+C+D \neq 0 \quad (2)$$

が成立する。つまり、窓の少なくとも 1 か所に番号のついた点が存在することになる。このときには、 A, B, C, D のうちただ 1 つだけが 0 でなければ、その番号を X に入れるし、2 つ以上が 0 でないときには、

*) 以下の手順では、もとの図形の黒の部分を表わす 1 と、 X に入れる番号の 1 とは混同を生じない。

Fig. 1 Examples of input patterns

Fig. 2 Diffusion of color ink

A	B	C
D	X	

Fig. 3 Scanning window

0	0	0
0	1	

Fig. 4 Encounter with a new pattern

Fig. 5 Flow chart of the algorithm
 CN: Color number
 N_1 : Number to be stored
 N_2 : Number already stored
 AN: Memory address
 $N(AN)$: Number stored at AN

たとえば、アルファベット順に優先順位を定めて番号を入れる。

ところで、後者の場合には、0 でない番号が同一番号でなければ異なった色のインクが混合し始めることを意味するから、それらを等しい（同値と呼ぶ）とみなすことを記憶する必要がある。なお、用いている窓の形から、意味のある（それまでに同値であることが記憶されていない）番号はただだか2種類であることがわかる。

したがって、ここでは新たに同値関係を記憶しなければならない可能性のある場所 ($A \neq 0$ なら A と C , $A=0$ で $B \neq 0$ なら B , $A=B=0$ なら C と D) だけを調べれば十分である。ある点 X での処理をフロー・チャートで示すと **Fig. 5** のようになる。Fig. 5 で CN は色の代わりに用いる番号であり、Next Step は窓を1つずらせることを意味している。作表手続きは後述。

2.3 等色表

2色混合が検出されたとき、それらの番号の同値関係を記憶するため、 N 個の色番号に対応して1から N までの番地のついた記憶場所を用意する（等色表）。ここで、 N は分離する図形の数に比べて十分大きい数とし、各記憶場所には最初に0を入れておく。

同値関係の記憶は、等しいとみなされる番号の一方

Table 1

	I	II	III	IV
1	0	0	0	0
2	0	0	1	1
3	0	2	2	1
4	2	2	2	1
5	0	0	0	0
6	0	0	5	5
7	0	0	6	5
8	0	0	0	0
9	0	0	0	0
.	.	.	.	.
.	.	.	.	.

の番地に他方の番号を蓄えることによって行なう。ただし、あとの便宜のため、規則 1:2 つの番号のうち大きい番号の番地に小さい番号を蓄える。たとえば、6番と5番が等しければ、6番地に5を記憶する。

ところで、このようにして同値関係を記憶する場合に問題となるのは、誤まってインクを落とすのは2色には限らないため、何個もの番号が等しくなり、上の規則にしたがって記憶しようとする、同じ番地にくつもの数を記憶しなければならない事態の生ずることである。

たとえば、**Fig. 6** の0点では等色表に同値関係が**Table 1** 第I欄のように記憶されているところで4番と3番が等しくなり、すでに番号の入っている4番地に新たに3を記憶したい事態となる。このときには、つぎの規則にしたがって記憶を行なえば、各番地には数字を1個だけ記憶すればよいことになる。今、ある番地 AN にすでに記憶されている番号を N_2 、新たに記憶したい番号を N_1 とすると、この規則は以下のようなになる。

規則 2: $N_1 > N_2$ なら N_1 番地に N_2 を記憶する。

規則 3: $N_1 = N_2$ なら記憶に関しては何も行なわな

Fig. 7 Flow chart of tabulating procedure

い。

規則 4: $N_1 < N_2$ なら N_2 番地に N_1 を記憶する。

規則 5: 記憶番地に等しい番号は記憶しない。

規則 6: 規則 2, 4 で「記憶する」とある個所についても、該当番地にすでに番号が記憶されていれば上の規則にしたがって処理する。

この規則にしたがって処理する作表手続きをフロー・チャートに示したものが、Fig. 7 である。Fig. 7 の Next Step も Fig. 5 と同じである。

この規則にしたがえば、上例の場合 Table 1 第 II 欄が得られる。Fig. 6 の P 点についても同様の処理をして、画面の一走査後には等色表は第 III 欄のようになる。このように画面を一走査した時点で等色表が完成し、図形を分離するための情報はすべて求めたことになる。また、この時点で画面に含まれる図形の数も知ることができる。すなわち、色分けのために総計 M 個の色を用いたものとすれば、図形の個数は等色表の 1 から M 番地までのうち、番号が記憶されていない (0 の入っている) 番地の個数として求めること

ができる。たとえば、Fig. 6 の場合 M は 7 で Table 1 の第 III 欄から図形の数 は 2 個であることがわかる。なお、この M は画面を走査し終えた時点での CN (Fig. 5) から、 $M = CN - 1$ として求まる。

2.4 等色表編集と分離

上述のように用いた色の間の同値関係が知られるので、これを用いて等しい色のところを同じ色に統一する。そのために等色表をつぎの手順で編集する。つまり、3 番地から順に、記憶されている (0 でない) 番号があればそれを、その数の番地に記憶されている (0 でない) 番号に入替えるという手続きを M 番地まで実行する。この手順にしたがって番号を入替えると、たとえば Table 1 第 III 欄は第 IV 欄となる。

こうして等色表を編集したのち、走査のときもとの図形につけた番号を編集後の等色表にしたがって入替えれば、1 つの図形は 1 つの色で塗り分けられることになる。たとえば、Fig. 6 の場合は、第 IV 欄にしたがって番号のつけ替えをすれば番号は 1 と 5 だけになり、しかも、1 つながりの図形の各部には同じ番号がつけられることがわかる。こうしてのち、同一番号をもつものを抜出せば、1 つながりの図形は他のものと分離されて取出されることになる。

上の方法で行なった図形の分離の例を Fig. 8 に挙げておこう。Fig. 8 (a) はもとの図形、(b) に画面走査を終了した時点での状態、(c) に等色表を編集したのち等しい色を統一して色分けをした状態を示す。図では印刷の都合上、上述の番号に対応してアルファベット大文字を使用している。(c) から、同じ色をもつものを抜出せば 1 つながりの図形を他から分離して取出せることがわかる。なお、Fig. 8 (b) で丸印の部分は、新しい図形の出現とみなされた場所、矩形の印は色の混合が起きる場所を示している。

2.5 セグメント法

上に述べた方法は 1 点ごとの連結性に着目しており、それが許される状況ではよい結果を与える。た

Fig. 8 Result of segmentation

Fig. 9 Segments

だ、実際に図形を観測し、量子化するには雑音の混入することは避けがたい。このとき上の方法にしたがうと、雑音と識別・処理したい図形との区別ができないう欠点がある。ここでは、この欠点を免がれるために、一種の平滑化 (smoothing) を行なう方法を考える。つまり、図形を、ある長さをもつセグメントで構成されるものと考えて、ある閾値より小さいセグメントは雑音とみなして無視するのである。

ここでは、横 (x 軸) 方向に連なった黒の点を 1 まとめにしてセグメントと呼び、左右両端の x 座標 p, q を用いて $\langle p, q \rangle$ と表わす^{*}。縦 (y 軸) 方向の位置と、セグメントにつける番号 α を同時に表現するときには $\langle \langle p, q \rangle, y, \alpha \rangle$ と表わすこととなる[†]。例としてセグメントに分けた図形を Fig. 9 に示す^{††}。

セグメントの左右両端の x 座標は、量子化された図形を、たとえば 1×2 の窓で走査して、それぞれ 01, 10 に対応する部分での x 座標として求まる。ただし、1 ビットだけ黒の抜けた部分は本来はつながっていたものとみなして穴埋めを行なうことにする。

図形を分離するには、このセグメント同士がつながっているかどうかを調べるわけである。上下に隣り合った 2 つのセグメントの上のほうを u , 下を l と呼びそれぞれ $\langle p_u, q_u \rangle, \langle p_l, q_l \rangle$ で表わされているものとしよう。このとき、セグメント u と l が接続しているための条件は

$$p_u \geq q_l \quad \text{かつ} \quad p_l \geq q_u \quad (3)$$

である。

このようにしてセグメント間の連結性が調べられれば、分離の手順は上述の場合と同様に進めることができる。なお、このとき、

$$|p - q| \leq n \quad (4)$$

を満たすセグメントは雑音として無視し、登録しないことにすれば、平滑化を実行できる。ここで、 n は図形を構成する線の太さと雑音の大きさを考慮して定める値である。

雑音のある入力について、(4)式による平滑化の効果を示したものが Fig. 10 で、この場合 $n=1$ となっている。

Fig. 10 Noise reduction

Fig. 11 Types of segment connection and corner points

3. 特徴抽出

上に述べた図形分離の手続きの途中で、少し余分の手間をかければ得られる特徴について考える。ここでは、セグメント法を用いた、文字などの識別に有用と思われる部分的特徴の求め方について述べる。図形の重心、外接矩形、直交する 2 つの対称軸、対称軸回りのモーメントなどの全体的特徴も容易に求められるが、ここでは省略する。

部分的特徴はセグメント間のつながり方 (接続形) から求まり、これはセグメント間の接続を調べるとき、ついでに $p_u - p_l, q_u - q_l$ を計算して知ることができる。

(a) 接続形

(i) 一様接続: 線分や曲線部分の滑らかな接続を表わし、特徴点とはみなさない。これは、

$$|p_u - p_l| \leq m \quad \text{かつ} \quad |q_u - q_l| \leq m \quad (5)$$

あるいは

$$|q_u - q_l - (p_u - p_l)| \leq m \quad (6)$$

のいずれかを満たす接続 (Fig. 11 (a) 参照) で、 m は網目の大きさと図形に含まれる線分の勾配から定める値である^{*}。

(ii) 逆 Γ 接続: Fig. 11 (b) のような接続で、

^{*} m の値は大きすぎると特長を見落とすので、あとの処理は多少煩雑にはなるが、小さ目にとっておくほうがよい。

^{*} ただし、 q の座標は右端の黒の点の座標を x_r として、 $q = x_r + 1$

[†] 以下の実験では、1 画面を 64×64 の網目で構成し、計算機の記憶番地に y を対応させ、1 語 (24 ビット) のうち 18 ビットを用いて、 p, q, α をそれぞれ 6 ビットで記憶している。

^{††} x, y 座標は、図のようにとるものとする。以下の議論は、すべてこれに基づいて行なう。

Fig. 12 Loops

Fig. 13

→
Original patterns, obtained
features and reconstructed
patterns

(a)

(b)

(c)

このとき

$$p_u - p_l > m \quad (7)$$

この接続の位置は、内・外角点（それぞれ、Fig. 11 (b) の ⊙, ⊗）で示すこととする。内角点の座標は (p_u, y_l) であり、外角点のそれは (p_l, y_u) である。

以下、接続形とそれが満たす条件ならびに内・外角点の座標を列挙すると、

(iii) 逆 L 接続 (Fig. 11 (c) 参照)

$$p_l - p_u > m \quad (8)$$

外角点: (p_u, y_u) , 内角点: (p_l, y_u)

(iv) Γ 接続: (Fig. 11 (d) 参照)

$$q_u - q_l > m \quad (9)$$

外角点: (q_u, y_u) , 内角点: (q_l, y_u)

(v) L 接続: (Fig. 11 (e) 参照)

$$q_l - q_u > m \quad (10)$$

外角点: (q_l, y_l) , 内角点: (q_u, y_l)

(b) 上下端

あるセグメントの上に接続するセグメントがない場合、このセグメントを上端セグメントとし、その左端右端をそれが属する図形の左上端、上右端と呼ぶ。一方、あるセグメントの下に接続するセグメントがない場合、このセグメントを下端セグメントといい、その左、右端をそれぞれ下左端、下右端と呼ぶ。

上、下端は1つの図形に属するセグメントについて y 座標値の最大、最小値から求められ、それに対応するこれらの端点も容易に求めることができる。

なお、ここでは、新しい番号の付されるセグメントをすべて上端としており、したがって、1つの図形について上端が複数個存在することもありうる。下端についても同様。

(c) ループ

ループとしては、たとえば Fig. 12 のような種々

のものがあろうが、いずれにしても、同じ番号あるいはすでに等価であることが知られている番号をもつセグメントが合流して1つになる場合には、その図形の中にループが存在することがわかる。

窓を用いる方法の場合、Fig. 5, 7 のフロー・チャートで丸印の部分でこの合流を検出できる。セグメント法の場合も、同様に簡単にこの合流を知ることができるから、ループの存在と個数、合流点の x, y 座標値を求めることができる。

これらの値は単純なループの場合にはきわめて有用な情報を提供し、たとえば、6, 8, 9 については、この y 座標値とループの数から、これらのうちのいずれであるかを識別することができる。

(d) 特徴による図形の再現

以上の特徴は、その種類を表わす符号と、それが属する色番号、その位置 (x, y 座標値) を1まとめにして記憶している。

ところで、これらの特徴がどの程度元の図形のもつ情報を担っているかを以下に実例によってみてみよう。すでに述べたように、各特徴点は種類と位置が知られているから、それぞれの位置に特徴点の形にしたがって、たとえば、L 形外角点の位置には「」内角点の位置には「」などを書き入れてみれば元の図形がかなりよく再現されるものと思われる。この場合、各特徴点間のつながり方は、直接には特徴抽出段階で取入っていないが、特徴点の種類からつながり方がある程度推定できるわけである。

簡単な図形として ISO OCR-B フォントの数字 0 ~ 3 を例にとり、元の図形、得られた特徴、特徴から再生した図形を順に Fig. 13 (a) ~ (c) に示す。今の場合、(16) ~ (21) の m は経験的に2にとってある。Fig. 13 (c) で、●は上端、○は下端、⊙⊗はそれぞ

Table 2 Number of beature points

OCR-B font Features	0	1	2	3	4	5	6	7	8	9
Loop	1	0	0	0	0	0	1	0	2	1
Upper end	1	1	1	2	2	1	1	1	1	1
Lower end	1	2	2	1	1	1	1	1	1	1
L-type connection	1	0	1	1	1	1	2	0	2	1
Γ -type connection	1	1	2	1	0	1	1	1	2	1
$\neg$ -type connection	1	1	1	1	1	1	1	0	2	2
L-type connection	1	0	1	1	1	1	1	0	2	1

れ内・外角点を示している。各特徴点を出る太線は特徴点の種類で決まるセグメントの接続形を表わしており、特徴点間を結んだ破線はこれから推定されるつながりを示している。

なお、図で破線を決定する際、実際には、 Γ 、逆 L (L 、逆 Γ) 形接続がこの順に同一セグメント上にあるとき、この部分でのセグメントは π (逆 π) 形接続をすることを用いている。

図から知られるように、上述の特徴抽出法により、基本的な特徴はほぼ抽出されることがわかる。3については、中央の部分のセグメントのつながり方を表現する特徴が設定されていないために、元の図形は再生されていない。ISO OCR-B フォントの数字については、3を除いて、他はすべて再生されることが確かめられた。上の方法で実際に得られた特徴点の数を種類別に表にすると Table 2 のようになり^{*)}、これだけでも上の数字に関しては識別が可能である。ただ、数字が傾くと表中の数字は変化するため、傾斜まで考慮する場合には、特徴点間の相対的位置関係まで用いる必要がある^{**)。}

4. 結びと謝辞

画面の中で1つながりの図形を1つのものとして分離する手法と、このとき同時に特徴を抽出する手法について考察した。これらは、連結性を分離の条件として採用してよい場合には、きわめて有効であることを確かめた。得られる特徴も簡単な図形については、識

別にも十分なものであることを示した。さらに、この手法は、図形の数を数える目的にも利用することができる。

得られる特徴を分離の手順にフィード・バックして、漢字などの場合のように、より高度の分離ができるようにすること、ハーフ・トーンの図形への適用などが今後の課題である。

日頃御指導を賜わる東京大学磯部孝教授に謝意を表す。また、本研究にあたり、プログラムその他で有益な助言と援助をいただいた渡辺泰助氏（現在、松下技研）に謝意を表す。

参 考 文 献

- 1) C. T. Zahh: Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans., C-20, 1, 68/86, (1971)
- 2) R. L. Hoffman et al.: Segmentation methods for Recognition of Machineprinted Characters, IBM J. Res. Dev., 15, 3, 153/165, (1971)
- 3) C. Arcelli et al.: Picture Processing and Overlapping Blobs, IEEE Trans., C-20, 9, 1111/1115, (1971)
- 4) R. L. Ledley et al.: FIDAC: Film Input to Digital Automatic Computer and Associated Syntax Directed Pattern Recognition Programming System, Optical and Electro-Optical Information Processing ed. by J. T. Tippett et al. MIT Press, 591/613, (1965)
- 5) 藤村貞夫: 図形分離の1方法, 第14回自動制御連合講演会前刷, 393/394, (1971)
- 6) 藤村貞夫: 任意図形の分離と計数, 電子通信学会論文誌, 55-D, 1, 63/64, (1972)
- 7) 嵯峨山茂樹: 図形の分離と特徴抽出, 昭和47年, 東京大学工学部計数工学科卒業論文。
- 8) P. Weston: Photocell field counts random objects, Electronics, 34, 9, 46/47, (1961)

^{*)} 表では、上左端、上右端を合わせて上端1個と数えてある。下端についても同様。

^{**) ことでの場合、表の数字がそのまま成立するのは3~4°までである。}