

Mathematical Morphology 演算の 高速化アルゴリズムの比較

仁保 勉^{†,☆} 江 浩^{†,☆☆} 山本 眞司[†]

画像解析に有効な手段である mathematical morphology には、フィルタサイズの増大にともない処理時間が莫大になるといった問題点がある。この問題を解決する手段として、大きなフィルタをより小さなフィルタに分割して繰返し処理を行う分割繰返し型や、2次元フィルタを1次元フィルタに分解して処理を行う1次元分解型が提案されている。本論文では上記両アルゴリズムの計算量・処理時間をフィルタの形状ごとに求め、比較を行った。その結果、計算量・処理時間の縮小に関して対称形状のフィルタでは1次元分解型は非常に有効であるが、そうでないフィルタでは分割繰返し型の方が有効であることが分かった。また、分割繰返し型で問題となる小フィルタのサンプリング誤差の累積が、処理結果画像にどのような影響を及ぼすのかを実験的に調べた。その結果、処理結果画像に現れる誤差は、フィルタ輪郭部分のサンプリング誤差が繰返し dilation 演算で累積されることが主な原因であることが判明し、分割小フィルタ群の選択によりこの誤差をある程度軽減できることが分かった。以上のことから、計算量・処理時間の削減のためにはフィルタ形状ごとに最適なフィルタリング手法を選択することが必要であり、本論文はそのためのひとつの指針を与えるものである。

Comparison of Efficient Algorithms of Mathematical Morphology

TSUTOMU NIHO,^{†,☆} HAO JIANG^{†,☆☆} and SHINJI YAMAMOTO[†]

Mathematical morphology is an useful tool for image analysis, because it can use an element of object image as a filter. But there is a problem that computation time is large if the size of filter is large. Two filtering methods (One is the 1D filtering method which divides 2D filter into 1D filters, the other is the repetition filtering method which decomposes original filter to smaller filters) had been proposed to solve this problem. In this paper, we evaluated the performance of the two filtering methods on different shapes and sizes of filters. We found that the 1D filtering method shows better performance than repetition filtering method if the filter has symmetric shape, and repetition filtering method is more effective than 1D filtering method if the filter is not symmetric. We also evaluated the appearance of the error using repetition filtering method. From the experimental results, we found that repeated dilation causes the accumulation of sampling errors in the edge of decomposed filter and this error can be reduced by choosing the correct set of decomposed filters.

1. ま え が き

mathematical morphology は、対象物（画像）の部分形状を処理用のフィルタ（通常、構造要素と呼ぶ）としてそのまま採用することができる特徴があり、画像解析に様々な形で応用され^{1)~6)}、また、morphology

演算による画像処理用のハードウェアも開発されている^{7),8)}。

ところで、morphology 演算がフィルタ（構造要素）の形状やサイズに強く依存するという事は、逆にそのフィルタの表現方法によって処理時間に大きな違いが発生するという事で、特にフィルタサイズを大きくしたときに処理時間が莫大となる。そこで、処理時間を短縮するために大きなフィルタをより小さなフィルタに分割して繰返し処理を行う^{9),10)}、2次元フィルタを1次元フィルタに分解して処理を行う¹¹⁾などの改良が提案されている。しかしながら、これらのフィルタリング法による計算量の差に関する検討はまだ十分に行われておらず、処理アルゴリズムを構成する際、

[†] 豊橋技術科学大学知識情報工学系

Knowledge Based Information Engineering Department, Toyohashi University of Technology

[☆] 現在、横河電機株式会社

Presently with Yokogawa Denki Ltd.

^{☆☆} 現在、郵政省通信総合研究所

Presently with Communications Research Laboratory, Ministry of Posts and Telecommunications

最適フィルタリング法の選択には不明点が数多く存在する。文献7)は、ハードウェアを構築する立場から3種類のアルゴリズムを比較しているが、上記の小フィルタによる繰返し処理を行う方法と、1次元フィルタに分解して処理を行う方法との差について評価していない。

また、小フィルタによる繰返し処理を用いる場合、離散系においてはサンプリング誤差の累積が当然予想されるが、それがどの程度、どのように影響するかは十分調べられていない。

そこで、本論文ではまず上記2つのフィルタリング法による morphology 処理の計算量について定量的に評価を行い、同時に計算量のフィルタ形状依存性について考察する。次に、小フィルタによる繰返し処理法で発生する誤差について実験的に評価し、同時に誤差を極力削減する一手法について提案する。

まず2章で mathematical morphology の定義と、フィルタリングアルゴリズムについて述べる。3章では、フィルタリングアルゴリズムについて、計算量という立場から評価を行う。4章では、小フィルタ繰返し法における誤差の発生について調べ、誤差抑制方法としてフィルタ分割方法の改良を検討する。

2. Morphology におけるフィルタリングアルゴリズム

2.1 Mathematical Morphology の基本演算

まず、mathematical morphology の基本演算について簡単に述べる。以下の定義式は文献1)に沿っている。

mathematical morphology には dilation, erosion の2つの基本演算がある。また、それらを組み合わせた opening, closing もよく用いられる。この4つの演算の定義は次のとおりである。

入力の2次元濃淡画像を $f(x, y)$ 、その定義域を F 、フィルタ関数を $g(x_1, y_1)$ 、その定義域を G とすると、2次元画像に対する morphology 演算は式(1)~(4)で表される。

$$\text{dilation } f \oplus g = \max_{\substack{(x_1, y_1) \in G \\ (x-x_1, y-y_1) \in F}} \{f(x-x_1, y-y_1) + g(x_1, y_1)\}. \quad (1)$$

$$\text{erosion } f \ominus g = \min_{\substack{(x_1, y_1) \in G \\ (x+x_1, y+y_1) \in F}} \{f(x+x_1, y+y_1) - g(x_1, y_1)\}. \quad (2)$$

$$\text{opening } f \circ g = \{f(x, y) \ominus g(x_1, y_1)\} \oplus g(x_1, y_1). \quad (3)$$

$$\text{closing } f \bullet g = \{f(x, y) \oplus g(x_1, y_1)\} \ominus g(x_1, y_1). \quad (4)$$

dilation 処理は、画像 f に対するフィルタ g による最大値フィルタリングとして定義されており、濃淡画像に対する穴埋め効果や太め処理を行う。これに対し、erosion 処理は最小値フィルタリングとして定義されており、濃淡画像に対する微小ノイズ除去や、画像の細め処理を行う。これらの演算の組合せである opening は濃淡画像の平滑化、すなわち突起部分の除去作用があり、また closing は、切れ込み部分の埋め込み作用がある。

2.2 フィルタリングアルゴリズム

式(1)~(4)で定義されたフィルタリング演算は、たとえば線形フィルタであるコンボリューション演算と同等、あるいはそれ以上に処理時間がかかる。特にフィルタ定義域 G が大きくなるとその効果が大きくなる。そこで、この処理時間を短縮する工夫が必要になるが、本論文では以下に示す2種類のアルゴリズムについて検討を行った。なお本論文では、フィルタ関数 $g(x_1, y_1)$ として2値関数の場合のみ扱う。したがって、以下では $g(x_1, y_1)$ を2値関数 B で表現する。

また、フィルタリングは主に原画像の参照と重み付け計算に処理時間を要する。原画像の参照と重み付け計算は同時に行われるため、今回は原画像の参照回数、すなわち1画素のフィルタリング処理を実行するとき用いる原画像の参照回数を調べ、それを計算量と定義して処理時間の比較を行う。

2.2.1 分割繰返し型フィルタリング

分割繰返し型フィルタリングとは、元フィルタを複数の小フィルタの繰返しフィルタリングで代用するものである^{9), 10)}。すなわち元フィルタ B (2値関数) に対して、小フィルタ b_i (2値関数) が存在し、

$$B = b_1 \oplus b_2 \oplus \dots \oplus b_N \quad (5)$$

の関係を満たすとき (一般に上式を満たす b_i の組は多数存在する)、dilation 演算に関して (図1参照)

$$f \oplus B = f \oplus (b_1 \oplus b_2 \oplus \dots \oplus b_N) \\ = ((\dots (f \oplus b_1) \oplus b_2) \oplus \dots \oplus b_N) \quad (6)$$

の関係が成立する。したがって、画像 f にフィルタ B を作用させる代わりに、 $b_1, \dots, b_i, \dots, b_N$ の小フィルタの dilation を次々に行うことにより、目的を達成する。

erosion についても同様であり、式(5)の条件下で、

$$f \ominus B = f \ominus (b_1 \oplus b_2 \oplus \dots \oplus b_N) \\ = ((\dots (f \ominus b_1) \ominus b_2) \ominus \dots \ominus b_N) \quad (7)$$

が成立する。なお、opening 演算、closing 演算については式(6)、(7)を組み合わせることで実現できる。

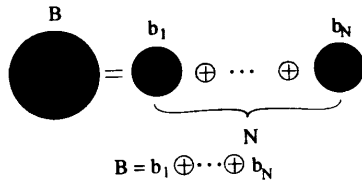


図1 フィルタの分割

Fig.1 Decomposition of filter.

上記の処理は、元フィルタ B を小フィルタ b_i に分割するため、個々のフィルタ定義域が縮小され、結果的に計算量の短縮が期待できる。計算量の短縮効果は次のとおりである。

2次元座標系においては、非繰返し型の計算量は以下の式で表すことができる。

$$S = O(R^2) \tag{8}$$

すなわち計算量 S はフィルタ定義領域内の画素数(面積)を反映する尺度 R (たとえば、円の半径、正方形一辺の長さ、など)の自乗のオーダーに比例する。

一方、分割繰返し型の計算量は以下の式で表すことができる。

$$S' = \sum O(R_i^2) \tag{9}$$

すなわち計算量 S' は元のフィルタを分割したすべての小フィルタによる計算量の総和である (R_i は各小フィルタ定義域内の画素数を反映する尺度)。

分割条件の式 (5) および dilation 演算の性質から

$$R = R_1 + \dots + R_s + \dots + R_N \tag{10}$$

であるため、式 (8) より

$$\begin{aligned} S &= O((R_1 + \dots + R_s + \dots + R_N)^2) \\ &= O\left(\sum R_i^2 + 2 \times \sum \sum R_i R_j\right) \\ &= \sum O(R_i^2) + 2 \times O\left(\sum \sum R_i R_j\right) \end{aligned} \tag{11}$$

となる。

よって、式 (9) と (11) から分かるように分割繰返し型の計算量 S' は、非繰返し型の S に比べて $2 \times O(\sum \sum R_i R_j)$ の分だけ少ない。

特に、分割フィルタの径がすべて等しいとき、つまり、

$$R_1 = R_2 = \dots = R_N = R_s$$

ならば、

$$S = O(R^2) = O((NR_s)^2) = N^2 \times O(R_s^2)$$

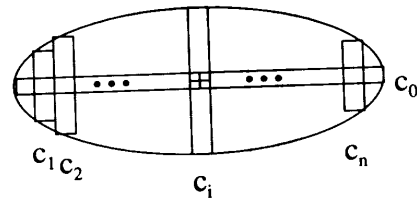
式 (9) より

$$S' = N \times O(R_s^2)$$

よって、

$$S'/S = 1/N \tag{12}$$

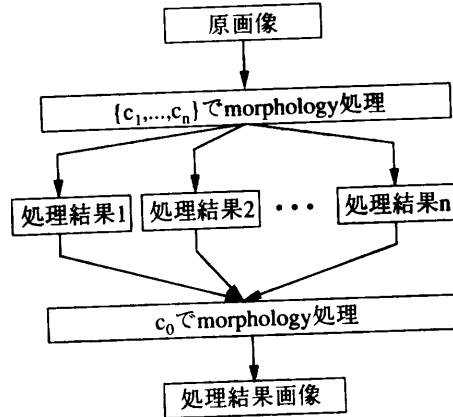
つまり、分割繰返し型の計算量は非繰返し型の $1/N$



+ 元フィルタの注目点

(a) 2次元フィルタの1次元分解

(a) 1D decomposition of 2D filter.



(b) 1次元分解型アルゴリズム

(b) flow-chart of 1D filtering method.

図2 1次元分解型アルゴリズムの概念

Fig.2 1D filtering algorithm.

に縮小される。

ところで以上の議論の出発点は式 (6)、式 (7) にあるが、 B, b_i が連続系の場合はこの式は厳密に成立する。また小フィルタ b_i の具体的選択は通常この連続系の中で行ってしまう。ところが、実際の使用にあたっては、 B, b_i を離散的サンプル点のセットで表現するため、式 (6)、(7) は厳密には成立せず、近似表現になる。このため、分割繰返し型フィルタリングでは近似誤差が発生する。このことについては4章で詳細に述べる。

2.2.2 1次元分解型フィルタリング

1次元分解型フィルタリングは、フィルタを複数の1次元フィルタに分解し、その処理結果からフィルタ全体の結果を求める方式であり¹¹⁾、元のフィルタを1次元フィルタに分割できる条件などは文献11)に説明されている。またこの原理に基づく画像処理用ハードウェアの開発も進められている⁷⁾。このアルゴリズムを用いた dilation 演算の概要を図2に示す。まず、2次元フィルタを図2(a)のように n 個の1次元フィルタ $\{c_1, \dots, c_n\}$ に分解する(分解する方向は、通常 y 軸(垂直)方向または x 軸(水平)方向が用いられる)。処理手順は図2(b)に示すとおりであり、まず、

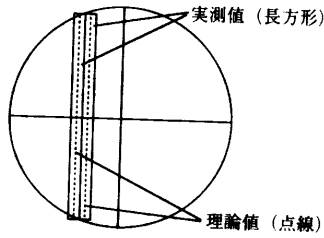


図3 1次元分解の理論値と実測値

Fig. 3 Difference of 1D decomposition in analog and digital space.

原画像に対して垂直方向の $\{c_1, \dots, c_n\}$ の各フィルタごとに1次元 dilation (erosion) 処理を行う。そして、得られた1次元配列データに対し水平方向フィルタ c_0 で1次元 dilation (erosion) 処理を行う。この c_0 による処理結果が元フィルタによる2次元 dilation (erosion) 処理に対応している。以上の処理を画像のすべての座標点で実行するが、1次元フィルタ c_i の中にその長さや重みが等しいなどの共通性がある場合は、ある座標点を中心とする1次元フィルタリング結果を記憶しておくことにより、別の座標点を中心とした1次元フィルタリング処理時の計算を省略できる場合がある。

たとえば、図2(a)の1次元フィルタの長さや重みがすべて等しい場合には、各画素ごとに c_i を1回計算して記憶するだけで十分であり、あとは c_0 の演算のみで処理ができるため演算時間が大幅に短縮される。また別の例として、フィルタが c_i を中心として左右対称である場合は、少なくとも元のフィルタの半分の計算量まで短縮できる。また、サンプリング誤差のため、連続系において1次元分解フィルタの長さが異なっても、離散系においては同じになることがある。たとえば、図3に示したように、円を境界とするフィルタにおいては縦方向の1次元分解の際、 c_i の長さが同じになることがある。したがって、計算量の短縮効果は元のフィルタの形、サイズに依存し、連続系における理論計算値と離散系における実測値が異なることがある。

3. 各アルゴリズムの計算量比較

計算量の比較のため、フィルタ形状には、円のような2次曲線を境界線とするディスク、定義域内に穴を持つリング、両軸方向に対称な正方形、1軸に関してのみ対称である正三角形、どちらの軸にも対称でない回転正三角形の5種類を採用した(表1)。表1において R, w, L はサイズパラメータ、 N は1次元フィルタの個数である。分割繰返し型では分割数を2と

表1 実験に用いたフィルタ
Table 1 Filters used in experiment.

フィルタ リング 手法 形状	非繰 返し型	分割 繰 返し型	1次元 分解型
ディスク フィルタ	R=40	R1=20 R2=20	N=18
リング フィルタ	R1=41 R2=10	R3=25 R4=15	N=29
正方形 フィルタ	w=35	w1=18 w2=17	N=2
正三角形 フィルタ 回転角0度	L=62	L1=31 L2=31	N=55
正三角形 フィルタ 回転角45度	L=62	L1=31 L2=31	N=79

し、2個の小フィルタが元フィルタに相似でかつ、なるべく元フィルタの半分のサイズになるように分割した(等分割)。ただし、リングフィルタは、小フィルタに元フィルタの相似形を選ぶと式(5)が成立しないため、円周部分のみを定義域に持つフィルタとディスクフィルタの2種類のフィルタに分割した。

比較のために、フィルタに小さい手を加えないでそのままフィルタリングを行う“非繰返し型”についても計算量を調べた。

各アルゴリズム、フィルタにおける計算量の理論値(連続系にて計算)と実測値を表2に示す。表2において S, S^* はそれぞれフィルタ定義域面積(計算量)の理論値、実測値である。括弧内に、非繰返し型アルゴリズムの計算量に対する個々の計算量の比をフィルタ形状ごとに求めた。

どのフィルタの理論値においても分割繰返し型と1次元分解型は、非繰返し型に比べ計算量が抑えられている。特に正方形フィルタでは、1次元分解型の計算量が非繰返し型の0.029倍と非常に低くなっており、分割繰返し型の0.5倍と比べても大きな差がある。ところが、1次元分解型は非繰返し型に比べて、リングフィルタでは、0.5倍、正三角形フィルタでは、0.5倍、回転正三角形フィルタでは、等倍と大きくなっており、むしろ分割繰返し型よりも性能が劣る。このことから、

表2 各アルゴリズム、フィルタにおける計算量の理論値と実測値、および計算時間
Table 2 Computation time for each algorithm and its corresponding filter.

フィルタリング方式 フィルタ形状		非繰り返し型	分割繰り返し型	1次元分解型
ディスク	理論値	$S=\pi R^2$ =5024 (1.000)	$S=\pi(R_1^2+R_2^2)$ =2512 (0.500)	$S=(1/2)\pi R^2$ =2512 (0.500)
	実測値	$S^*=5025$ (1.000)	$S^*=2514$ (0.500)	$S^*=1361$ (0.271)
リング	理論値	$S=\pi(R_1^2-R_2^2)$ =4964 (1.000)	$S=2\pi R_3+\pi R_4^2$ =864 (0.174)	$S=(1/2)\pi(R_1^2-R_2^2)$ =2482 (0.500)
	実測値	$S^*=4472$ (1.000)	$S^*=874$ (0.195)	$S^*=1576$ (0.352)
正方形	理論値	$S=4w^2$ =4900 (1.000)	$S=4(w_1^2+w_2^2)$ =2452 (0.500)	$S=4w$ =140 (0.029)
	実測値	$S^*=5041$ (1.000)	$S^*=2594$ (0.515)	$S^*=142$ (0.028)
正三角形 回転角0度	理論値	$S=(3\sqrt{3}L^2)/4$ =4997 (1.000)	$S=[3\sqrt{3}(L_1^2+L_2^2)]/4$ =2499 (0.500)	$S=(3\sqrt{3}L^2)/8$ =2499 (0.500)
	実測値	$S^*=5048$ (1.000)	$S^*=2498$ (0.495)	$S^*=2655$ (0.526)
正三角形 回転角45度	理論値	$S=(3\sqrt{3}L^2)/4$ =4997 (1.000)	$S=[3\sqrt{3}(L_1^2+L_2^2)]/4$ =2499 (0.500)	$S=(3\sqrt{3}L^2)/4$ =4997 (1.000)
	実測値	$S^*=4969$ (1.000)	$S^*=2464$ (0.496)	$S^*=3879$ (0.781)

1次元分解型はフィルタの形状によって計算量に大きく差がでることが分かる。

次に実測値について検討してみる。分割繰返し型においては理論値と実測値がほぼ一致するが、1次元分解型についてはかなりの誤差が生じた。これは2.2.2項に説明したように、連続系における1次元分解と離散系における1次元分解の差によるものと考えられる。

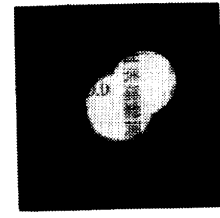
4. 分割繰返し型における誤差解析

4.1 分割小フィルタにおける近似誤差

離散系での分割繰返し型アルゴリズムでは処理結果画像に誤差が現れるが、その原因は用いるフィルタを離散的サンプル点のセットで表現することから発生する(式(6), (7)が厳密には成立しない)。すなわち、フィルタを離散的サンプル点のセットで表現した場合、分割繰返し型アルゴリズムでは、振幅変化の急激な箇所、言い換えればフィルタの輪郭部分で比較的大きな近似誤差が発生する。この誤差を繰返し dilation 演算の回数分だけ累積(増幅)するため、非繰返し型フィルタとの間に無視できない程度の誤差が発生することがある。その一例を図4に示す。処理対象画像を図5(a)とし、半径40のディスクフィルタで dilation 処理を行った結果である。(a)は非繰返し型での処理結



(a) 対象画像



(b) ディスクフィルタによる dilation 処理結果画像

(a) original image. (b) dilated output of (a) by disc filter.

図4 対象画像と dilation 処理結果画像

Fig. 4 Original image and its dilated output.

果画像, (b)は分割数を6としたときの分割繰返し型での処理結果画像, (c)がここで問題となるそれらの差画像である。この差画像の濃度値の総和を画像誤差と呼ぶことにする。

この画像誤差とフィルタ分割数の関係を図6に示す方法で調べた。すなわち、原画像を非繰返し型で処理した結果 F と、分割繰返し型で処理した結果 G との差画像を求める。その差画像の濃度値総和を F の濃度値総和で正規化したものを画像誤差比率 ϵ として分割数との関係を求めた。すなわち、

$$F = f(i, j), G = g(i, j),$$

$$(i = 1, \dots, n, j = 1, \dots, n)$$

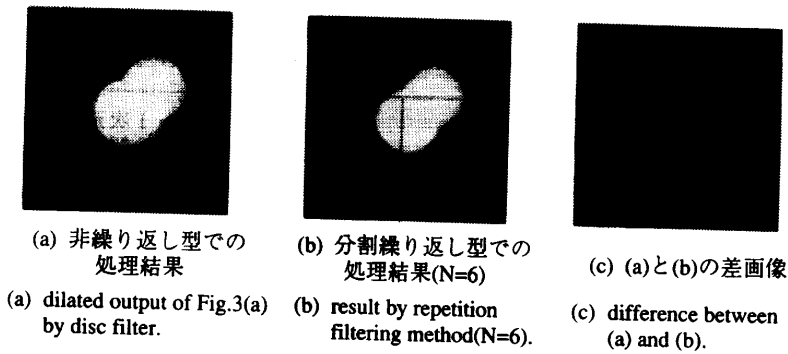


図5 分割繰り返し型で発生する誤差
Fig. 5 Error in resulting image using repetition filtering method.

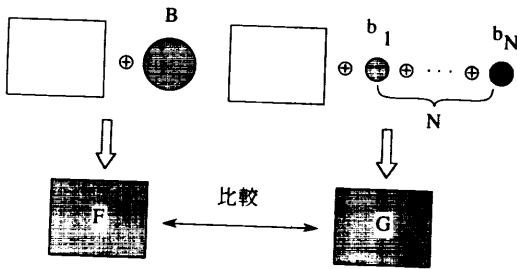


図6 分割数を N としたときの画像誤差の求め方
Fig. 6 Means in obtaining the error in resulting image when the number of repetition is N.

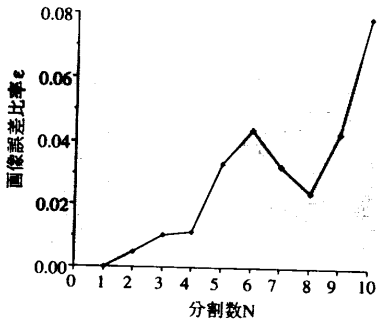


図7 分割数と画像誤差の関係
Fig. 7 Relation between the number of repetition and the error in output images.

とすると,

$$\epsilon = \frac{\sum_{i=1}^n \sum_{j=1}^n |f(i, j) - g(i, j)|}{\sum_{i=1}^n \sum_{j=1}^n f(i, j)} \quad (13)$$

であり、通常の S/N 比の逆数に相当するものである。ディスクフィルタ B の半径を 40 と固定したときの結果を図 7 に示す。

この結果、画像誤差比率はフィルタを細かく分割するほど大きくなるのが分かる。しかし、分割数が 5 から 8 の間で画像誤差比率が上下しており、予想された単調増加ではない。

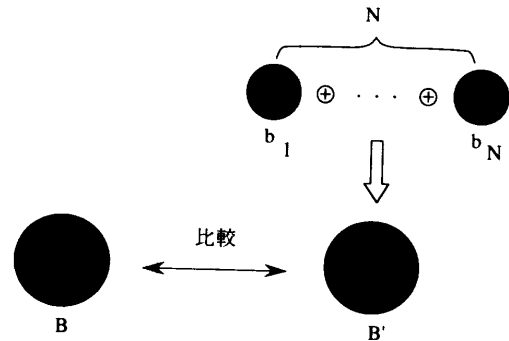


図8 分割数を N としたときのフィルタ誤差の求め方
Fig. 8 Means in obtaining the error in filters when the number of repetition is N.

そこで次に画像誤差比率と分割数の関係を詳しく調べるために、小フィルタの繰り返し dilation 演算によって求められるフィルタと元フィルタとの関係を調べてみた。図 8 に方法を示す。まず、小フィルタのサイズを r で固定する。この小フィルタの N 回の繰り返し dilation 演算によりフィルタ B' を作成する。これとは別にサイズ R のフィルタ B を作成する。B と B' の差を見ることでサイズ r のフィルタによる N 回の繰り返し dilation 演算で蓄積されるサンプリング誤差を計ることができる。ここで、B と B' の差をフィルタ誤差と呼ぶことにする。このフィルタ誤差の画素数総和をフィルタ B の画素数総和で正規化したものをフィルタ誤差比率 ϵ' とする。すなわち、

$$B = B(i, j), B' = B'(i, j),$$

$$(i = 1, \dots, n, j = 1, \dots, n)$$

とすると,

$$\epsilon' = \frac{\sum_{i=1}^n \sum_{j=1}^n |B(i, j) - B'(i, j)|}{\sum_{i=1}^n \sum_{j=1}^n B(i, j)} \quad (14)$$

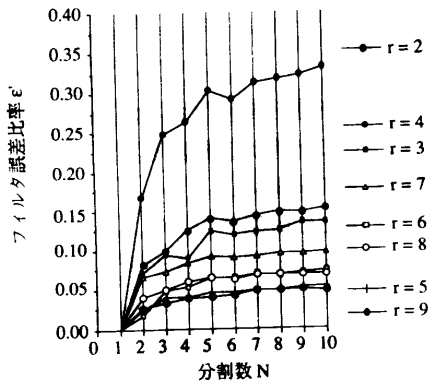


図9 分割数とフィルタ誤差の関係

Fig. 9 Relation between the number of repetition and the error in filters.

であり、これも通常の S/N 比の逆数に相当するものである。

図9は、図8で示した実験で、小ディスクフィルタのサイズ r を2から9としたときのフィルタ誤差比率をプロットしたグラフである。これより、フィルタ誤差比率は dilation 演算の繰返し回数の増加とともにある一定値で飽和することが分かる。また、その飽和値は小ディスクフィルタのサイズごとに異なることが分かる。

小フィルタごとにフィルタ誤差比率の飽和値が異なることから、図7の画像誤差と分割数の関係が説明できる。図7で画像誤差が低く抑えられている分割数8では、 $r=5$ の小フィルタ、すなわち図9のフィルタ誤差比率飽和値の低いフィルタを繰り返し使用していることが分かる。一方、図7でピーク値を示す分割数6では、 $r=7$ 、すなわち図9のフィルタ誤差比率飽和値が $r=5$ のときよりも約2倍大きいフィルタを繰り返し使用していることが分かる。このことから、小フィルタの誤差の累積が最終結果に大きな影響を与えていることが判明した。

図8にはディスクフィルタの結果のみを示したが、正方形フィルタ以外の代表的な多角形フィルタ(表1)についても同じ傾向が観測され、上記結果には一般性があることを確認した。なお、正方形フィルタのみ例外であるが、これは離散系上でサンプリング誤差を持たないことに起因している。

4.2 分割方法の改良

小フィルタのサイズによってフィルタ誤差の現れ方が異なることから、フィルタ分割の際の小フィルタの選択の仕方に工夫を要することが明らかになった。

すなわち、これまででは、元フィルタを等分割することで小フィルタ群を得ていたが、むしろ誤差を少なく

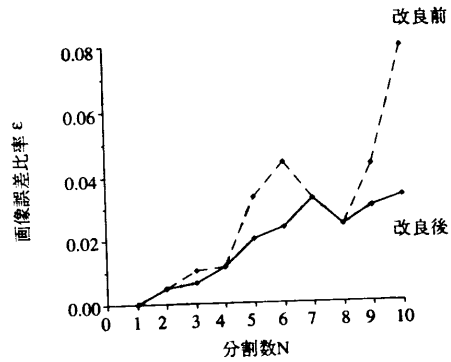


図10 分割方法改良による画像誤差の変化

Fig. 10 Resulting errors in output image before and after modification.

表3 分割方法改良による計算量の変化

Table 3 Difference of computation time between two filter decomposition methods.

N	改良前	改良後
5	985 (1.00)	1057 (1.07)
6	822 (1.00)	894 (1.09)
7	727 (1.00)	727 (1.00)
8	648 (1.00)	648 (1.00)
9	569 (1.00)	593 (1.04)
10	490 (1.00)	550 (1.12)

するような最適の組合せを選択することが重要である。そこで、以下のような改良アルゴリズムを考案した。

図9の各小フィルタのフィルタ誤差比率に飽和現象が見られるが、その飽和値は小フィルタごとに異なるので、その飽和値を該当する小フィルタの評価値とした。そして、小フィルタ群の繰返し dilation 演算で得られる復元フィルタの評価値を、小フィルタ群の評価値の総和で近似した。サイズ R のフィルタを N 個の小フィルタに分割するという条件を満たすすべての小フィルタの組合せに対してこの総和値を求め、その中で復元フィルタの評価値が最小になるものを最適の組合せとして選んだ。

この方法でフィルタを最適分割した結果を使って、図4と同様の画像に対して dilation 処理を行ったときの画像誤差比率を図10に実線で示す。図中の破線で従来の等分割方法を用いたときの画像誤差比率を示しており、今回の改良分割法が良い傾向を示すことが分かる。

次に分割方法を改良したことで、計算量がどのように変化するかを調べた結果を表3に示す。表3から

分かるように分割方法改良により計算量が極端に変化することはなかった。なお表2と同様に改良後と改良前の相対値は下段の括弧内に書いている。

5. 考 察

5.1 アルゴリズムの選択

3章では原画像の参照回数を計算量として、各アルゴリズム、各フィルタの計算量を調べた。その結果、フィルタの形状によって処理時間短縮に有効なアルゴリズムが異なるということが分かった。

特に1次元分解型でこの影響が顕著である。たとえば、正方形フィルタでの計算量が非常に小さいのに対し、回転正三角形フィルタでは目立った計算量の縮小は見られない。正方形フィルタは、垂直、水平両軸に関して対称であり、1次元フィルタを2種類しか必要としないのに対し、回転正三角形フィルタはどちらの軸に関して対称でないので、必要な1次元フィルタの種類が多いということが原因である。すなわち、少数の1次元フィルタに分解できる（たとえば、対称性を持ち、かつ凸形状のもの）フィルタについて1次元分解型は有効であると考えられるが、そうでないフィルタについては必ずしも1次元分解型は適しておらず、次に述べる分割繰返し型よりも劣る場合がある。

次に分割繰返し型について見てみると、リングフィルタを除くすべてのフィルタで非繰返し型の約0.5倍、すなわち $1/N$ （この場合 $N=2$ ）、と安定した計算量の低減がなされている。これはリングフィルタ以外のフィルタが元フィルタの縮小相似形の2つに分割していることに起因しており、フィルタの面積効果がそのまま効いている。なお、リングフィルタの場合は円周フィルタとディスクフィルタという特殊な分割を行ったため、上記面積効果以上の高い効果を実現しており、1次元分解型と比較しても優位である。したがって、処理時間短縮をねらう場合、フィルタの形状にあったアルゴリズムをその都度注意深く選択する必要があることが明らかになった。

5.2 分割繰返し型での最適分割

4章で分割繰返し型で発生する誤差とフィルタ分割方法の改良について述べた。画像誤差は小フィルタを離散系上で表現するときのフィルタ輪郭部分のサンプリング誤差が繰返しdilation演算で累積されることが主な原因であることが明らかとなった。したがって、一般的にはフィルタを細かく分割すればするほどフィルタ誤差が大きくなるといえる。しかし、小フィルタのサイズによって誤差発生の比率が異なるため、小フィルタの選択にあたっては、分割数一定下で最適の組合

わせがあることが判明した。

6. む す び

morphology演算を高速化するための2種類のアルゴリズムについて各種比較検討を行った。得られた結論は以下のとおりである。

(1) 1次元分解型アルゴリズムは、フィルタの形状によって計算量の縮小効果が大きく変動する。たとえば、フィルタが対称形状である場合、あるいは矩形フィルタのようにわずかな種類の1次元フィルタに分解できる場合には非常に効率的なアルゴリズムである。一方、分割繰返し型アルゴリズムは、フィルタ形状による計算量の縮小効果が比較的安定している。たとえば分割フィルタの径がすべて等しい N 個に分割する場合は計算量は非分割型の $1/N$ に縮小される(2次元画像の場合)。

(2) 分割繰返し型アルゴリズムでは、分割数を大きくするにつれ画像誤差が増大する。その増大の仕方は一様ではなく、分割小フィルタの形状・サイズに強く依存する。この画像誤差を極力抑えるための分割小フィルタの選択方法について新たに提案し、その有効性を示した。

以上のとおりであり、具体的応用にあたってはそれぞれの長所、短所を知ったうえで、対象とするフィルタ形状ごとに最適選択を行う必要があること、また各方式の短所が実用上問題とならないよう工夫することが必要であることが分かった。

謝辞 高速化アルゴリズムについてご討論いただいた豊橋技術科学大学情報工学系今井正治教授、本沢邦明氏、金川英一氏に感謝いたします。

参 考 文 献

- 1) Haralick, R.M., Stemberg, S.R. and Zhuang, X.: Image Analysis Using Mathematical Morphology, *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol.PAMI-9, No.4, pp.532-550 (1987).
- 2) Haralick, R.M.: Mathematical Morphology and Computer Vision, *Proc.22nd. Asilomar. Conf. Signals, System and Computers*, pp.468-479 (1988).
- 3) Serra, J. and Vincent, L.: An Overview of Morphological Filtering, *Circuits Systems Signal Process*, Vol.11, No.1, pp.47-108 (1992).
- 4) 松野下純一, 赤松茂雄, 山本眞司: トップダウン型画像認識システム TOPS とその MRI 画像認識への応用, 電子情報通信学会論文誌 (D-II), Vol.J-76-D-II, No.2, pp.304-314 (1993).
- 5) 磯部義明, 大久保なつみ, 山本眞司, 鳥脇純一

郎, 小畑秀文: 孤立性陰影抽出用 Quoit フィルタの性質とその乳癌 X 線陰影抽出への応用, 電子情報通信学会論文誌 (D-II), Vol.J-76-D-II, No.2, pp.279-287 (1993).

- 6) 金 華栄, 小畑秀文: 多重構造要素を用いたモルフォロジーフィルタによる微小石灰化画像の抽出, 電子情報通信学会論文誌 (D-II), Vol.J-75-D-II, No.7, pp.1170-1176 (1992).
- 7) 小島昭二, 海老澤嘉伸, 宮川達夫: 大きな構造要素が使える画像の高速モルフォロジーハードウェア, 電子情報通信学会論文誌 (D-II), Vol.J76-D-II, No.6, pp.1106-1113 (1993).
- 8) Stenberg, S.R.: Biomedical Image Processing, *IEEE Comput.*, Vol.16, No.1, pp.22-34 (1983).
- 9) Zhuang, X. and Haralick, R.M.: Morphological Structuring Element Decomposition, *CVGIP*, Vol.15, No.3, pp.370-382 (1986).
- 10) Xu, J.: Decomposition of Convex Polygonal Morphological Structuring Elements into Neighborhood Subsets, *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol.13, No.2, pp.153-162 (1992).
- 11) 横井茂樹, 鳥脇純一郎, 福村晃夫: 画像処理のための 2 次元フィルタリングの 1 次元分解について, 電子情報通信学会論文誌 (D), Vol.J61-D, No.7, pp.512-513 (1978).
- 12) 仁保 勉, 江 浩, 山本眞司: 各種モルフォロジー演算アルゴリズムの演算速度の比較, 電子情報通信学会技術研究報告, PRU, Vol.94, No.51, pp.25-32 (1994).
- 13) 仁保 勉, 江 浩, 山本眞司: 分割繰返し型モルフォロジー演算の諸性質について, 第 49 回情報処理学会全国大会論文集 (2), pp.159-160 (1994).

(平成 7 年 1 月 5 日受付)

(平成 8 年 7 月 4 日採録)



仁保 勉

平成 5 年豊橋技術科学大学知識情報卒業。平成 7 年同大学院修士課程修了。在学中は画像処理の研究に従事。現在、横河電機(株)に勤務。



江 浩 (正会員)

昭和 57 年中国合肥工業大学・機械卒業。昭和 63 年福井大学大学院修士課程修了。平成 4 年名古屋大学大学院博士課程修了。同年豊橋技術科学大学助手。平成 8 年郵政省通信総合研究所主任研究官, 現在に至る。工学博士。パターン認識, 画像処理, および, その医用画像への応用に関する研究に従事。



山本 眞司 (正会員)

昭和 15 年生。昭和 37 年名古屋大学工学部電子工学科卒業。同年日立製作所中央研究所入所。昭和 49 年同所主任研究員, 昭和 55 年同所研究部長, 昭和 62 年日立製作所那珂工場開発部長, 平成 2 年豊橋技術科学大学教授, 現在に至る。文字認識の研究, 医用画像処理の研究などに従事。工学博士。著書(共著)「パタン認識とその応用」, 「医用画像処理」ほか。