

修士論文

題目

SIFT特徴量抽出に対応する  
並列プロセッサ構成法の研究

指導教員

近藤利夫 教授

平成23年度

三重大学大学院 工学研究科 情報工学専攻  
計算機アーキテクチャ研究室

韓 楊 (410M5B1)

## 内容梗概

近年、SIFT(Scale Invariant Feature Transform) 特徴量を利用する画像認識手法が注目されている。しかし、専用ハードウェアにより、実時間の SIFT 処理を実現する手法では SIFT 以外の処理に全く対応出来ない等、汎用性と柔軟性に欠ける問題が存在する。これに対し、本論文では、SIFT 以外の画像処理にも適用できるだけの汎用性を備えた SIMD 型の並列プロセッサ構成を検討した。また、SIFT 特徴量抽出の主要な処理を全部 SIMD 並列処理で実行することを可能とした。さらに、MIMD 型並列プロセッサに劣らぬ性能を得るために処理ネックとなる部分に、新たな高機能命令セットを定め、性能をはかった。具体的には、Gaussian フィルタ演算処理で、フィルタ処理を 3 倍高速化した。極値検出で、注目画素の近傍の比較演算を 3 倍高速化した。オリエンテーション算出（ヒストグラム生成を含む）で、ハードウェア構成の追加によって、8 並列で実行により 7 倍高速化した。また、性能モデルを用いて汎用性を備えた SIMD 型の並列プロセッサ構成の性能評価を行った。

# Abstract

Currently, with the development of image-recognition technique, to recognize a large number of objects everyday, has been developed by various research methods. However, such an environment to move against the actual camera itself, there is a problem that would have greatly diminished accuracy. It is difficult to recognize the impact of external environmental changes and the presence or absence of light and other lighting, even in the background of major changes in video camera. Therefore the recognition technique with quantity of SIFT characteristic is nominated for a masterpiece. However, dedicated hardware implementations of the high processing speed is obtained, there is a little exhibity issue. Furthermore, the parallel computation system which is real-time processing correspondence and exhibity is necessary. In this paper, I proposed a method to construct robust to withstand poor conditions, based on dedicated hardware configuration of the existing law. And I performed the basic design of highly parallel processors for general purpose image recognition that can be implemented in other process.

# 目次

<b>1</b>	<b>まえがき</b>	<b>1</b>
<b>2</b>	<b>SIFT 特徴量とその抽出処理法</b>	<b>2</b>
2.1	SIFT 特徴量の概要 . . . . .	2
2.2	SIFT 特徴量抽出法 . . . . .	2
2.2.1	Difference-of-Gaussian 処理 . . . . .	5
2.2.2	極値検出 . . . . .	5
2.2.3	ローカライズ . . . . .	5
2.2.4	オリエンテーション算出 . . . . .	6
2.2.5	ヒストグラム作成 . . . . .	7
<b>3</b>	<b>SIFT 特徴量抽出への SIMD 型並列処理の適用性</b>	<b>8</b>
<b>4</b>	<b>SIFT 特徴量抽出対応の SIMD プロセッサ構成法の提案</b>	<b>10</b>
4.1	ライン/ブロック処理単位切り替え構成 . . . . .	10
4.2	Systolic Algorithm による SIFT 処理の前処理部の実現方法	11
4.3	オリエンテーション算出用行単位メモリ . . . . .	13
4.4	ヒストグラム算出機構 . . . . .	13
<b>5</b>	<b>ハードウェア構成の設計</b>	<b>14</b>
5.1	全体構成 . . . . .	14
5.2	PE の構成 . . . . .	15
5.3	制御部の構成 . . . . .	15
<b>6</b>	<b>SIMD プロセッサ部の命令セットの設計</b>	<b>16</b>
6.1	Difference-of-Gaussian 処理に対応する命令 . . . . .	16
6.2	極値検出に対応する命令 . . . . .	16
6.3	オリエンテーション算出に対応する命令生成 . . . . .	17
<b>7</b>	<b>性能評価方法</b>	<b>17</b>
7.1	Gaussian フィルタに対応する命令 . . . . .	18
7.2	極値検出における命令セット . . . . .	19
7.3	オリエンテーション算出における命令セット . . . . .	19
7.4	ヒストグラムにおける命令セット . . . . .	20
7.5	評価結果 . . . . .	21

<b>8</b>	<b>関連研究</b>	<b>23</b>
8.1	SIFT 特徴量を用いた画像認識 . . . . .	23
8.1.1	SIFT 特徴量 . . . . .	23
8.1.2	特徴量マッチング . . . . .	24
8.2	Look up table . . . . .	24
8.3	PE Array . . . . .	25
<b>9</b>	<b>あとがき</b>	<b>26</b>
	謝辞	27
	参考文献	28

## 目次

2.1	SIFT 特徴量を用いた認識手法の対象物 . . . . .	3
2.2	SIFT 特徴量を用いた認識手法の入力画像 . . . . .	3
2.3	対象物のオリエンテーション . . . . .	4
2.4	SIFT 特徴量を用いた認識手法の生成例 . . . . .	4
2.5	DoG 処理 . . . . .	5
2.6	極値検出 . . . . .	6
2.7	ヒストグラム作成 . . . . .	7
3.8	SIMD 型並列処理の適用性 . . . . .	9
4.9	TWIST TORUS 接続 . . . . .	10
4.10	ライン単位とブロックのメモリアクセス例 . . . . .	12
4.11	ライン単位アクセスによるアドレスの生成例 . . . . .	12
5.12	全体構成 . . . . .	14
5.13	PE の構成 . . . . .	15
7.14	Gaussian フィルタのサイクル数の生成例 . . . . .	18
7.15	極値検出の所有のサイクル数 . . . . .	19
7.16	オリエンテーションの算出のステップ数 . . . . .	21
7.17	所用ステップ数の評価 . . . . .	22

# 表 目 次

# 1 まえがき

近年画像認識技術はその着実な進歩により、ITS 分野での自動車システムへの応用が始まっている。特に自動車分野においては安全運転支援を主な目的として車外、車内に複数のカメラが搭載されるようになってきている。しかし、自動車システムの応用には、照明の有無や外光の影響による外部環境変化により、カメラの背景映像が大きく変わるような条件にも耐える頑健性が必要になる。

従来、道路標識認識には、テンプレートマッチングを利用する手法が研究されてきた。しかし、車載カメラで撮影された画像中には様々な大きさで出現するだけでなく、傾き、オクルージョン(樹木などで隠れる事)、照明変化等を伴っている場合も多く。このため、拡大、縮小、回転を行い変形させたテンプレート画像を多数用意する必要があり、膨大なマッチングの処理コストが必要となる欠点がある。また、その他にも色の違いを識別する手法が数多く研究されてきたものの、太陽光の照明変動により様々な色に変化するため正確に認識するのは非常に困難である。

このような状況の中、一般物体認識の有効なアプローチとして、輝度の勾配情報から算出される SIFT(Scale Invariant Feature Transform) を用いた局所特徴量を利用する認識手法が注目されている。この SIFT 特徴量を用いた認識手法は、回転・スケール変化不変・照明変化への高い耐性を有している。しかし、SIFT 特徴量を計算するためには膨大な時間を必要とする。文献調査によって、Intel Core2 Duo E8400 で高解像度の画像(1024x768p) に対して実行時間は 7.28s かかることも分かった。また、専用の構成のハードウェアが実現されているものの、ほかの処理に全く対応出来ない等、汎用性と柔軟性に欠ける問題が存在する。これに対し、性能の向上が著しいものの SIFT をリアルタイム処理するには非力である。このため、汎用性との両立にはプロセッサの並列化が必須になる。

そこで、本研究では既存の専用ハードウェア構成法をベースに SIFT 手法以外の画像処理にも適用できるだけの汎用性を備えた並列プロセッサの構成を明らかにすることを目的にする。

具体的には、既存の SIFT 処理専用ハードウェア構成法をベースにして他の種々の画像処理にも対応可能な、汎用性の高い SIMD 型並列プロセッサの構成を明らかにすることを目的にする。

## 2 SIFT 特徴量とその抽出処理法

### 2.1 SIFT 特徴量の概要

SIFT 特徴量を用いた画像認識とは Lowe らが提案した輝度勾配方向の情報を用いてオリエンテーションを求め、正規化を行う事で得られる回転、拡大縮小等に不変な特徴量記述する二段階からなる。検出段階では、スケールスペースの極値検出 (Scale-space peak description) 及びキーポイントのローカライズ (keypoint localization) 処理は行われ、記述段階では、オリエンテーション割り当て (Orientation assignment) と SIFT 記述子によりキーポイントの特徴量抽出 (keypoint descriptor) が行われる。SIFT 特徴量を用いた画像認識処理の例を図 2.1～図 2.4 に示す。入力の机上画像 (図 2.2) から特徴量算出 (キーポイント、オリエンテーション、勾配情報など) を行って、対象物の書籍とマッチングを行った、最後に、結果の画像 (図 2.4) が出力されている。また、キーポイントのオリエンテーションの生成例は図 2.3 に示す。具体的な SIFT を用いた認識手法の画像出力方法については 7 章の関連研究で参照できる。

### 2.2 SIFT 特徴量抽出法

SIFT 特徴量抽出は、特徴量 (キーポイントと呼ぶ) の検出 (detection) と特徴量の記述 (description) のからなり、主な処理は以下に示す。

- detection
  1. Difference-of-Gaussian 処理
  2. 極値検出
  3. キーポイントのローカライズ
- description
  1. オリエンテーション算出
  2. ヒストグラム作成

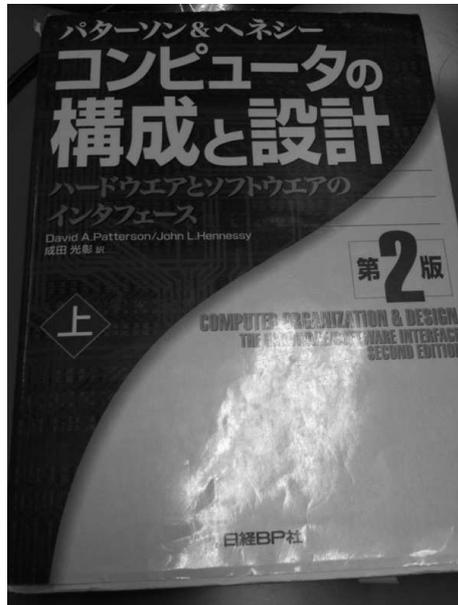


図 2.1: SIFT 特徴量を用いた認識手法の対象物

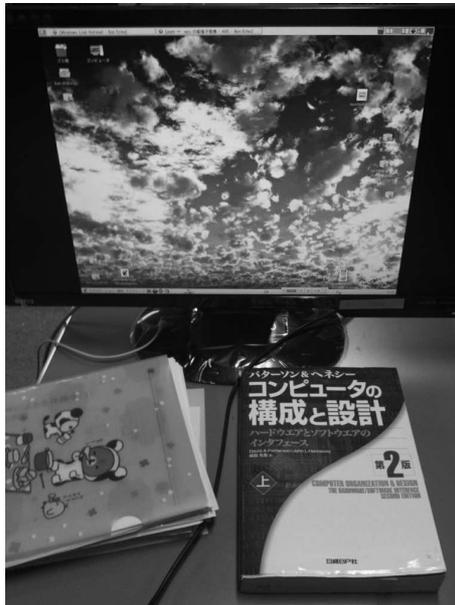


図 2.2: SIFT 特徴量を用いた認識手法の入力画像

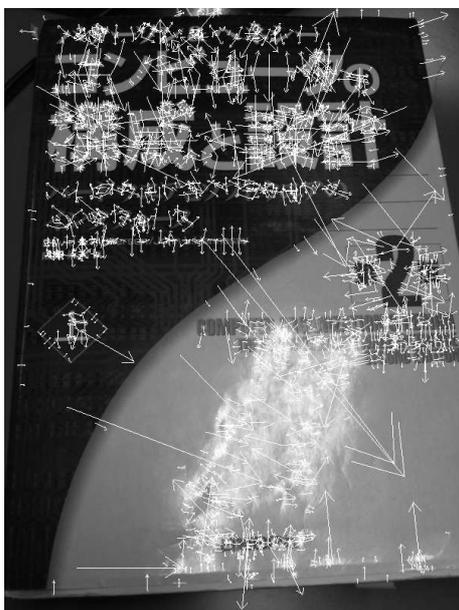


図 2.3: 対象物のオリエンテーション

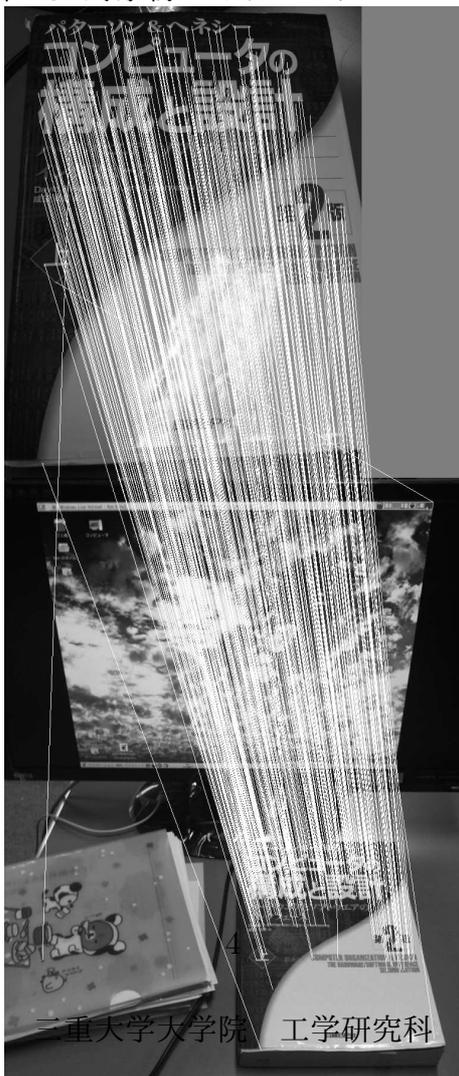


図 2.4: SIFT 特徴量を用いた認識手法の生成例

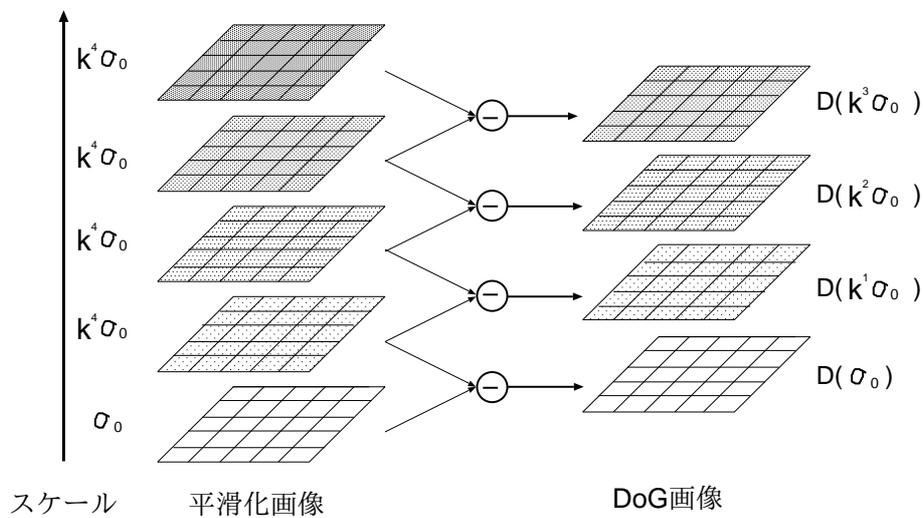


図 2.5: DoG 処理

### 2.2.1 Difference-of-Gaussian 処理

Difference-of-Gaussian 処理は、主に入力画像と異なるスケールのがうす関数を畳み込んだ平滑化画像の差分（DOG 画像）から求める処理となる、その処理の流れを図 2.5 に示す。

### 2.2.2 極値検出

極値検出は、DOG 画像を三枚セットで行う。注目画素とその近傍の 26 画素を比較演算をする。極値である場合はキーポイントとなる、その処理は図 2.6 に示す。

### 2.2.3 ローカライズ

キーポイントのローカライズは、検出されたキーポイントの中には、エッジ上の点がノイズの影響を受けやすい。そこで、主に微分エッジ検出（2 次のエッジ検出）処理を行うことで、エッジ位置を求めることができるので、キーポイントを絞り込むことができる。また、二次エッジ検出は、2 次導関数のゼロ交差する検出手法に基づくヘッセ行列の微分演算により実現される。

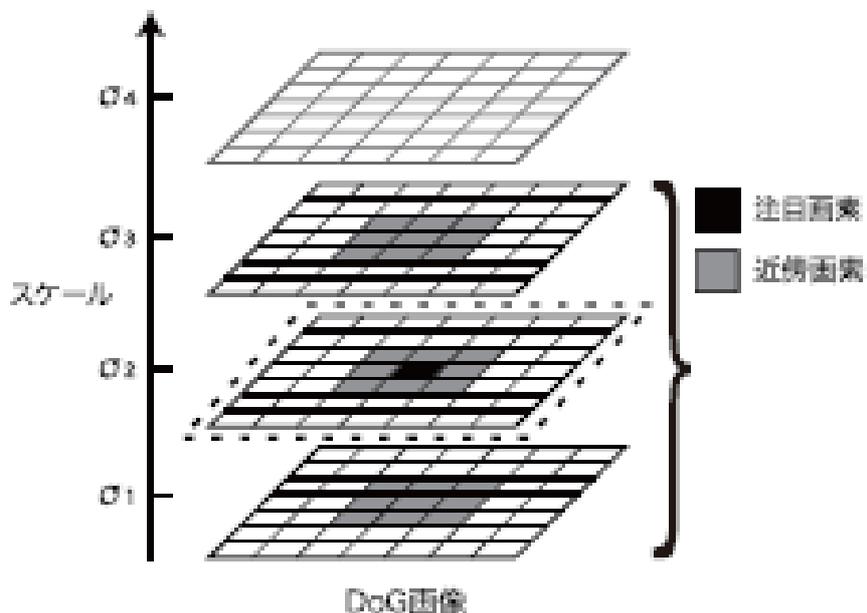


図 2.6: 極値検出

#### 2.2.4 オリエンテーション算出

オリエンテーション算出は、検出された各キーポイントに対して、輝度勾配方向を割り当てることである。キーポイントが検出された平滑画像  $L(x, y)$  の各画素の勾配強度  $m(x, y)$  とその勾配方向  $\theta(x, y)$  は、以下の式より計算できる。

$$m(u, v) = \sqrt{f_u(u, v)^2 + f_v(u, v)^2} \quad (1)$$

$$\theta(u, v) = \tan^{-1} \frac{f_v(u, v)}{f_u(u, v)} \quad (2)$$

$$\begin{cases} f_u(u, v) = L(u + 1, v) - L(u - 1, v) \\ f_v(u, v) = L(u + 1, v) - L(u, v - 1) \end{cases} \quad (3)$$

オリエンテーションはキーポイントにおける方向を表し、特徴量記述の際にオリエンテーションにより向き正規化を行うことで、回転に不変となる。

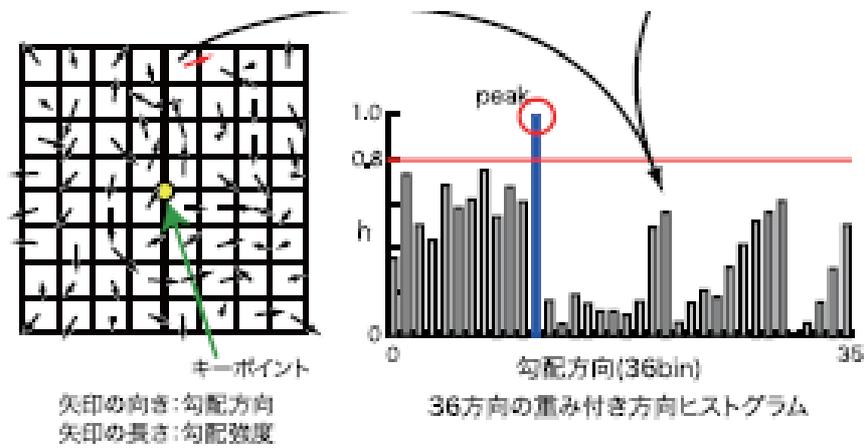


図 2.7: ヒストグラム作成

### 2.2.5 ヒストグラム作成

オリエンテーション算出で求められた勾配強度  $m$  と勾配方向  $\theta$  から、図のような 36 方向に分割した重み付け方向ヒストグラム  $h$  を以下の式より作成する、そのイメージ図は図 2.7 で示す。

$$h'_\theta = \sum_x \sum_y w(x, y) * \delta[\theta', \theta(x, y)] \quad (4)$$

$$w(u, v) = G(x, y, \sigma) * m(x, y) \quad (5)$$

ここで、 $h_\theta$  は、全 36 方向に量子化したヒストグラムである。

### 3 SIFT 特徴量抽出への SIMD 型並列処理の適用性

SIFT 特徴抽出の主要な処理である DoG 処理、極値検出、ローカライズ、オリエンテーション算出とヒストグラム作成の SIMD 型並列処理への適用性を 1024x768 画素の評価画像で調べた結果を図 3.8 に示す。ここで、処理時間の割合とは gcov コードの実行結果のプロファイリングに基づく。

SIFT の前処理は、全画素に対し逐次処理を行い、全体処理時間の 3 割ほどを占めているが分かった。アクセス速度を向上するため、ライン単位で並列化することを提案した。後処理は全処理の六割以上を占めている、ブロック単位の処理であるため、前処理のライン単位から後処理のブロック単位へ切り替える構成が必要とする。ここで、ライン単位とは行方向の 64 画素、ブロック単位とは 8x8 画素を示す。途中で処理単位が変わるので SIMD プロセッサによる全処理の対応には、ライン単位からブロック単位へ切り替えを可能とする SIMD 型の構成が必要となる。

ここで、均一とは、各画素と近傍画素との間で同じ処理を行うことを示す。均一性の高い処理は、SIMD 型並列処理に適する。これに対し、ヒストグラム作成の勾配強度を積み上げる処理は不均一であり、近傍の画素データ参照だけでは実行できないため、ハードウェア構成の工夫を必要とする。表に示していない処理は、不均一の処理も存在するが、プロファイリングによる時間割合が 0 % に近い結果になったので、別個に設ける汎用プロセッサにより実現する。

System block	Function	Time (%)	Unit	Uniformity
DoG	Difference of Gaussian	8.12%	Line	Uniformity
Detection extremums	Comparing of 3x3x3 pixel neighborhood	5.42%	Line	Uniformity
Localize of features	Differential edge detection (Computes 3D hessian matrix)	0.72%	Line	Uniformity
Orientation	Calculate Orientation and Gradient magnitude	46.21%	Block	Uniformity
Create a Histogram	A orientation histogram of 36 bin	21.29%	Block	Non-uniformity

図 3.8: SIMD 型並列処理の適用性

上の表の Function（機能）の概要を以下に示す。

1. Difference-of-Gaussian 処理は、Gaussian フィルタ平滑化画像の差分
2. 極値検出は、近傍の 3 D 空間の比較演算
3. キーポイントのローカライズは、3 D のヘッセ行列演算
4. オリエンテーション算出は、勾配方向の三角演算と勾配強度のフィルタ演算
5. ヒストグラム作成は、36 方向の画素強度の割り当て処理

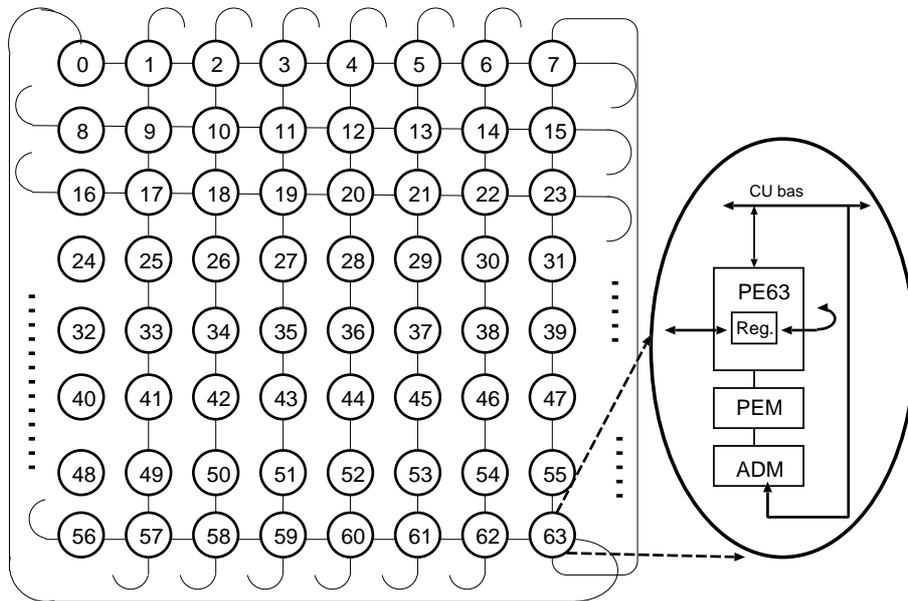


図 4.9: TWIST TORUS 接続

## 4 SIFT 特徴量抽出対応の SIMD プロセッサ構成法の提案

本論文では、最適オリエンテーション算出と異なるスケールに対する DoG 処理を効率的に実行可能なハードウェアの構成を検討した。また、そのハードウェア構成で SIFT の全体の処理方法を明らかにした。

### 4.1 ライン/ブロック処理単位切り替え構成

ILLIAC IV は、二次元配列形の代表的な SIMD プロセッサである。命令を解釈する 1 つのコントロールユニット (CU) のもとに 64 個の演算装置 (PE) があり、CU の制御によって全 PE が同一演算を同期して行う構成を採る。PE 間が図 4.10 に示すように接続されているため、64 画素のライン単位と 8X8 ブロック単位のアクセスを両立できる利点がある。また、ADM は個別にアドレスを修飾するためのユニットである。

図 4.11 に示すように、PE アレイは一次元アレイとも見なせ、64 画素単位で折り畳むラインの集合としてメモリ PEM に格納される二次元の画像を処理する。その処理対象の二次元画像は一ライン毎に 8 画素分だけ右方向にずらして、格納している。これにより、全 PE 同一のアドレス

を指定することで、DoG をシストリックアルゴリズムで高速処理するのに必要となる、8 画素ずつずれて行く 64 画素長のラインがアクセスされる。その結果、次節で述べるようにガウシアンフィルタ演算を高効率で実行全画素可能になる。

一方、オリエンテーション算出に必要な任意の 8X8 ブロック単位のアクセスは、アドレスをインクリメントしながら、8 画素分ずつアクセスすることで、キーポイントがランダムに分配しても、任意位置の 8X8 ブロックサイズの画素データのアクセスを可能としている。ライン単位の格納形式の画像データから、ブロック単位のアクセスを可能とするメモリモジュールごとのアドレスの与え方を図 4.10 に示している。この図に示されるように 8x8 のメモリモジュールに格納される任意位置のブロックデータは、8 画素幅のブロック構成ライン毎にインクリメントするアドレスを与えるので、アドレスのを衝突無しでアクセスできる。

## 4.2 Systolic Algorithm による SIFT 処理の前処理部の実現方法

### 1. ガウシアンフィルタ演算処理

2 次元畳み込みのフィルタ演算 (積和演算) より実現される。フィルタカーネルの係数と読み込まれた画素の積和を取りながら、その結果を PE 間で列方向にシフト転送することで、自動的な累算処理を可能にする。二次元の PE によって、同時に 64 画素分ずつ実行されて、すべての画素データに対して、高並列処理が可能になる。

### 2. 極値検出

SystolicAlgorithm を用い 6 4 個の PE ごとに注目画素と近傍の 2 6 画素と比較演算をしながら、隣の PE にシフト転送する仕組みより実現される。

### 3. ローカライズ

DoG スケールスペース・ピラミッドの全ピクセルのために、3D ヘッセ行列の計算を行うことにより実現される。画素に対する斜めの引き算の 4 方向のを行うには、同様に SystolicAlgorithm を用い 6 4 個の PE ごとに引き算をしながら、隣の PE にルーティングするような仕組みで実現する。

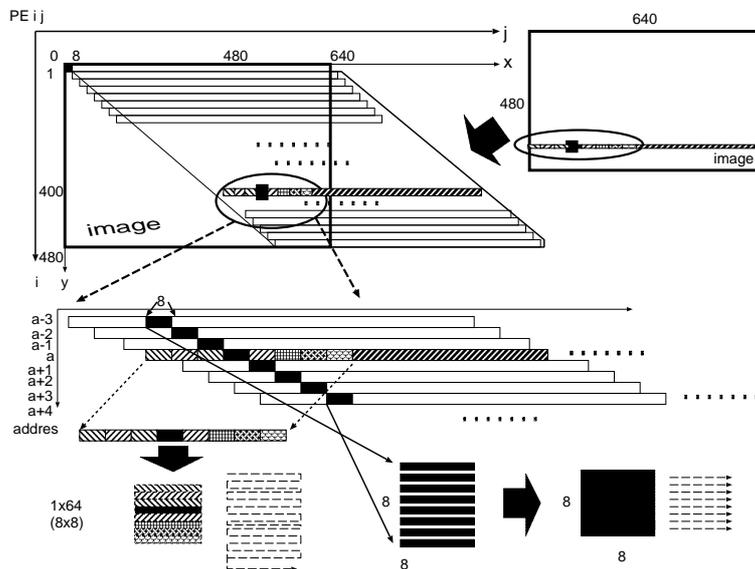


図 4.10: ライン単位とブロックのメモリアクセス例

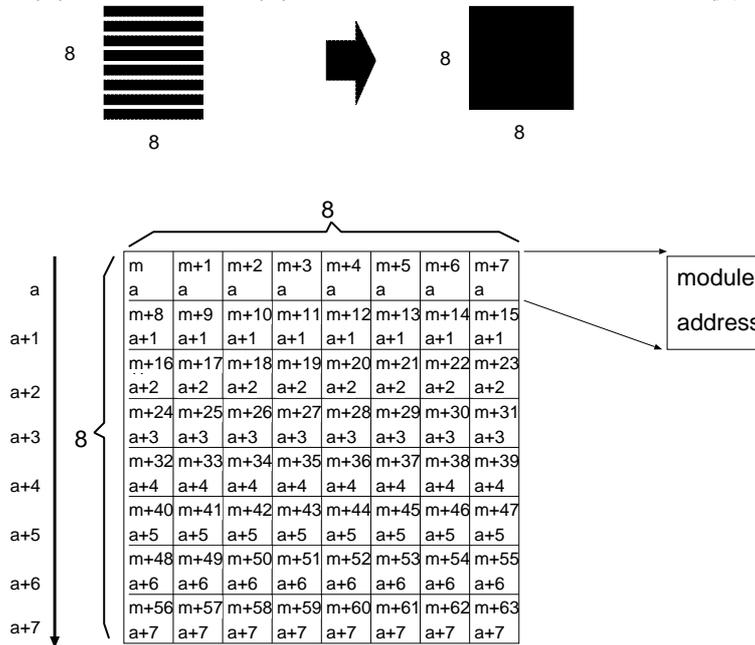


図 4.11: ライン単位アクセスによるアドレスの生成例

### 4.3 オリエンテーション算出用行単位メモリ

オリエンテーション算出では、勾配情報を利用して、勾配強度と方向を算出し、ベクトル化した結果を36方向ごとのヒストグラムを作成する必要がある。具体的な実現方法は、全体のPEアレイにあるLUT（共有メモリ）から外部コントローラによって、方向 $\theta$ の先頭アドレスを一齐に読み出し、レジスタに格納する。さらに、ALUで方向の勾配の $u$ と $v$ が連結して生成する、二つのアドレス(8bit)を加算することで、実際の $\theta$ 方向のアドレスを求められる。生成したアドレスを順番に八個のPEに経由して共有メモリに与える。LUT（共有メモリ）中で取得されたベクトルの値、PEのレジスタに格納する。

### 4.4 ヒストグラム算出機構

オリエンテーションの算出結果から、36の各方向の出現頻度を求める。具体的な実現方法では、各PEで、LUT出力の方向値と1から36（10°から360°）とを $\theta$ と比較し一致したPEの保有する勾配の総合を取り、ヒストグラムのメモリに積み上げられる。ここで、PE間にまたがる勾配の総和はAdder Treeを利用して、行単位の並列で加算し、さらにその結果を列単位で並列に加算して、求める。

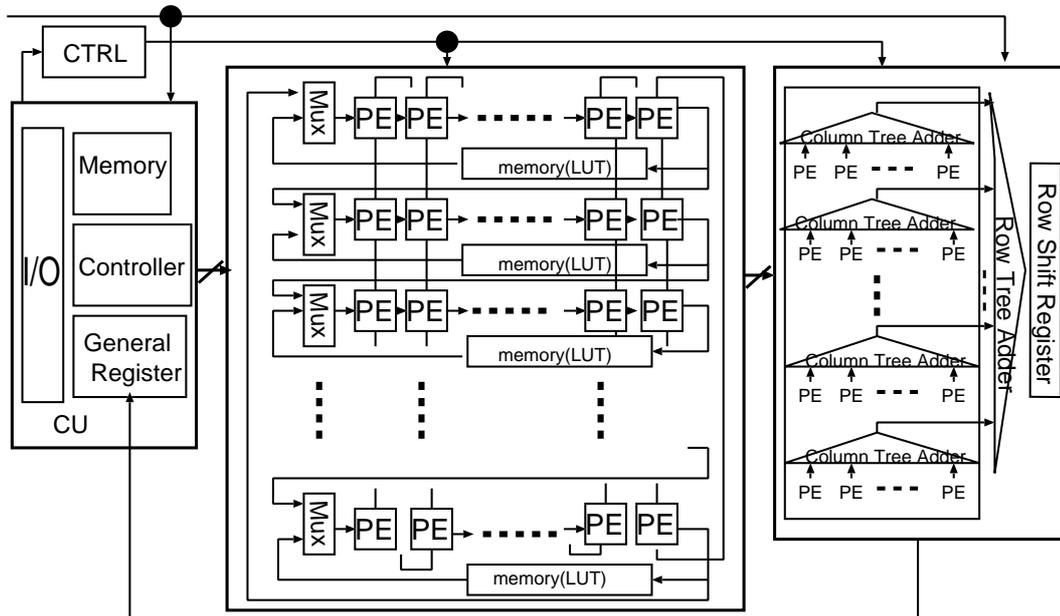


図 5.12: 全体構成

## 5 ハードウェア構成の設計

この章では全体のブロック構成を図 5.11 に示す。SIMD プロセッサ全体の制御プロセッサでは PE アレイを使用し、次元の異なる制御構成を実現した汎用性を備えた SIMD 型並列処理アーキテクチャである。

### 5.1 全体構成

全体構成は、1つのコントロールユニット (CU) のもとに 64 個の演算装置 (PE) が同一演算を同期して行う典型的な SIMD 構成である。行ごとの PE 配列には Adder Tree を付加している。PE 間の接続は ILLIAC IV の Twisted Torus 接続構成を採ることで、64 画素のライン単位と 8x8 画素のブロック単位の並列処理構成が切り替えられるようにしている。

また、個別にアドレスの収束が可能な PE ごとのローカルメモリを内蔵し、ライン単位とブロック単位のメモリアクセスを、競合無しで可能にしている。さらに、LUT 用メモリを行単位に付けることより、最小限のメモリ容量でオリエンテーション (勾配とその方向) 算出を効率よく実行できるようにしている。最後に、不均一処理となるヒストグラム作成を Adder Tree を用いたことにより、並列で実行できるようにした。

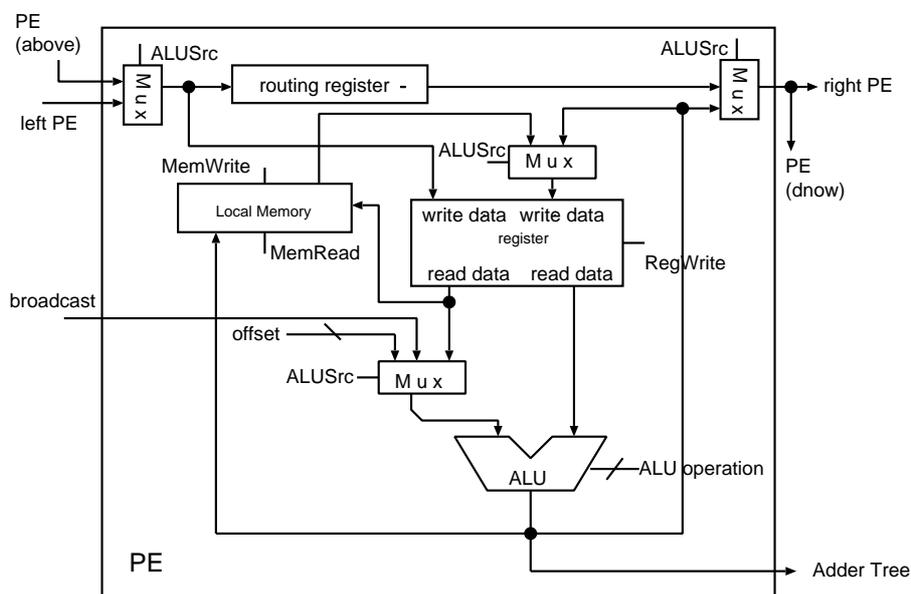


図 5.13: PE の構成

## 5.2 PE の構成

PE の構造は、MIPS の R3000 プロセッサの基本構造をベースにして、設計を行った。

内部構造は図 5.13 の示すように、ローカルメモリ、レジスタ、ALU、MUX とルーティングレジスタから構成されており、PE ごとでアドレスを決定する。具体的には、broadcast の外部通信により共通のアドレスを配り、PE ごとにあらかじめ 0 から 63 番をセットする、これよりアドレス生成を行い、64 の PE アレイのアクセス出来るようになり、全てプログラムにより制御される。プログラムの制御は外部の単一のコントローラで行い、すべての PE は一度に同一の命令を実行出来る。また、64 個 PE が 8x8 の PE アレイ状に並べられ、行ごとに Adder Tree 備え付けられている。

## 5.3 制御部の構成

制御プロセッサと PE アレイからなる制御部、メモリ転送制御部、ビデオ入出力制御部、CPU バス I/F から構成される。PE 部の詳細は前節で参照できる。

## 6 SIMD プロセッサ部の命令セットの設計

専用の MIMD 型並列プロセッサに劣らぬ性能が得るために処理ネックとなる部分に新たな高機能命令セットを定め、性能を向上させる。

### 6.1 Difference-of-Gaussian 処理に対応する命令

Difference-of-Gaussian 処理は、まず Gaussian フィルタの実現が必要とする。本アーキテクチャによる Gaussian フィルタでは、隣り合うガウシアン平滑化処理ではフィルタカーネルの係数と読み込まれた画素とを乗算しながら、その結果を PE 間列方向にシフト転送することで、自動的な累算を可能にする。実現方法は列と行、二段階の処理により効率化して、全画素とフィルタ係数の列方向を畳み込みその総和を求める（乗算と累算）。その処理結果を総和を求める（総和演算）。その効率的な処理に対応するために設計した命令は以下に示す。

1 SMDmra t0, t1, t2

1 サイクルで実行する乗算結果に隣接 PE からの入力 (累算結果) を加え、出力レジスタに格納する命令である。

Gaussian フィルタ演算では、ウィンドウサイズが  $7 \times 7$  (スケールは 1.3 の時) の場合、64 画素分ごとの Gaussian フィルタの演算量は 14 サイクルまでに低減できる。Gaussian フィルタ平滑画像の差分の DOG 画像の生成は、シストリックアルゴリズムを用いて、PE に初期画像の構成画素をロードすることから始まる。具体的には、Gaussian フィルタ平滑化画像から 64 画素  $x_2$  を読み出し、ALU により DoG 画像 64 画素を生成する。1 サイクルで差分を計算出来る。しかし、アドレス計算やロード命令が必要ため、高性能で実現するのが困難である。この 64 画素を毎サイクル PE に入力する。

### 6.2 極値検出に対応する命令

極値検出では、64 個の PE に DoG 画像データを蓄積される DOG 画像データから、極値 (最大値と最小値) を検出し出力する。その検出処理を効率的に実行する命令である。

1 SMDslgt

等しいとより小さい判定を行い、判定結果は隣接 PE に出力する。

## 2 ORS

最大値検出を効率よく実行するための命令で、隣接 PE からの判定結果を順に OR 累算する。ここで、OR 累算とは画素値のビット列の 0 ビット目に OR の演算し、ALL 0 かどうかを判定し、出力結果の最大値を近傍に隣接 PE に転送する。

## 3 ANDS

最小値検出を効率よく実行するための命令で、隣接 PE からの判定結果を順に AND 累算する。ここで、AND 累算とは画素値のビット列の 8 ビット目に AND の演算し、ALL 1 であれば、出力結果の最小値を近傍に隣接 PE に転送する。

これらの命令を組み合わせで、極値件検出処理は、3 サイクルで行えるようになる結果、64 画素分の極値検出が 26 サイクルで実行される。

### 6.3 オリエンテーション算出に対応する命令生成

オリエンテーション算出に対応するおける命令セットの生成勾配情報を利用し勾配とその方向を算出するオリエンテーション算出は、SIMD プロセッサにとり、SIFT 特徴量抽出処理中で最も重い処理である。

その処理に基づく生成した命令は以下ようになる。

#### 1 cdpe

共有メモリから読み出されたデータを、順番に PE 配列に取り込む命令である。

## 7 性能評価方法

シミュレータの構築方法については、専用の MIMD 型並列プロセッサに劣らぬ性能が得るために処理ネックとなる部分に新たな高機能命令セットを定め、性能を向上させる方法になる。評価方法について、SIFT 処理が必要とするアセンブリの命令を生成し、フィルタ演算部に置き換えるように演算量の評価を行った。

準備作業としては、SIFT 処理が必要とするアセンブリの命令を生成し、SIMD プロセッサの命令で演算部に置き換え、性能評価を行える。

本論文では、全体構成の基本評価を行った。主要な処理ごとに生成したアセンブリ言語の命令セットを利用し、パイプライン実行しない場合の評価対象と、所用のクロックサイクル数について評価を行った。

	1	2	3	4	5	6	7	8	9	10
1	ld	mul	ac	rou						
2		ld	mul	ac	rou					
3			ld	mul	ac	rou				
4				ld	mul	ac	rou			
5					ld	mul	ac	rou		
6						ld	mul	ac	rou	
7							ld	mul	ac	rou

図 7.14: Gaussian フィルタのサイクル数の生成例

## 7.1 Gaussian フィルタに対応する命令

Gaussian フィルタ演算では、係数との重み計算（積和演算）の並列処理（列と行、二段階の処理による効率化全画素とフィルタ係数の列方向を畳み込む（掛け算と累算））より実現できる。具体的に、演算を行う時に、Gaussian ウィンドの一行目総和演算を実現するために、階層的パイプライン処理で所用のサイクル数は図 7.14 に示す。また、フィルタウィンドウサイズが 7x7 の場合（スケールは 1.3 の時）、一行目のフィルタ演算をアセンブリ命令生成サイクル数の生成例は図 7.14 に示す。その他の列は同様にパイプラインで実行される。ただし、縦軸はウィンドウサイズの一行分を示して、横軸は逐次的な命令列による所用のサイクル数を示している。全体の処理時間の評価結果では、64 画素分ごとの Gaussian フィルタの演算量は 70 サイクルまでに低減できる。また、生成した SMDmra の命令をパイプラインで実行するのは、パイプラインは次の 5 段階に分割されていて、それぞれのステージが以下のように示している。

- E1 : ld、レジスタからデータを演算器に持って来る
- E2 : mul、乗算演算
- E3 : ac、隣接 PE からの入力（累算結果）を加える演算
- E4 :rou、PE 間で列方向にシフト転送する

	1	2	3	4	5	6	...	...	...	24	25	26
1	SMD slgt	Ors	ands									
2		SMD slgt	Ors	ands								
3			SMD slgt	Ors	ands							
4				SMD slgt	Ors	ands						
...							...	...				
...							...	...				
								SMD slgt	Ors	ands		
									SMD slgt	Ors	ands	
26										SMD slgt	Ors	ands

図 7.15: 極値検出の所有のサイクル数

## 7.2 極値検出における命令セット

極値検出を実現するための命令とサイクル数の生成例は図 7.15 に示している。処理をオーバーロードできるので、26 方向にはパイプラインで実行できるので、その結果、極値検出の所用のサイクル数は 26 サイクルになる。また、パイプラインは次の 3 段階に分割されていて、それぞれのステージは以下のようなになる。

E1 : SMDslgt、等しいとより小さい判定を行い、判定結果は隣接 PE に出力する。

E2 : ORS、隣接 PE からの判定結果を順に OR 累算する。

E3 : ANDS、隣接 PE からの判定結果を順に AND 累算する。

## 7.3 オリエンテーション算出における命令セット

SIFT 特徴量演算量の最も処理ネットとなるオリエンテーション算出部は勾配情報を利用し強度と方向を算出処理となる。メモリその処理に基づく生成したアセンブリ命令は以下のようなになる。

1 lw t0, 0(s2); メモリからレジスタへ方向の先頭アドレスを t0 にロードする。

2 lw t1, 4(s3); メモリからレジスタへ方向の勾配情報アドレスを t1 にロードする。

3 sll t2, t1,8; 方向の先頭アドレスと方向の勾配情報アドレスの連結演算には、8ビット左シフトする。

4 or t0, t1,t2; シフトしてから、2つのレジスタのOR命令でビット連結の演算をできる。

5 sw t0, 0(s1); レジスタからメモリへ連結したもの。

cdpe (capture data to PE) 共有メモリから読み出されたデータを、順番にPE配列に取り込む命令である。

8 PEを同時に行えると、LUTで方向のアドレス生成をするので、全部で9サイクルで実現できる。また、8PEごとの8並列で実行することで64PEの場合では8倍化高速できる。

## 7.4 ヒストグラムにおける命令セット

ヒストグラム作成による命令セットの生成は以下のように、1から36の整数を使って比較信号を行う、未満の時は0を出力、一致したら1を出力する。三つの命令セットを36回を行うので、36 X 3サイクルをかけることになる。

sltiu,rs,rt,immediate : 16ビットimmediateを符号拡張して、rtの内容と比較する。32ビットの整数として両方の数を判定、rtがimmediate未満の時には結果は1、そうでなければ結果が0になる。結果はrsに格納する。ヒストグラムの方向加算に応用。

```
addi t2, s1, 0;
sltiu t2, t2, 1;
sw t2, 100(s0);
... ..
addi t2, s1, 0;
sltiu t2,t2, 36;
sw t2, 100(s0);
```

add t0, t2, t0 最後には、ヒストグラム用のメモリに積み上げれる。

所用ステップ数の見積もりを行った。また、オリエンテーションの算出の評価方法については、以下の図7.16に示している。

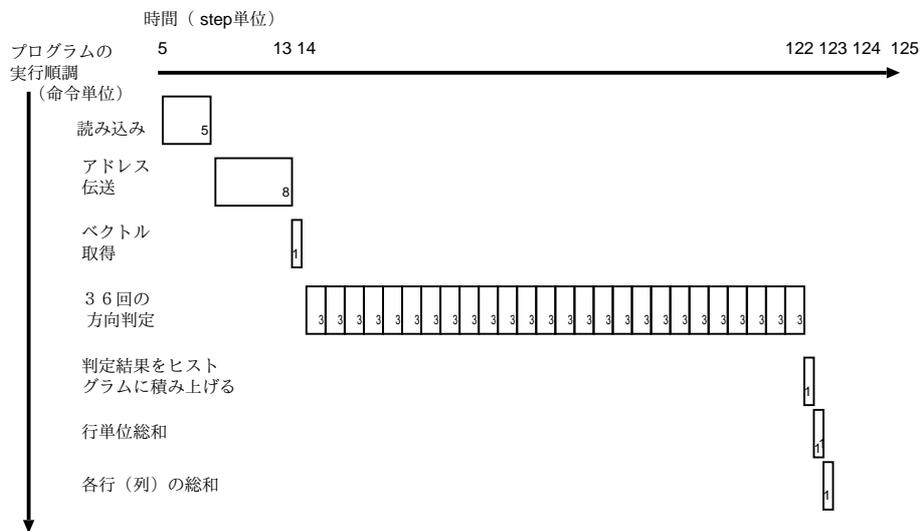


図 7.16: オリエンテーションの算出のステップ数

## 7.5 評価結果

評価結果は図 7.17 に示している。横軸は所用のサイクル数を表しており、バーは低いほうが高性能である。Gaussian フィルタ演算処理で、フィルタ処理を 3 倍高速化した。極値検出で、注目画素の近傍の比較演算を 3 倍高速化した。オリエンテーション算出（ヒストグラム生成を含む）で、ハードウェア構成の追加によって、8 並列で実行により 7 倍高速化した。具体的に、各処理が命令を生成おける所用サイクル数の内訳を以下に示している。また、評価モデルは、SIMD プロセッサがパイプライン実行しない場合になる。

— Gaussian フィルタ : —

ここで、評価対象は Systolic Algorithm を用いていない場合とする

no pipeline :  $196 = 4 \times 7 \times 7$

This Work :  $70 = 10 \times 7$  (パイプラインで実行する)

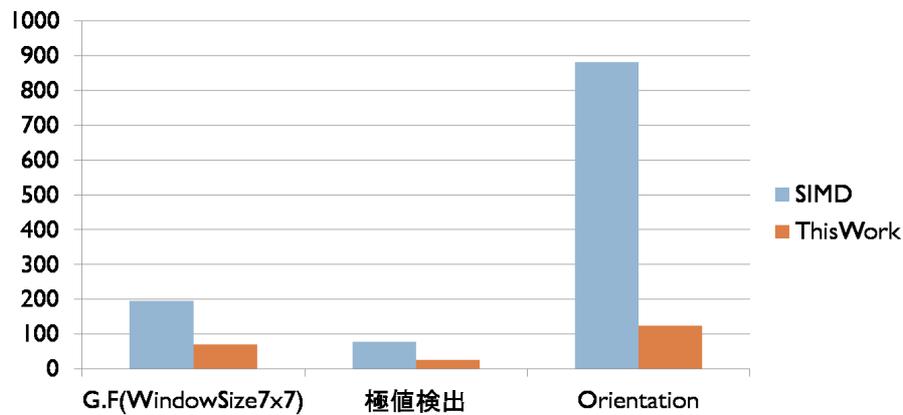


図 7.17: 所用ステップ数の評価

— 極値検出 : —

ここで、評価対象は Systolic Algorithm を用いていない場合とする

no pipeline :  $78 = 3 \times 26$

This Work : 26 (パイプラインで実行する)

— オリエンテーション (ヒストグラム) : —

評価対象は Systolic Algorithm を用いていない場合

no Systolic Algorithm :  $881 = 5 + 8 + 1 + 3 \times 36 \times 8 + 3$

This Work :  $125 = 5 + 8 + 1 + 3 \times 36 + 3$

(8PE + LUT + AdderTree = 64PE の 8 並列の Systolic Array)

## 8 関連研究

### 8.1 SIFT 特徴量を用いた画像認識

#### 8.1.1 SIFT 特徴量

実行ファイルは book.pgm の画像ファイルの特徴量求める。book.key ファイルとしてキーポイントを求められ、記述されたキーポイントは book.key ファイルに生成され、出力される。これにより、合計 128 次元の特徴ベクトルをキーポイントの特徴量として使う。

対象物：

```
./sift < book.pgm > book.key  
Finding keypoints...  
2254 keypoints found.
```

入力画像：

```
./sift < desk2.pgm > desk2.key  
Finding keypoints...  
5434 keypoints found.
```

以下は book.key ファイルにより生成例を説明する。一列目はキーポイントの数と特徴ベクトルの次元数であり、二列目は 1 つめキーポイントの X 座標, Y 座標, スケール, 向きであり、三列目はインデントされている数字が 128 次元ベクトル (20 個で改行が入る) 2254 128

/\* 1 つめのキーポイント \*/

```
542.10 338.67 97.25 -0.608  
26 1 0 0 0 0 26 51 1 0 0 0 0 6 161 2 0 0 0  
0 0 10 53 0 0 0 0 0 0 0 92 42 13 0 0 0 0 17  
178 40 4 0 0 2 35 178 47 7 2 13 37 29 106 154 0 0 0 1  
6 1 0 0 74 38 4 0 0 0 0 178 178 14 3 1 1 3 40  
57 123 35 45 53 11 9 16 0 0 0 3 9 0 0 0 4 5 0 0  
0 0 0 0 39 100 0 0 0 0 0 0 3 24 1 0 0 0 0 0  
0 0 0 0 0 0 0
```

/\* 2 つめのキーポイント \*/

```
309.71 264.30 67.58 -0.873  
5 0 0 0 0 2 7 31 11 2 0 0 0 4 65 40 3 0 0  
0 0 5 69 1 0 0 0 0 0 5 5 52 7 2 4 0 0 2 36
```

```
87 30 19 31 3 0 3 45 160 20 4 1 0 2 53 160 11 0 0 0
12 44 121 50 25 25 18 12 0 0 0 5 134 42 28 51 1 0 0 3
160 160 11 3 1 3 12 57 45 48 31 60 81 45 29 19 0 0 1 1
0 0 0 0 28 45 5 3 0 0 0 0 111 160 34 0 0 0 0 0
5 98 85 33 6 0 0 0
```

### 8.1.2 特徴量マッチング

対象画像と入力画像で生成した、book.key と desk2.key ファイルを利用して対応点をマッチングする。

出力画像：

```
./match -im1 book.pgm -k1 book.key -im2 desk2.pgm -k2 sk2.key >out3.pgm
Found 413 matches.
```

画像 book と画像 desk2 が上下に並べて、対応点を連結した画像が描写され、出力画像は out3.pgm になる。図 3.6 に示すように。

## 8.2 Look up table

LUT [ look-up table ] 輝度変換を行うための入出力対応テーブルのことです。LUT は、Look up Table の略で、入力輝度に対する出力輝度の割当テーブルのことである。通常は LUT のテーブル容量 (格納数) は階調分だけ存在する。例えば 256 階調なら 256 個のデータ格納数を持っている。このテーブルに、入力輝度に対する出力輝度をあらかじめ作成しておき、そのテーブルを参照することで輝度変換を行う。例えばテーブルの 64 番目に 120 という数字を入れたと仮定すると、入力輝度が 64 の画素の出力輝度はテーブル参照によって瞬時に 120 に置き換えられる LUT は、明るさの変更などに利用されている。

さらに、処理効率をよく参照や変換をする目的でつくられた配列や連想配列などのデータ構造のこともいう。例えば大きな負担がかかる処理をコンピュータに行わせる場合、あらかじめ先に計算できるデータは計算しておき、その値を配列 (ルックアップテーブル) に保存しておくこと。

### 8.3 PE Array

並列コンピュータの構造は、集中共有メモリ方式と、分散共有メモリ方式に構成さえる。並列コンピュータの構成では、プロセッサ PE がメモリを共有の集中と分散によって通信や同期のような性能が決められる。

部分共有メモリ方式では、それぞれの PE が専用ローカルメモリを持ち、また、共有メモリが結合網を通じて接続させるは部分共有メモリ方式になる。ローカルメモリがそのバスに接続された PE だけがアクセスできる。

アレイ方式では、PE が隣どうしだけで通信できる特徴である。PE にはメモリが含まれており、PE 間にも通信用バッファとしてメモリが置かれていて共有メモリ型通信をする。

そして、今回の研究でも採用した手法であり、しかし隣同士しか通信できない欠点を工夫して、NXN の PE が全体にも通信出来るような構成を設計した。

## 9 あとがき

本研究では、SIFT 特徴量ベースの画像認識に対応する汎用性の高い並列プロセッサの基本設計を試みた。SIFT 特徴量抽出の主要な処理を全部 SIMD 並列処理可能としたことで、SIFT ベースの画像認識の機能を効率的な処理が可能になる。現段階では、基本設計に基づくシミュレーターの構築方法については確立した。今後の課題としては、基本設計の結果に基づくシミュレーターを完成し、それによる機能構成の検証と構成の最適化を行う。

## 謝辞

本研究を進めるにあたり、ご指導を頂いた卒業論文指導教員の近藤利夫教授、並ぶに多く助言を頂きました大野和彦講師、佐々木敬泰助教授に深く感謝致します。また、日常の議論を通じて多くの知識や示唆を頂いた計算機アーキテクチャ研究室の皆様感謝いたします。

## 参考文献

- [1] 藤吉弘宣 ”一般物体認識のための局所特徴量 (SIFT と HOG) ”  
PCSJ/IMPS2008 ナイトセッション.
- [2] 小室孝, 鏡慎吾, 石川正俊 ”ビジョンチップのための動的再構成可能な SIMD プロセッサ”, 電子情報通信学会論文誌 D-II, Vol.J86-D-II, No.11, pp.1575-1585 (2003)
- [3] 岡崎信一郎, 京昭倫, 古賀拓也, 肥田野文之: 車載組込み用画像認識プロセッサ IMAPCAR, NEC 技報 Vol.60 No.2/2007 17 組込みシステム・プラットフォーム.
- [4] Bonato, V. / Marques, E. / Constantinides, G A. ”A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection”  
IEEE transactions on circuits and systems for video technology. - Vol. 1, no. 1 (Mar. 1991)-. - New York, N.Y. : Institute of Electrical and Electronics Engineers , c1991-.
- [5] Yoichi MURAOKA “The Configuration of Main Memory for Illiac IV  
「Information Processing Society of Japan」 Vol.16 No.4