

## 画素間の拘束関係にもとづく連想法

合田智一, 石川聖二, 加藤清史

(九州工業大学工学部)

あらまし コンピュータビジョンにおける重要な課題の一つにオクルージョン問題がある。その解決の手がかりとして脳の基本的機能である連想に的を絞り、画素間の局所的な拘束を利用した連想方式先に提案した。これは、ネットワーク拘束解析問題を抽象したコンシステントラベリングの概念を用いている。この連想方式では、局所的な拘束の取り方が連想の信頼性を左右するという問題があった。そこで本稿では、ランダムに取っていた局所領域について、信頼性の高い連想を行うための局所領域の設定法を与えた。また連想される画像パターンは、縦型探索を用いて求めていたが、これは時間のかかる処理なので、縦型探索の回数を減らすための処理法を提案した。これらの手法に基づき2値画像パターンの連想実験を行い、満足のいく結果を得た。

## An Association Technique Based on the Constraint Relation among Pixels

Tomokazu GODA, Seiji ISHIKAWA, Kiyoshi KATO

Department of Electrical, Electronic and Computer Engineering  
Kyushu Institute of Technology  
Tobata, Kitakyusyu, 804 JAPAN

Abstract The occlusion problem is one of the unsolved difficulties in computer vision. As a possible cue for solving this difficulty, we have focused our attention on association which is one of the basic functions of the brain, and proposed a new technique for associating images employing sets of local constraints among pixels on an image. This technique utilizes the concept of consistent labeling abstracted from individual constraint satisfaction problems. But it was not easy to select those local constraints that may not cause interference among memorized images when recollecting. In the present paper, we propose a selection scheme of the local constraints in order to realize noninterference among memorized images. An associated image is given in the form of the solution of depth first search, which may cause time explosion. Therefore we also propose a technique for reducing the amount of the depth first search to a large extent. Simulation of the association based on these proposals was performed employing synthetic binary images, and the result was satisfactory.

## 1. はじめに

脳の情報処理機能を工学的に実現する為の研究は、人工神経回路網理論<sup>1)</sup>や知識工学<sup>2)</sup>として、近年活発に行われている。本稿では、脳の基本的機能の一つである連想を取り上げる。連想はコンピュータビジョンにおけるオクルージョン問題等の解決につながる重要な機能である。連想機能の工学的実現の試みは、既に Nakano<sup>3)</sup>, Kohonen<sup>4)</sup>, Hirai<sup>5)</sup>によって人工的な神経回路網の立場から行われているが、我々は、ラベリング問題という立場からのアプローチを試みる。我々は、画像上の画素間の局所的な拘束関係を利用し、コンシステントラベリングの概念を用いて連想を実現する方式を提案した<sup>6)</sup>が、局所領域の取り方によっては誤った連想が生じることがあり、信頼性に問題があった。本稿では、2値画像パターンに対する、連想方式、連想パターンの信頼性を保証する局所領域の取り方についての理論、及びそれに基づく連想実験の結果について報告する。また本連想方式においては、縦型探索を用いて連想パターンを求めるが、これは時間のかかる処理である。本稿ではその探索量を減らすための処理についても提案する。

## 2. コンシステントラベリングを用いた連想方式

ここで問題としている連想は、連想の中でも自己想起と呼ばれる部類のものである。これはあらかじめ何らかの処理を施されて、メモリに記録されているパターンに対して、そのパターンの一部を入力すると、そのパターンの残りの部分が想起されるという形態である。

ネットワーク拘束解析問題の一つの表現であるコンシステントラベリング問題<sup>7)</sup>とは与えられた対象をユニット  $u$ 、その集合をユニット集合  $U$ 、ユニットに付けられる名前をラベル  $l$ 、その集合をラベル集合  $L$ 、相互に拘束されたユニットの  $n$  組をユニット拘束関係  $t$ 、その集合をユニット拘束集合  $T$ 、またそれにラベルも兼ね合わせユニット・ラベルのペアの  $n$  組をユニット・ラベル拘束関係  $r$ 、その集合をユニット・ラベル拘束集合  $R$  としたとき、全てのユニットに対して無矛盾なラベル付け、言い換えればコンパチビリティモデル  $(U, L, T, R)$  に対するコンシステントラベリングを求めることである。

$U$  を  $N_u$  個のユニットの集合、

$$U: \{u_1, u_2, \dots, u_{N_u}\}$$

$L$  を  $N_l$  個のラベルの集合、

$$L: \{l_1, l_2, \dots, l_{N_l}\}$$

とすると

$T, R$  は次のように表現される。

$$T: \{(u_{i1}, u_{i2}, \dots, u_{in}) \mid u_i \in U\} \subseteq U^n$$

$$R: \{(u_{i1}, l_{i1}, u_{i2}, l_{i2}, \dots, u_{in}, l_{in}) \mid u_i \in U, l_i \in L\} \subseteq (U \times L)^n$$

図1においてユニット、ラベル、ユニット拘束関係、及びユニット・ラベル拘束関係の概念を表す。ユニット1、

2, 3, 4 に対してラベル  $a, b, c, d$  が与えられている。

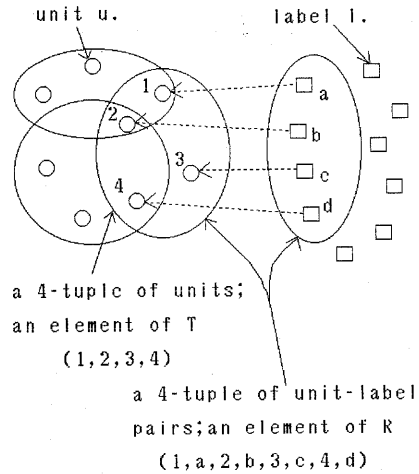


図1 コンシステントラベリングの概念

コンシステントラベリングを用いた連想方式は以下のように表される。デジタル画像  $I$  は、画素  $(i, j)$  とその濃度値  $f_{ij}$  のペアの集合、即ち  $I = \{(k, f_k) \mid k = (i-1)N + j; i=1, 2, \dots, M; j=1, 2, \dots, N; M, N$  は正の整数) と考えることができる。画像  $I$  上で  $n$  個の画素  $k_1, k_2, \dots, k_n$  をとれば、 $(k_1, f_{k_1}), (k_2, f_{k_2}), \dots, (k_n, f_{k_n})$  は画像  $I$  に関して互いに拘束し合う。例えば、図2において  $\square$  の中の画素に付いて左から  $k_1, k_2, k_3, k_4, k_5$  とすると、 $(k_1, \square), (k_2, \square), (k_3, \bullet), (k_4, \square), (k_5, \square)$  という拘束があるとみることができる。

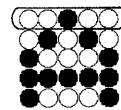
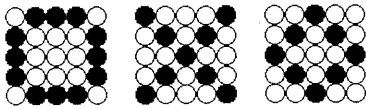


図2 画素間の拘束

そこで、画素  $k$  をユニット  $u$  ( $u \in U$ )、濃度値  $f_k$  をラベル  $l$  ( $l \in L$ )、 $n$  組の画素  $\tau_n = (k_1, k_2, \dots, k_n)$  をユニット拘束関係  $t$  ( $t \in T$ )、 $n$  組の画素・濃度値のペア  $r_n = (k_1, f_{k_1}, k_2, f_{k_2}, \dots, k_n, f_{k_n})$  をユニット・ラベル拘束関係  $r$  ( $r \in R$ ) とする。ユニット拘束集合  $T$  に対して画像  $I$  に関するユニット・ラベル拘束集合  $R_i$  とするとき  $N$  個の画像が記録された状態は、 $R = \bigcap_{i=1}^N R_i$  と表せる。

図3 (a) の3種類の2値画像パターンに対する記録状態例を図3 (b) に示す。但し  $U = \{1, 2, \dots, 25\}$  とし  $L = \{-1$  (白),  $1$  (黒) とする。



(a)

$T = \{(1, 2, 6, 7),$   
 $(2, 3, 8, 9),$   
 $\dots\}$   
 $R_o = \{(1, -1, 2, 1, 6, 1, 7, -1),$   
 $(2, 1, 3, 1, 8, -1, 9, -1),$   
 $\dots\}$   
 $R_x = \{(1, 1, 2, -1, 6, -1, 7, 1),$   
 $(2, -1, 3, -1, 8, -1, 9, 1),$   
 $\dots\}$   
 $R_\diamond = \{(1, -1, 2, -1, 6, -1, 7, 1),$   
 $(2, -1, 3, 1, 8, -1, 9, 1),$   
 $\dots\}$   
 $R = R_o \vee R_x \vee R_\diamond$   
 $= \{(1, -1, 2, -1, 6, -1, 7, 1),$   
 $(1, -1, 2, 1, 6, 1, 7, -1),$   
 $(1, 1, 2, -1, 6, -1, 7, 1),$   
 $(2, -1, 3, 1, 8, -1, 9, 1),$   
 $(2, 1, 3, 1, 8, -1, 9, -1),$   
 $(2, -1, 3, -1, 8, -1, 9, 1),$   
 $\dots\}$   
 (b)

図3 画像パターンの記銘状態

(U, L, T, R) は画像 I のコンパチビリティモデルによる表現とみなせる。ユニット・ラベル拘束集合 R から集合 U 内の各ユニットに対して無矛盾なラベル付けが求められたとき、これは (U, L, T, R) に対するコンシステントラベリングとなり、これが連想されるパターンである。今、ある記銘画像を与えるユニット・ラベル拘束集合 R に対し、不完全な入力画像 X に矛盾しない R の要素の集合を  $R^*$  ( $\subseteq R$ ) で表す。R の要素を  $r = (u_{r1}, l_{r1}, u_{r2}, l_{r2}, \dots, u_{rn}, l_{rn})$  で表し、X の要素を  $x = (u_{x1}, l_{x1}, u_{x2}, l_{x2}, \dots, u_{xn}, l_{xn})$  と表し、また  $s = (u_{s1}, l_{s1}, u_{s2}, l_{s2}, \dots, u_{sn}, l_{sn})$  に対し、 $s \mid \tau = (u_{s1}, u_{s2}, \dots, u_{sn})$  と定義すれば、R の要素が X に矛盾するとは  $r \mid \tau = x \mid \tau$  であるようなある x に対し、 $l_{ri} \neq l_{xi}$  かつ  $l_{xi} \neq \text{nil}$  であるような i が存在することである。

ユニット・ラベル拘束集合  $R^*$  を生成するためのアルゴリズム REMOVE を以下に示す。

```

REMOVE
begin
  copy R to R*;
  for all r ∈ R* do
    for all x ∈ X do
      if (r | τ = x | τ)
        then if (lri ≠ lxi) and (lxi ≠ nil)
          then remove x from R*
    end.
end.
  
```

集合  $R^*$  によるコンパチビリティモデル (U, L, T,  $R^*$ ) に対するコンシステントラベリングは、 $R^*$  に縦型探索を適用することによって求められる (図4 参照)。

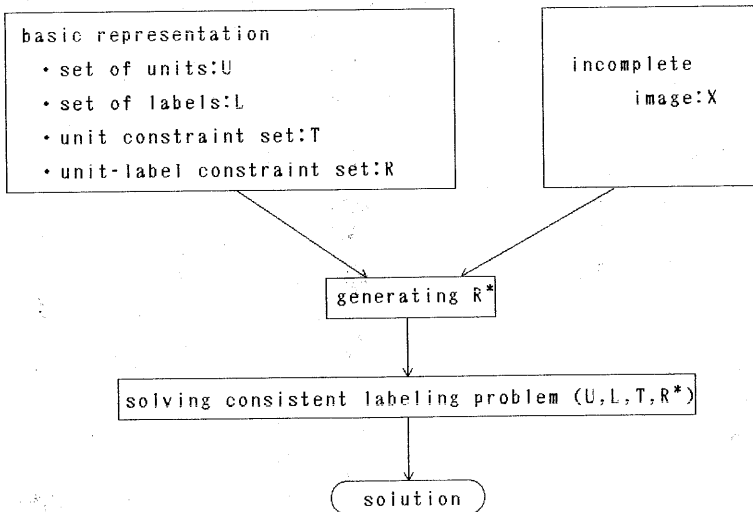


図4 コンシステントラベリングによる連想手順

### 3. フィルタリング

コンシステントラベリングの解は $R^*$ に対する縦型探索によって求められるが、これは時間のかかる処理であり、その時間は当然 $R^*$ の要素数が増えるにつれ増大する。そこである画素について、画素値が決定するたびにREMOVE処理を行い $R^*$ を更新することにより、 $R^*$ の要素数を減少させる処理を導入する。画素に対し画素値が決定する、即ちユニットに対してラベルが付けられることを $u \leftarrow 1$ で表す。またユニット拘束集合の部分集合 $T(u)$ を

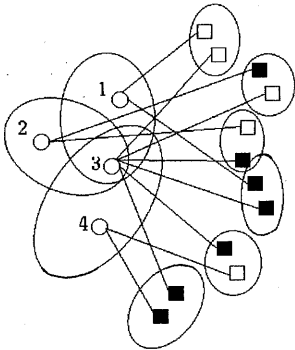
$$T(u) = \{t \mid t \in T, t \text{ contains } u\}$$

のように選ぶ。これはユニット $u$ を含む全てのユニット拘束関係の集合を意味する。このとき $u \leftarrow 1$ は以下のように定義される。

[定義]

$$u \leftarrow 1 \text{ iff } \forall t \in T(u), \exists r \in R \\ \text{s. t. } r \uparrow t = t, r = (\dots, u, 1, \dots)$$

図5にラベル付けの例を示す。ユニット3にラベル■が付けられている。



$$T = \{(1, 3), (2, 3), (3, 4)\}$$

$$R = \{(1, \square, 3, \square), (1, \blacksquare, 3, \blacksquare), \\ (2, \blacksquare, 3, \square), (2, \square, 3, \blacksquare), \\ (3, \blacksquare, 4, \square), (3, \blacksquare, 4, \blacksquare)\}$$

$$3 \leftarrow \blacksquare$$

$$\therefore (1, \blacksquare, 3, \blacksquare) \\ (2, \square, 3, \blacksquare) \\ (3, \blacksquare, 4, \square)$$

図5 ラベル付けの例

以上のようにして、画素に対して画素値が決定するが、これは並列的な処理を意図しており、ソフトウェア上で

逐次に行う場合、 $T(u)$ の検索にかなり時間がかかるものと思われ非能率的である。そこで、2値画像に対してはラベルは2種類(白または黒)なので、あるユニットを含む $t$ を全て検索するかわりに、各 $t$ について、その中のユニットでラベルが1通りしかないものについては、もう1つのラベル付けの可能性は断たれているものとして、そのラベルをユニットに割り当てるという方式を用いる。例えば図5においては(3, 4)のユニット拘束関係について、3のラベルは■の可能性しかないため、 $3 \leftarrow \blacksquare$ となる。新たにラベルを割り当てられたユニットを用いてREMOVE処理を行う。この処理をフィルタリングと呼ぶ。そのアルゴリズムFILTERを以下に示す。

#### FILTER

```
begin
  iff  $\exists t \in T,$ 
     $\forall r \in R(t),$ 
       $l_{j1} = l_{j2} = \dots = l_{jmt}$ 
    then  $u_j \leftarrow l_{j1};$  call remove;
end;
```

ここで、 $t = (u_1, u_2, \dots, u_n)$

$$R(t) = \{r \mid (u_1, l_{11}, u_2, l_{21}, \dots, u_n, l_{n1}) \\ = \{(u_1, l_{11}, u_2, l_{21}, \dots, u_n, l_{n1}) \\ (u_1, l_{12}, u_2, l_{22}, \dots, u_n, l_{n2}) \\ \dots \\ (u_1, l_{1m}, u_2, l_{2m}, \dots, u_n, l_{nm})\}$$

但し  $n(R(t)) = m_t$

以上の処理を $R^*$ が更新されなくなるまで繰り返す。そしてそのうち縦型探索を行う。またこの時すでに全ての画素について画素値が決定しているなら、縦型探索を行う必要はなく連想パターンが出力される。 $R^i$ にREMOVE処理を行ったものを $R^{i+1}$ で表し、 $X^i$ に画素値が決定した画素が加わったものを $X^{i+1}$ で表すことにすれば、フィルタリングの処理過程は、図6のように示される。但し同図で●は集合の大きさとその変化を表している。

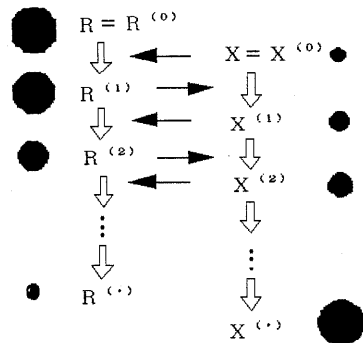


図6 フィルタリングの処理過程

#### 4. 局所領域の取り方

局所領域，即ちユニット拘束集合Tの取り方は重要である。取り方によっては記銘画像パターン同士が結合して，未記銘の画像パターンが連想されることがある。今，記銘画像パターンの中に，図7(a)に示されるような2つの画像パターンがあるとき，図7(b)の左のような不完全な入力画像パターンが与えられると，同図右上の正しい画像パターンと共に，右下のように記銘されていない画像パターンが出力されることがある。これはユニット拘束集合Tの取り方が悪く，ユニット・ラベル拘束集合R\*において縦型探索を行った際に，誤った画像パターンが無矛盾にラベル付けされる為である。

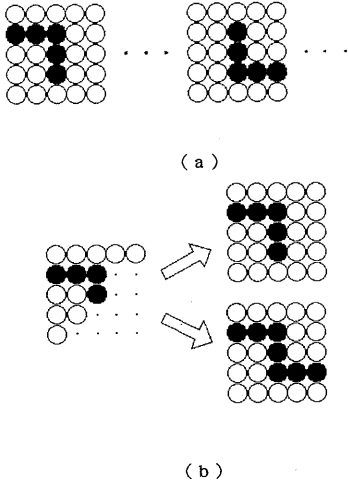


図7 正しい連想と誤った連想

記銘された2値画像パターン間の分離性を保証するTの取り方について以下の事が言える。2値画像Iの黒い画素の集合を $I_F$ で表し，N個の2値画像 $I_i (i=1, 2, \dots, N)$ に対し，

$$c = \max_{i < j} n(\{e | e \in I_{F,i} \wedge I_{F,j}\})$$

と定義する。これは画像パターン $I_1, I_2, \dots, I_N$ 間の各ペアに共有される黒い画素の最大個数である。このcを用いて $I_i$ と $I_j$ 間の分離子 $\tau_{c+2,i,j}$ を次のように定義する。

$$\tau_{c+2,i,j} = E_1 \vee E_2 \vee E_3$$

但し

- (i)  $E_1 = \{e | e \in I_{F,i} \wedge I_{F,j}\}$
- (ii)  $E_2 = \{e | e \in I_{F,i} \wedge I_{F,j}\}, n(E_2) \geq 1$
- (iii)  $E_3 = \{e | e \in I_{F,i} \wedge I_{F,j}\}, n(E_3) \geq 1$
- (iv)  $n(E_1 \vee E_2 \vee E_3) = c + 2$

ここで $n(S)$ は集合Sの要素の個数である。

また画像 $I_i$ に関する記述子 $\tau_{c+2,i,\cdot}$ を次のように定義する。

$$\tau_{c+2,i,\cdot} \subset I_{F,i}, n(\tau_{c+2,i,\cdot}) = c + 2$$

このとき次の定理が成立する。

#### [分離定理]

ユニット拘束集合Tを以下の手続きのように取れば，記銘画像パターン同士が結合して連想されることはない。

- (1) 記銘画像パターンに対し， $I_1, I_2, \dots, I_N$ の各ペア $I_{F,i}, I_{F,j}$ の分離子 $\tau_{c+2,i,j}$ を取る。
- (2) 各画像パターンの記述子 $\{\tau_{c+2,i,u} | u=1, 2, \dots, N_i\}$ を $n(\tau_{c+2,i,u} \wedge \tau_{c+2,i,v}) = c + 1$ かつ， $\forall \tau_{c+2,i,u} = I_{F,i}$ となるように取る。
- (3)  $\tau_{c+2,i,j}$ の全ての画素を含む $I_{F,k}$ があるときは， $\tau_{c+2,i,j}$ を $I_k$ の記述子 $\tau_{c+2,k,u}$ として取る。

図8に分離子，記述子の例を示す。同図(a1)は，交わり部分のある2つの画像パターンに対し，枠内の画素を分離子に取った場合である。同図(a2)はこれら2つの画像パターンに対する分離子のユニット・ラベル拘束関係を表している。同図(b2)は画像パターン(b1)に対する記述子の取り方を示している。

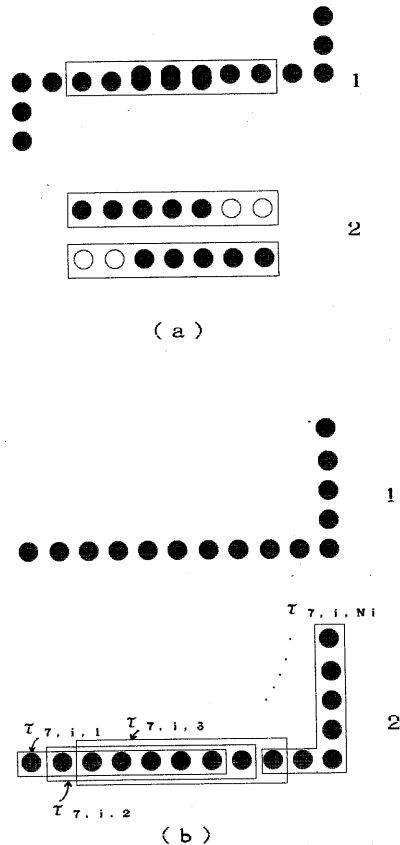


図8 分離子，記述子の例

分離定理の有効性を概論的に示す。

<証明の概略>

この証明においてはTの要素は全て $c+2$ 個の画素を持つので、 $\tau_{c+2,ij}$ 、及び $\tau_{c+2,i..}$ はそれぞれ簡単に $\tau_{ij}$ 、 $\tau_{i..}$ と表すことにする。証明の主旨は $I_i$ と $I_j$ が結合しないことを示すことにあるが、 $I_i$ と $I_j$ が結合するということは、言い替えれば $\tau_{ij}$ 上の画素が全て黒でラベル付けされるということである。ここではそれが起こり得ないことを2つの場合に分けて示す。

i) 第1のケースは図9(a)に示されるような場合である。 $I_i$ と $I_j$ が重なっていて、分離子 $\tau_{ij}$ はその重なり部分を含んでいる。分離子部分のそれぞれのパターンが同図(b)に示されている。この第1のケースにおいては $\tau_{ij}$ 上の全ての画素を含んでいるような記銘パターンはないものとする。この時、図9(a)の左端の灰色で示されている部分のように、 $I_i$ の一部分が与えられると、記述子同士は $c+1$ の交わりでつながっているので、 $\tau_{i,1}$ 、 $\tau_{i,2}$ 、...と順に連鎖していき、 $\tau_{i,N_i}$ まで達する。しかし $\tau_{j,1}$ までは連鎖しない。なぜなら $\tau_{i,N_i}$ と $\tau_{j,1}$ が同時に全て黒でラベル付けされるには $\tau_{ij}$ 上の全ての画素を含む記銘パターンが必要だからである。

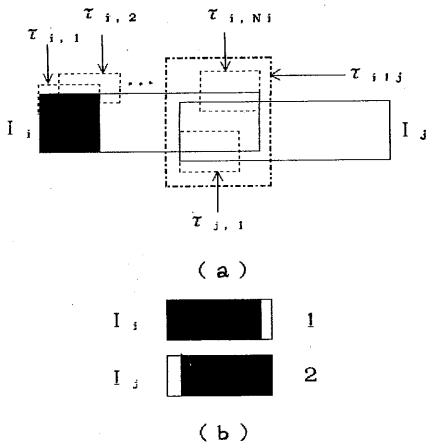


図9 i) の場合

ii) 第2のケースは図10(a)に示されているような場合である。この場合第1のケースと違い、 $\tau_{ij}$ 上の全ての画素を含んでいるような記銘パターン $I_k$ がある。しかしこの場合も $I_i$ と $I_j$ の結合が起きないことを示す。この場合3つの記述子 $\tau_{ij}$ 、 $\tau_{jik}$ 、 $\tau_{kji}$ についてそれぞれの部分のパターンは図10(b)のようになる。

$I_i$ と $I_k$ 間、 $I_j$ と $I_k$ 間についてはi)の場合と同様である。i)同様 $I_i$ の一部分が与えられた時、 $\tau_{i,1}$ 、 $\tau_{i,2}$ 、...と順に連鎖していき、 $\tau_{i,N_i}$ まで達する。しかし今度は $\tau_{ij}$ 上の全ての画素を含むパターン $I_k$ がある為、 $\tau_{j,1}$ まで連鎖可能に思えるが、図10(b)を見ると同図(b1)、同図(b4)、及び同図(b7)がコンシステントラベリングとなり同図(b3)は取られないことがわかる。同図(b3)を取ってしまうと $I_k$ が連鎖しようとするが、 $I_i$ と $I_k$ 間の分離子上の画素を全て含むパターンはないため、結局連鎖不可能となる。よって $I_i$ が他の記銘パターンと結合するようなことはない。

結局 $I_i$ と $I_j$ の分離子上の全ての画素を含むようなパターンがないときは、 $I_i$ と $I_j$ が結合して分離子上の画素が全て黒でラベル付けされることは不可能となるので結合は生じない。またそのようなパターン $I_k$ があるときも $I_k$ と $I_i$ の分離子上の全ての画素を含むようなパターンが無いため、 $\tau_{k..}$ は連鎖できない。よって $\tau_{ij} = \tau_{k,u}$ 上の画素を全て黒でラベル付けすることはできない。従って $I_i$ と $I_j$ が結合することはない。

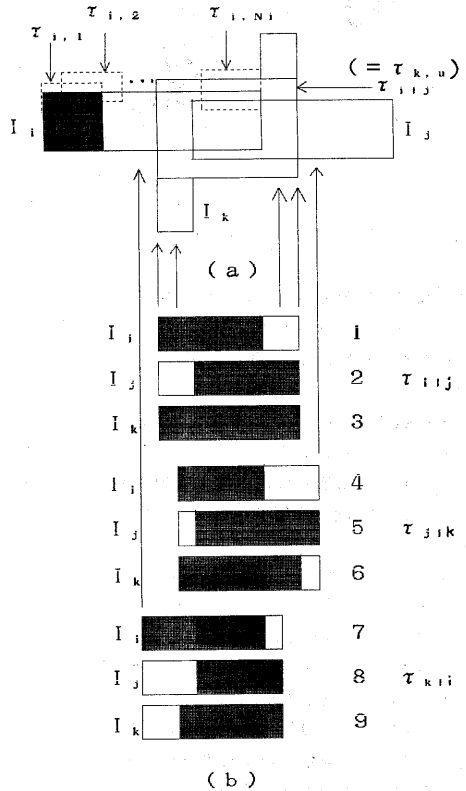


図10 ii) の場合

## 5. 記述子フラグ

分離定理に基づくユニット拘束集合  $T$  の取り方により、記銘画像パターン同士の黒画素の部分  $I_F$  が結合した誤った画像パターンは連想されないが、これは入力パターンにおいて黒で与えられた画素について、その画素を含む記述子内、どれか一つ以上が必ず全て黒でラベル付けされるという仮定に基づいている。これは言い換えれば黒で与えられた画素は、必ずあるパターンの  $I_F$  の要素であるということを表す。そこで黒で与えられた画素全てについて、フラグを設ける。そしてその画素を含む記述子がどれか一つでも全て黒になれば、フラグを立てるという方式にする。黒で与えられた全ての画素にフラグが立てば連想画像パターンが出力されるようにする。このフラグがなければ、入力画像パターンにおいて与えられた黒画素の部分以外は全て白であるような画像パターンが連想されることがある。例えば、記銘画像パターンの中に、図11(a)に示されるような3つの画像パターンがあるとき、図11(b)の左のような不完全な入力画像パターンが与えられると、同図右下の正しい画像パターンと共に、右上のように未記銘の画像パターンが出力されることがある。これは、記述子がどれも全て黒とならなかったときに、どの画像パターンも黒い部分が連想されなかった結果である。即ち、記銘画像パターン同士の白い部分が組合わされた画像パターンである。

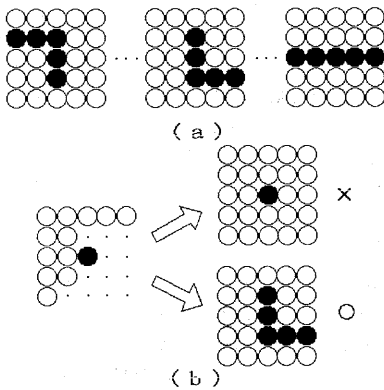


図11 正しい連想と誤った連想

## 6. 実験

5 × 5 画素の2値画像パターンを使って連想実験を行った。記銘パターンは、図12の8種類の画像パターンである。ユニットを  $u_i$  ( $i=1, 2, \dots, 25$ )、ラベルは画像が2値パターンであるから、 $L = \{-1, 0, 1\}$  とする。1が黒、-1が白、0はnil(不明)を表す。本実験では  $c = 2$  であるから、ユニット拘束集合  $T$  は画素の4つ組(4-tuple)を要素とした。分離定理に基づいて拘束集合を設定したところ、 $n(T) = 77$ 、 $n(R) = 432$  であった。実験結果の一部を図13に示す。各組とも

左の不完全入力画像パターンから連想された画像パターンが右の画像パターンである。

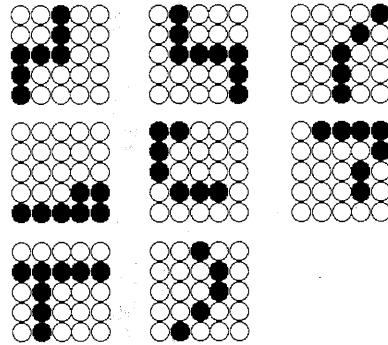


図12 実験で用いた画像パターン

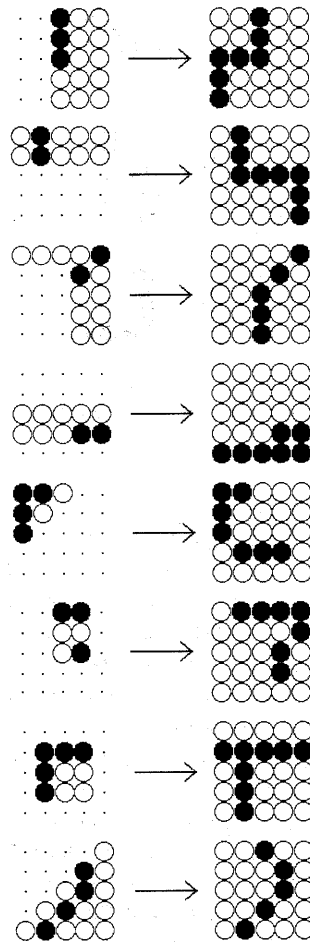


図13(a) 実験結果の一部

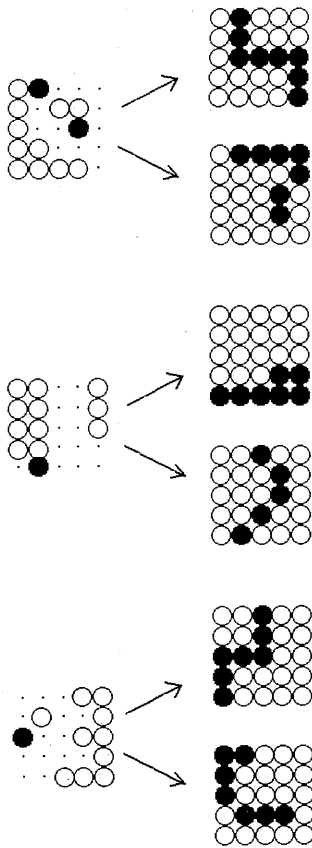


図13((b)) 実験結果の一部

## 7. 考察

実験結果を見ると図13(a)より、各パターンに固有の部分入力を与えると正しくそのパターンのみが連想され、また同図(b)より2つのパターンの共通部分を入力すると、その2つのパターンが正しく連想されていることが示されている。記銘画像パターン同士が干渉して生じる無記銘の誤ったパターンは想起されていない。この想起は、 ${}_8C_2 = 28$ 個の分離子と49個の記述子の計77個のユニット拘束関係に基づいて行われている。これは $5 \times 5$ 画素中の可能な4-tupleの組合せの数 ${}_{25}C_4 = 12,650$ 個のわずか0.61%である。実際、本実験で用いた記銘画像パターンは8個であり、多くはないが、記銘画像パターンの数が増えても分離定理に基づいて拘束関係を取らば、記銘画像パターン間の結合を防ぐことができる。また同時に記述子フラグのチェックを用いて連想を行うと、白画素に起因する記銘画像パターン間の干渉も防ぐことができ、信頼性の非常に高い連想を行うことができる。

パターン連想がコンシステントラベリング問題として

記述されることを示したが、その解は縦型探索によって求められる。記銘画像パターンの増加に伴いユニット・ラベル拘束関係が増えると縦型探索にかかる時間はかなり増えるものと思われる。本稿で提案したフィルタリング処理は、そのユニット・ラベル拘束関係の数を減らし、縦型探索の探索量を減少させるための処理であるが、1つのパターンのみ想起のときには、フィルタリング処理のみで画像パターンの想起が完了してしまうというような効果も得られた。

しかし本稿で述べられたアルゴリズムはノイズに弱いという問題がある。今後ノイズの入った部分画像パターンに対しても柔軟な連想を行うためのアルゴリズムの構築が必要となるであろう。

## 8. まとめ

画素間の局所的な拘束関係を用いて、画像の連想を行うための手法について述べ、記銘画像間に結合が生じないように、ユニット拘束関係の設定法を示した。またフラグを用いることによって、より信頼性の高い連想を実現した。2値画像パターンについて実験を行い、その動作を検討したところ満足のいく結果が得られた。コンシステントラベリングの解(連想されるパターン)は、ユニット・ラベル拘束集合 $R^*$ に対する縦型探索によって求められるため、 $R^*$ の要素数が増大すると、探索時間が増大する。そこでフィルタリング処理を導入し $R^*$ の要素数の低減を行ったが、このときフィルタリング処理のみで画像パターンが連想されるといった副次効果も得られた。今後、記銘画像パターンを増やしての連想実験、実際の画像パターンへの適用の為の並列ハードウェア化等による実行時間の短縮、及びノイズに対する除去処理が課題となる。

## 参考文献

- 1) D.E. Rumelhart and J.L. McClelland (甘利俊一訳) : PDPモデル, 産業図書(昭和61年)。
- 2) 大須賀節雄: 知識情報処理, オーム社(昭和61年)。
- 3) 中野 馨 : アソシアトロン, 昭晃堂(昭和54年)。
- 4) Kohonen, T: Correlation Matrix memories, IEEE Trans. Computers, C-21, 4, 353-359(1972)。
- 5) Hirai, Y: A model of human associative processor (HASP), IEEE Trans. Syst. Man, Cybern, SMC-13, 5, 851-857(1983)。
- 6) 黒川, 石川, 加藤: コンシステントラベリングを用いたパターン連想, 情報処理学会論文誌, 31, 3, 511-515(1990)。
- 7) R.M. Haralick and L.G. Shapiro : The Consistent Labeling Problem: Part 1, IEEE Trans. Pattern Anal. Mach. Intell., PAMI-1, 173-184(1979)。