3R-4

グラフィックスハードウェアを用いた 多値モルフォロジーの一考察

司馬　浩之†　　手島　裕詞‡　　西尾　孝治†　　小堀　研一†

大阪工業大学†　　静岡理工科大学‡

1. はじめに

近年，画像処理の需要が大きくなってきており，フィルタリングや特徴抽出など様々な分野で利用されている．その中でも最近注目されている画像処理手法としてモルフォロジー演算[1]がある．モルフォロジー演算は，構造関数と呼ばれるフィルタを自由に設定できるため，処理目的に応じて適切に設定できれば，他の画像処理手法よりも効果が望める場合が多い．しかし，モルフォロジー演算は，計算量が非常に大きいといった欠点がある．そこで，本研究では，グラフィックスハードウェア(Graphics Processing Unit，以下GPUと呼ぶ)を用いて多値モルフォロジー演算を高速に行う手法を提案する．

2. モルフォロジー演算

モルフォロジー演算には，ミンコフスキー和とミンコフスキー差と呼ばれる二つの基本演算があり，それぞれ式(1)，(2)で定義されている．式中の f は処理対象画像，g は構造関数，$f(x)$, $g(x)$はそれぞれの要素を表す．図 1 に基本演算の一例を示す．

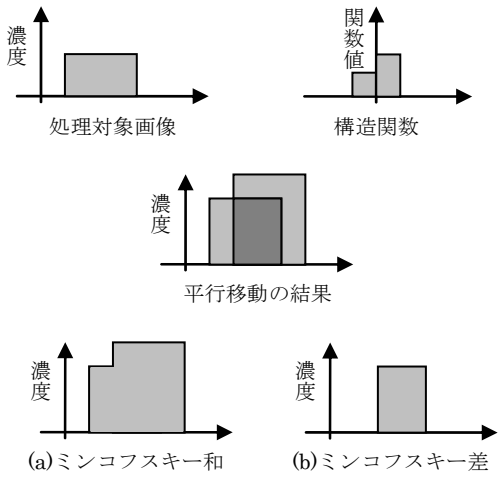

図 1　基本演算

$$[f \oplus g](x) = \max_{\substack{x-u \in F \\ u \in G}} \{ f(x-u) + g(u) \} \qquad (1)$$

$$[f \ominus g](x) = \min_{\substack{x-u \in F \\ u \in G}} \{ f(x-u) - g(u) \} \qquad (2)$$

3. 提案手法

提案手法の処理の流れを図 2 に示す．提案手法では，モルフォロジー演算に用いる構造関数(以下，フィルタと呼ぶ)を正方形の集合に分割する．

分割した正方形を大きさについて昇順にソートし，それぞれの正方形を用いてモルフォロジー演算を行う．また，提案手法では，二つのバッファを用いる．一つは，最終結果を保存するための最終演算結果用バッファ，もう一つは，分割したフィルタの演算結果を保存する正方形演算結果用バッファである．

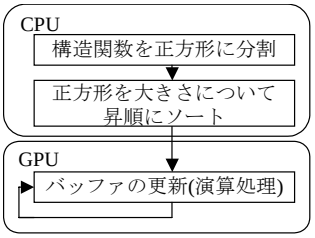

図 2　処理の流れ

3. 1　フィルタの分割

フィルタの分割は，フィルタの定義域を正方形の集合で表現する．また，それぞれの正方形内のフィルタ値は，正方形内の最小値をとる．フィルタの分割例を図 3 に示す．同図の xy 平面はフィルタの定義域，z 軸はフィルタ値を表す．

分割したフィルタの原点は，図 4 のように正方形の左下とする．そして，分割前のフィルタと分割したそれぞれの正方形フィルタの原点が異なるために，フィルタの原点から分割した正方形フィルタの原点までのベクトルを保存する．

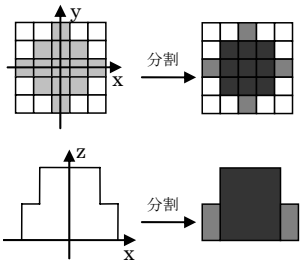

図 3　フィルタの分割

A Study on Multiple-value Mathematical Morphology by Using Graphics Hardware.
Hiroyuki SHIBA, Yuji TESHIMA, Koji NISHIO and Ken-ichi KOBORI.
† Osaka Institute of Technology.
‡ Shizuoka Institute of Science and Technology.

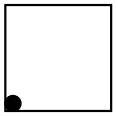

図 4　正方形フィルタ

3. 2　バッファの更新

前節の処理で行った分割結果を元に GPU を用いてモルフォロジー演算を行う．

正方形演算結果用バッファの更新は，分割結果の中で[最も大きな正方形の一辺の長さ－1]回行われる．また，最終演算結果用バッファの更新は，分割した正方形の数と同じ回数だけ行われる．

3.2.1　正方形演算結果用バッファの更新

正方形演算結果用バッファは，図 4 に示したフィルタを用いて処理した結果を保存する．たとえば，正方形演算結果用バッファに，図 5 に示すフィルタを適用することで，一回り大きな正方形で処理した結果が得られる．

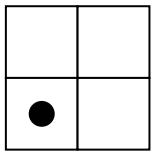

図 5　拡大用のフィルタ

3.2.2　最終演算結果用バッファの更新

モルフォロジー演算は，式(3)の分配則が成り立つ．ここで，式(3)右辺の論理演算は，式(4)で計算できる．このために，前項で得られた演算結果と最終演算結果用バッファの結果を式(4)の f_1 と f_2 として演算を行い，最終演算結果用バッファを更新する．この時に，正方形演算結果用バッファを位置ベクトルに従って，平行移動させる．

$$f \oplus (g_1 \vee g_2) = (f \oplus g_1) \vee (f \oplus g_2) \quad (3)$$

$$(f_1 \vee f_2)(x) = \max\{f_1(x), f_2(x)\} \quad (4)$$

4.　実験および考察

提案手法の有効性を検証するために，実験を行った．実験に用いたフィルタは，正規分布であり，定義域が円，リング，正方形，長方形とした．また，実験環境は，CPU が Pentium4 3.2GHz，メモリが 1GB，GPU が GeForce 6800 Ultra である．

処理対象画像の解像度を 64×64 から 2048×2048 に変化させた場合の処理時間を測定した．従来法は，CPU で演算を行う手法である．その結果を図 6 に示す．

2048 の高解像度である場合に提案手法は，従来法の 5%の処理時間で処理できる．また，フィルタを 2 値とした場合では，荻原らの手法[2]に対して 23%で処理できる．

提案手法は，処理対象画像の解像度の増加が 4 倍になっても，処理時間が約 3 倍しか増加しない．処理対象画像が高解像度になるほど提案手法が有効であることを確認した．また，2 値のフィルタに対しても有効であることを確認した．

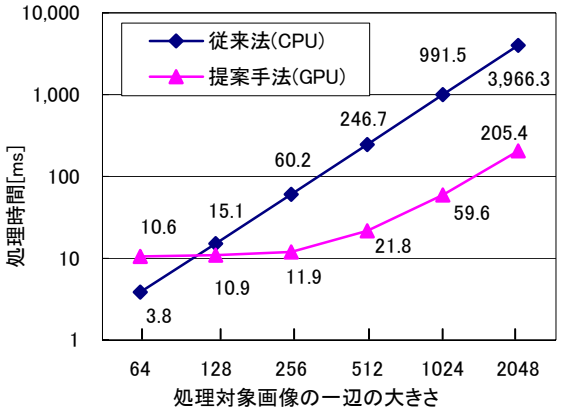

図 6　処理対象画像の解像度を変化させた場合

次にフィルタの定義域を 2 倍，4 倍と変化させた場合の処理時間を測定した．その結果を図 7 に示す．分割なしでは，処理時間が約 3.6 倍に増加しているのに対して，提案手法では，約 3 倍の増加に抑えられている．フィルタを分割することで高速に処理できることを確認した．

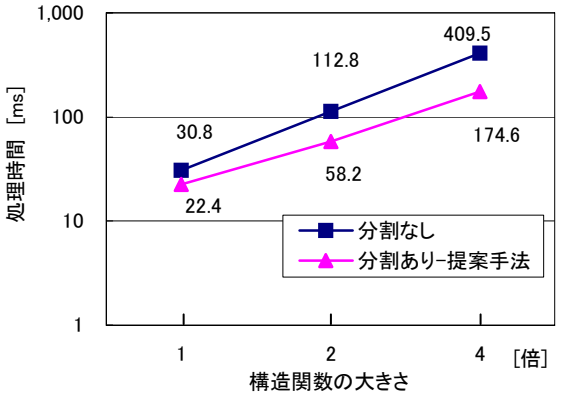

図 7　フィルタの解像度を変化させた場合

5.　おわりに

提案手法では，フィルタを正方形に分割することで高速に処理する手法を提案した．今後の課題として，提案手法の GPU で処理を行っている間の CPU を有効活用する手法の開発や，より有効な分割手法の考案などが挙げられる．

参考文献

[1] 小畑秀文：“モルフォロジー”，コロナ社，1996.

[2] 萩原義裕，小畑秀文：“高速モフォロジーフィルタリングアルゴリズム”，電子情報通信学会論文誌 Vol.J85-D-Ⅱ No.8 pp.1341-1350，2002.