

## 正規化相関演算の単調関数化による高速テンプレートマッチング

池田 光二<sup>†</sup>      吉田 昌司<sup>†</sup>      中島 啓介<sup>†</sup>      浜田 長晴<sup>†</sup>  
 依田 晴夫<sup>††</sup>

High Speed Template Matching by Monotonized Normalized Correlation

Mitsuji IKEDA<sup>†</sup>, Shoji YOSHIDA<sup>†</sup>, Keisuke NAKASHIMA<sup>†</sup>, Nagaharu HAMADA<sup>†</sup>,  
 and Haruo YODA<sup>††</sup>

あらまし 正規化相関に基づくテンプレートマッチングを高速化するため、相関演算を効率良く打ち切る方法を提案している。本方法は、正規化相関の近似式ではなく、本来の式そのものを単調関数化することによって実現しているため、有効な演算途中で誤って打ち切ることを確実に回避できる。実画像を用いた性能評価実験の結果、本方法を適用したことによって処理画素対の数、処理時間ともに、打ち切りを行わない従来方法と比べて、約1/2~1/10に低減することができ、パターンマッチングの高速化に有効であることを確認できた。

キーワード テンプレートマッチング, パターン認識, 位置合せ, 正規化相関, SSDA

### 1. ま え が き

リモートセンシングやファクトリオートメーションなど数多くの分野では、対象となる画像を探索するための手法として、テンプレートマッチング手法が広く用いられている[1], [2]。テンプレートマッチングは、探索したい対象画像を参照画像情報(テンプレート)として用い、被探索対象画像における様々な位置(参照位置)において、テンプレートとの類似性を計算することにより、参照画像と一致した画像を探し出すマッチング手法である。

このような画像間の類似性を表す評価尺度として様々なものが提案されている[1]~[9]が、中でも、正規化相関[7]と差分総和[3]が代表的である。

正規化相関は、対応する画素値の平均と標準偏差によって正規化された絶対測度であり、-1 から 1 までの値をとる。1 に近いほど類似性が高いことを表す。また、正規化相関は、画素値の線形変換に対して一定の評価値を提供する。このため、照明変動など外乱による影響を受けにくい評価尺度として知られている。

一方、差分総和は、テンプレートと探索対象画像の対応する画素値の差の絶対値を合計した尺度であり、値が小さいほど類似性が高いことを表す。差分総和は、正規化相関とは異なり、画素値の線型変換に対して変動する。しかし、差分総和は、それを求める過程において単調増加する尺度であるため、あらかじめ特定のしきい値を定めておけば、計算の打ち切りによる計算時間短縮手法を用いることができる。この計算時間短縮手法は SSDA (Sequential Similarity Detection Algorithms)[3] という名前で知られている。このため、差分総和は、特に、PC やワークステーションをベースとするソフトウェア処理によるテンプレートマッチングに利用されている[10], [11]。

絶対測度であり、かつ、線形変換に対してロバストな正規化相関に対しても SSDA を用いることができれば、高速かつ有効なテンプレートマッチングの実現が期待できる。しかし、通常の正規化相関では処理画素数に対して単調減少しないと考えられていたため、そのままでは SSDA を適用できない[1]。

筆者らは、正規化相関を求める手順を工夫することにより、SSDA を適用する手法を開発した[12], [13]。同時期に足立らも逐次演算部を SSDA 化する手法を提案している[14]が、逐次演算に用いる係数処理に時間がかかり、探索対象画像が毎回異なるような一般的な応用に対して高速化が妨げられていた。

<sup>†</sup>(株)日立製作所日立研究所, 日立市

Hitachi Research Laboratory, Hitachi, Ltd., 1-1 Omika-cho 7-chome, Hitachi-shi, 319-1292 Japan

<sup>††</sup>(株)日立製作所計測器グループ, ひたちなか市

Instruments, Hitachi, Ltd., 882 Ichige, Hitachinaka-shi, 312-0033 Japan

このような係数処理の効率化にも本論文の手法は有効であり、高速なテンプレートマッチングが必要な幅広い応用分野に適用できる。本論文では、開発した演算法の概要と評価結果について報告する。

## 2. テンプレートマッチングにおける従来の正規化相関演算

テンプレートマッチングでは、探索対象画像の各参照位置において、テンプレートと参照位置の対応する画素対を評価尺度に基づいて計算することによって、参照位置がどれくらいテンプレートに類似しているかを評価する。

本論文では、テンプレートを  $t(i)$  ( $i = 0, \dots, N-1$ )、探索対象画像を  $F(i)$  ( $i = 0, \dots, L-1$ )、探索対象画像  $F(i)$  における参照位置  $x$  の、テンプレートと同サイズとなる部分画像を  $f_x(i)$  ( $i = 0, \dots, N-1$ )、参照位置の総数を  $S$  と表す。

テンプレート  $t(i)$  ( $i = 0, \dots, N-1$ ) と部分画像  $f_x(i)$  ( $i = 0, \dots, N-1$ ) との正規化相関  $R_x$  は式(1)で与えられる。

$$R_x = \frac{\sum_{i=0}^{N-1} (f_x(i) - f_{xM}) \times (t(i) - t_M)}{\sqrt{\sum_{i=0}^{N-1} (f_x(i) - f_{xM})^2} \sqrt{\sum_{i=0}^{N-1} (t(i) - t_M)^2}} \quad (1)$$

ただし、

$$f_{xM} = \frac{1}{N} \sum_{i=0}^{N-1} f_x(i), \quad t_M = \frac{1}{N} \sum_{i=0}^{N-1} t(i).$$

正規化相関  $R_x$  は  $-1$  から  $1$  までの値をとり、値が大きいほど類似性の高さを表す絶対尺度である。すべての  $i$  ( $i = 0, \dots, N-1$ ) に関して、

$$f_x(i) = a \times t(i) + b \quad (\text{ただし } a > 0)$$

が成立するとき、 $R_x = 1$  となる。また、すべての  $f_x(i)$  ( $i = 0, \dots, N-1$ ) に関して、

$$g(i) = a \times f_x(i) + b \quad (\text{ただし } a > 0)$$

という線形変換を行った画像  $g(i)$  とテンプレート  $t(i)$  の正規化相関は、線形変換を行う前のものと同じである。このように、正規化相関は、画像の輝度の線形変化に対して安定したロバストな評価尺度といえる。

式(1)は、このままの形では画素対ごとに式(1)の  $\Sigma$  の中を演算することになり、単調関数ではないため、SSDAを適用することができず、計算打ち切りによる高速化はできないと考えられていた。

## 3. 正規化相関演算の単調化による高速化手法

### 3.1 原理

$\sum_{i=0}^{N-1} t(i)$ ,  $\sum_{i=0}^{N-1} \{t(i)\}^2$ ,  $\sum_{i=0}^{N-1} f_x(i)$  及び  $\sum_{i=0}^{N-1} \{f_x(i)\}^2$  を、簡単のため、それぞれ、 $A$ ,  $B$ ,  $C_x$  及び  $D_x$  と表すと、式(2)のように表すことができる。

$$R_x = H_x - \frac{\sum_{i=0}^{N-1} \{f_x(i) - t(i)\}^2}{K_x} \quad (2)$$

ただし、

$$H_x = \frac{D_x + B - \frac{2(C_x \times A)}{N}}{K_x},$$

$$K_x = 2\sqrt{D_x - \frac{\{C_x\}^2}{N}} \sqrt{B - \frac{A^2}{N}}.$$

式(2)において、 $H_x$  及び  $K_x$  をあらかじめ計算しておくことにより、 $R_x$  は、画素対の評価を進めるに従って単調減少する関数とみなすことができる。したがって、正規化相関を用いたテンプレートマッチングにもSSDAを適用できる。

$H_x$  及び  $K_x$  を効率良く求める手法を説明するため、1回当たりの加減算、乗算、除算、平方根演算、及び、比較の計算量を、それぞれ、add, mult, div, sqrt, 及び, cmp で表し、複合処理の計算量を  $N$ ,  $L$  及び  $S$  を用いた1次結合式で表記する。例えば、 $S \times N$  回の乗算及び  $2 \times S \times N$  回の加算は、 $S \times N \times (2 \times \text{add} + \text{mult})$  あるいは  $S \cdot N \cdot (2\text{add} + \text{mult})$  と記述する。また、 $N$ ,  $L$  及び  $S$  に依存しない演算は無視する。

式(2)に示すように、 $H_x$  及び  $K_x$  は、 $N$ ,  $A$ ,  $B$ ,  $C_x$  及び  $D_x$  から構成されている。 $A$  及び  $B$  はすべての参照位置で共通であり、テンプレートマッチングの中で1度計算すればよい。 $A$  の計算量は  $N \times \text{add}$ ,  $B$  の計算量は  $N \times (\text{add} + \text{mult})$  である。

$C_x$  は、隣接した参照位置における同一サイズ領域の画素値総和  $C_x - 1$  から、減少した幅1画素分の短冊領域の画素値総和を減算し、増加した幅1画素分の短冊

領域における画素値総和を加算することにより求めることができる。幅 1 画素分の短冊領域における画素値総和は、縦方向に 1 画素ずれた同一サイズの領域から、減少した 1 画素を減算し、増加した 1 画素を加算することにより求めることができるので、すべての上記短冊領域の画素値総和計算量はたかだか  $2L \times \text{add}$  である。幅 1 画素領域の画素値総和を用いることにより、すべての参照位置  $x$  における総和  $C_x$  の計算量はたかだか  $2L \times \text{add}$  である。したがって、合計  $4L \times \text{add}$  の計算量で、すべての参照位置における  $C_x$  を求めることができる。

$D_x$  は、すべての画素を 2 乗する以外は  $C_x$  と同様に求めることができるので、計算量は  $L \times \text{mult} + 4L \times \text{add}$  である。

すべての参照位置における  $K_x$  及び  $H_x$  は、上記  $C_x$  及び  $D_x$  を用いることにより、それぞれ、 $S \times (\text{add} + 3 \times \text{mult} + \text{sqrt})$ 、 $S \times (2 \times \text{add} + \text{mult} + \text{div})$  で求めることができる。

上述の考えに基づいて SSDA を適用した正規化相関テンプレートマッチングアルゴリズム MNCM1 (Monotonized Normalized Correlation Matching algorithm 1) を図 1 に示す。

図 1 において、 $R_{th}$  は演算を打ち切るための正規化相関のしきい値、 $TH$  は  $R_{th}$  に対応して決定される実際の打ち切りしきい値、 $X$  は逐次演算経過である。

図 1 のアルゴリズムでは、演算の効率化を考慮して、逐次演算を式 (2) の右辺とするのではなく、単に、 $\sum \{f_x(i) - t(i)\}^2$  としている。このため、打ち切りしきい値もそれに対応したしきい値  $TH$  を  $R_{th}$  から求めている。したがって、 $\sum \{f_x(i) - t(i)\}^2$  がしきい値  $TH$  を超過することと、上述式 (2) の右辺の値がしきい値  $R_{th}$  を下回ることは等価である。

まず、 $A$ 、 $B$ 、 $C_x$  及び  $D_x$  を計算する。また、しきい値  $R_{th}$  の初期値を与える。次に、各部分画像ごとに、しきい値  $R_{th}$  と  $H_x$  及び  $K_x$  から逐次演算項のしきい値  $TH$  を求める。部分画像とテンプレートの各画素対に対して、差の 2 乗和を計算しながらしきい値  $TH$  と比較し、差の 2 乗和がしきい値  $TH$  を超過した時点で、その部分画像に対する処理を打ち切り、未対象の部分画像を選択して同様の処理を行う。すべての画素対に対して 2 乗和を計算してもしきい値  $TH$  を超過しない場合は、正規化相関  $R_x$  を求めるとともに、新たなしきい値  $R_{th}$  として更新し、未対象の部分画像を選択して同様の処理を行う。

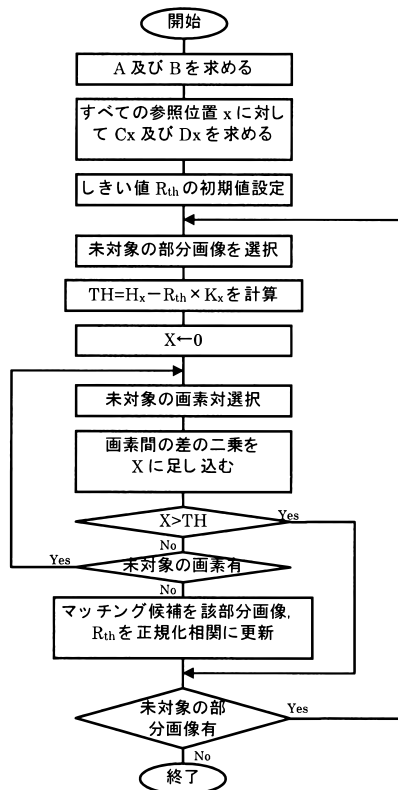


図 1 アルゴリズム MNCM1  
Fig. 1 Flowchart of MNCM1.

### 3.2 その他の SSDA アルゴリズム

前節の例では、探索対象画像とテンプレートの画素対の差の 2 乗和を逐次計算することで正規化相関を求めたが、ほかにも正規化相関を単調関数を用いて求める方法が考えられる。ここでは、正規化相関を求めるために用いる他の SSDA アルゴリズムを導出し、打ち切りの効率化や計算量に関して考察する。

前述の式 (1) は、下記式 (3) ~ (6) のようにも変形することができる。

$$R_x = H_{2x} - \frac{\sum_{i=0}^{N-1} \{f_x(i) - f_{xM} - (t(i) - t_M)\}^2}{K_x} \quad (3)$$

ただし、

$$H_{2x} = \frac{D_x + B - \frac{(C_x)^2 + (A)^2}{N}}{K_x},$$

$$R_x = 1 - \frac{1}{2} \sum_{i=0}^{N-1} \left( \frac{f_x(i) - f_{xM}}{\sqrt{D_x - \frac{\{C_x\}^2}{N}}} - \frac{t(i) - t_M}{\sqrt{B - \frac{A^2}{N}}} \right)^2 \quad (4)$$

$$R_x = H_{3x} - \frac{\sum_{i=0}^{N-1} (f_x(i) \times \tilde{t}(i))}{\tilde{K}_x} \quad (5)$$

ただし，

$$\tilde{t}(i) = 255 - t(i), \quad \tilde{A} = \sum_{i=0}^{N-1} \tilde{t}(i), \quad \tilde{B} = \sum_{i=0}^{N-1} \{\tilde{t}(i)\}^2,$$

$$H_{3x} = \frac{C_x \times \tilde{A}}{\tilde{K}_x},$$

$$\tilde{K}_x = \sqrt{D_x - \frac{\{C_x\}^2}{N}} \sqrt{\tilde{B} - \frac{\tilde{A}^2}{N}},$$

$$R_x = H_{4x} - \frac{\sum_{i=0}^{N-1} (f_x(i) - F_{\min}) \times (\tilde{t}(i) - \tilde{t}_{\min})}{\tilde{K}_x} \quad (6)$$

ただし，

$$F_{\min} = \min_{i=0}^{L-1} \{F(i)\}, \quad \tilde{t}_{\min} = \min_{i=0}^{N-1} \{\tilde{t}(i)\},$$

$$H_{4x} = \frac{(C_x - N \times F_{\min}) \times (\tilde{A} - N \times \tilde{t}_{\min})}{\tilde{K}_x}.$$

上記各式から導出されるSSDAアルゴリズムを、それぞれ、MNCM2, MNCM3, MNCM4, 及び MNCM5 と名づける。MNCM2~5を以下に概説する。

[MNCM2] あらかじめ計算した  $A, B, C_x$  及び  $D_x$  より  $H_{2x}, K_x, f_{xM}$  及び  $t_M$  を求める。次に、各部分画像ごとに、しきい値  $R_{th}$  と  $H_{2x}$  及び  $K_x$  から逐次演算項のしきい値  $TH_2 = K_x(H_{2x} - R_{th})$  を求める。部分画像とテンプレートの各画素対  $(f_x(i), t(i))$  に対して、 $\{f_x(i) - f_{xM} - (t(i) - t_M)\}^2$  を加算しながらしきい値  $TH_2$  と比較し、しきい値  $TH_2$  を超過

した時点で処理を打ち切る。最後までしきい値  $TH_2$  を超過しない場合は、正規化相関  $R_x$  を求めるとともに、新たなしきい値  $R_{th}$  として更新する。

[MNCM3] あらかじめ計算した  $A, B, C_x$  及び  $D_x$  より  $\frac{t(i) - t_M}{\sqrt{B - \frac{A^2}{N}}}, \frac{1}{\sqrt{D_x - \frac{\{C_x\}^2}{N}}}, f_{xM}$  を求める。部分画像とテンプレートの各画素対  $(f_x(i), t(i))$  に対して、 $\left( \frac{f_x(i) - f_{xM}}{\sqrt{D_x - \frac{\{C_x\}^2}{N}}} - \frac{t(i) - t_M}{\sqrt{B - \frac{A^2}{N}}} \right)^2$  を加算しながらしきい値  $TH_3 = 2 - 2R_{th}$  と比較し、しきい値  $TH_3$  を超過した時点で処理を打ち切る。最後までしきい値  $TH_3$  を超過しない場合は、正規化相関  $R_x$  を求めるとともに、新たなしきい値  $R_{th}$  として更新する。

[MNCM4] あらかじめ、 $\tilde{t}(i) = 255 - t(i)$  を計算し、 $\tilde{A}, \tilde{B}, C_x, D_x, H_{3x}$  及び  $\tilde{K}_x$  を求める。次に、各部分画像ごとに、しきい値  $R_{th}$  と  $H_{3x}$  及び  $\tilde{K}_x$  から逐次演算項のしきい値  $TH_4 = K_x(H_{3x} - R_{th})$  を求める。部分画像とテンプレートの各画素対  $(f_x(i), \tilde{t}(i))$  に対して、 $f_x(i) \times \tilde{t}(i)$  を加算しながらしきい値  $TH_4$  と比較し、しきい値  $TH_4$  を超過した時点で処理を打ち切る。最後までしきい値  $TH_4$  を超過しない場合は、正規化相関  $R_x$  を求めるとともに、新たなしきい値  $R_{th}$  として更新する。

[MNCM5] あらかじめ、 $\tilde{t}(i) = 255 - t(i)$  を計算し、 $\tilde{A}, \tilde{B}, C_x, D_x, H_{4x}$  及び  $\tilde{K}_x$  を求める。また、 $F_{\min}, \tilde{t}_{\min}$  を求める。次に、各部分画像ごとに、しきい値  $R_{th}$  と  $H_{4x}$  及び  $\tilde{K}_x$  から逐次演算項のしきい値  $TH_5 = K_x(H_{4x} - R_{th})$  を求める。部分画像とテンプレートの各画素対  $(f_x(i), \tilde{t}(i))$  に対して、 $(f_x(i) - F_{\min}) \times (\tilde{t}(i) - \tilde{t}_{\min})$  を加算しながらしきい値  $TH_5$  と比較し、しきい値  $TH_5$  を超過した時点で処理を打ち切る。最後までしきい値  $TH_5$  を超過しない場合は、正規化相関  $R_x$  を求めるとともに、新たなしきい値  $R_{th}$  として更新する。

MNCM2~5とMNCM1との違いは、初期値及び逐次演算項である。アルゴリズムでは、効率化のため、逐次演算項を単純化したがる、もともとは、式(2)~(6)における、右辺の第1項が初期値、第2項が逐次演算項である。

## 4. SSDA アルゴリズムの比較

### 4.1 演算処理回数による処理時間の評価

前章で導出したSSDAアルゴリズムの演算処理時間について検討する。

処理時間比較のために、図 2 に示す SSDA でないテンプレートマッチングアルゴリズム NCM を定義する。NCM では、 $A, B, C_x$  及び  $D_x$  を求めた後、画素対の積和を逐次計算で求める。打切りはしない。対象

画素対の積和計算を終えると、既に計算済みの  $A, B, C_x$  及び  $D_x$  を用いて正規化相関を求める。

これらのアルゴリズムの計算量は、前処理の計算及び後処理の計算量と逐次計算の計算量の合計で与えられる。前処理では、初期値や逐次計算に用いる係数及びしきい値を、後処理では最終的な正規化相関値を計算する。逐次計算の計算量は、画素当りの計算量と処理画素対の個数の積となる。MNCM1 から MNCM5 及び SSDA [14] の処理画素対の個数をそれぞれ、 $n_1, n_2, n_3, n_4, n_5$ 、及び、 $n_6$  として、各アルゴリズムにおける、計算量の比較を表 1 に示す。

NCM では、前処理として、 $A, B, C_x$  及び  $D_x$  を求め、 $\sqrt{D_x - \frac{\{C_x\}^2}{N}} \sqrt{B - \frac{A^2}{N}}$  及び  $\frac{C_x \times A}{N}$  を計算する。 $A, B, C_x$  及び  $D_x$  の計算量は  $N \times (2add + mult) + L \times (8add + mult)$ 、 $\sqrt{D_x - \frac{\{C_x\}^2}{N}} \sqrt{B - \frac{A^2}{N}}$  及び  $\frac{C_x \times A}{N}$  の計算量は  $S \times (add + 4mult + sqrt)$  である。後処理として、正規化相関計算に  $S \times (add + div)$ 、最大相関値との比較に  $S \times cmp$  かかる。

画素当りの逐次計算量は、 $add + mult$  であり、処理画素対の個数は  $S \times N$  である。

MNCM1 では、前処理として、 $A, B, C_x$  及び  $D_x$  を求め、 $TH$  を計算する。 $TH = K_x(H_x - R_{th})$  より、 $TH$  の計算量は  $S \times (4add + 5mult + div + sqrt)$  となる。後処理の計算量は  $S \times (add + div + cmp)$  である。

画素当りの逐次計算量は  $2add + mult + cmp$  である。

MNCM2 では、前処理として、 $A, B, C_x$  及び  $D_x$

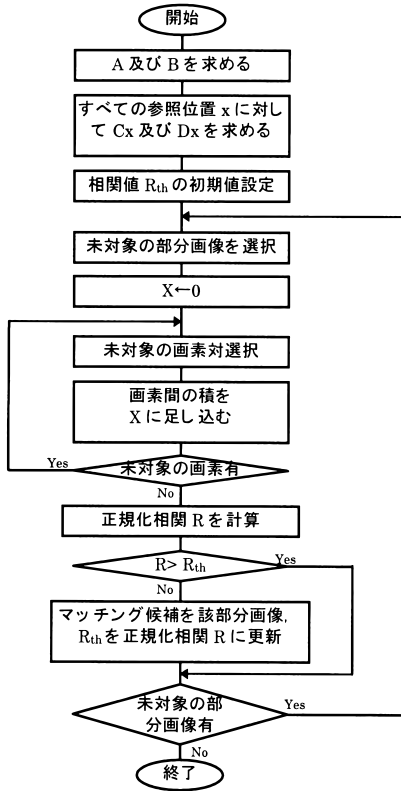


図 2 アルゴリズム NCM  
Fig.2 Flowchart of NCM.

表 1 各アルゴリズムの計算量  
Table 1 Time complexity for each algorithm.

Algorithm	pre/post_process part	sequential process part
NCM	$N \cdot (2add + mult) + L \cdot (8add + mult) + S \cdot (2add + 4mult + sqrt + div + cmp)$	$S \cdot N \cdot (add + mult)$
MNCM1	$N \cdot (2add + mult) + L \cdot (8add + mult) + S \cdot (5add + 5mult + sqrt + div + cmp)$	$n_1 \cdot (2add + mult + cmp)$
MNCM2	$N \cdot (2add + mult) + L \cdot (8add + mult) + S \cdot (5add + 6mult + sqrt + 2div + cmp)$	$n_2 \cdot (3add + mult + cmp)$
MNCM3	$N \cdot (2add + mult) + L \cdot (8add + mult) + S \cdot (3add + 4mult + sqrt + div + cmp)$	$n_3 \cdot (3add + 2mult + cmp)$
MNCM4	$N \cdot (2add + mult) + L \cdot (8add + mult) + S \cdot (2add + 4mult + sqrt + 2div + cmp)$	$n_4 \cdot (add + mult + cmp)$
MNCM5	$N \cdot (2add + mult + cmp) + L \cdot (8add + mult + cmp) + S \cdot (3add + 4mult + sqrt + div + cmp)$	$n_5 \cdot (2add + mult + cmp)$
SSDA [14]	$N \cdot (2add + mult + cmp) + S \cdot N \cdot (2add + mult) + S \cdot (sqrt + 2div)$	$n_6 \cdot (3add + 2mult + cmp)$

を求め、 $TH_2$  を計算する。 $TH_2 = K_x(H_{2x} - R_{th})$  より、 $TH_2$  の計算量は  $S \times (4\text{add} + 6\text{mult} + \text{div} + \text{sqrt})$  となる。後処理の計算量は  $S \times (\text{add} + \text{div} + \text{cmp})$  である。

画素当りの逐次計算量は、 $3\text{add} + \text{mult} + \text{cmp}$  である。

MNCM3 では、前処理として、 $A, B, C_x$  及び  $D_x$  を求め、 $\frac{t(i)-t_M}{\sqrt{B-A^2/N}}, \frac{1}{\sqrt{D_x-\{C_x\}^2/N}}, f_{xM}$  を計算する。 $TH_3$  はすべての参照位置で共通であるため、計算量は無視できる。 $\frac{t(i)-t_M}{\sqrt{B-A^2/N}}$  を求める計算量が  $N \times (\text{add} + \text{mult})$ 、 $\frac{1}{\sqrt{D_x-\{C_x\}^2/N}}$  を求める計算量が  $S \times (\text{add} + 2\text{mult} + \text{div} + \text{sqrt})$ 、 $f_{xM}$  を求める計算量が  $S \times \text{mult}$  である。後処理の計算量は  $S \times (\text{add} + \text{mult} + \text{cmp})$  である。

画素当りの逐次計算量は  $3\text{add} + 2\text{mult} + \text{cmp}$  である。

MNCM4 では、前処理として、 $\tilde{t}(i) = 255 - t(i)$ 、 $\tilde{A}, \tilde{B}, C_x$  及び  $D_x$  を求め、 $TH_4$  を計算する。 $\tilde{t}(i)$  を計算する時間が  $N \times \text{add}$ 、 $TH_4 = \tilde{K}_x(H_{3x} - R_{th})$  より、 $TH_4$  の計算量は  $S \times (\text{add} + 4\text{mult} + \text{div} + \text{sqrt})$  となる。後処理の計算量は  $S \times (\text{add} + \text{div} + \text{cmp})$  である。

画素当りの逐次計算量は、 $\text{add} + \text{mult} + \text{cmp}$  である。

MNCM5 では、前処理として、 $\tilde{t}(i) = 255 - t(i)$ 、 $F_{\min}, \tilde{t}_{\min}, \tilde{A}, \tilde{B}, C_x$  及び  $D_x$  を求め、 $TH_5$  を計算する。 $F_{\min}$  の計算量が  $L \times \text{cmp}$ 、 $\tilde{t}_{\min}$  の計算量が  $N \times \text{cmp}$  である。 $TH_5 = \tilde{K}_x(H_{4x} - R_{th})$  より、 $TH_5$  の計算量は  $S \times (2\text{add} + 4\text{mult} + \text{div} + \text{sqrt})$  となる。後処理の計算量は  $S \times (\text{add} + \text{div} + \text{cmp})$  である。

画素当りの逐次計算量は  $2\text{add} + \text{mult} + \text{cmp}$  である。

以上より、NCM, MNCM1 から MNCM5 までの前処理及び後処理の計算量は  $(L + S + N)$  のオーダーとなり、[14] における前処理の計算量が  $(S \times N)$  のオーダーであることと比べて効率的であることがわかる。一方、逐次計算の計算量は  $(S \times N)$  のオーダーであり、NCM, MNCM1 から MNCM5 に関しては、逐次計算が支配的であるといえる。

画素当りの逐次計算量を比較すると、NCM が最も少なく、MNCM4, MNCM1 及び MNCM5, MNCM2, MNCM3 の順に多くなっている。実際の計算時間は、実行するプロセッサの命令セットアーキテクチャ、キャッ

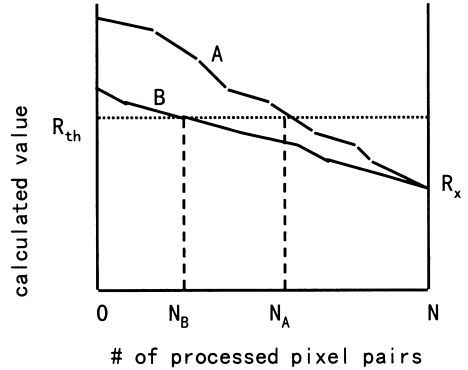


図3 初期値と打ち切り効率の関係  
Fig. 3 Relation between initial value and efficiency.

シュ構成、コンパイラ等によって異なるため、上記計算量だけから比較するのは難しい。ただし、演算数が多く、浮動小数点演算を多用する MNCM3 が、最も画素当りの逐次計算時間がかかると考えられる。

次に、SSDA アルゴリズムの処理画素対の個数について考察する。

MNCM3 と SSDA [14] の初期値及び逐次演算項は同一であるため、 $n_3 = n_6$  となる。

SSDA アルゴリズムでは初期値が小さいほど打ち切りの効率が良いと考えられる。図3に、初期値の異なる2種類の SSDA アルゴリズム A, B の、ある参照位置における計算推移を示す。図3の横軸は処理画素対数、縦軸はそのときの計算値を表している。すべての画素対を処理した時点で求まる正規化相関値は同じであるので、初期値の小さい B の方が A より早くしきい値に到達する可能性が高いと考えられる。

MNCM1, MNCM2, MNCM3 の初期値  $H_x, H_{2x}, 1$  を比較すると、 $H_x \geq H_{2x} \geq 1$  であるため、 $n_1 \geq n_2 \geq n_3$  と考えられる。また、MNCM4, MNCM5 の初期値  $H_{3x}, H_{4x}$  を比較すると、 $H_{3x} \geq H_{4x}$  であるため、 $n_4 \geq n_5$  となることが期待される。MNCM1 から MNCM3 の初期値と、MNCM4 から MNCM5 の初期値とでは、必ずしも大小関係は成立しない。

#### 4.2 評価実験

本手法を用いたテンプレートマッチングの性能評価実験を行った。実験1では、種類及びサイズの異なる画像に対して、上記各アルゴリズムの処理画素対の数、処理時間を評価し、実験2では、テンプレートの種類及び位置の違いによる処理効率を評価した。実験に用いた画像を図4に示す。図4(a)及び(b)は  $256 \times 256$

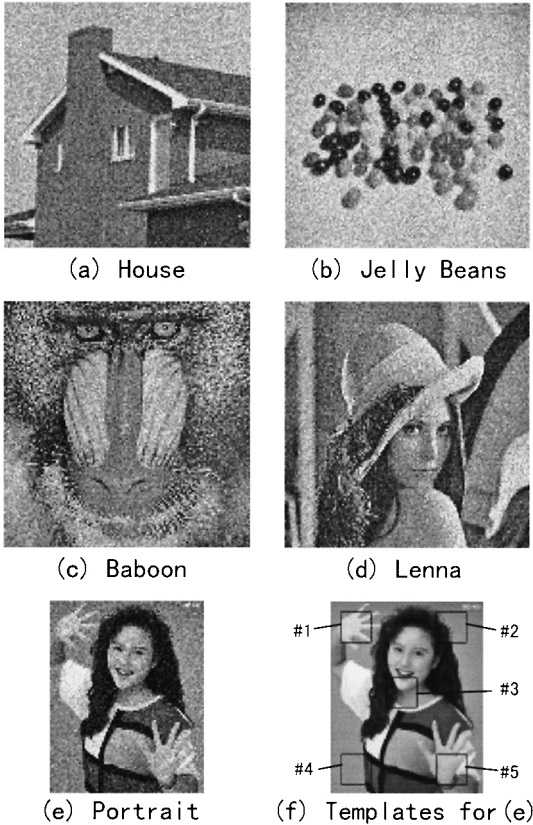


図4 評価実験に用いた画像  
Fig. 4 Image for experiments.

画素サイズの画像, 図 4 (c) 及び (d) は  $512 \times 512$  画素サイズの画像, 図 4 (e) は  $160 \times 120$  画素サイズの画像である。これらの画像には, センサから入力されたことを模擬するため, ノイズ量 16 のガウス雑音を付加している。

評価装置は SUN UltraSPARC-II (CPU 動作周波数 296 MHz; 主記憶 1G バイト; SunOS 5.5.1) を用いた。また, コンパイラは gcc ver.2.7.2.3 を用い, 最適化オプション-O2 でコンパイルした。

各テンプレートマッチングに関して, しきい値の初期値  $R_{th}$  を 0 とし, 求めた正規化相関がしきい値を上回るごとに, 求めた正規化相関値を新たなしきい値として更新した。参照位置は左下から開始して横優先順に選択し, 処理対象となった画素対の総和と処理時間を評価した。実験 1 では, 図 4 (a) から (e) を用いてテンプレートマッチングを行った。テンプレートと

しては, 雑音を付加する前の図 4 (a) から (e) に対応するオリジナル画像の中央から, 図 4 (a) 及び (b) に対しては  $64 \times 64$  画素サイズの画像, 図 4 (c) 及び (d) に対しては  $128 \times 128$  画素サイズの画像, 図 4 (e) に対しては  $32 \times 32$  画素サイズの画像をそれぞれ切り出して用いた。

表 2 に実験 1 の結果を示す。上段は各参照画像 (image) に対する正解位置での正規化相関値  $\times 100$  (score) である。下段は, 参照画像に対応して処理手法 (Algorithm) ごとの測定結果を, 処理画素対の数 (processed pixel pair) を百分率, 前処理を含めた処理時間 (processing time) を秒, 逐次演算における画素当りの処理時間 (processing unit time) をナノ秒で示したものである。

評価したすべての画像に対して, 処理画素対は, MNCM3, MNCM2, MNCM1, MNCM5, MNCM4 の順に大きくなるという結果が得られ, MNCM1 と MNCM5 の関係以外は, 4.1 で検討した内容と一致した。一般の画像では, MNCM5 より MNCM1 の方が有利であるといえる。

処理時間に関しては, 4.1 の予想に反して, MNCM3 の画素当りの処理時間が最も短かった。これは, MNCM3 では予想どおりのアセンブラコードが生成されていたのに, 整数演算を用いた他のアルゴリズムでは, 想定以外のコードが多く生成されていたことが原因であった。より厳密な性能比較のためには, 今後, コンパイラの性能に起因するような部分の評価も必要である。

今回の評価実験では, 画像及びテンプレートサイズの違いによる優位差は認められなかった。

実験 2 では, 図 4 (e) を用いてテンプレートマッチングを行った。テンプレートとしては, 図 4 (f) に示す #1 から #5 を用いた。探索位置が早く出現すると, しきい値も早く高くなるため, 打ち切り効率が増えると考えられる。図 4 (f) のテンプレートの探索順序は, #4, #5, #3, #1, #2 の順である。実験 2 の結果を表 3 に示す。

実験結果では, 探索順序の早いテンプレートに対して打ち切り効率が高くなる傾向があることが確認できた。ただし, アルゴリズムによって, 打ち切り効率の順序にばらつきがみられた。また, 探索順序が遅くても打ち切り効率が高いものもあり, その原因は, テンプレートの選び方にあるものと考えられる。

基板実装検査や IC リードの位置決めなど探索対象

表2 実験1の評価結果  
Table 2 Results of evaluation experiment #1.

image	Portrait	House	Jelly Beans	Baboon	Lenna
score	96.8	91.9	95.4	86.7	95.6

Algorithm	processed pixel pairs [%]					processing time [s]					Processing unit time [ns]
	Portrait	House	Jelly Beans	Baboon	Lenna	Portrait	House	Jelly Beans	Baboon	Lenna	
NCM	100	100	100	100	100	2.82	22.2	22.4	358	356	147
MNCM1	41.4	46.8	50.5	58.6	34.1	2.08	16.8	19	332	196	229
MNCM2	30.9	22.7	38.2	38.0	29.2	1.28	6.3	10.7	171	132	182
MNCM3	27.3	20.7	31.4	31.0	26.0	0.65	3.5	5.2	85	70	111
MNCM4	79.6	94.9	91.2	93.0	89.7	4.3	34	33	532	514	235
MNCM5	73.1	94.2	84.2	87.6	87.6	4.0	35	34	525	530	247

表3 実験2の評価結果  
Table 3 Results of evaluation experiment #2.

template	#1	#2	#3	#4	#5
score	85.8	95.5	96.8	92.5	95.6

Algorithm	processed pixel pairs [%]					processing time [s]				
	#1	#2	#3	#4	#5	#1	#2	#3	#4	#5
NCM	100	100	100	100	100	2.88	2.84	2.82	2.86	2.88
MNCM1	77.5	36.1	41.4	38.1	50.1	3.58	1.84	2.08	1.90	2.38
MNCM2	60.9	23.8	30.9	18.0	15.1	2.51	0.99	1.28	0.76	0.64
MNCM3	53.6	20.7	27.3	9.1	6.6	1.27	0.49	0.65	0.24	0.17
MNCM4	90.7	81.3	79.6	85.5	71.6	4.8	4.3	4.3	4.6	3.8
MNCM5	86.0	60.5	73.1	58.4	59.4	4.7	3.3	4.0	3.1	3.2

画像のパターンがあらかじめわかっている場合は、検査対象部分とテンプレートとして選択する部分画像との相対座標を記憶することができる。これによって、任意の部分画像をテンプレートとして選択することができるので、適切なテンプレート画像を選択することで、より打ち切り効率の高いテンプレートマッチングの実現が期待できる。このようなテンプレートの特徴と打ち切り効率の定量的な評価に関しては、今後更に検討が必要である。

以上の実験結果から、特に MNCM3 は、打ち切りを行わない従来の NCM と比べて、処理画素対の数及び処理時間がほぼ  $1/2 \sim 1/10$  に低減できていることを確

認できた。また、適切なテンプレートの選択、参照位置の選択順序の変更等により、処理時間の更なる低減が期待できることもわかった。

## 5. む す び

テンプレートマッチングにおける正規化相関演算の単調関数化を導出し、これを適用した SSDA アルゴリズムを提案した。本アルゴリズムによって、従来正規化相関では不可能と考えられていた無駄処理のみを打ち切る高速化が可能になった。

実画像を用いた評価実験により、従来の打ち切りを行わない場合と比べて、処理画素対の数、処理時間とも



に、約  $1/2 \sim 1/10$  に低減することができ、本手法の有効性を確認できた。また、適切なテンプレートの選択、参照位置の選択順序の変更等により、処理時間の更なる低減が期待できることを見出した。

今後の課題としては、コンパイラの性能に起因する要因の定量評価や正規化相関演算を効率良く打ち切るためのテンプレート選択法の検討などが挙げられる。

## 文 献

- [1] 高木幹雄, 下田陽久, 画像解析ハンドブック, pp.707-712, 東京大学出版会, 1991.
- [2] 江尻正員, 画像処理産業応用総覧 (株)フジ・テクノシステム, 1994.
- [3] D.I. Barnea and H.F. Silverman, "A class of algorithms for fast digital image registration," IEEE Trans. Comput., vol.C-21, no.2, pp.179-186, Feb. 1972.
- [4] M. Svedlow, C.D. McGillem, and P.E. Anuta, "Experimental examination of similarity measures and preprocessing methods used for image registration," Proc. Symposium on Machine Processing of Remotely Sensed Data, Westville, Ind., pp.4A-9, June 1976.
- [5] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," Proc. Int. Joint Conf. in Artificial Intelligence, pp.659-663, 1977.
- [6] W.K. Pratt, Digital Image Processing, John Wiley & Sons, New York, 1978.
- [7] D.H. Ballard, Computer vision, pp.65-70, Prentice-Hall, Inc., Englewood Cliffs, 1982.
- [8] A. Rosenfeld and A. Kak, Digital picture processing, vol.I and II, Academic Press, Orlando, Fla., 1982.
- [9] E. De Castro and C. Morandi, "Registration of translated and rotated images using finite Fourier Transforms," IEEE Trans. Pattern Anal. & Mach. Intell., vol.PAMI-9, no.5, pp.700-703, Sept. 1987.
- [10] 中尾敏康, 兼吉昭雄, "デジタルモザイク処理を利用したパーソナル向け画像入力," 画電学第 156 回研究会講演予稿, pp.9-12, March 1997.
- [11] 藤岡利和, 水澤和史, 田中武久, 中 基孫, 森 真人, "形状特徴を用いたテンプレートマッチング法による複数移動物体の実時間追跡," 信学技報, PRMU96-178, March 1997.
- [12] 池田光二, 吉田昌司, 中島啓介, 桂 晃洋, 依田晴夫, "パターンマッチングの高速化手法について," 画電学第 156 回研究会講演予稿, pp.5-8, March 1997.
- [13] 池田光二, 吉田昌司, 中島啓介, 桂 晃洋, 依田晴夫, "テンプレートマッチング高速化のための正規化相関演算打ち切り方法," 信学総大, 分冊 2, no.SD-4-1, pp.439-440, March 1997.
- [14] 足立一秀, 岩佐英彦, 竹村治雄, 横矢直和, "正規化相互相関テンプレートマッチングの KL 展開を用いた SSDA

化," 信学技報, PRMU96-173, March 1997.

(平成 10 年 10 月 8 日受付, 11 年 3 月 17 日再受付)



池田 光二 (正員)

昭 58 阪大・基礎工・情報卒。昭 60 同大学院修士課程了。同年 (株)日立製作所入社。以来、論理回路のテスト生成、画像処理・認識技術などの研究開発に従事。現在、同社日立研究所情報制御第一研究部主任研究員。情報処理学会会員。



吉田 昌司 (正員)

昭 63 東大・工・計数卒。同年 (株)日立製作所入社。以後、同社日立研究所に勤務。現在、情報制御第一研究部に所属。RISC プロセッサ、コントローラ、及び画像処理に関する研究に従事。



中島 啓介 (正員)

昭 53 静岡大・工・電子卒。昭 55 同大学院修士課程了。同年 (株)日立製作所日立研究所入社。以来、文書画像処理の研究を中心に、ファクシミリ、スキャナ用の LSI、ソフトウェアの製品化に従事。現在、同社日立研究所情報制御第一研究部主任研究員。情報処理学会、画像電子学会、IS&T 各会員。画像電子学会編集委員。



浜田 長晴 (正員)

昭 40 鹿児島大・工・電気卒。同年 (株)日立製作所入社。日立研究所にて、グラフィックス&イメージ処理プロセッサとマルチメディアシステムの研究開発に従事。工博。情報処理学会、画像電子学会、IEEE、SID 各会員。



依田 晴夫 (正員)

昭 42 早大・理工・電気卒。昭 44 東大大学院修士 (計数) 了。同年 (株)日立製作所入社。以来、画像認識システム、電子線描画装置などの研究開発に従事。昭 51 本会学術奨励賞, 昭 62 本会論文賞受賞。工博。現在、同社計測器グループ副技師長。