

## 選択画素符号化方式と選択画素を用いた図形処理†

森田 啓義††\* 有本 卓††

デジタル2値画像の一表現方法として、画像から抽出された特徴点による表現について論じた。この特徴点は、2値画像を図形としてながめたときの“角”に密接に関係して定義され、選択画素と呼ばれる。一般に2値画像はそれ自身に含まれるすべての角により一意的に表現されるが、選択画素による画像表現は角による表現より簡潔であることが示される。そこで選択画素による画像表現を2値画像の符号化に適用し、符号化効率を実際の画像データ（漢字パターン）に対して調べた。その結果、画像の統計的性質に依存しない簡単な符号構成をもつ提案方式によって、統計的な手法を用いる従来のものと同等あるいはそれ以上のデータ圧縮効果が得られることが明らかになった。ところでこの選択画素による画像表現には元の画像のもつ位相的な性質に関する情報が含まれている。そこで選択画素の情報のみから2値画像に含まれる連結成分およびホールを同時検出アルゴリズムを導いた。このアルゴリズムは1パス走査による連結成分およびホールの検出を実現している。

### 1. はじめに

近年、計算機による画像処理技術が発達し、画像データ・ベースやロボットの視覚系などへ幅広く応用されるにつれて、膨大な量の画像データをそれ自身に含まれる情報を失わない方法で圧縮して記憶すること、いわゆる画像の符号化が重要になってきている<sup>1-10)</sup>。

本論文では、計算機において扱われる機会の多い回路図・機械部分の設計図・漢字パターンなどの幾何学的なパターンから構成される2値画像に対する符号化について考察する。

従来の2値画像符号化方式としては、ランレングス符号化方式<sup>3)</sup>・予測符号化方式<sup>4),5)</sup>・MR方式<sup>6)-8)</sup>など数多くの方式が提案されている。これらの方式においては、ラスタ走査方向に色の変化する変化点の位置あるいはその間隔（ランレングス）を符号化の対象とし、それぞれ独自の方法で符号化している。

ここではデータ圧縮効果を高めると同時に簡単な符号構成を実現するために、従来の方式とは異なり、符号化の対象となる画素そのものの数を少なくすることによって2値画像の符号化を行う。このためにはまず、画像から選び出された少数の画素の集りを用いて元の画像を一意に復元できる画像表現方法が必要になる。その一つの方法に、2値画像の図形としての“角”に相当した選択画素による表現方法<sup>9)</sup>がある。

本論文では、この選択画素による画像表現方法に基づいて選択画素符号化方式 (Selective Element Coding Technique, 略して SECT) を導き、そのデータ圧縮効果について従来の方式との比較・検討を行う。

さらに SECT では、DF-表現方式<sup>2)</sup>・線図形に対するチェーン符号化方式<sup>10)</sup>などと同様に、図形の平行移動・拡大・縮小を符号化データのままで行うことができる。これは選択画素による画像表現自体に元の画像の幾何学的な特徴が含まれていることに起因する。

選択画素による画像表現のこの側面を利用して、2値画像に含まれる連結成分の検出を画像全体ではなく、選択画素だけを用いて行うことができる。この検出方法は、画像の各画素を一定順序で一度だけ走査して処理すれば連結成分とホールを同時に検出できるという点で、従来の方法<sup>11)-14)</sup>以上に実用に適している。

2章ではまず、本文で用いる記号と語句の説明を行い、ついで選択画素の定義を与えて2種類の画像表現方法について論じる。3章では前章の議論に基づいて SECT を導出し、圧縮効果を調べるために漢字パターンの符号化を行う。4章では選択画素による画像表現を用いた1パス走査形式による連結成分とホールの同時検出方法について述べる。なお本文中ではアルゴリズムの記述に文献<sup>15)</sup>の片言 ALGOL を用いた。

### 2. 選択画素の定義と選択画素による2値画像表現

本文中で用いる語句と記法をまとめておく。対象とするデジタル白黒2値画像を、黒画素が1、白画素が0に対応した大きさ  $M \times N$  の2値行列  $P$  として

† Selective Element Coding Technique and Image Processing by Using Selective Elements by HIROYOSHI MORITA and SUGURU ARIMOTO (Department of Mechanical Engineering, Faculty of Engineering Science, Osaka University).

†† 大阪大学基礎工学部機械工学科

\* 現在 豊橋技術科学大学工学部生産システム工学系

表す.  $P$  の第 1 列の左, 第  $N$  列の右と第 1 行の上に, 値が 0 の列  $p(i, 0)$ ,  $p(i, N+1)$ ,  $1 \leq i \leq M$  と行  $p(0, j)$ ,  $0 \leq j \leq N+1$  を加えた  $(M+1) \times (N+2)$  行列を  $P_{ex}$  とする.  $P_{ex}$  全体の集合を  $\mathcal{P}_{ex}$  とおく. 座標  $(i, j)$ ,  $0 \leq i \leq M$ ,  $0 \leq j \leq N+1$  を画素,  $P_{ex}$  の要素  $p(i, j)$  を画素値と呼ぶ. 画素  $(i, j)$  が左隣りの画素  $(i, j-1)$  と異なる値をもつとき,  $(i, j)$  を変化点という. また,  $(i, j)$  が,  $p(i, j) = p(i-1, j-1)$ ,  $p(i, j) \neq p(i-1, j)$ ,  $p(i, j) \neq p(i, j-1)$  を満たすとき,  $(i, j)$  を交差点という. 画素  $(i_1, j_1)$  と  $(i_2, j_2)$ ,  $0 \leq i_1, i_2 \leq M$ ,  $0 \leq j_1, j_2 \leq N+1$  の間には

$$(i_1, j_1) \leq (i_2, j_2) \Leftrightarrow i_1 < i_2 \text{ あるいは } i_1 = i_2, j_1 \leq j_2$$

の意味での辞書式順序を与え, 画素の位置関係に大小比較を許す.

**定義 1** (選択画素  $I$  点,  $R$  点):  $P_{ex}$  における, 区間  $\{(i, j) | 1 \leq i \leq M, 1 \leq j \leq N+1\}$  内の画素に対し,  $I$  点,  $R$  点を次のように定義する.

- ①  $(i, j)$  は  $I$  点である.  
 $\Leftrightarrow p(i-1, j-1) = p(i, j-1)$  かつ  $p(i-1, j) \neq p(i, j)$
- ②  $(i, j)$  は  $R$  点である.  
 $\Leftrightarrow p(i-1, j-1) \neq p(i, j-1)$  かつ  $p(i-1, j) = p(i, j)$

さらに変化点である  $I$  点 ( $R$  点) を  $TI$  点 ( $TR$  点), 変化点でない  $I$  点 ( $R$  点) を  $CI$  点 ( $CR$  点) と記す. 行の両端には値 0 の画素が存在するので, 明らかに, 一つの行には  $I$  点と  $R$  点が対をなして存在する. また, 幾

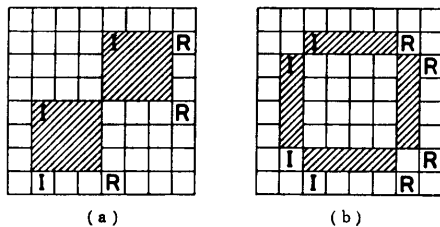


図 1 選択画素の例

Fig. 1 An example of selective elements.

```

begin
1  Q ← (M+1) × (N+2) の 0-行列;
2  for n ← 0 until K do
3    if n は偶数 then DECODE ((In, Jn), (In+1, Jn+1-1), 0)
4    else DECODE ((In, Jn), (In+1, Jn+1-1), 1);
  return (Q)
end
procedure DECODE ((Is, Js), (IE, JE), FLAG)
1 if FLAG = 0 then for (u, v) ← (Is, Js) until (IE, JE) do q(u, v) ← q(u-1, v)
   else for (u, v) ← (Is, Js) until (IE, JE) do q(u, v) ← 1 - q(u-1, v)

```

図 2 アルゴリズム 1:  $(I, R)$ -表現からの元の画像の復元

Fig. 2 Algorithm 1: Reproducing the original image from an  $(I, R)$ -representation.

何学的には  $I$  点・ $R$  点は図形の角に対応している (図 1 参照). 画像  $P_{ex}$  に  $K$  個の  $I$  点と  $R$  点が存在するとき, これらを大きさの順に並べた系列  $D_p = \{(I_n, J_n), n=1, \dots, K\}$  を画像  $P$  の  $(I, R)$ -表現と呼ぶ. ここで  $K$  は常に偶数であることに注意する.  $(I, R)$ -表現から元の 2 値画像  $P$  は次のアルゴリズム 1 を用いて復元できる.

**アルゴリズム 1:** ある 2 値画像  $P$  の  $(I, R)$ -表現  $D_p$  を入力とし,  $P_{ex}$  を出力する. アルゴリズムでは, 大きさ  $(M+1) \times (N+2)$  の 2 次元配列  $Q$  と  $(I_0, J_0) = (1, 0)$ ,  $(I_{K+1}, J_{K+1}) = (M, N+1)$  なる定数を使用して, 部分区間  $[(I_n, J_n), (I_{n+1}, J_{n+1}-1)]$ ,  $0 \leq n \leq K$  ごとに順次復元してゆく. アルゴリズムを図 2 に示す.

**命題 1** 2 値画像  $P$  の  $i-1$  行 ( $1 \leq i \leq M$ ) の画素と  $i$  行の  $I$  点と  $R$  点の位置が与えられれば,  $i$  行の画素値はアルゴリズム 1 から正確に復元される.

(証明) アルゴリズム 1 によって復元された  $i$  行の画素値を  $q(i, j)$ ,  $0 \leq j \leq N+1$  とおく.  $P_{ex}$  の定義から  $(I, R)$ -表現の偶数番目の画素  $(I_{2n}, J_{2n})$  は  $R$  点で, 奇数番号の画素  $(I_{2n-1}, J_{2n-1})$  は  $I$  点なので, 復元された  $i$  行の  $I$  点と  $R$  点の位置は元の画像  $P$  の  $i$  行のそれらと完全に一致している. さて, 復元された画素値が  $P$  の  $i$  行の値と異なっていると仮定する.  $j^* \triangleq \min \{j | p(i, j) \neq q(i, j)\}$  とおく. 明らかに,  $p(i, j) = q(i, j)$ ,  $0 \leq j \leq j^*-1$ . もし,  $p(i-1, j^*-1) = p(i, j^*-1)$  ならば,  $(i, j^*)$  は元の画像と復元画像のいずれか一方において  $I$  点となり, 他方では選択画素ではない. またもし  $p(i-1, j^*-1) \neq p(i, j^*-1)$  ならば, 元の画像と復元画像のいずれか一方において  $(i, j^*)$  は  $R$  点となり, 他方では選択画素ではない. これは復元された行と元の行では選択画素が一致するという仮定に反する. すなわち,  $p(i, j) = q(i, j)$ ,  $0 \leq j \leq N+1$  が成立する. (証了)

命題 1 を繰返し用いることにより, アルゴリズム 1 は  $D_p$  から  $P$  を復元することを容易に示すことができる.

$(I, R)$ -表現は一つの画像を表現するには十分すぎる. すなわち, 選択画素の定義を少し変更することによって, 画像を表現する選択画素の数を角の数 (= 辺数) より確実に少なくすることができる. ここで, 2 値画像を表す 0-1 行列  $P$  における辺とは, 同じ値をとる変化点が縦に 1 列に並んだもの (縦辺), あるいは真上の画素と異なる値をとる画素で同じ値のものが横に 1 列に並んだもの (横辺)

を指す。

**定義 2** (選択画素  $i$  点,  $r$  点,  $C$  点)

①  $(i, j)$  は  $i$  点である。

⇔ i)  $(i, j)$  は  $TI$  点である。あるいは,

ii) 次の 3 条件が同時に成立する。

a)  $(i, j)$  は  $CI$  点である。

b)  $\bar{j} \triangleq \min \{t | (i, t) \text{ は } R \text{ 点}, t > j\}$  とおくととき,  
 $(i, \bar{j})$  は  $CR$  点である。

c)  $(i, j)$  と  $(i, \bar{j})$  の間には交差点が存在しない。

②  $(i, j)$  は  $r$  点である。⇔  $(i, j)$  は  $TR$  点である。

③  $(i, j)$  は  $C$  点である。⇔  $(i, j)$  は交差点である。

**命題 2** 2 値画像  $P$  に含まれる図形の辺数を  $L_p$  とし,  $P$  に含まれる  $i$  点,  $r$  点,  $C$  点の総数を  $N_p$  とすると, 次の不等式が常に成立する。

$$L_p/2 \leq N_p \leq 3L_p/4$$

証明は付録で行う。

画像  $P$  に  $i, r, C$  点が  $K$  個含まれるとき, その位置  $(I, J)$  と種類  $X(=i, r, C)$  を組にして位置の大きさ順に並べた系列  $d_p = \{(I_1, J_1, X_1), \dots, (I_K, J_K, X_K)\}$  を  $P$  の  $(i, r, C)$ -表現とよぶ。  $i$  点,  $r$  点,  $C$  点と  $(i, r, C)$ -表現に含まれない  $I$  点と  $R$  点に対して次の関係が成立する。

**命題 3**

(1)  $X_n = i$  または  $C$  のとき, もし,

$$J_n^* = \min \{j | (I_n - 1, j) \text{ は変化点},$$

$$(I_n - 1, J_n) < (I_n - 1, j) < (I_{n+1} - 1, J_{n+1})\}$$

が存在するならば,  $(I_n, J_n^*)$  は  $CR$  点である。

(2)  $X_n = r$  または  $C$  のとき, もし,

$$J_n^* = \max \{j | (I_n - 1, j) \text{ は変化点},$$

$$(I_{n-1} - 1, J_{n-1}) < (I_n - 1, j) < (I_n - 1, J_n)\}$$

が存在するならば,  $(I_n, J_n^*)$  は  $i$  点とは異なる  $CI$  点である。

(証明)  $X_n = i$  または  $C$  で  $J_n^*$  が存在すると仮定する。このとき  $J_n^*$  の定義から,

$$p(I_n - 1, j) = p(I_n - 1, J_n), J_n \leq j \leq J_n^* - 1 \quad (1)$$

が成立する。また仮定より,

$$p(I_n, J_n) = 1 - p(I_n - 1, J_n) \quad (2)$$

である。さらに, 選択画素  $r$  点は  $(I_m, J_m)$ ,  $1 \leq m \leq K$  以外の位置には存在しないことから,

$$p(I_n, j) = p(I_n, J_n), J_n \leq j \leq J_n^* \quad (3)$$

が成立する。①, ②, ③から  $(I_n, J_n^*)$  は  $CR$  点である。

$X_n = r$  または  $C$  で  $J_n^*$  が存在する場合には,  $(I_n, J_n^*)$  から  $(I_n, J_n)$  と  $(I_n - 1, J_n^*)$  から  $(I_n - 1, J_n)$  までの画

素値を上と同様の方法で調べることにより, 題意が示される。(証了)

**命題 4** 任意の  $CR$  点  $(I, J)$  に対し, 適当な  $n$  が存在し,  $I_n = I, J_n^* = J, X_n = i$  または  $C$  が成立する。同じく,  $i$  点とは異なる  $CI$  点  $(I, J)$  に対し, 適当な  $n$  が存在し,  $I_n = I, J_n^* = J, X_n = r$  または  $C$  が成立する。

(証明) 任意の  $CR$  点  $(i, j)$  を一つ固定する。  $p(i, j-1) = 1 - p(i-1, j-1)$  かつ  $p(i, 0) = p(i-1, 0)$  なので,

$$j_* \triangleq \max \{k | (p(i, k), p(i-1, k))$$

$$\neq (p(i, j-1), p(i-1, j-1)), 0 \leq k < j-1\}$$

が必ず存在する。このとき,  $p(i, j_*)$  と  $p(i-1, j_*)$  の値に関しては,

$$(1) \quad p(i, j_*) = p(i, j_* + 1)$$

かつ

$$p(i-1, j_*) = 1 - p(i-1, j_* + 1)$$

$$(2) \quad p(i, j_*) = 1 - p(i, j_* + 1)$$

かつ

$$p(i-1, j_*) = p(i-1, j_* + 1)$$

$$(3) \quad p(i, j_*) = 1 - p(i, j_* + 1)$$

かつ

$$p(i-1, j_*) = 1 - p(i-1, j_* + 1)$$

の三つの場合が考えられる。①②の場合,  $(i, j_* + 1)$  は  $i$  点, ③の場合には  $C$  点なので, ある  $n$  が存在して,  $(I_n, J_n) = (i, j_* + 1)$  である。また  $(i, j)$  は  $CR$  点なので  $(i-1, j)$  は変化点であることから,  $J_n^*$  は必ず存在する。さらに  $j_*$  の定義から区間  $[(i-1, j_* + 1), (i-1, j-1)]$  には変化点は存在しない。これは区間  $[(I_n - 1, J_n + 1), (I_{n+1} - 1, J_{n+1} - 1)]$  における最小の変化点が  $(i-1, j)$  であることを意味する。すなわち  $J_n^* = j$  である。上と同様の議論から, 任意の  $i$  点でない  $CI$  点  $(i, j)$  に関してはある  $n$  が存在して  $(I_n, J_n) = (i, j_E)$  が導かれる。ここで  $j_E$  は

$$j_E = \min \{k | p(i, k), p(i-1, k))$$

$$\neq (p(i, j), p(i-1, j)), j < k \leq N+1\}$$

で定義される。さらに  $J_n^* = j$  であることが容易に示される。以上により定理の主張が成立する。(証了)

命題 3 と 4 は, 区間  $[(I_n, J_n), (I_{n+1}, J_{n+1} - 1)]$  には  $CR$  点  $(I_n, J_n^*)$  と  $i$  点でない  $CI$  点  $(I_{n+1}, J_{n+1}^*)$  がたかだか一つずつ存在することを示している。

$(i, r, C)$ -表現から元の画像を復元するには次のアルゴリズムを用いる。

**アルゴリズム 2:** ある  $(i, r, C)$ -表現  $d_p$  を入力と

```

begin
1  Q←(M+1)×(N+2) の 0-行列;
2  for n←0 until K do
      begin
3    CURPEL←(In, Jn);
4    if (Xn は i または C) then
5      if (Jnf が存在する) then
          begin
6        DECODE (CURPEL, (In, Jnf-1), 1);
7        CURPEL←(In, Jnf)
          end;
8    if (In+1>In) then
          begin
9        DECODE (CURPEL, (In+1-1, M), 0);
10       CURPEL←(In+1, 1)
          end;
11   if (Xn+1 は r または C) then
          begin
12     FLAG←1;
13     if (Jn+1b が存在する) then
          begin
14       DECODE (CURPEL, (In+1, Jn+1b-1), 0);
15       CURPEL←(In+1, Jn+1b)
          end
          end
          end
16   else FLAG←0;
17   DECODE (CURPEL, (In+1, Jn+1-1), FLAG)
          end;
18 return (Q)
end

```

図 3 アルゴリズム 2: (i, r, C)-表現からの元の画像の復元  
 Fig. 3 Algorithm 2: Reproducing the original image from an (i, r, C)-representation.

し,  $P_{ex}$  を出力する. 方法としては, 区間  $[(I_n, J_n), (I_{n+1}, J_{n+1}-1)]$ ,  $0 \leq n \leq K$  ごとに (i, r, C)-表現を (I, R)-表現に直しながらアルゴリズム 1 と同じ方法で復元を行う. ここで  $(I_0, J_0, X_0) \leftarrow (1, 0, r)$ ,  $(I_{K+1}, J_{K+1}, X_{K+1}) \leftarrow (M, N+2, i)$  とおく. アルゴリズムを図 3 に示す.

命題 5 アルゴリズム 2 は, ある (i, r, C)-表現  $d_r$  から元の画像  $P$  を正確に復元する.

(証明)  $m-1$  行 ( $m \geq 1$ ) の  $P_{ex}$  の画素値が正確に

復元されているとする (0 行は  $P_{ex}$  の条件から常に正確に復元される). このとき  $m$  行の画素値が正確に復元されることを示せばよい. もし  $m$  行に  $i$  点・ $r$  点・ $C$  点のいずれも存在しなければ, ある  $n (\geq 0)$  に対し,  $I_n \leq m < I_{n+1}$  が成立する. ただし等号成立は  $m=1$ , すなわち  $n=0$  の場合に限る. このとき  $m$  行はアルゴリズム 2 のステップ 8 の実行中において値が定まる. このステップでの操作はアルゴリズム 1 の場合と同じなので,  $m$  行は正確に復元される. 次に,  $m$  行に  $i$  点・ $r$  点・ $C$  点のいずれかが存在するとする. その一つを  $(I_p, J_p)$ ,  $I_p = m$  とする. 命題 3 によれば,  $(I_p, J_p)$  の前後に  $CR$  点あるいは  $i$  点でない  $CI$  点が存在する可能性がある. これらの画素が存在するかしないか, さらに存在するとすればどの位置にかは,  $m-1$  行の値と  $X_p, J_{p-1}, J_{p+1}$  から正確に判定できる. アルゴリズム 2 ではこの判定をステップ 5 とステップ 13 において行っている. 判定後の操作は, (I, R)-表現が得られている場合にアルゴリズム 1 で行う操作と等しいので,  $m$  行は正確に復元される. (証了)

### 3. 選択画素符号化方式

#### 3.1 符号化・復号化アルゴリズム

(i, r, C)-表現は, 2 値画像に描かれた図形の斜め  $45^\circ$  の直線部分において他の場所以上に選択画素が多く現れる傾向をもつ (図 4 (a) 参照). そこで, この章では, 斜め  $45^\circ$  の直線上にある  $i$  点,  $r$  点を省いても元の画像を一意に表現できるアルゴリズムを示し, その方法を用いて 2 値画像のデータ圧縮を行う. まず選択画素を求めるアルゴリズムを次のように定める.

アルゴリズム 3 (符号化過程): 入力を  $(M+1) \times (N+2)$  の 2 値画像  $P_{ex} \in \mathcal{P}_{ex}$  の第  $M$  行の下に, 値 0 の行  $p(M+1, j)$ ,  $0 \leq j \leq N+1$  が加わった行列  $P'_{ex}$  と

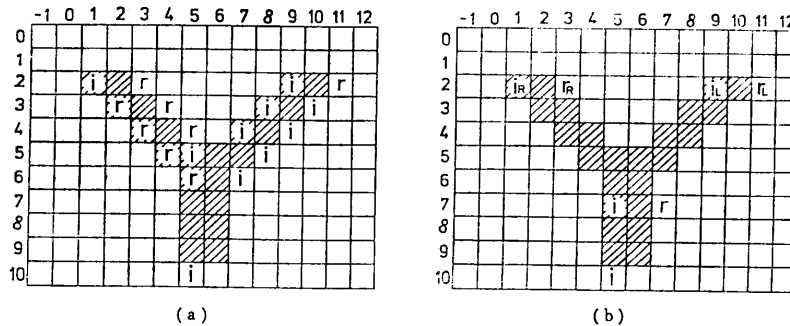


図 4 (i, r, C)-表現とその変形表現の例  
 Fig. 4 Examples of (i, r, C)-representation and its modified version.

し、 $P'_{xx}$  に含まれる選択画素の位置と種類のリスト ENCODE を出力する。このアルゴリズムでは、 $P$ 、ENCODE の他に二つの長さ  $N$  の 1 次元配列 LEFT と RIGHT を使用する。LEFT と RIGHT の各要素は値 0 または 1 をとり、その初期値は 0 とする。さて、アルゴリズムは各  $i$  行ごとに  $i-1$  行と  $i+1$  行の画素値を用いて選択画素を捜し、その位置と種類をリスト ENCODE に登録する。 $i$  行の処理が終わったら、次行の処理に移る前に手続き、RUNSHIFT ( $i$ ) を行う。この手続きは、RIGHT、LEFT のいずれかの値が画像  $P$  の  $i$  行に存在する 1-ランまたは 0-ランの先頭の位置において 1 である場合に限り、該当するランの先頭を RIGHT が 1 なら右へ、LEFT が 1 なら左へ一つ移す。それと同時に、配列 RIGHT、LEFT 内で 1 をとる場所もランの先頭の移動に伴い同じ方向へ一つずらす。アルゴリズムは図 5 に示す。

アルゴリズム 3 では  $i_R, i_L, r_R, r_L, C_R, C_L$  の 6 種類の画素を新たに選択画素として定めている。 $i_R, r_R, C_R$  を選択画素の右変形、 $i_L, r_L, C_L$  を左変形と呼ぶ。アルゴリズム 3 によって、図 4 a の画像から選ばれる選択画素は図 4 b となる。一般に、斜め  $45^\circ$  の直線上に

```

1 for (i,j)←(1,1) until (M,N) do
  begin
2   n←0;
3   if (i,j) は選択画素 S=(i,r,C) である then
     begin
4     X←S;
5     if (ρ(i,j)=ρ(i+1,j-1) かつ ρ(i+1,j-1) は変化点) then
       begin
6         X を左変形 SL に変える;
7         LEFT[j]←1
8       end
9     else if (ρ(i,j)=ρ(i+1,j+1) かつ ρ(i+1,j+1) は変化点) then
       begin
10        X を右変形 SR に変える;
11        RIGHT[j]←1
12      end;
13    n に 1 を加え、ENCODE[n]←(i,r,X) とする
14  end;
15  if j=N then RUNSHIFT(i)
16 end
17 procedure RUNSHIFT(i)
18 begin
19   for j←until N do
20     if (i,j) は変化点でない then RIGHT[j] と LEFT[j] を 0 とする;
21   for j←1 until N-1 do LEFT[j]←LEFT[j+1];
22   LEFT[N]←0;
23   for j←1 until N do
24     if (LEFT[j]=1 または RIGHT[j]=1) then ρ(i,j)←1-ρ(i,j);
25   for j←N step -1 until 2 do RIGHT[j]←RIGHT[j-1];
26   RIGHT[1]←0
27 end

```

図 5 アルゴリズム 3: SECT 符号化過程

Fig. 5 Algorithm 3: Encoding process of SECT.

ある  $i$  点または  $r$  点は先頭の一つを除いて省かれることに注意する。

アルゴリズム 4 (復元化過程): リスト ENCODE を入力とし、原画像  $P$  を出力する。アルゴリズム 3 と同様に初期値 0 の長さ  $N$  の配列、LEFT と RIGHT を用いる。アルゴリズムはアルゴリズム 2 に次の動作を付け加える。

- ① ステップ 3 で  $X_n$  を調べる際、 $X_n$  がある選択画素の右変形なら RIGHT [ $J_n$ ]←1、あるいは左変形なら LEFT [ $J_n$ ]←1 とおく。
- ② 同様にステップ 10 で  $X_{n+1}$  を調べる際に、 $X_{n+1}$  がある選択画素の右変形なら RIGHT [ $J_{n+1}$ ]←1、あるいは左変形なら LEFT [ $J_{n+1}$ ]←1 とおく。
- ③ ステップ 8 において各  $i$  行 ( $I_n \leq i \leq I_{n+1}$ ) の復元が終わった時点で RUNSHIFT ( $i$ ) を実行する。

命題 6 アルゴリズム 4 は、ENCODE から元の画像を正確に復元する。

(証明) ENCODE に登録されている最初の右変形あるいは左変形を  $(I_i, J_i)$  とする。 $(1, 0)$  から  $(I_i, N+1)$  までのアルゴリズム 4 の動作は、 $I_i$  行において右変形と左変形がリスト ENCODE から取り出される度ごとに、RIGHT および LEFT の対応する位置を 1 にすることを除いては、アルゴリズム 2 のものと本質的に同じである。したがって、 $(1, 1)$  から  $(I_i, N)$  までは正確に復元できる。 $I_i+1$  行の復元の際には、 $I_i$  行の値は RUNSHIFT ( $I_i$ ) によって修正されているが、符号化時においてもこれと同じ修正が行われているので、復元された  $I_i+1$  行の画素値は、元のものに等しい。 $I_i+2$  以降についても、上の議論を繰返し用いることにより、正確に復元できることが示される。(証了)

### 3.2 漢字パターンの圧縮

SECT の適用例として漢字パターン圧縮を行う。比較のために、漢字パターン圧縮においてよく知られた次の符号化方式による圧縮も行う。

#### 1) 1 次元ランレングス符号化方式

2 値画像  $P$  の各行の 1 ランと 0 ランの長さを、それらの頻度分布に基づいて構成された 2 種類のハフマン符号によって符号化する。

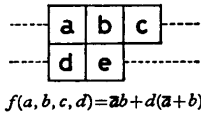


図 6 実験で用いた予測関数

Fig. 6 A prediction function used in the experiment.

2) 2次元予測符号化方式

まず、各画素の値を近傍の画素値の関数によって予測する (図 6 参照)。この関数は実測の漢字パターンにおいて、(a, b, c, d) という近傍のもとで大きい確率頻度をもつ値を表している。もし、画素 (i, j) の予測値が  $p(i, j)$  と等しければ 0, 異なれば 1 とする。結果的に、もとの画像 P はある 0-1 行列 E に変換される。次にこの行列 E を 1 次元ランレングス符号によって符号化する。

3) ブロック符号化方式<sup>16)</sup>

画面を縦・横  $n \times m$  の小ブロックに分割して、すべての画素値が 0 である小ブロックには 0, それ以外のブロックには長さ  $nm$  の系列の前に 1 をつけ加えた長さ  $nm+1$  の系列を対応させる。さらに圧縮効果を上げるために、各小ブロックを一つのシンボルとみなし、実験データから得られた頻度分布をもとにしてハフマン符号で符号化する方法も試みる。

4) SECT

(a) 直接アドレス法: 選択画素の位置を縦・横合わせて、 $\lceil \log_2 M \rceil + \lceil \log_2 N \rceil$  ビットで符号化する。選択画素の種類を符号化においては、r 点と C 点に同じ符号語を割り当てる。なぜなら、もし (i, j) が r 点なら、定義より  $p(i-1, j-1) = p(i-1, j)$ 、一方、(i, j) が C 点なら、 $p(i-1, j-1) \neq p(i-1, j)$  であるので、前行の値を参照することにより (i, j) が r 点か C 点かの区別がつく。右変形と左変形の場合も同様である。したがって、選択画素の種類には、 $i \dots 000, r, C \dots 001, i_R \dots 010, i_L \dots 011, r_R, C_R \dots 100, r_L, C_L \dots 101$  と各 3 ビットの符号語を対応づける。

(b) 行番号消去法: (a) において、行番号の表示を省略する。その代わりに、行末の選択画素の列番号の後には行の終りを示すエンド・マーク 11 を挿入する。

(c) エンドマーク消去法: (b) において、ある行の最後 (右端) の選択画素の列番号を  $J_E$ 、次の行の最初 (左端) の選択画素の列番号を  $J_S$  とするとき、もし、 $J_S \leq J_E$  なら、次の行に移行したことがわかるので、その間のエンドマークを省略する。

実験に用いた画像データは当用漢字のなかから無作

表 1 各種符号化方式の圧縮効果の比較

Table 1 Comparison of data compression performance for some encoding schemes.

符号化方式	1次元ランレングス符号化方式	2次元予測符号化方式	ブロック符号化方式		SECT		
			2×2	2×4	a	b	c
平均符号化ビット数	568	547	650 606	666 503	752	527	487

為に抽出された 115 個の漢字パターンで、これを自作の画像入力装置により、 $32 \times 32$  のデジタル 2 値画像に変換した。この画像データを SECT および 1)~3) の各種方式により符号化した結果を表 1 に示す。ここで、ブロック符号の欄の値は、上が本来のブロック符号化によるもので、下はハフマン符号を用いた場合のものである。SECT の欄の a, b, c は順に、直接アドレス法、行番号消去法、エンドマーク消去法による場合に相当する。表から、エンドマーク消去法を用いた SECT がいちばん高い圧縮効果を挙げており、SECT が実用面からも有意義な符号化方式であることが示された。

4. 連結成分の検出への応用

4.1 連結成分およびホールの定義

連結成分の定義の仕方には、4-連結あるいは 8-連結による方法の 2 種類がある<sup>17)</sup>。4-連結の場合は、一つの画素とそれを囲む上下左右の画素とが同じ値をもつかどうかで連結性が定められる。一方、8-連結の場合はこれに対角方向の 4 画素が加わる。たとえば、4-連結による連結成分の定義によれば、図 1 (a) には 2 個の連結成分が含まれるとみなされるが、8-連結の場合は連結成分の数は 1 個である。ところで、図 1 (b) のような画像に対し連結成分とホールの両方を同じ 4-連結あるいは 8-連結で定義すると矛盾が生じるので、両者を互いに異なる連結性によって定義する必要がある。したがって以後、連結成分を 4-連結 (8-連結) で定義する場合は自動的にホールは 8-連結 (4-

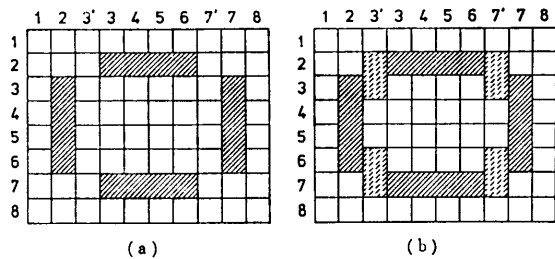


図 7 交差点の取扱い方

Fig. 7 The treatment of cross-point elements.

表 2 各連結性における交差点の解釈

Table 2 The interpretation of cross-point elements in each of 4- and 8-connectivities.

	4-連結	8-連結
1-交差点	CR/TI	TR/CI
0-交差点	TR/CI	CR/TI

連結)で定義されるものとする。

連結成分を 4-連結で定義するか 8-連結で定義するかによって交差点 (C点) の意味に違いが生じる。

例として、図 1 (b) で第 2 列と第 3 列の間と、第 6 列と第 7 列の間に 3' 列と 7' 列が存在するものとして図 7 に示す二つの画像を考える。このとき、4-連結性 (8-連結性) の定義のもとでは、図 7 (a) (同図 (b)) に含まれる連結成分とホール数は元の図形のそれに等しい。すなわち、この修正によって図形の連結性は保存されて、元の画像の C 点は修正した画像の R 点と

I 点が重なったものとしてみなすことができる。このとき、一つの C 点に対応する R 点と I 点は連結性の定義と C 点の画素値によって変化点である場合とそうでない場合に分かれる。たとえば、連結成分を 8-連結で定義した場合、C 点の画素値が 0 ならば、この C 点は左側に存在する I 点 (あるいは C 点) に対しては CR 点で、右側に存在する R 点 (あるいは C 点) に対しては TI 点である。このことを、CR/TI と記す。連結性と C 点の画素値のすべての組合せに対する C 点の解釈の仕方を表 2 にまとめておく。

#### 4.2 連結成分の検出

アルゴリズム 5: 2 値画像  $P$  に含まれる I 点, R 点と C 点の位置  $(I_n, J_n)$  と種類  $X_n (=TI, TR, CI, CR, \text{または } C)$ , さらに  $p(I_n, J_n)$  の値  $V_n, n=1, \dots, 2K$  を入力とする。ここで C 点に関しては連結性の定義に従って表 2 に示す二つの選択画素が重なりあっておりとみなし、添字  $n$  も C 点においては 2 度数える。出力は

```

1 for n←1 until K do
  begin
2     if  $(X_{1n-1}=TI \text{ かつ } X_{1n}=TR)$  then  $B[J_{1n-1}]$  と  $B[J_{1n}]$  をそれぞれ、1 と -1 にする;
3     if  $(X_{1n-1}=TI \text{ かつ } X_{1n}=CR)$  then
4         begin
5              $B[J_{1n-1}]←B[J_{1n}]$ ;
6              $B[J_{1n}]←0$ ;
7         end;
8     if  $(X_{1n-1}=CI \text{ かつ } X_{1n}=TR)$  then
9         begin
10             $B[J_{1n}]←B[J_{1n-1}]$ ;
11             $B[J_{1n-1}]←0$ ;
12        end;
13    if  $(X_{1n-1}=CI \text{ かつ } X_{1n}=CR)$  then
14        begin
15            if  $(B[J_{1n-1}]=1 \text{ かつ } B[J_{1n}]=-1)$  then
16                if  $V_{1n}=0$  then  $CCOUNT←CCOUNT+1$ 
17                else  $HCOUNT←HCOUNT+1$ ;
18            if  $(B[J_{1n-1}]=1 \text{ かつ } B[J_{1n}]=1)$  then  $B[\text{MIN}(J_{1n})]←-1$ ;
19            if  $(B[J_{1n-1}]=-1 \text{ かつ } B[J_{1n}]=-1)$  then  $B[\text{MAX}(J_{1n-1})]←-1$ ;
20             $B[J_{1n-1}]$  と  $B[J_{1n}]$  を 0 にする
21        end
22    end
23  procedure MIN(J)
24  begin
25      SUM←B[J];
26      j←J;
27      repeat j←j+1 until  $(SUM←SUM+B[j])=0$ ;
28      return (j)
29  end
30  procedure MAX (J)
31  begin
32      SUM←B[J];
33      j←J;
34      repeat j←j-1 until  $(SUM←SUM+B[j])=0$ ;
35      return (j)
36  end

```

図 8 アルゴリズム 5: 連結成分とホールの検出アルゴリズム

Fig. 8 Algorithm 5: Finding algorithm of connective components and holes.

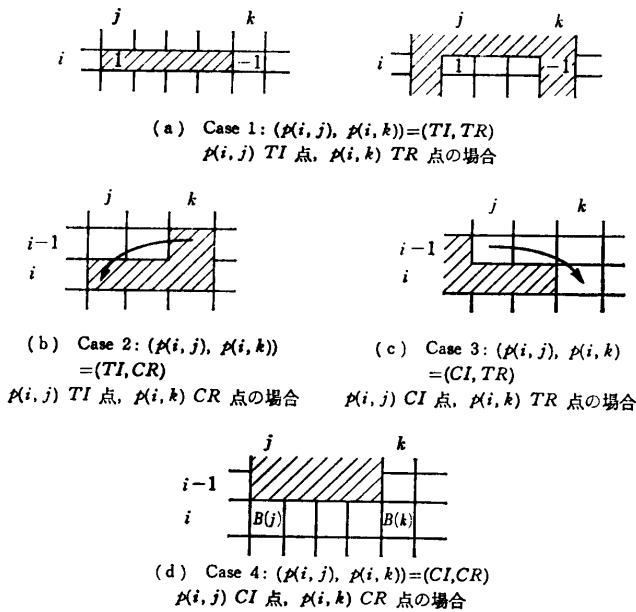


図 9 選択画素と局所的な画像との関係

Fig. 9 The relation of selective elements and local image.

連結成分の個数 CCOUNT とホール の個数 HCOUNT である。このアルゴリズムでは長さ  $N$  の 1 次元配列  $B$  を使用する。  $B$  は連結成分あるいはホールの境界の位置を示す。各配列要素は  $-1, 0, 1$  のいずれかの値をとるものとし、その初期値はすべて 0 とする。アルゴリズムを図 8 に示す。

**命題 7** アルゴリズム 5 は 2 値画像  $P$  に含まれる連結成分とホール の数を正確に求めることができる。

(証明)  $(I_{2n-1}, J_{2n-1})$  をある  $I$  点の位置、  $(I_{2n}, J_{2n})$  を次の  $R$  点の位置とおく  $(I_{2n-1} = I_{2n})$ 。  $(I_{2n-1}, J_{2n-1})$  と  $(I_{2n}, J_{2n})$  が変化点かそうでないかによって  $(I_{2n-1}, J_{2n-1})$  と  $(I_{2n}, J_{2n})$  付近の図形は図 9 に示すように、 0 と 1 の反転画像を除けば一意に確定する。アルゴリズム 5 のステップ 2 ~ 8 までは連結成分およびホールの上端を見いだして、左側の境界と右側の境界を同時に追跡する動作を行う。問題は、アルゴリズム 5 のステップ 9 の条件が成立するとき、すなわち、図 9 (d) のようにある連結成分またはホールの下端に走査が及んだ場合の動作の正当性を示すことにある。

$i$  行のある選択画素  $(I_{2n-1}, J_{2n-1})$  と  $(I_{2n}, J_{2n})$ 、  $I_{2n-1} = I_{2n} = i$  において、アルゴリズム 5 のステップ 9 の条件が成立したとする。このとき、前行までの処理によって、  $B[J_{2n-1}]$  と  $B[J_{2n}]$  はいずれも 0 でない値をとる。一般に、  $B[j] = 1$  ならば、画素  $(i, j)$  は第 1 行から第  $i$  行までの範囲内で境界をたどることによりあ

る連結成分あるいはホールの上端の左側に到達できる。また、  $B[k] = -1$  ならば、画素  $(i, k)$  は同じ範囲内である連結成分あるいはホールの上端の右側に到達できる。画素  $(i, j)$  と  $(i, k)$  が第 1 行から第  $i$  行までの範囲である連結成分あるいはホールの上端を介して境界部分をたどることによって互いに到達できることをたんに到達可能という。連結成分とホールは入れ子状にしか配置できないことより、どの  $B[j] = 1$  なる  $(i, j)$  と、どの  $B[k] = -1$  なる  $(i, k)$  とが到達可能であるかの判定は次のようにして行える。すなわち、  $B[j] = 1$  なる  $(i, j)$  に対しては、ある区間  $[j, t]$  内で  $B$  の 1 と  $-1$  の数が等しくなる最小の  $t$  を求める。この  $(i, t)$  が  $(i, j)$  から到達可能な画素である。また、  $B[k] = -1$  なる  $(i, k)$  に対しては、ある区間  $[s, k]$  内で  $B$  の 1 と  $-1$  の数が等しくなる最大の  $s$  を求める。この  $(i, s)$  が  $(i, k)$  から到達可能な画素である。アルゴリズム内では、これらの操作は手続き MIN と MAX によって行われる。も

し、  $B[J_{2n-1}] = 1$  かつ  $B[J_{2n}] = -1$  ならば、上の議論から  $(I_{2n-1}, J_{2n-1})$  と  $(I_{2n}, J_{2n})$  は互いに到達可能なので、一つの連結成分かホールが存在する。いずれであるかは  $V_{2n}$  の値によって判定される。これ以外の場合には、  $(I_{2n-1}, J_{2n-1})$  から到達可能な画素  $(i, j)$  と  $(I_{2n}, J_{2n})$  から到達可能な画素  $(i, k)$  とが互いに到達可能であるので、次にステップ 9 の条件が満足されるまで、  $(i, j)$  と  $(i, k)$  から境界の追跡を行ってゆく必要がある。このとき、  $B$  の定義から、  $(i, j)$  と  $(i, k)$  のうち左側に位置するものの  $B$  の値を 1、右側に位置するものの  $B$  の値を  $-1$  としておく必要がある。このためには、もし、  $B[J_{2n-1}] = 1$  かつ  $B[J_{2n}] = 1$  ならば、  $B[\text{MIN}(J_{2n})]$  を 1 に変え、また、もし、  $B[J_{2n-1}] = -1$  かつ  $B[J_{2n}] = -1$  ならば  $B[\text{MAX}(J_{2n-1})]$  を  $-1$  に変えればよい。アルゴリズム 5 のステップ 10 ~ 15 ではこの要求を満たす処理を行っている。したがって最終的には、すべての連結成分とホール の数をアルゴリズム 5 によって求めることができる。(証了)

アルゴリズム 5 の特徴としては、

- ① 検出に必要な選択画素の情報を得るためには画像を正確に一度走査すればよい、
- ② 処理に必要な記憶容量は画像の一部を記憶するのに  $2N$  ビット、配列  $B$  を記憶するのに  $2N$  ビットである、
- ③ 4-連結と 8-連結による連結成分の定義の違い



に対しては、交差点の解釈を変更するだけでよく、アルゴリズムの記述を変える必要はない、の3点を挙げることができる。

## 5. む す び

画像を特徴づける画素として選択画素を定義し、この選択画素を用いた符号化方式 SECT を提案した。SECT はライン間の相関を利用して符号化すべき画素をできるだけ少なくする符号化方式である。SECT を漢字パターンの符号化に適用し、この方式がデータ圧縮に対し有効であることを示した。さらに、選択画素に含まれる図形の位相的性質を利用して、符号化データのままで、従来明確な形で与えられていなかった逐次型境界追跡方式による連結成分の検出法を導出した。なお、本文中では言及しなかったが、図形の平行移動や回転、拡大、縮小等の操作も符号化データを処理して容易に行うことができる。選択画素はさまざまな画像処理を行う上で基本的な要素として考えることができ、今後さらに多くの図形処理技術への応用が期待できる。最後に、画像入力装置の製作に協力していただいた北島浩二氏（現東芝）と選択画素の定義に関して有意義なご指摘をいただいた金沢工業大学教授林彬氏に深く感謝します。

## 参 考 文 献

- 1) Frank, A. J.: High Fidelity Encoding of Two-Level, High Resolution Images, *ICC '73, Conf. Rec.*, pp. 26.5-26.10 (1973).
- 2) 遠藤, 河口: DF-画像表現の性質と情報圧縮への応用, *信学論 (D)*, Vol. J62-D, No. 2, pp. 141-148 (1979).
- 3) Meyr, H., Rodolsky, H. B. and Huang, T. S.: Optimum Run-Length Codes, *IEEE Trans. Comm.*, Vol. COM-22, No. 6, pp. 826-835 (1974).
- 4) Mussmann, H. G. and Preuss, D.: A Redundancy Reducing Facsimile Coding Scheme, *Nachrichtentech. Z.*, Vol. 26, No. 2, pp. 91-94 (1973).
- 5) 高木, 津田: 二次元予測を用いたファクシミリ帯域圧縮, *信学論 (D)*, Vol. J56-D, No. 3, pp. 170-177 (1973).
- 6) 若原, 山崎, 寺村, 中込: ファクシミリ信号の変化点相対アドレス符号化方式の圧縮率, *画像電子*, Vol. 5, No. 3, pp. 92-100 (1976).
- 7) Yamada, T.: Edge-Difference Coding—A New, Efficient Redundancy Reduction Technique for Facsimile Signals, *IEEE Trans.*

*Comm.*, Vol. COM-27, No. 8, pp. 1210-1217 (1979).

- 8) Yasuda, Y.: Overview of Digital Facsimile Coding Techniques in Japan, *Proc. IEEE*, Vol. 68, No. 7, pp. 830-845 (1980).
- 9) 森田, 有本: デジタル画像の選択符号化方式, *情報処理学会第22回全国大会*, pp. 607-608 (1981).
- 10) Freeman, H.: On the Encoding of Arbitrary Geometric Configurations, *IRE Trans.*, EC-10, No. 2, pp. 260-268 (1961).
- 11) Pratt, W.: *Digital Image Processing*, pp. 542-546, John & Wiley, Canada (1975).
- 12) Rosenfeld, A.: Connectivity in Digital Picture, *ACM*, Vol. 17, No. 1, pp. 146-160 (1970).
- 13) 横井, 鳥脇, 福村: 2値図形収縮のための逐次形アルゴリズムについて, *信学論 (D)*, Vol. J62-D, No. 8, pp. 537-542 (1979).
- 14) Rosenfeld, A. and Milgram, D. L.: Parallel/Sequential Array Automata, *Information Processing Letters* 2, pp. 43-46 (1973).
- 15) Aho, A. V. et al.: *The Design and Analysis of Computer Algorithms*, pp. 33-39, Addison-Wesley, Reading, MA (1974).
- 16) Kunt, M. and Johnsen, O.: Block Coding of Graphics: Tutorial Review, *Proc. IEEE*, Vol. 68, No. 7, pp. 770-786 (1980).
- 17) Pratt, W.: *Digital Image Processing*, pp. 514-517, John & Wiley, Canada (1975).

## 付 録 (命題2の証明)

縦辺の上端の画素  $(I, J)$  は変化点であり、しかも、 $(p(I-1, J-1), p(I-1, j)) \neq (p(I, J-1), p(I, J))$  を満足するので、選択画素  $i, r, C$  点のいずれかである。このことと縦辺の数が  $L_p/2$  であることより左側の不等号が導かれる。一方、縦辺の真下の画素  $(K, L)$  は、 $C$  点、 $CI$  点、 $CR$  点のいずれかに限られる。 $(K, L)$  が  $C$  点ならば、それはすでにある縦辺の上端に対応づけられている。また、 $(K, L)$  が  $CR$  点であれば定義から  $r$  点ではない。そこで、 $(K, L)$  が  $CI$  点のみならず  $i$  点であると仮定する。この場合、定義から  $(K, L)$  の右隣りある縦辺の下に位置する  $CR$  点  $(i, k)$  が存在する。これは、 $(K, L)$  に対し、二つの縦辺が対応することを意味する。すなわち、縦辺の上端に対応していない選択画素の数はたかだか  $L_p/4$  である。したがって、選択画素  $i$  点、 $r$  点、 $C$  点の総数はたかだか  $3L_p/4$  である。

(証了)

(昭和57年7月14日受付)

(昭和58年4月19日採録)