

C\_012

# 高速画像認識のための超並列ビジョンプロセッサの設計

## Design of a Massively Parallel Vision Processor for High Speed Image Recognition

山口 光太† 渡辺 義浩† 小室 孝† 石川 正俊†  
 Kota Yamaguchi Yoshihiro Watanabe Takashi Komuro Masatoshi Ishikawa

### 1. まえがき

車載やセキュリティ用途をはじめとして実環境における画像認識の応用が近年広がっている。実環境での画像認識は高度かつ膨大な演算が必要なため、その処理の高速化が課題となっている。フレームレートや認識率の向上のためには高速な処理を実現するプロセッサが必要である。

画像認識を想定したプロセッサには次のような研究例がある。NECで開発されたIMAPアーキテクチャは128個のPE(Processing Element)を一列に並べ、画像の列単位での並列処理を行う[1]。また、東芝は3つのVLIWプロセッサモジュールを搭載した画像認識プロセッサ Viscontiを開発し、データ流および命令レベルでの並列処理を実現している[2]。

これらのプロセッサは高度な処理を実装し、車載用途の画像認識などで一定の成果を上げている。しかし空間、物体、人物の認識など、将来的に実環境でロボットなどに搭載して画像認識を行うために十分な性能とは言えず、更なる処理の高速化が課題であると考えられる。

単純な処理の繰り返しで実現されるような画像処理においては、これまでSIMD (Single Instruction Multiple Data)型アーキテクチャを用いたビジョンプロセッサによって処理の高速化が図られてきた。特に画素単位でPEを置く超並列SIMD型プロセッサにより逐次処理に比べて大幅に処理速度を向上可能であることが知られている。東京大学の石川らの研究グループはイメージセンサの画素毎にPEを置いた超並列ビジョンチップを開発し[3]、高速な各種フィルタ処理、対象の面積や位置の抽出を実装している。これにより従来不可能であった実時間での視覚計測や視覚フィードバック制御などが実現されている。また、Gealowらは画素並列のPEを用いた低コスト高速な画像処理を目指している[4]。

しかし、超並列プロセッサを画像認識に用いた研究例はこれまで多くは見られなかった。画像認識で行われる処理は非常に多岐に渡り、その複雑な演算アルゴリズムを超並列プロセッサに実装するのが困難であったこと、超並列プロセッサの入出力がボトルネックとなってしまう利用しにくいことなどが理由と考えられる。そのため画像認識では汎用の逐次型プロセッサを用いるか、前述の並列度の小さなSIMDモジュールを搭載したプロセッサが用いられることが多かった。

画像認識の高速化という問題に対し、我々は超並列SIMD型アーキテクチャの高い処理能力を様々な画像処理に利用可能とするビジョンプロセッサの設計を行った。このプロセッサは複数の並列度の異なるSIMD型プロセッサが一つのメモリを共有するマルチSIMDアーキテクチャをもつ。

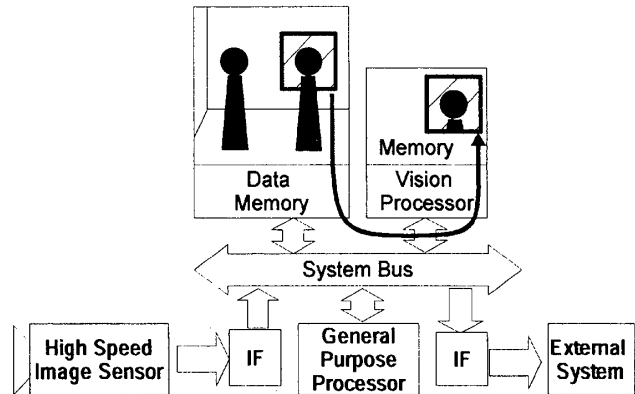


図1 想定する高速画像認識システム

これにより画像認識に必要なとされる多様な演算を汎用プロセッサに比べて効率よく実現可能である。

提案するプロセッサは図1に示すような高速画像認識システムに用いることを想定している。画像認識では必ずしも画像全体に処理を適用するのではなく、局所的に切り出された画像に処理を適用する場合が多い。この局所的な画像に対して超並列処理を行うことで、高速な画像認識が実現されると考えられる。

本論文ではまずこれまでの超並列SIMD型プロセッサにおける問題点を説明し、続いて我々の設計したビジョンプロセッサのアーキテクチャについて述べる。次にシミュレーションによってマルチSIMDアーキテクチャによる演算量の削減量を評価し、最後にメモリ回路の構成や今後の方針などを述べる。

## 2. アーキテクチャの設計

### 2.1. 画像認識に必要な処理

空間、物体、顔など、画像認識には対象に応じて様々な手法が存在する。その多くは以下のような画像処理を組み合わせることで実現される。

- 前処理： 2値化、2Dフィルタリングなど
- 中間処理： スカラー特徴量抽出、図形描画、アフィン変換、直交変換など

これらの処理によって抽出された特徴量を基に識別などの高次処理を行う。実環境での画像認識のためには、その解空間の探索にかかる膨大な演算を効率化する必要があった。

### 2.2. 従来の超並列SIMD型処理の問題

前述の画像処理で多用される演算のうち代表的なものを図2に示す。例えば図形描画は座標値演算と単一画素演算を組み合わせることで実現される。画素並列にPEを配

†東京大学大学院情報理工学系研究科, Graduate School of Information Science and Technology, The University of Tokyo

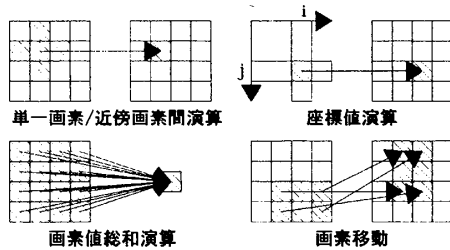


図2 画像認識の処理で多用される演算

置した従来の超並列 SIMD 型プロセッサでは単一画素/近傍画素間演算を高速に実行することができる。しかしその他の演算では様々なデータ流を扱うため超並列 SIMD 型プロセッサでは高速処理が難しかった。高速な画像認識のためにはこれらの演算の効率化することが課題となる。

また、超並列 SIMD 型プロセッサでは入出力方法も問題となる。SIMD 型プロセッサでは通常 PE ごとに局所メモリを持つが、それだけのバンド幅の I/O ポートを用意することは現実的でないため、効率的に入出力を実現する方法が必要となる。

### 3. マルチ SIMD アーキテクチャ

従来の超並列 SIMD 型プロセッサでの問題から、我々は図3に示すマルチ SIMD アーキテクチャを設計した[5]。

このアーキテクチャはメッシュ状に PE を結合した 2 次元 SIMD モジュール、直線状に PE を並べた 1 次元 SIMD モジュール、逐次処理を行う 0 次元 CPU が 1 つのメモリを共有する構造を持つ。0 次元 CPU はデータ処理だけでなくコントローラとしての機能も持たせることを想定している。メモリ共有により各 SIMD モジュールと 0 次元 CPU の間のデータ転送のコストを軽減することができる。

2 次元/1 次元 SIMD モジュール間と 1 次元 SIMD/0 次元 CPU 間にはそれぞれ図4に示す総和演算機構、ブロードキャスト機構が設けられている。総和演算機構は 1 次元データ列の総和を計算するもの、ブロードキャストはスカラー値を 1 次元データ列にコピーする操作である。1 次元 SIMD モジュールには更に任意の PE 間のデータの並べ替え機構を用意する。これらの機構を用意することで従来の超並列 SIMD 型プロセッサでは効率の悪い演算も高速化が可能となる。

なお、各 SIMD モジュールと CPU が扱える演算のビット幅は 2 次元 ≤ 1 次元 ≤ CPU となるように構成する。

### 4. アーキテクチャ評価

提案したアーキテクチャを用いることで、従来の SIMD 型アーキテクチャや逐次処理の場合と比較して削減される演算量の評価を行った。以下、シミュレーション条件、画像処理アルゴリズム、評価結果について述べる。

#### 4.1. 演算回数見積もり

逐次処理 (0D)、列並列 SIMD (1D)、画素並列 SIMD (2D)、マルチ SIMD (M-SIMD) の 4 種類のアーキテクチャで 6 種類の画像処理を実行するのにかかる演算回数を見積もった。

2D の PE は 1 ビット、1D の PE と 0D のプロセッサは任意のビット幅の整数を 1 クロックで処理できるものとした。

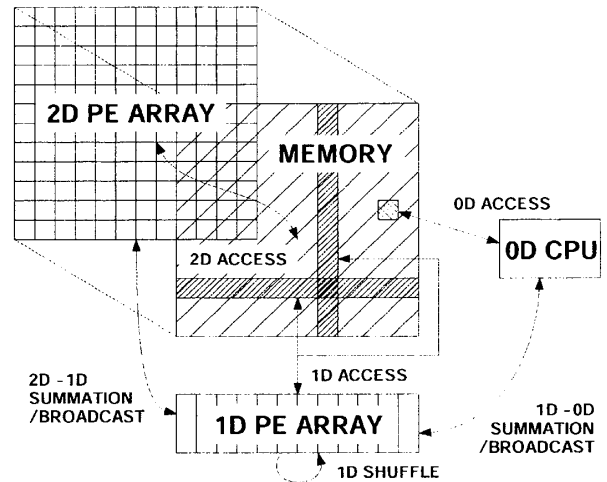


図3 マルチ SIMD アーキテクチャ

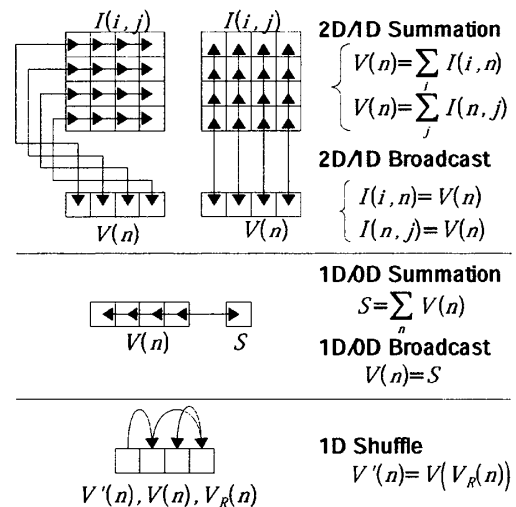


図4 総和演算・ブロードキャスト・一次元並べ替え

M-SIMD が扱えるデータ幅はこれらを組み合わせたものとした。M-SIMD の総和演算、ブロードキャスト、一次元並べ替えはそれぞれ 1 クロックで処理できるものとした。

PE アレイのサイズは対象画像から処理のために切り出された局所画像のサイズと同じものである。今回はそのサイズ  $64 \times 64$  と  $256 \times 256$  の 2 種類について演算量見積もりを行った。

#### 4.2. 評価に用いたアルゴリズム

##### •2 値化

2 値化は多値画像中の対象物抽出などに用いられる。画像  $I(i, j)$  に対する 2 値化画像  $I'(i, j)$  は次式で表される。

$$I'(i, j) = \begin{cases} 0 & (I(i, j) < A) \\ 1 & (I(i, j) \geq A) \end{cases}$$

$A$  は閾値である。各プロセッサは以下のように演算を行った。

0D: 各点で逐次的に演算した。

1D: 行単位で並列に演算した。

2D: 画素並列に演算した。

M-SIMD: 2D と同様に画素並列に演算した。

●エッジ検出

エッジ検出により対象物体の輪郭を抽出できる。画像  $I(i, j)$  に対するエッジ検出画像  $I'(i, j)$  は次式で表される。

$$I'(i, j) = |I(i, j) - (I(i-1, j) + I(i+1, j) + I(i, j-1) + I(i, j+1)) / 4|$$

各プロセッサは次のように演算を実行した。

- 0D: 各点で逐次的に演算した。
- 1D: PE 間の通信を用いて行単位で並列に演算した。
- 2D: PE 間の通信を用いて画素並列に演算した。
- M-SIMD: 2D 同様に画素並列に演算した。

●円の描画

画像中から特定領域を切り出したり、図形マッチングにより形状の識別を行うために図形描画が必要となる。中心  $(i_0, j_0)$ 、半径  $r$  の円の領域は次式で表される。

$$\{(i, j) | C(i, j) = (i - i_0)^2 + (j - j_0)^2 - r^2 < 0\}$$

したがって閾値として  $C(i, j)$  を選んで2値化を行うことで円が描画される。 $C(i, j)$  の計算は以下のように行った。なお各々の PE は予め座標値を持っているものとした。

- 0D: 各点で逐次的に  $C(i, j)$  を求めた。
- 1D: 行毎に並列に計算して  $C(i, j)$  を求めた。
- 2D: 画素並列に  $C(i, j)$  を求めた。
- M-SIMD:  $(i - i_0)^2 - r^2$  と  $(j - j_0)^2$  を1次元 SIMD モジュールで計算しこれを列/行ブロードキャストによって2次元 SIMD モジュールへ送信して  $C(i, j)$  を求めた。

●モーメント特徴量抽出

モーメント特徴量は物体の重心や形状、姿勢などを推定する上で重要なスカラー量である。バイナリ画像  $I(i, j)$  の  $p+q$  次モーメント特徴量は次式で定義される。

$$m_{pq} = \sum_i \sum_j i^p j^q I(i, j)$$

ここでは2次モーメント量  $m_{11}$  の計算を行った。

- 0D: 逐次的に各点の値を足し合わせて  $m_{11}$  を計算した。
- 1D: 行並列に  $jI(i, j)$  の加算を繰り返し  $\sum_j jI(i, j)$  を求め、行並列に  $i$  を乗じてから行方向の近傍通信を用いて加算を行って  $m_{11}$  を求めた。
- 2D:  $jI(i, j)$  を計算してから近傍通信を用いて  $\sum_j jI(i, j)$  を求め、以降 1D と同様の手順で  $m_{11}$  を求めた。
- M-SIMD: 画素並列に  $jI(i, j)$  を計算して列方向の総和演算により  $\sum_j jI(i, j)$  を求め、これに  $i$  を乗じて1次元 SIMD モジュールの総和演算により  $m_{11}$  を求めた。

表1 演算量のオーダー

	0D	1D	2D	M-SIMD
2値化	$O(N^2)$	$O(N)$	$O(a)$	$O(a)$
エッジ検出	$O(N^2)$	$O(N)$	$O(a)$	$O(a)$
円の描画	$O(N^2)$	$O(N)$	$O(a^2)$	$O(a)$
モーメント特徴量抽出	$O(N^2)$	$O(N)$	$O(aN)$	$O(a)$
バイナリ画像の回転	$O(N^2)$	$O(N^2)$	$O(aM^2)$	$O(aN)$ or $O(M^2)$
1次元FFT	$O(N^2)$	$O(N)$	$O(N^2)$	$O(N)$

●バイナリ画像の回転

画像の回転はアフィン変換の一部であり、物体や空間の認識で重要な処理である。画像  $I(i, j)$  の回転  $I'(i, j)$  は次式のように表される。

$$I'(i, j) = I(i_{ref}(i, j), j_{ref}(i, j))$$

$$i_{ref}(i, j) = (i - i_0) \cos \theta + (j - j_0) \sin \theta + i_0$$

$$j_{ref}(i, j) = -(i - i_0) \sin \theta + (j - j_0) \cos \theta + j_0$$

$(i_0, j_0)$  は回転の中心座標、 $\theta$  は回転角である。座標値  $(i_{ref}(i, j), j_{ref}(i, j))$  は円の描画と同様の手順で計算した。画素値の置換は次のように行った。

- 0D: 各点で逐次的に画素値の置換を実行した。
- 1D: 並列処理が困難なため外部に画像を転送して逐次処理により画素値の置換を行うものとした。
- 2D: 近傍通信を用いて次式のように処理を実行した。

$$I'(i, j) = \begin{cases} I(i \pm p, j \pm q) & (i_{ref}(i, j) = i \pm p, j_{ref}(i, j) = j \pm q) \\ I(i, j) & (i_{ref}(i, j) \neq i \pm p, j_{ref}(i, j) \neq j \pm q) \end{cases}$$

$(0 \leq p \leq M, 0 \leq q \leq M)$

$M$  は近傍通信を行う範囲を表す値で、画像サイズ  $N \times N$  や回転角  $\theta$ 、回転の中心座標  $(i_0, j_0)$  に依存する。全ての  $(p, q)$  の組み合わせについて処理を行った。ただし  $M$  が大きい場合は 1D 同様に処理した。

M-SIMD: 次式のように行単位で並列に処理を実行した。

$$I'(i, l) = \begin{cases} I(i, l) & (l \neq j_{ref}(i, l)) \\ V(i) & (l = j_{ref}(i, l)) \end{cases}, \quad V(i) = V(V_{ref}(i))$$

$$V(i) = I(i, l), \quad V_{ref}(i) = \sum_j m(i, j) i_{ref}(i, j)$$

$m(i, j)$  は  $j_{ref}(i, j)$  と処理する行座標値  $l$  を用いて作られるマスクである。 $0 \leq l \leq N$  のすべての行について処理を繰り返すことで回転を行った。

●1次元FFT

周波数領域で特徴量を抽出するためにはフーリエ変換を行う必要がある。画像  $I(i, j)$  の列方向1次元離散フーリエ変換  $I(i, v)$  は次式で表される。

$$I(i, v) = \sum_{l=0}^{N-1} I(i, j) W_N^{lv}$$

ただし  $W_N = e^{-2\pi j / N}$  である。これを高速に演算するアルゴリズムが1次元FFTであり、以下のように演算を実行した。

- 0D: 各点で逐次的に演算した。
- 1D: 行単位で並列に演算した。
- 2D: 行並列での多ビット演算を多用する処理の実装が困難なため、外部に画像を転送して逐次的に演算を行えるものとした。

M-SIMD: 1D 同様に並列に演算した。

以上6種類の画像処理の、サイズ  $N \times N$  の画像に対する演算量のオーダーを表1に示す。 $a$  は演算値のビット幅を表す。

4.3. 評価結果

図5にサイズ  $256 \times 256$  の画像についての各プロセッサの演算量のシミュレーション結果を示す。グラフは0Dの

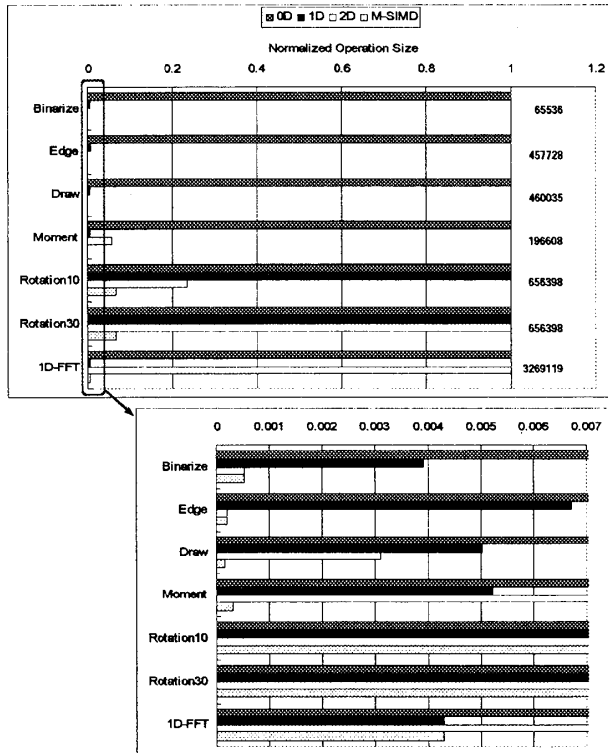


図5 各プロセッサでの画像処理演算量の見積もり

演算量を1に正規化してある。

M-SIMDは0Dに比べ演算量をおよそ1/10から1/3000程度に削減可能であることが確認された。特に1Dや2D単独で達成可能な演算量よりも、1次元と2次元のSIMDモジュールを組み合わせたM-SIMDの演算量が小さいことが確認された。これらのことから複雑な画像処理を実行するために提案アーキテクチャが有効であると言える。

なお、サイズ64×64での見積もりについてもほぼ同様な傾向が見られた。

### 5. 共有メモリ回路

メモリ共有型マルチSIMDアーキテクチャでは0次元/1次元/2次元にデータアクセス可能なメモリが必要となる。一般的なRAMでは利用可能なアクセスポート数が限られるため、図6に示す専用の共有メモリ回路の設計を行った。

図中の $n_0$ ,  $n_1$ ,  $n_2$ はそれぞれ0次元CPU、1次元/2次元PEが扱うビット幅を、 $n_b$ はメモリバンクの持つビット線の数を表す。 $N$ は画像の幅、 $N_w$ はワード線の数である。また台形で囲んだスイッチは3状態セレクタを表し、 $N$ 本のスイッチ入力のうち1つだけがonとなるか、全てがon/offとなるように制御信号を入力して使う。例えば1次元SIMDモジュールがアクセスする場合、ワード線と0D&1Dcol制御線、1Dcol制御線で各々1本だけonにし、他の制御線を全てoffにする。また0次元モジュールからの書き込み時に0Dcol制御線を全てonにすることで全列に同一データをブロードキャストすることが可能となる。

### 6. むすび

画像認識に必要なとされる各種画像処理を高速に実行可能なマルチSIMDアーキテクチャを提案し、シミュレーション

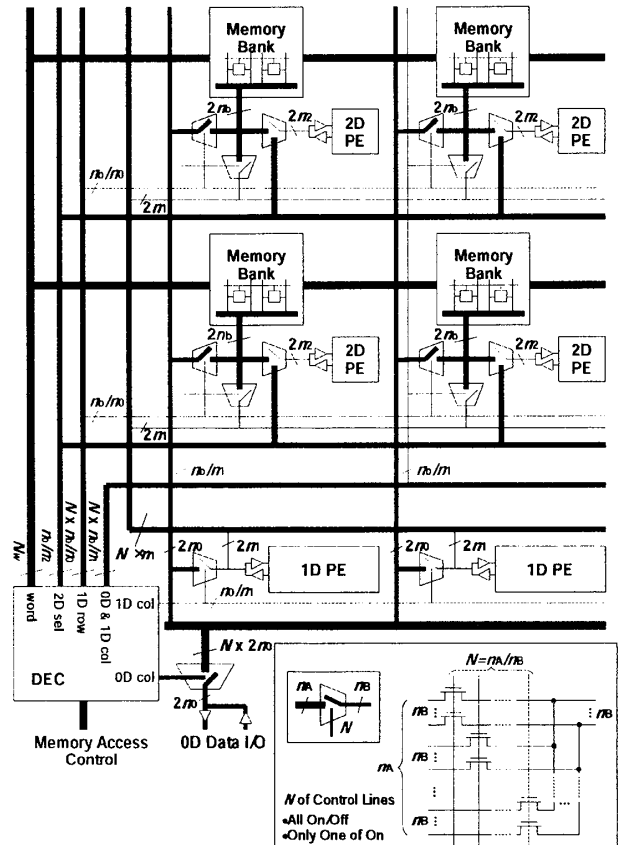


図6 共有メモリ回路

ョンによりその演算量の削減効果を示した。また、本アーキテクチャの実装に必要な共有メモリ回路の構成を示した。今後より詳細な仕様を決定し、画像認識システムのプロトタイプ試作を行う予定である。

### 参考文献

- 1) S. Kyo, S. Okazaki and T. Arai: "An integrated memory array processor architecture for embedded image recognition systems", Proceedings of the 32nd International Symposium on Computer Architecture, pp. 134-145 (2005).
- 2) J. Tanabe, Y. Taniguchi, T. Miyamori, Y. Miyamoto, H. Takeda, M. Tarui, H. Nakayama, N. Takeda, K. Maeda and M. Matsui: "Visconti: Multi-VLIW image recognition processor based on configurable processor", Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, pp. 185-188 (2003).
- 3) 石川正俊, 小室孝: "デジタルビジョンチップとその応用", 電子情報通信学会論文誌C, J84-C, 6, pp.451-461 (2001).
- 4) J. Gealow and C. Sodini: "A pixel-parallel image processor using logic pitch-matched to dynamic memory", IEEE Journal of Solid-State Circuits, 34, 6, pp.831-839 (1999).
- 5) 山口光太, 渡辺義浩, 小室孝, 石川正俊: "メモリ共有型マルチSIMD型アーキテクチャを有する高性能ビジョンプロセッサの設計", 電子情報通信学会技術報告, 106, 92, (ICD2006-56), pp.89-94 (2006)