

# マトリクスアーキテクチャ型超並列 SIMD 演算プロセッサを用いた モルフォロジカルパターンスペクトラムの実装と評価

Implementation and evaluation of a morphological pattern spectrum  
using an highly-parallel SIMD matrix processor

塚田靖史<sup>‡</sup> 竹田知弘<sup>‡</sup> 本多隼也<sup>‡</sup> 熊木武志<sup>‡</sup> 小倉 武<sup>‡</sup> 藤野 毅<sup>‡</sup>

Yasushi Tsukada Tomohiro Takeda Toshiya Honda Takeshi Kumaki Takeshi Ogura Takeshi Fujino

## 1. まえがき

近年のコンピュータ技術の進歩により、画像や映像等のデジタルデータは高画質、高解像度のもので取り扱える様になり、より鮮明度が増している。またデジタル化の普及に伴い、画像の伝送・認識のためのデジタル画像処理は幅広く用いられている。これにより、様々な手法で画像、映像データから情報抽出を行うことが可能となり、テキスト解析やオブジェクト判別といった解析が一般化してきた。画像等の視覚情報から認識を行う上で重要となる特徴の 1 つに、細かな形状が多数集まり形成されているテキストと呼ばれる模様があり、人間はその違いを捉え認識を行う。また、きめの細かさの違いから風景の遠近感を得られるため、テキストの特徴化・解析は重要視されている [1]。オブジェクト判別は画像に写る物が何であるのかを特定するものであり、監視カメラや航空技術等で幅広く扱われてきている。これらを実現するためのアルゴリズムは様々なものが知られているが、そのデジタル信号の解析技術の一種であるモルフォロジカルパターンスペクトラム [2] と呼ばれるものがあり、注目画素とその近傍画素とで簡単な演算処理を行う事で効果的な解析が可能となるアルゴリズムである事が知られている。このアルゴリズムを用いた既存の研究として、テキスト解析 [3] に加え、画像に写る人物のシルエットからパターンスペクトラムを行い、ズボンやスカートまたは胸部の大きさや体型などの情報を抽出し、男女判別を行う [4] 等の応用例が報告されてきた。

パターンスペクトラムは、画像の解析技術として、広い応用が期待できる技術である。しかし、このアルゴリズムは、大量の画素間演算を行う必要があり、オープニングやクロージング、輝度値計算等を幾度と行う。またメモリアクセスが頻繁に行われるため、膨大な処理時間が要求される。このため、一般的なプロセッサによる逐次処理では演算性能に限界があり [5]、実用面に関して問題があるのが現状である。一方、スマートフォン、携帯電話、及びデジタルカメラ等の、モバイル機器においては、カメラ機能を搭載することが当たり前となってきており、効果的な画像の解析技術が必要となってきている。しかしながら、これらの機器は、消費電力、及び実装面積等、低コストに目的の処理を実現するため、演算機能やハードウェア量に制限のある組み込みプロセッサが用いられることが多い。以上の背景より、本研究では C 言語ベースのプログラムによるソフトウェアベースプロセッサでありながら、効果的

な並列処理が可能な LSI コアであるマトリクスアーキテクチャ型超並列 SIMD (Single Instruction Multiple Data) 演算プロセッサ MX-1 [6] を用いて実装、評価を行う。モルフォロジカルパターンスペクトラムの解析能力を組み込み機器に実装させ、高い画像解析能力を獲得しつつ、低消費電力、低コストに処理を実現させ、組み込み機器への実用性を高めることを目指した。MX-1 は、モバイル機器におけるマルチメディア処理に適した組み込み向け LSI アーキテクチャであり、低消費電力で最大 1,024 並列の演算処理を実行できる。我々は 1,024 並列かつ、演算ユニット間的高速データ転送が可能な MX-1 の処理能力を活かし、ソフトウェアベースのプロセッサでは難しいとされてきた、モルフォロジカルパターンスペクトラムの隣接画素間演算並列化を効率的に行う手法を提案する。

以降、2 章ではモルフォロジカルパターンスペクトラムについての概要、3 章ではマトリクスアーキテクチャ型超並列演算プロセッサ MX-1 の概要と仕組みについて述べ、4 章では MX-1 を用いたモルフォロジカルパターンスペクトラムの並列実装法について詳述し、5 章で実装結果とその評価について述べることにする。

## 2. モルフォロジカルパターンスペクトラム

### 2.1. モルフォロジー演算

モルフォロジー演算 (Morphological operation) は画像処理の体系の一つとして考案され、画像が特徴的な構造の集まりであると捉えることで、対象とする画像構造の大きさや形状を見出すことを可能とするものである。モルフォロジー演算では「対象画像」と「構造要素」と呼ばれる小図形との間の集合演算を行う。また、それぞれの画素値が 2 値 (0 と 1) か、多値 (0 ~ 255) かによって、表 1 に示す 3 つ (SP: Set Processing, FSP: Function and Set Processing, FP: Function Processing) に分類され、それぞれ隣接画素との演算が異なる [7]。画像認識の研究においては、一般的に SP が用いられていることが多いが、SP の場合、対象画像を 2 値化しなければならない。そのため、対象画像に含まれる情報を削り取ってしまい、アプリケーションで利用する際に、十分な解析を行えない可能性がある。そのため、元の画像情報を保持しつつ、FP よりもメモリ使用量を軽減させるために、本論文では FSP を用いて解析を行う。FSP の処理については基本的な演算が下記の通り定義されている。原画像を  $X$ 、その中に含まれる画素を  $x$ 、Opening 画像中の画素  $y$  として、構造要素を  $B$ 、 $B$  の反転を  $B^S = (-b : b \in B)$  と定義する。

<sup>‡</sup>立命館大学理工学研究所 〒 525-8577 滋賀県草津市野路東 1 丁目 1-1

$$\begin{aligned}
 \text{Dilation} &: (X \oplus B^S)(x) = \max X(y) : y \in (B^S)_x \\
 \text{Erosion} &: (X \ominus B^S)(x) = \min X(y) : y \in (B^S)_x \\
 \text{Opening} &: (X \ominus B^S) \oplus B^S(x) \\
 \text{Closing} &: (X \oplus B^S) \ominus B^S(x)
 \end{aligned}$$

オープニングは図形の辺縁を内側から滑らかに、クロージングは図形の辺縁を外側から滑らかにする一種の平滑化処理である。オープニングはエロージョンの後ディレイションを行い、クロージングはディレイションの後エロージョンを行う。これらの演算イメージを図1に示す。

表 1: モルフォロジー処理の分類。

	SP	FSP	FP
対象画像	2値	多値	多値
構造要素	2値	2値	多値
演算	AND, OR	比較演算	加算, 減算 比較演算

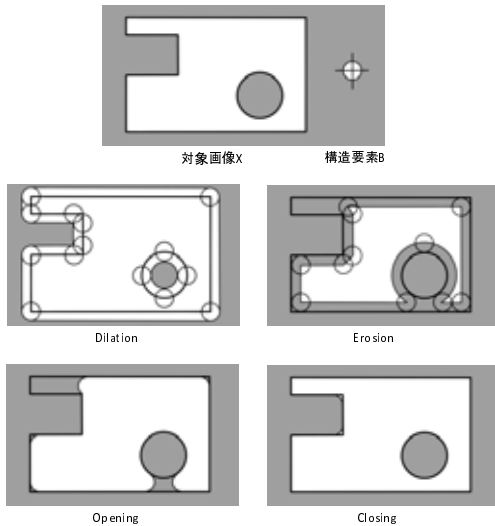


図 1: モルフォロジー処理の基本演算イメージ。

## 2.2. パターンスペクトラムアルゴリズム

パターンスペクトラム (Pattern Spectrum) アルゴリズムは、構造要素が原画像中のどの部分を表現しているかを、形状やサイズの分布として表したものであり、視覚的な構造を識別する効果的なアルゴリズムとして、画像の特徴抽出に利用される。図2にモルフォジカルパターンスペクトラムを用いた処理の流れを示す。構造要素を  $n$  回スケールアップしたものに対するオープニングを  $X_{nB}$  として、任意の回数までオープニング処理を行い、スケールアップした構造要素に対するオープニングの変化を調べるため、 $X_{nB} - X_{(n+1)B}$  差分図形を求め、差分図形の面積を画像の特徴として捉える。この結果がパターンスペクトラムとなる。

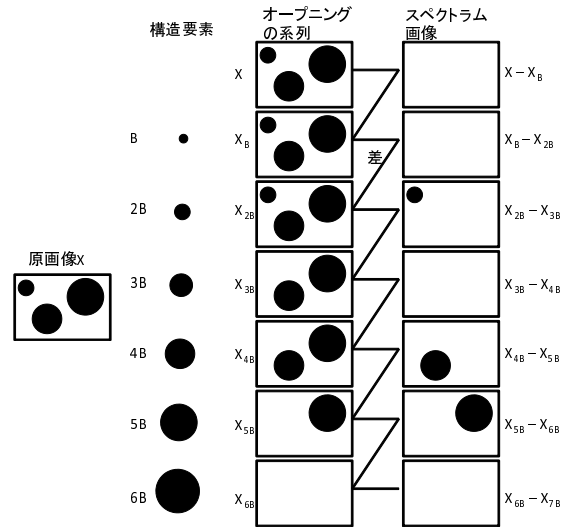


図 2: モルフォジカルパターンスペクトラム処理の流れ。

## 3. 組み込み機器向け超並列 SIMD 型演算プロセッサ MX-1

本章では、組み込み機器向け超並列 SIMD 型演算プロセッサである MX-1[6] について述べる。MX-1 は、ルネサスエレクトロニクス社が画像データ等に対し数値・論理演算を行う際の、効率的な並列処理を実現するために研究・開発を行ってきた SIMD 型演算プロセッサである。SRAM をベースとし、演算器の配置、データの処理ベクトルを工夫することにより、従来のプロセッサと比べて並列度を大幅に向上させ、演算器とデータレジスタ領域を密結合することによって、PE、メモリ間の I/O 転送にかかる消費電力を削減している。90nm 7Cu CMOS テクノロジーの実装結果では、面積 3.1 mm<sup>2</sup>、消費電力 250mW であり、16 ビットの加算処理では動作周波数 200MHz で 40GOPS (Giga Operation per Second) の性能を得ることが可能である [6]。我々は、これまでに画像情報変換処理、電子透かし処理、及び暗号化処理等、様々なアプリケーションを実装してその効果を確認してきた [8, 9, 10, 11]。

MX-1 は C 言語ベースのプログラムで様々なアプリケーションを処理するためのハードウェア構成を採用しており、図3に示すように、大きく分けて、SDRAM、制御用 CPU コア、DMA (Direct Memory Address)、及び MX コアの 4 つで構成されている。MX コアは 1,024 個の演算器 (PE: Processing Element) が、2 つの SRAM (512 ビット × 1,024 エントリ) に挟まれた形で配置され、外部メモリとのバスであるインターフェースモジュール、MX コアを制御するコントローラを備えており、1 命令で最大 1,024 のデータを 2 ビット単位で並列に処理することが可能である。また、制御用 CPU はプログラムのフロー制御、及び逐次処理実行部分に使用されるため、CPU と MX コアを効率よく動作するようにプログラムを作成することが重要となる。命令の実行は、制御用コントローラ内部の命令メモ

リから制御線を通して PE に命令が送られた後, 1,024 個の PE が左右の SRAM から読み出した値を演算することによって行われる. MX コアの各 PE は加算器, 乗算器, 及び論理演算器等から構成されており, Valid flag(V フラグ) が備わっている. 演算が行われる PE を V フラグで指定でき, 1 が格納されているエントリは演算可能となり, 0 が格納されているエントリは演算を行わない. 演算器と左右のエントリは水平チャンネル (Horizontal channel) で接続され, これが 1,024 本垂直方向に並んでいる. これにより単一命令で 1,024 並列の演算を可能にしている. 垂直方向でデータをやり取りする際には, 垂直チャンネル (Vertical channel) を利用する. 演算処理の方法は, 従来のメディアプロセッサや専用プロセッサの多くがビットパラレル・ワードシリアル (Bit-Parallel Word-Serial) であるのに対し, ビットシリアル・ワードパラレル (Bit-Serial Word-Parallel) の手法をとっている. そのため, インターフェースモジュールは, 外部 SDRAM に格納されている演算対象データの処理ベクトルを直交に変換して, SIMD 型演算モジュールを効率よく動作させる働きを担う. コントローラは内部に命令メモリを持ち, アプリケーションにあわせてプログラムを入れ替えることで柔軟にマルチメディアアプリケーションを処理することが可能となっている.

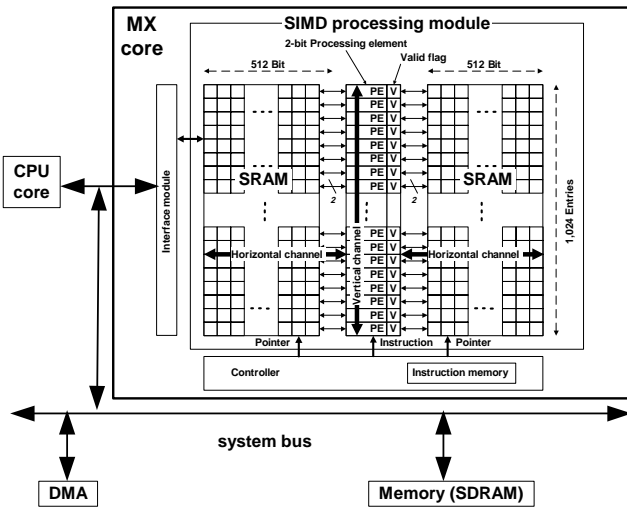


図 3: MX-1 の全体構成 .

#### 4.MX-1 によるモルフォロジカルパターンスペクトラムの実装手法

本章では, モルフォロジカルパターンスペクトラムに対して, MX-1 を用いて効率的な並列処理を実現させる提案手法について述べる.

##### 4.1. モルフォロジー演算の処理手順

パターンスペクトラムの算出のためには, モルフォロジー演算の基本処理であるオープニングを行う. MX-1 で処理を実装する手順を説明する. 例として, 正方形の構造要素を  $3 \times 3$  ピクセルとしてモルフォロジー演算の処理を説明し, その後一般化する. 対象となる対

象画像 ( $32 \times 32$  ピクセル) とその画素を図 4 のように表現する .

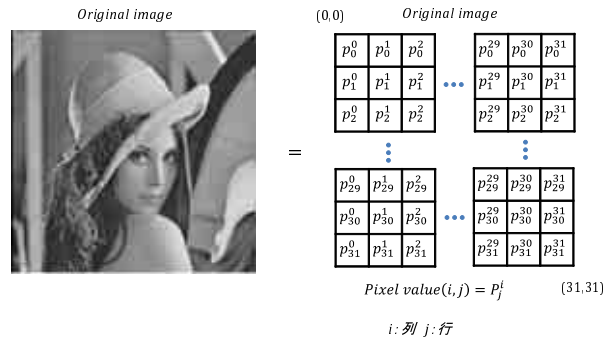


図 4: 原画像データの画素表現 .

モルフォロジー演算の流れは以下の手順をとる .

手順 (1) MX の SRAM 内に画像データを格納する . SRAM の 1 つのアレイに図 4 に示す画素を 1 行目から順に縦に並べていく ( $p_0^0, p_0^1, p_0^2, \dots, p_{31}^{31}$ ). 対象画像の各画素は SRAM の各エントリに対応する形でインターフェースモジュールを介して図 5 のようにデータを格納する .

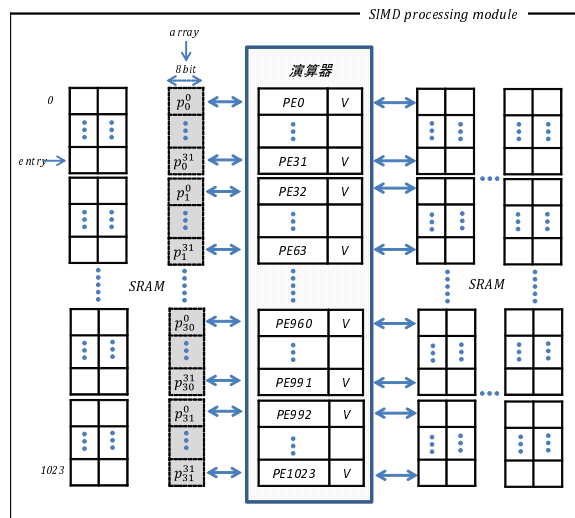


図 5: 処理画素と MX コアエントリとの関係 .

手順 (2) 構造要素の形状に合わせて, 画素データを SRAM 内で移動させ, データ配置を行う. MX-1 で並列処理を効率よく行うには, 画素データをどのように配置するのが重要となる. モルフォロジー演算を並列に行うため, 手順 (1) で入力された画素データを演算用に構造要素に合わせて展開する必要がある .

$3 \times 3$  ピクセルの正方形形状構造要素でモルフォロジー演算を行う場合, 並列度を向上させるため, 図

6 に示すように画素を横一列に配置する．図中の番号付けは構造要素の位置付けのためのものである．演算注目画素は「5」に対応しており，その周りの近傍セル画素との間で最小値・最大値探索を行う．この際に図 6 のように，水平に配置して探索を行うことで，1 画素に対する近傍セルとの探索演算が一つのエントリ内で行うことができ，1,024 画素分の演算を一度に行うことが可能となる．

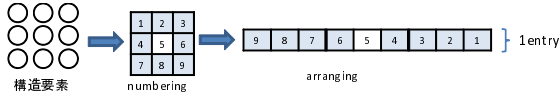


図 6: 構造要素 3 × 3 での処理工程．

実際，構造要素 3 × 3 の場合の対象画像に対する適用は，図 7 に示すように，注目画素とその隣接画素が全要素に当てはまるパターン A，上下左右のどれか一列がはみ出て処理されるパターン B，上下の一行と左右の一行がはみ出て処理されるパターン C があり，3 種類のいずれかとなる．対象画像のパターン A (処理対象画素  $p_1^1$ )，パターン B (処理対象画素  $p_{31}^1$ )，パターン C (処理対象  $p_{31}^1$ ) の部分を注目画素群とし，また適用された近傍セルの画素情報を図 8 のように配置する．

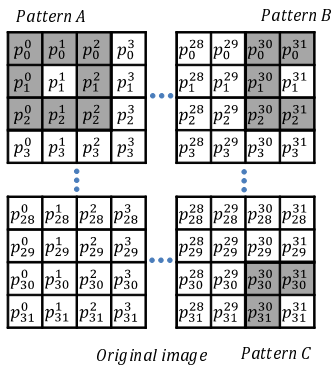


図 7: 正方形の構造要素時の 3 パターン．

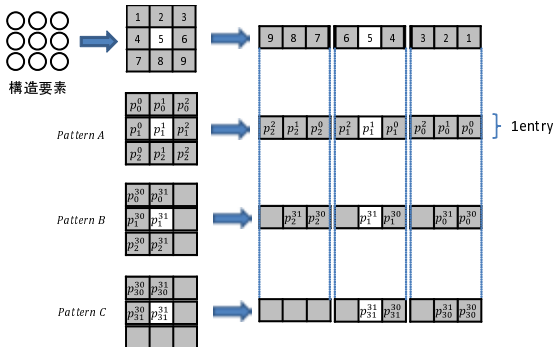


図 8: パターン別での画素配列．

以上の手順で全画素に構造要素を適用すると，注目画素と隣接画素は図 9 のように SRAM 内で展開される．MX コアは垂直チャネルを用いて縦方向のデータ転送を行うことができるため，手順 (1) で入力された画像データを縦方向にずらして，他のアレイに転送することで展開を行う．

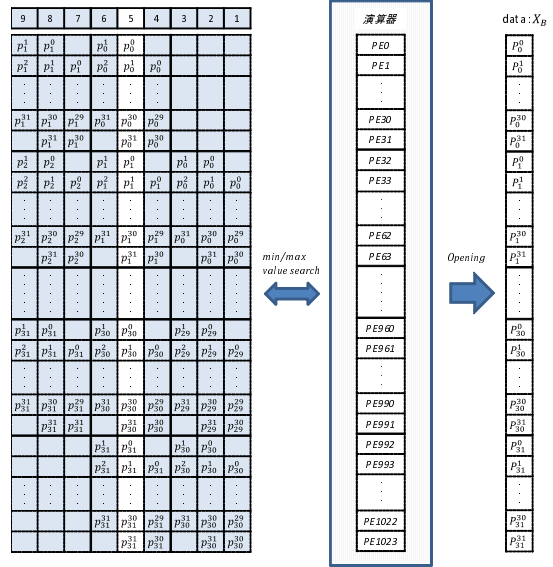


図 9: 注目及び隣接全画素の配置と演算結果の出力 (構造要素 3 × 3)．

手順 (3) 図 9 の配置が 3 × 3 の構造要素に対する MX コアでの画像データ全体のメモリ配置となり，1 つのエントリ毎で最小値・最大値探索を行うことで，モルフォロジー演算による各画素のデータ遷移を 1,024 並列で実現できる．

手順 (4) 手順 (1) ~ (3) の流れでエロージョン (最小値探索) を行った後に手順 (1) に戻り，手順 (3) でディレイション (最大値探索) を行い，オープニング処理を実現する．図 9 に示すように，オープニング処理が行われたデータは SRAM 内の任意のアレイに出力される．図中の  $X_B$  は，対象画像  $X$  を構造要素  $B$  (3 × 3 ピクセル) でオープニング処理したデータである．

以上の手法は構造要素 (3 × 3 ピクセル) の場合であるが，モルフォロジカルパターンスペクトラムでは構造要素のスケールアップを任意の回数まで行い，その度にオープニング処理を行うため，構造要素の大きさによるデータ配置を一般化する必要がある．構造要素  $N \times N$  ピクセルの場合は，画素が図 10 のように配列される．図中の  $E$  は前述と同様に，構造要素の位置付けするためのものである．

全体の MX メモリの展開配置を一般化したものを，図 11 に示す．このように画素群を配置し，1 つのエントリ内で最小値・最大値探索を行うことで，オープニング処理を MX コアの最大エントリである 1,024 並列に行うことが可能となる．

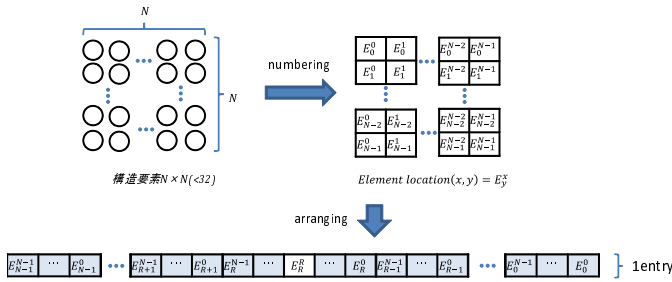


図 10: 構造要素  $N \times N$  での処理工程 .

4.2. パターンスペクトラムの処理手順

モルフォロジカルパターンスペクトラムは、まず 4.1 で述べたモルフォロジー演算のオープニング処理を行い、原画像  $X$  と生成したオープニング画像群  $X_B \sim X_{15B}$  を用いて、減算処理  $X_{nB} - X_{(n+1)B}$  を行うことで実現する。その様子を図 12 に示す。画像  $X_{nB}$  の画素を  $A_j^i$ 、画像  $X_{(n+1)B}$  の画素を  $B_j^i$  と表す。減算処理は各エントリ毎に 2 つの画像データの各画素同士で行われる。その後、各減算結果の全輝度値を構造要素のスケール毎に集計してパターンスペクトラムとする。実験結果については 5.1 節で詳述する。

4.3. MX-1 による実装上の高速化について

MX-1 内部の制御用 CPU は通常の逐次処理に加えて、MX ライブラリの命令呼び出し動作も逐一行っているため、通常のプログラムでは制御用 CPU の負担が重くなり、速度に大きく影響を及ぼす。

本論文では、CPU と MX コアを並列に動作させて処理速度を向上させるために、DMA を利用しデータ転送時の CPU の負担を軽減する。また、CPU に対しては、MX 命令の呼び出し動作を最小限に留め、プログ

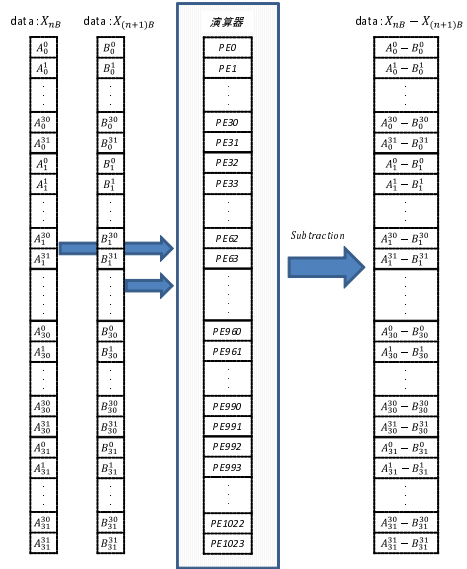


図 12: 減算処理の手順 .

ラム内の変数計算等、並列処理と関係のない処理に集中させ、負荷を軽減させる。これらを実現させるため、データ配置と比較演算を行うモルフォロジー演算、減算処理と輝度値集計を行うパターンスペクトラム演算の MX 動作命令をユーザマイクロコードとしてまとめ、制御用 CPU から MX に対する呼び出し命令数の削減を実現させる。ユーザマイクロコードを生成するためには、制御用 CPU、DMA、そして MX コアの各動作命令をプログラム内にて混在させずに記述する必要があり、各処理工程をそれぞれのコアに担当させ、並列動作させるようにする。ここで、CPU と MX コアによる並列動作の概念を図 13 に示す。並列動作を考慮しない

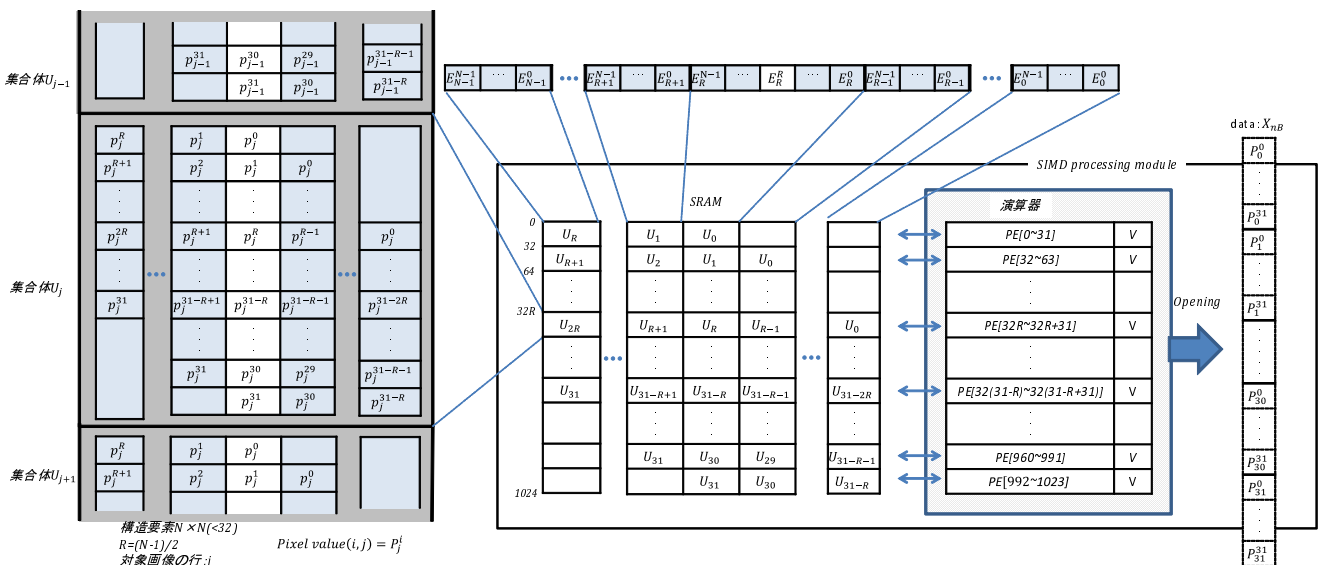


図 11: 全画素の配置 (構造要素  $N \times N$ ) .

通常の実装 (Straight forward implementation) では MX 動作と並行して, CPU の制御動作に加え, 呼び出し (Polling) 動作を行い, CPU 動作時に MX コアでは動作待ちの状態になっているが, MX 動作命令をユーザマイクロコードとしてまとめた最適化実装 (Optimized implementation) では, CPU のポーリング動作が削減されている. その結果, CPU と MX コアの効率的な並列動作が可能となり, 実行時間の短縮が実現できる. この検証結果については, 5.2 節にて述べる.

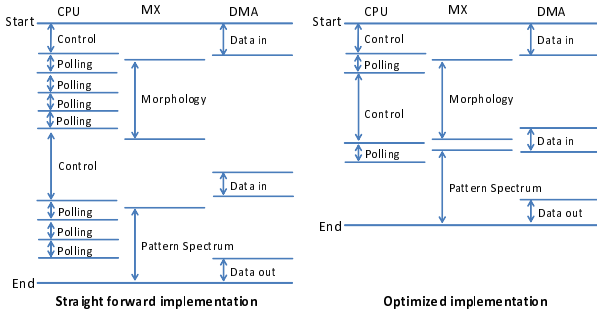


図 13: CPU と MX コアによる並列動作の概念.

5. 実装結果と評価

MX-1 による並列実装の効果を検証するため, ルネサスエレクトロニクス総合開発環境である High-performance Embedded Workshop (HEW) [12] と評価ボードを用いて実装を行った. 尚, MX コアの動作を HEW にてサポートするため, HEW には専用のデバッカ, コードジェネレータを組み込んでいる.

5.1. MX-1 によるモルフォロジカルパターンスペクトラム処理の実装結果

今回 2 種類の対象画像を用意し, 各画像のモルフォロジカルパターンスペクトラムの画像結果と輝度値結果を, 図 14, 15, 及び図 16, 17 に示す. 今回は FSP 処理を行っているため, 輝度値の合計を算出する. 図 14, 15 において, オープニング処理による各画素の状態遷移は構造要素のサイズの変化によって変化する. その変化の差分を視覚的に表したものが, スペクトラム画像になる. 図 16, 17 では, スペクトラム画像の全画素の輝度値を集計したものであり, 構造要素サイズに対するオープニング画像の変化を示している. なお, 構造要素は 14 回スケールアップ (n=15) している.

5.2. MX-1 によるモルフォロジカルパターンスペクトラム処理の時間測定

MX-1 によるパターンスペクトラムを評価ボードとシミュレータ (HEW) で実装し, 時間計測を行った. 時間測定は, パターンスペクトラム処理をデータ入力 (対象画像: 32 x 32 ピクセル), オープニング処理 (スケールアップ回数 n=15), そして減算処理までを行い計測した.

評価ボードと HEW の実装環境を表 2 に示す. 評価ボードと HEW 上で同様のプログラムを動作させた処理時間と HEW 上で高速化を行った処理時間の結果を

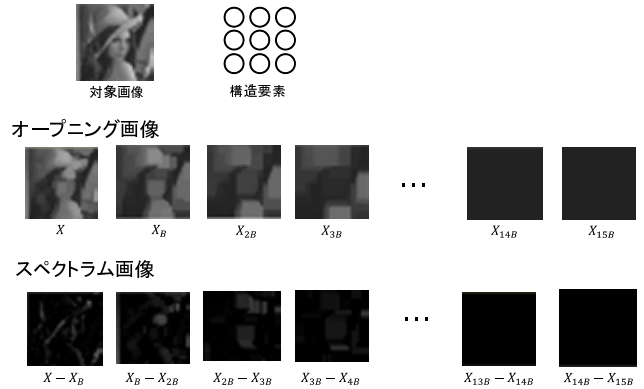


図 14: MX-1 によるパターンスペクトラム演算の画像結果 (Lenna).

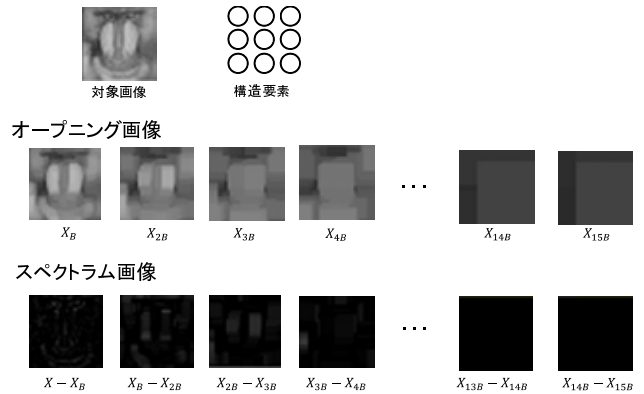


図 15: MX-1 によるパターンスペクトラム演算の画像結果 (Mandrill).

表 3 に示す. 表 2 の通り, CPU の種類が異なり, 動作周波数にも違いがあるため, 処理時間に違いが生じた.

次に, HEW 上で 4.3 節で述べた, ユーザマイクロコードを生成した処理の最適化プログラムによる効果の検証を行った. 表 3 より, CPU と MX コアを並列に動作させた場合, 約 5 倍処理速度が向上していることが分かる. また, 通常プログラムと最適化プログラムの各処理にかかる MX コアと制御用 CPU のサイクル数についての結果を図 18 に示す. MX コア側のサイクル数はほとんど変化しないが, 全体の占める制御用 CPU サイクル数の割合が削減され, 全体の処理サイクル数が大幅に減少していることが分かる. MX-1 は制御用 CPU と MX コアが並列して動作しており, 制御用 CPU が MX コアの動作処理を行っているため, 制御用 CPU の制御動作の負荷を最小限にしたことで, 効率的な並列動作を可能とした. 以上より, CPU と MX コアの並列動作効果を確認することができた.

5.3. 既存プロセッサとの処理性能比較

MX-1 の処理性能を客観的に評価するため, 他のプロセッサと比較するにあたり, 実際にノート PC やデジタルカメラ等のモバイル機器に用いられているプロセッ

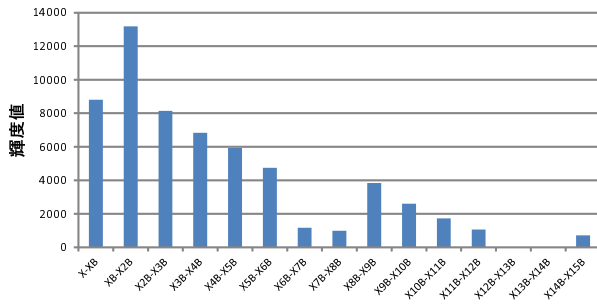


図 16: スペクトラム情報の輝度値 (Lenna) .

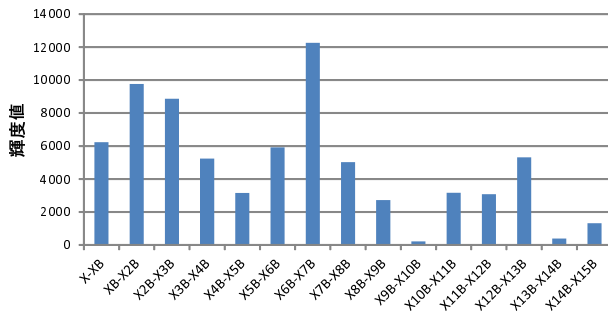


図 17: スペクトラム情報の輝度値 (Mandrill) .

サとの比較を行った。既存のプロセッサには、MX-1 と同じくソフトウェアベースのシステムであり、低消費電力下で動作可能なハードウェアコストの低いものを選んだ。比較対象として、ARM 社の Cortex-A8, AMD 社の Geode, インテル社の Atom を用いた。それぞれの仕様を表 4 に示す。

TI DM3730 (ARM AM3715 Cortex-A8) は ARM 社のコアを用い、一般に普及率の高いプロセッサであり、超小型組込みボード BeagleBoard [13] 等に搭載されている。また NEON テクノロジーと呼ばれる SIMD 機構が組み込まれ、MX-1 と同様に並列処理が可能となっている。更にノート PC に使われている Geode LX800 [14], Atom(TM) N450 [15] で実装を行い、処理速度におけるモルフォロジカルパターンスpektrラムの検証を行った。

各プロセッサ上において、MX-1 で実装したものと同じ動作内容 (画像: 32 x 32 ピクセル, スケールアップ

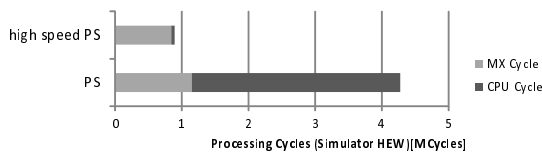
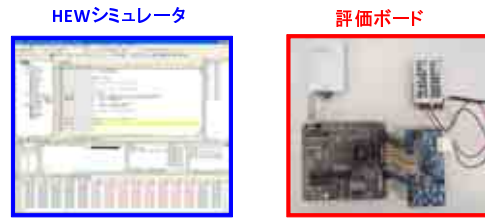


図 18: パターンスpektrラムと高速化パターンスpektrラムでの処理サイクル数 .

表 2: MX-1 におけるシミュレータと評価ボードの環境詳細 .



制御用 CPU	種類	SH-2A (HEW)	M32R (評価ボード)
	動作周波数	200MHz	81MHz
MXコア	動作周波数	200MHz	162MHz
	消費電力	123.5mW	200mW

表 3: パターンスpektrラムと高速化パターンスpektrラムの計測結果 .

	Program	Processing time[ms]
実機	通常プログラム	51.716
HEW	通常プログラム	21.371
	高速化	4.466

回数 n=15) のプログラムを動作させ、時間計測を行った。MX-1 の評価ボード、及びその他プロセッサでの実装に加えて、ARM Cortex-A8 の NEON テクノロジーを用いた実装も行い、MX-1 との並列処理での比較も行った。その結果を図 19 に示す。図 19 より、MX-1 が明らかに処理が高速であることが分かる。MX-1 の評価ボードは ARM Cortex-A8 (NEON) より約 22 倍、Geode LX800 より約 85 倍、Atom N450 より約 21 倍速い処理速度を得ることができ、モルフォロジカルパターンスpektrラムの並列実装効果を確認することができた。各プロセッサの動作周波数を比べると MX-1 が劣ってはいるものの並列度に特化した構造から、実行速度の優位性が獲得でき、この結果からモルフォロジカルパターンスpektrラム処理において、並列化実装し、処理を効率良く行う事が可能であることが分かった。

また、MX コアの並列度を 1,024 から更に増加させた場合には、解析できる画像のサイズを拡張することもできるため、高画質での解析が可能である。

表 4: 既存のモバイルプロセッサの環境仕様 .

Processor	動作周波数
TI DM3730(ARM AM3715 Cortex-A8)	1Ghz
Geode LX800	0.50GHz
Intel(R) Atom(TM) CPU N450	1.66GHz

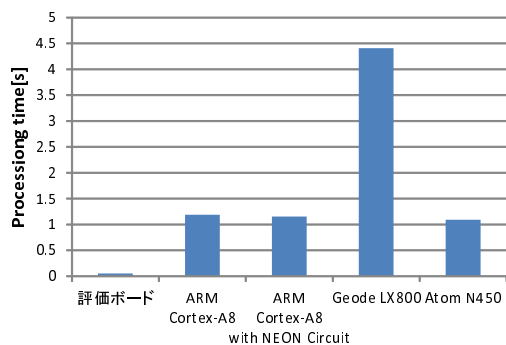


図 19: パターンスペクトラム実装の処理時間比較結果。

## 6. まとめ

本研究では、超並列 SIMD 演算プロセッサ MX-1 を用いて、画像解析技術の一種であるモルフォロジカルパターンスペクトラムの並列実装を実現し、MX-1 実装結果については、プログラムのユーザマイクロコード化により処理速度が約 5 倍向上した。更に、実装評価として一般的に広く普及しているプロセッサと比較を行い、ARM Cortex-A8 より約 22 倍、Geode LX800 より約 85 倍、Atom(TM) N450 より約 21 倍速い処理速度を得ることができた。

## 謝辞

本研究の一部は、日本学術振興会科学研究員補助金 若手研究 (B) (No.24710190)、科学技術振興機構 A-STEP (FS) (AS242Z03360H) の支援により行われた。

## 参考文献

- [1] <http://www.sice.jp/handbook/テクスチャ解析>
- [2] 浅野晃, 浅野千恵, 木森義隆, 棟安実治, 延原肇, 藤尾光彦, “非線形画像・信号処理,” 丸善株式会社, pp.43-67, Sep., 2010.
- [3] 宮川美穂, 浅野晃, 藤尾光彦, “パターンスペクトラムを用いたテクスチャ画像の特徴抽出,” 信学技報, 100 巻, 566 号, pp.123-128, Jan., 2001.
- [4] 数藤恭子, 大和淳司, 伴野明, “モルフォロジー処理によるパターンスペクトラムを特徴量に用いた男女識別法,” 信学論, D-II, vol.J80-D-II, No.5, pp.1037-1045, May, 1997.
- [5] T. Ikenaga, et. al., “Real-Time morphology processing using highly parallel 2-D cellular automata CAM<sup>2</sup>,” IEEE Trans. Image processing, Vol. 9, No. 12, pp. 2018-2026, Dec. 2000.
- [6] M. Nakajima, et. al., “A 40GOPS 250mW massively parallel based on matrix architecture,” ISSCC Dig. Tech. Papers, pp. 410-412, Feb., 2006.
- [7] P. Maragos, “Morphological-Part II: Their relations to median, order statistic, and stack filters,” IEEE Trans. Signal processing, Vol.35, No.18, pp.1170-1184, Aug. 1987.
- [8] T. Kumaki, M. Osawa, S. Itaya, T. Ogura, and T. Fujino, “Decomposition/Reconstruction Acceleration of Max-Plus Algebra-Based Morphological Wavelet Transform with Massive-Parallel SIMD Matrix Mobile Processor,” Journal of Signal Processing, vol. 15, no.6, pp. 425-434, Nov., 2011.
- [9] T. Kumaki, T. Koide, H. J. Mattausch, Y. Kuroda, T. Gyoten, H. Noda, K. Dosaka, K. Arimoto, and K. Saito, “Integration architecture of content addressable memory and massive-parallel memory-embedded SIMD matrix for versatile multimedia processor,” IEICE Trans. Electron., vol. E91-C, no.9, pp. 1409-1418, Sept., 2008.
- [10] 本多隼也, 望月陽平, 熊木武志, 藤野毅, “ストリーム暗号 CryptMT のデータ並列処理による高速化手法及び SIMD 型組込みプロセッサによる実装と評価,” 信学論 (D), vol. J96-D, No. 3, pp.495-505, Mar., 2013.
- [11] 望月陽平, 吉田直之, 松本直樹, 村上佑馬, 熊木武志, 藤野毅, “SIMD 型組込みプロセッサによる疑似乱数生成アルゴリズムの並列処理実装とその評価,” 信学論 (D), vol. J95-D, No. 3, pp.376-386, Mar., 2012.
- [12] [http://japan.renesas.com/products/tools/ide/hew/hew\\_mid\\_level\\_landing.jsp](http://japan.renesas.com/products/tools/ide/hew/hew_mid_level_landing.jsp)
- [13] 米田 聡, “新「BeagleBoard」で最強 PC を作る,” 日経 Linux, 第 11 巻, 第 7 号通巻 118 号, pp. 55-68, Jul., 2009.
- [14] <http://www.amd.com/us/products/embedded/processors/geode-lx/Pages/geode-lx-processor-family.aspx>
- [15] <http://ark.intel.com/products/42503/>