

積分画像をベースとした U-SURF アルゴリズムの超並列プロセッサへの並列実装手法

佐藤 好紀[†] 野田 英行^{††a)} 杉村 武昭^{††b)} 長崎 健^{†c)}

Integral-Image Based Implementation of U-SURF Algorithm for Embedded Super Parallel Processor

Yoshinori SATO[†], Hideyuki NODA^{††a)}, Takeaki SUGIMURA^{††b)},
and Takeshi NAGASAKI^{†c)}

あらまし 近年、画像認識において U-SURF のようなスケール不変特徴量抽出アルゴリズムが多く用いられている。U-SURF は、画像認識アプリケーションにとっては非常に有効であるが、実行には画像中に散在する多くの局所領域に対する処理が必要となり、超並列処理時において、処理並列度を高めることは困難である。そこで本研究では、U-SURF の組み込み超並列処理プロセッサに対する最適な並列実装手法を提案した。U-SURF の局所特徴量抽出処理を積分画像をベースとして並列化することで、効率的な並列処理アルゴリズムを実現した。提案手法の導入により、MX コア併用時の局所特徴量抽出処理の実行時間は約 1710 ms から約 234 ms へ短縮し、約 7.3 倍の高速化を実現した。また、汎用 RISC プロセッサのみ使用時と MX コア併用時の U-SURF、及びベクトルマッチングの実行時間は、40271 ms から 991 ms へ短縮し、約 40.6 倍の高速化を実現した。評価結果より、組み込み超並列処理プロセッサを使用したシステムにおける、提案手法の有効性を示した。

キーワード SURF, SIMD, 並列処理, 局所特徴量, 組み込みシステム

1. ま え が き

近年、デジタルカメラや道路標識認識 [1]、人間の追跡 [2]、侵入者の監視等のアプリケーションにおいて、画像認識が利用されている。このようなアプリケーションにおける画像マッチングや物体認識の手段として、Scale Invariant Feature Transform (SIFT) [3] アルゴリズムが提案されている。SIFT は、画像中から回転不変かつスケール不変な特徴点と、その周囲の特徴量を記述するアルゴリズムである。

SIFT はユニークな画像特徴量の抽出が可能である

が、実行には高性能なシステムが要求される。これは、記述する画像特徴量のスケール不変性獲得のため、ガウスフィルタを繰り返し適用することに起因する。Speeded Up Robust Features (SURF) [4] アルゴリズムは、SIFT におけるガウスフィルタを BOX フィルタに置き換えることで、SIFT を近似したものである。このため、SURF はより少ない計算量で、スケール変化に対してロバストな特徴点・特徴量の抽出が可能である。

しかし、この SURF であっても、組み込みシステムにおけるリアルタイム処理を実現するためには、比較的高性能なプロセッサが要求される。

この問題を解決するため、U-SURF の実行に、組み込み超並列処理プロセッサ “MX コア” [5] を利用する。U-SURF とは、SURF から回転不変性を排除したアルゴリズムである。MX コアの構成を図 1 に示す。MX コアは 1024 個の PE (プロセッサエレメント) と 1024 ビット × 1024 ワードのデータレジスタによって構成されている。一つの PE では、PE の左右にあるデータレジスタのデータに対し、1 クロックで 2 ビッ

[†] 公立はこだて未来大学システム情報科学研究科，函館市
Graduate course of System Information Science, Future University Hakodate, 116-2, Kamedanakano-cho, Hakodate-shi, 041-8655 Japan

^{††} ルネサスエレクトロニクス株式会社技術開発本部システムコア開発統括部，伊丹市

System Core Development Div., Technology Development Unit, Renesas Electronics Corporation, 4-1, Mizuhara, Itami-shi, 664-0005 Japan

a) E-mail: hideyuki.noda.pc@renesas.com

b) E-mail: takeaki.sugimura.wj@renesas.com

c) E-mail: nagasaki@fun.ac.jp

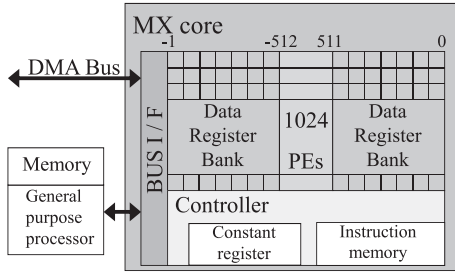


図 1 MX コア

Fig. 1 MX core.

トの加減算等を実行し、データレジスタに書き戻すことが可能である。PE が扱えるデータレジスタは、PE と同じ行にあるもののほか、2 の n 乗の位置にある他の行のデータレジスタも扱うことが可能である。また、同じ定数をすべての PE に適用する演算等が可能である。MX コアは Single Instruction, Multiple Data stream (SIMD) 型のプロセッサであり、一つの命令ですべての PE が同時に動作する。MX コアを利用することで、動作周波数 166 MHz 時の 16 ビット加算において、最大 17 GOPS の並列処理性能を実現している。演算に用いるデータや演算結果の転送には、ホスト CPU からの指示に従って、SDRAM からデータレジスタあるいは、データレジスタから SDRAM に対して DMA (Direct Memory Access) 転送を行う。

並列演算命令のデータは、ホスト CPU からの指示に従って、SDRAM から MX コア内部の命令メモリへ DMA 転送される。本研究の評価実験には、ホスト CPU として SH-2A プロセッサを使用している。

MX コアを利用した場合、オリジナルの U-SURF をそのままポーティングするだけでは、パフォーマンスを最大限に発揮することは困難である。そこで本研究では、MX コアに対する U-SURF の最適な並列実装手法を提案する。

2. U-SURF の概要

U-SURF は、画像の拡大率に依存しない画像中の特徴を表す点を検出し (図 2 の特徴点検出部)、検出されたそれぞれの特徴点について、それぞれの特徴点の特徴を表す局所特徴量を局所領域の輝度こう配のヒストグラムとして表し (図 2 中央の局所特徴抽出)、局所特徴量の比較を行うことで特徴点の対応を求める。以下では、U-SURF の、特徴点検出、局所特徴量抽出、特徴量ベクトルマッチングの三つの処理について

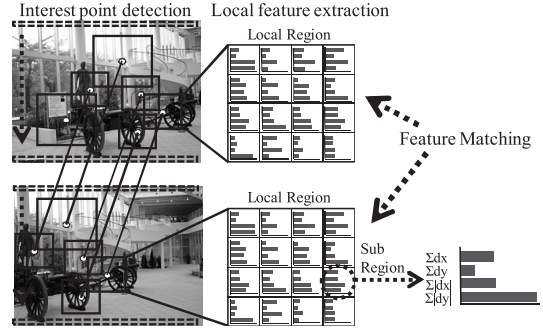


図 2 U-SURF の概要

Fig. 2 Outline of U-SURF.

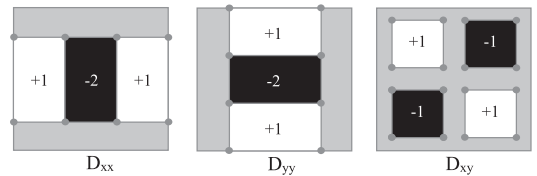


図 3 Box フィルタ

Fig. 3 Box filter.

概略を述べる。

2.1 特徴点検出

特徴点は、BOX フィルタ処理と、それによって得られた値を利用して計算された行列式である Hessian 値の極大点を探索することによって検出される。特徴点検出処理では、まず、元画像から積分画像 [6] $ii(x, y)$ を生成する。積分画像は、式 (1)、式 (2) に示すように、元画像の左上から任意の座標 (x, y) までの総和を計算することによって生成される。

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

ただし、 $i(x, y)$ は入力画像である。

次に、積分画像を利用して、BOX フィルタ (図 3) を計算する。BOX フィルタを適用することによって、ガウスフィルタと微分フィルタを畳み込んだ値の近似値である D_{xx} , D_{yy} , D_{xy} を得る。スケール不変性を得るため、BOX フィルタを 9×9 , 15×15 , $21 \times 21, \dots$ の様々なマスクサイズで適用し、マスクサイズに応じた D_{xx} , D_{yy} , D_{xy} を求める。BOX フィルタは、積分画像を利用することにより、マスクサイズにかかわらず、 D_{xx} は 8 点、 D_{yy} は 8 点、 D_{xy} は 16 点の画素値に対して単純な加減算のみで計算可能であるため、非常に高速に演算可能である。

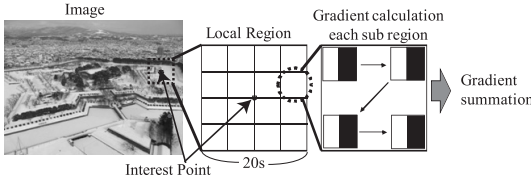


図 4 局所特徴量の抽出

Fig. 4 Extraction of a local descriptor.

次に、全スケールの D_{xx} , D_{yy} , D_{xy} から、Hesse 行列の近似値 H' の行列式 $\det(H')$ を計算する．ここで、 $\det(H')$ は Hessian と呼ばれる．Hessian によって、画像中の輝度値が成す曲面の形状を分類可能であり、 (x, y, s) の 3 次元空間内における Hessian 値の極大点で、釣鐘型の形状が周囲と比較して最もよく現れていると判断できる．この性質を利用して、Hessian 値を極大点でしきい値処理することで、特徴点を検出する．

特徴点検出処理では、BOX フィルタ処理、Hessian 値計算、Hessian 値の極大点検出処理をそれぞれ実行するが、これらの処理には多くの近傍領域処理が含まれている．それゆえ、これらの処理は超並列処理プロセッサへポーティングすることで、高い並列度を実現することが可能である．一方、上記の処理で用いられている積分画像生成アルゴリズムについては、超並列処理への並列化が可能である．

2.2 局所特徴量抽出

局所特徴量抽出処理では、検出された特徴点の周囲からユニークな記述子を抽出する．抽出された局所特徴量は、スケール変化に加えて照明変化に対してもロバストである．局所特徴量は、図 4 のように特徴点を中心とし、そのスケールに応じた周囲 $20s \times 20s$ の局所領域から抽出される (s は BOX フィルタで近似したガウスフィルタの分散を示すスケール値である)．ここで、局所領域を 4×4 に分割した各小領域を“サブ領域”と呼ぶ．

局所特徴量抽出処理では、まず、図 5 のような $2s$ の大きさの Haar-Wavelet フィルタを局所領域の全ピクセルに適用し、特徴点のスケールに対応した輝度こう配 d_x , d_y を計算する．このとき、Haar-Wavelet フィルタは、特徴点検出処理時の BOX フィルタと同様に、積分画像を利用して計算される．そのため、1 回のフィルタ演算には、フィルタサイズにかかわらず 6 点の画素値に対して単純な加減算のみで計算可能である．

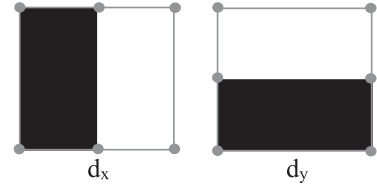


図 5 Haar-Wavelet フィルタ

Fig. 5 Haar-Wavelet filter.

次に、輝度こう配 d_x , d_y とその絶対値である $|d_x|$, $|d_y|$ のそれぞれを、局所領域内のサブ領域ごとに総和を計算することによって、4 次元ベクトル $\nu = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ を得る．4 次元ベクトル ν は、局所領域内の各サブ領域ごとに抽出され、各サブ領域の 4 次元ベクトルを合わせた 64 次元ベクトルを、特徴点に対応した局所特徴量として抽出する．

局所特徴量抽出処理では、画像中に散在する多くの方形領域に対する輝度こう配計算が必要である．この処理をそのまま超並列処理プロセッサへ実装しても、処理効率を高めることができない．その理由として、演算領域の重複が挙げられる．画像中に散在する局所領域は、同じ演算を実行する領域同士が重複していたり、同じライン上に存在する場合が非常に多い．MX コアを利用した場合、同じラインのデータを並列処理可能であるため、局所領域ごとに処理するようなアルゴリズムでは非効率である．

更に別な要因として、データ転送量が挙げられる．局所特徴量抽出処理では、積分画像を利用して Haar-Wavelet フィルタを計算するため、任意の 1 ラインに対する輝度こう配を計算する場合、 X 方向輝度こう配では 2 ライン分、 Y 方向輝度こう配では 3 ライン分の積分画像が必要である．積分画像のラインの位置は、計算する局所領域の位置やスケールによって変化し、これらのすべてのラインに対してデータ転送が発生する．このため、データ転送回数が増大し、それに比例して転送時間も増大する．

上記の問題点を解決するため、局所特徴量抽出処理を超並列処理プロセッサへ最適化することで処理効率を最大化する．

2.3 特徴量ベクトルマッチング

特徴量ベクトルマッチングでは、図 6 のように一方の画像から U-SURF によって抽出されたある一つの局所特徴量の 64 次元ベクトルに対し、もう一方の画像からあらかじめ抽出された局所特徴量同士の 64 次元ベクトルのすべてに対して、距離比較による対応付

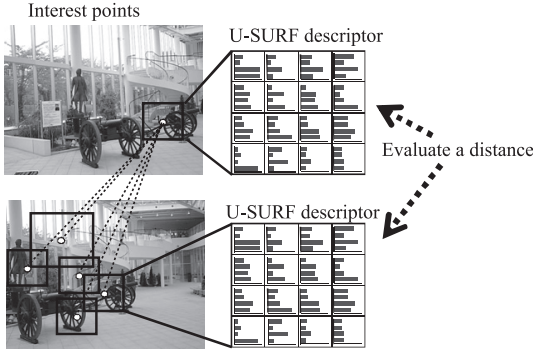


図 6 2 枚の画像の特徴量の比較

Fig. 6 Comparison of an U-SURF descriptor of one image and all U-SURF descriptors of the other.

けを行う．特徴量ベクトルマッチングにより，物体認識や追跡，その他の様々なアプリケーションが実現可能である．一般的に，ベクトルマッチングにはユークリッド距離が用いられるが，本研究では演算コストを優先し，マンハッタン距離を採用している．また，マッチング手法には，Nearest Neighbor 法としきい値処理を組み合わせた手法を利用している．特徴量ベクトルマッチング処理は，超並列処理への並列化が可能である．

3. 積分画像を用いた U-SURF の並列化とその実装手法

組み込み用途の画像認識アプリケーションへの応用に向けて，U-SURF の並列処理アルゴリズムを MX コアへ実装した．U-SURF の並列化のためには，特徴点ごとに演算が必要となる複数の局所処理を，大域処理を用いて実施する必要がある．一方で，U-SURF における特徴点抽出処理では前章で述べたように BOX フィルタに対し積分画像を利用して並列化することが可能であり，MX コアを用いた積分画像の生成も高速化が可能である．そこで，特徴点ごとに演算が必要な特徴量抽出処理に対して，積分画像を用いた大域処理に基づく並列化手法を提案する．本節では，U-SURF の並列化手法について要素処理ごとにその並列化手法と MX コアへの実装について述べる．

3.1 積分画像生成の並列化とその実装手法

積分画像生成では，まず入力画像幅 $\text{pixel} \times 1 \text{ pixel}$ で構成される水平方向 1 ラインの画像データに対して水平方向の 1 次元積分画像を並列に演算する．その演算結果を複数ラインに対して積算することで積分画像

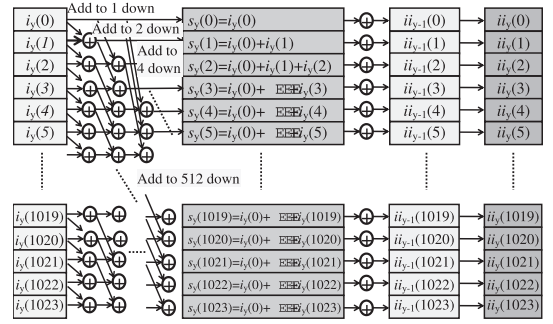


図 7 MX コアを用いた積分画像生成処理

Fig. 7 Integral image generation with the MX core.

を得る．この方法では，SIMD 演算を実行する演算器の数を N ，入力画像の横幅を w ，入力画像の縦幅を h として 1 次元積分画像の生成に要する演算時間を一定とすると，その演算時間は $O(wh/N)$ となる．ここでは，MX コアの並列度は 1024 並列と入力画像に対して十分大きいので， $O(h)$ となり入力画像のライン数に比例する．

MX コアを用いた 1 次元積分画像の並列演算の実装を図 7 に示す．積分画像生成処理では，演算対象の 1 ラインの入力画像 $i(x, y)$ を SDRAM から MX コアのデータレジスタへ転送し，MX コアに格納された 1 ラインの画像データ $i_y(x)$ に対し，以下の式 (3) を適用する．

$$s_y(x) = \begin{cases} i_y(x) & (x < 2^n) \\ i_y(x) + i_y(x - 2^n) & (x \geq 2^n) \end{cases} \quad (3)$$

ここで， i_y ， s_y は入力画像と計算結果の 1 ラインのベクトルをそれぞれ表している．また， n は $1 \leq n \leq \log_2 w$ を満たす整数である．図 7 に示すように，まず $i_y(x)$ を隣接画素のデータと加算し，更にその結果の 2^n 画素離れた結果との加算を繰り返すことで，0 画素から自画素までの画素値の合計を自画素のデータレジスタに格納した 1 次元積分画像を得る．このために必要な加算は $\log_2 w$ 回となり，逐次処理と比較して大幅な演算時間の削減が可能である．

次に，求めた $s_y(x)$ を，式 (4) に示すように前ラインまでの積算結果に加算して該当ラインまでの積分画像を得る．

$$ii_y(x) = ii_y(x - 1) + s_y(x) \quad (4)$$

ここで， ii_y は積分画像の 1 ラインのベクトルであり，この処理をライン数だけ繰り返すことで入力画

像の積分画像を得る．以上より，逐次処理の演算時間 $O(hw)$ に対して，並列化後の演算時間は $O(h \log_2 w)$ となる．このように，MX コアを用いた積分画像の生成では，並列化による演算時間の削減が可能であり，この手法は後述する局所特徴抽出処理でも利用する．

3.2 特徴点抽出の並列化とその実装手法

特徴点抽出では 2.1 で述べたように BOX フィルタ処理，Hessian 値計算，Hessian 値の極大点検出については単純なフィルタ処理であるため，並列化が有効に働く．これらの処理では，ライン単位で並列に実行しているため，計算量は画像のライン数 h にだけ依存し， $O(h)$ となる．

以上のフィルタ処理により画像中の特徴点座標の画素値として特徴点データを保持する特徴点画像が生成される．この特徴点画像から特徴点に対する局所特徴量の抽出のため，特徴点の ID (通し番号) と座標データを含む特徴点リストを作成する必要がある．逐次処理では特徴点画像をスキャンしてその画素が特徴点に該当する場合リストに加える処理を全画素に対して適用する必要がある．これに対して，提案する特徴点抽出の並列化手法を図 8 に示す．提案する手法では，外部メモリにある特徴点画像をラインごとに MX コアのデータレジスタに書き込み，並列処理に適したバイトニックソートを用いて座標も含む特徴点のデータを上位エントリ (若い番号の PE) に集めて配置する．この整列済みのデータを MX コアより外部メモリのラインバッファ領域に書き戻し，汎用プロセッサによる逐次スキャンを実施することで，大部分の画素についての参照処理を削減している．本手法では画像の 1 ラインを並列に処理しているため，バイトニックソートの計算量は $O((\log_2 w)^2)$ となる．また，1 ラインに存在する特徴点数は，1 ラインの画素数より十分少ないため，バイトニックソート後に探索する方が，通常的全画素の探索より少ない計算で特徴点の抽出が可能である．

この汎用プロセッサで行う処理の計算量は，検出された特徴点数を m とすると $O(m)$ となる．具体的な処理を見ても，最大で $2m$ 回の比較と特徴点リストを作成するための m 個の特徴点の座標値の転送処理を行うだけなので，軽い処理となっている．それに対し，MX コアで実行する処理は，バイトニックソートの計算量が $O((\log_2 w)^2)$ ，それ以外が $O(h)$ であり，計算の複雑さについては大きな差はないが，MX コアは多数のフィルタ処理等を行うため，実際には多くの時間

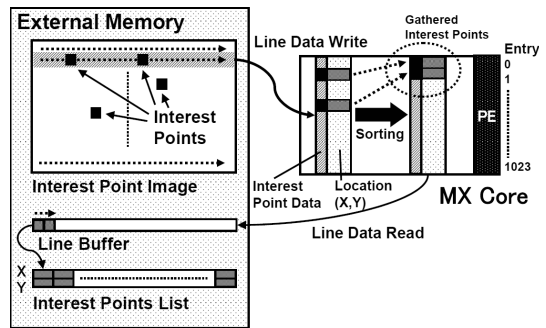


図 8 MX コアによる並列ソーティングを用いた特徴点抽出処理

Fig. 8 Interest point extraction utilizing parallel sorting with MX.

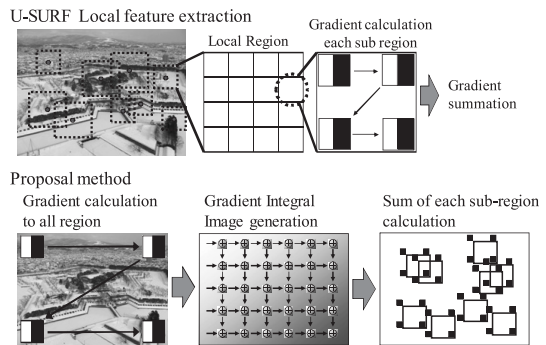


図 9 提案する特徴量抽出手法と従来手法の比較

Fig. 9 Comparison of proposed method of descriptor components extraction and conventional method.

を占めることになる．

3.3 積分画像ベース局所特徴量抽出アルゴリズム

特徴点リストをもとにその特徴点近傍の画素値を用いて特徴量を抽出する局所特徴量抽出処理では，特徴点近傍のサブブロック領域に対して Haar-Wavelet フィルタを適用し，その演算結果を用いて特徴量を演算する．この処理はすべての特徴点に対して必要であり，画像中に散在する多くの方形領域に対して同様の処理を実行する．この局所処理については，複数の領域を MX コアに書き込み処理を実行することで並列化は可能である．しかし，複数の特徴点に対する方形領域が重なる場合など，同一のデータを複数回 MX に転送することが必要であり，他の並列処理と比較してデータ転送時間の比率が高くなる．このため，積分画像を利用した，特徴点に依存しない大域処理を含む局所特徴量抽出アルゴリズムを提案する．

図 9 に従来的特徴量抽出手法と提案手法の比較を示

す．図 9 上段に示す従来手法では，すべての特徴点に対してその近傍画素データを用いて輝度こう配を演算し，その結果から局所領域ごとの 4 次元ベクトルを求める．これに対して，図 9 下段に示す提案手法では，あらかじめ入力画像全体に Haar-Wavelet フィルタを適用し，輝度こう配を得た後，輝度こう配画像より積分画像を生成する．積分画像を用いることで，前述のように任意の領域に対して 4 点の画素値からそれらの点で囲まれる領域の輝度こう配の総和を求めることが可能であり，MX コアを用いて処理を実行する場合は，1 領域につき 5 ラインの画像データを MX コアに書き込むことで，局所特徴量を演算できる．

この積分画像を用いた局所特徴量抽出処理の概要を図 10 に示す．まず，特徴点抽出に用いた積分画像より，画像の全領域，全スケールに対する輝度こう配 d_x ， d_y を求める．次に，輝度こう配 d_x ， d_y とその絶対値である $|d_x|$ ， $|d_y|$ をそれぞれ積分した，輝度こう配積分画像を作成する．提案手法では，図 10 の輝度こう配に対する積分画像の作成までは画像全体に適用する大域処理であるため並列化が可能であり，加えて輝度こう配積分画像を使用することによって，任意の方形領域に対する d_x ， d_y ， $|d_x|$ ， $|d_y|$ の総和演算に必要なデータは領域端点のみとなり，従来手法と比較すると MX コアへのデータ転送回数は大幅に削減される．最後に，この演算結果より汎用プロセッサで局所領域ごとの特徴ベクトル成分 ($\sum d_x$ ， $\sum d_y$ ， $\sum |d_x|$ ， $\sum |d_y|$) を演算し，大きさの規格化を行い必要な特徴量を得る．このように，大域処理と，演算に必要なデータ量を削減した局所処理を組み合わせることで，演算の並列度

を維持すると同時に，実行時間に対するデータ転送時間の比率を低く抑えることが可能である．

積分画像ベース局所特徴量抽出アルゴリズムの計算量は，画像幅 w と画像ライン数 h に対して，積分画像が 3.1 に示したとおり， $O(h \log_2 w)$ であり，Haar-Wevelet フィルタの計算が $O(h)$ となり，逐次処理より大きな改善が見られる．また，特徴ベクトルの作成には，汎用プロセッサと MX コアが協調して実行にあたる．64 次元の特徴量ベクトルを作成するために，ベクトルの一つの要素につき 4 点の画素の加減算を汎用プロセッサで行う．汎用プロセッサによって求められた 64 次元のベクトルの大きさの 2 乗を MX コアで求め，ベクトルの大きさを汎用プロセッサで求め，ベクトルの規格化のための割り算を MX コアで実装した．これらの処理は $O(m)$ となるが，汎用プロセッサにおいて計算すべきことが他の処理の部分より大きいため，汎用プロセッサの計算時間が局所特徴量抽出アルゴリズムの処理時間に占める割合は，大きな値となる．

3.4 特徴ベクトルマッチングの並列実装手法

MX コアを利用した特徴量ベクトルマッチング処理の概要を図 11 に示す．

特徴量ベクトルマッチングでは，まず，片方の画像から抽出された任意の特徴量ベクトルを MX コアへ格納する．このベクトルを比較対象ベクトル V_b と呼ぶ．ただし，任意の V_b ごとにデータ転送を行うのではなく，次のようにする．

(1) 特徴量ベクトルのうち，MX コア内部の 1 ラインに格納可能な数のベクトルを格納する．ここで，MX コア内部の PE 数が 1024 であり，かつ特徴量ベクトルの次元数が 64 次元であるため，一度に格納可能な特徴量ベクトルは 16 個分である．

(2) 格納された 16 個分の特徴量ベクトルのうち，

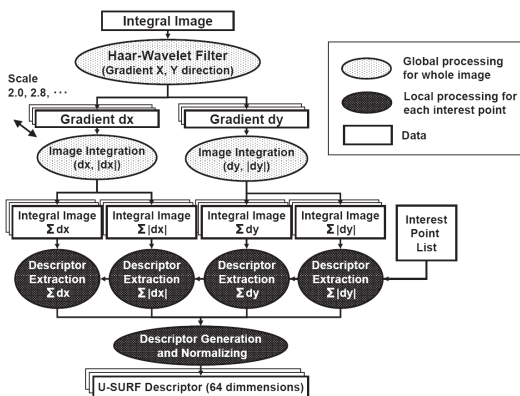


図 10 積分画像を用いた局所特徴量抽出アルゴリズム

Fig. 10 Local descriptor extraction algorithm utilizing integral image.

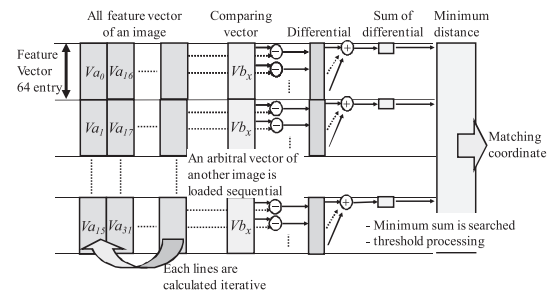


図 11 MX コアによるベクトルマッチング

Fig. 11 Vector matching with the MX core.

任意のベクトルを MX コア内の別領域へ展開し、1 ライン中にそのベクトルの 16 個分のコピーを作成する (図 11 中 Vb_x).

次に、もう一方の画像から抽出されたすべての特徴量ベクトル Va を MX コアへ格納する. このとき、1 ライン当り 16 個分のベクトルが複数のラインにわたって格納される (図 11 中 $Va_i, i = 0 \dots$).

次に、MX コアのデータレジスタに格納された 16 個の特徴ベクトル Va の任意のラインに対し並列に、 Vb とのベクトル成分ごとの差分を求める. 求められたベクトル成分ごとの差分を、各特徴量ベクトルごとに 64 次元分の総和を求める. 一つの特徴点の特徴ベクトル 64 次元の総和を求める計算では、 $O(\log_2 64)$ で実行でき、これを 16 個並列に実行が可能である. また、ここで求めた 16 個の差分総和を、 Va の前ラインについて求められた差分総和と比較し、値が小さい場合は更新することで、最小値の候補を MX コアのデータレジスタに保持する. 上記のベクトル差分総和演算を、すべての Va に対し、繰り返し実行する.

次に、保持されている 16 個の差分総和最小値の中から更に最小値を求め、ベクトル距離の最小値とベクトルの番号を SDRAM へ出力する. 最後に、汎用プロセッサを使用して、出力されたベクトル距離をしきい値処理することで、対応点を登録する. MX コアで実行している特徴ベクトル間の距離計算の計算量は、 $O(m^2)$ であり、汎用プロセッサで行っているしきい値処理の計算量は、 $O(m)$ である. 通常の利用状況下では、特徴点が多く存在するため、MX コアで実行される命令数より、汎用プロセッサで実行される命令数の方が少ない.

4. 提案手法の性能評価

MX コアを搭載した評価ボードへ、オリジナルの U-SURF と、提案手法を導入した U-SURF、及びベクトルマッチング処理をそれぞれポーティングした. 評価ボードの性能は、動作周波数 162 MHz、DMA 転送バス幅 64 bit であり、MX コア制御用のホスト CPU として汎用 RISC プロセッサである SH-2A を搭載している. SH-2A の動作周波数は、MX コアと同様の 162 MHz である. プログラムやデータは、DDR2-SDRAM に格納し、この SDRAM は MX コアと SH-2A が共有する構造をとっている. また、本評価実験に用いたプログラムでは、それぞれのフィルタが、外部の SDRAM のデータを MX コアに取り込

み、計算を行い、計算結果を外部の SDRAM に格納する構造をとった.

積分画像ベース局所特徴量抽出アルゴリズムの導入時、及び未導入時の実行時間を図 12 にそれぞれ示す. ここで示す実行時間は、いずれも積分画像生成の並列実装手法、及び特徴ベクトルマッチングの並列実装手法をそれぞれ導入した状態で測定した. ただし、検出特徴点数は 1000 点、入力画像サイズは 640×480 ピクセル、スケール数は 5 である. また、ベクトルマッチング処理時間は、あらかじめ片方の画像から抽出してある特徴量ベクトルとのマッチング処理に要した実行時間を示している. 図 12 に示すように、提案手法により局所特徴量抽出処理の実行時間が 1710 ms から 234 ms へ短縮され、約 7.3 倍の高速化を実現した.

次に、汎用プロセッサである SH-2A のみを用いたオリジナル U-SURF の実行時間と、MX コア併用時の提案手法を導入した U-SURF の実行時間をそれぞれ測定した. 各実行時間を図 13 に示す. ただし、検出特徴点数は 1083 点であり、ベクトルマッチング時に使用した比較対象の画像中の検出特徴点数は 835 点である. 図 13 に示すように、MX コアを使用した場合、各処理ごとに 36 倍から 54 倍の高速化を、処理

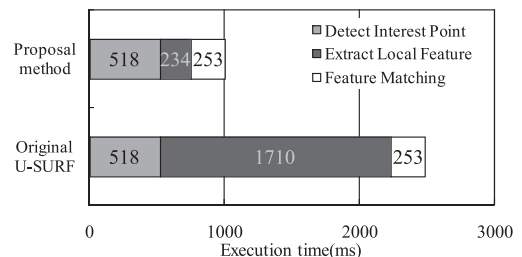


図 12 MX コア併用時の提案手法と従来手法の実行時間
Fig. 12 Execution time of the original method and the proposal method with MX core.

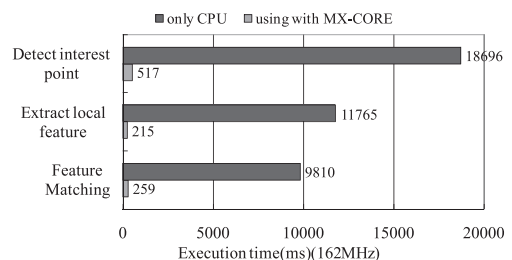


図 13 MX コア併用時と SH2A のみ使用時の提案手法の実行時間

Fig. 13 Execution time of proposal method using with the MX core and only the SH2A.

全体として約 40.6 倍の高速化を実現した。SH-2A のみで U-SURF を実行した場合、40270 ms の処理時間を要していたところを、MX コアを利用することで 992 ms まで短縮することに成功した。

5. 考 察

アルゴリズムの改善の有効性を示すために、各プロセッサの性能をもとに、並列プロセッサによる性能向上の見積値と実測値の比較を行った。SH2A は、2 ウェイスーパスカラであり、性能が 200 MHz で 360 MIPS と発表されているので、1 クロック当り 1.8 命令実行できる計算となる。一方、MX コアは一つの PE が 1 クロックで 2 ビットの演算のみ実行可能であるため、多ビットの演算にはビット数に比例し、数クロック必要となる。そのため、性能向上の比率は実際のプロセッサ数の比率より小さな値となる。

本提案手法では、特徴点の検出処理がすべて MX コアで実行されている。本実験では、 640×480 の画像に対して、1 ライン単位で並列処理を行っているため、実際には、640 プロセッサが同時に処理する実装となっている。積分計算には 19 ビット必要であるため、MX コアでは 1 クロック当り 64 ピクセルに対する演算が行われる計算となる。SH-2A はクロック当りの演算性能が 1.8 であるので、約 35.6 倍の演算性能の向上が見込まれている。それに対し、実測値はほぼ同じ値を示しており、CPU と MX コア間の同期のオーバーヘッドや、積分画像生成のようにプロセッサ数に対して対数オーダでしか改善の見られない処理等の存在を考慮したすると、本実装は効率良い並列実装ができている。特徴点抽出処理の中で SH-2A が主体的に行うしきい値処理の実行率の推定値は、特徴点抽出処理全体に対し、約 3% 以下であった。

特徴量の抽出処理については、オリジナルのアルゴリズムに忠実に MX コアに実装した場合、画像情報を MX コアに DMA 転送するとき、小さな局所画像のデータを多数回転送するため、DMA の初期化にかかるコストや、局所領域の 1 ラインごとのアドレス計算等により、画像の 1 ラインをそのまま転送する場合と比べ、約 7 倍の転送時間がかかっていることが実測値として得られた。オリジナルの U-SURF に沿った実装では、特徴点数に比例して各種フィルタ処理が必要であることに對して、本提案手法では、各種フィルタ処理が一定の計算量になるため、特徴点数が増えた場合、高い性能の改善が見込まれる [7]。本評価実験

で用いた 1000 点の特徴点が発出される場合は、オリジナルのアルゴリズムでは、1 点の特徴量を計算するとき、400 ピクセルに対して計算を行うので、一つの画像に対して 400,000 ピクセルを用いる [4]。しかし、画像の格納されているメモリとマイコンのバスの構造上、メモリから一つひとつのピクセルをアクセスする場合でも、利用しないピクセルも含めた連続転送の方が高速にアクセスできるため、実質的には 2 倍以上のピクセルデータの転送を行っている。それに対し、本提案手法では、全画素に対して 3 スケール分のフィルタ処理を適用する。そのため、 640×480 の画像の場合、3 スケール分の場合、合計で 1440 ラインのデータを扱うので、921,600 ピクセルについて計算を行う。本評価実験では、両者の扱うピクセル数がほぼ同じ転送量となるため、DMA 転送の効率の差がほぼ見える形となっている。

特徴量の抽出処理を SH-2A と MX コアの性能比について見た場合、特徴点の検出と同様に約 35.6 倍が期待されるが、オリジナルの U-SURF を MX コアに実装した結果と比較すると、実問題として転送時間の影響が大きく出ていることが分かる。本提案手法では、データ転送の改善を行ったことにより、性能比以上の改善が得られたことになる。更に、MX コアと SH-2A の実行時間の比率について見ると、画像サイズに依存する画像の全領域に対する処理は、すべて MX コアで実行しており、処理時間が約 103 ms であった。また、特徴点数に依存する処理のうち、MX コアの実行時間の推定値は約 56 ms で SH-2A の実行時間の推定値は約 97 ms となり全体の 37% を占めた。

特徴ベクトルのマッチング処理では、特徴点検出処理や特徴点の抽出処理と異なり、1024 プロセッサをすべて利用しているため、転送時間を考慮しないときの SH-2A と MX コアの単純な性能比を考えたとき、約 56.9 倍の性能向上が期待できることになる。これに対して、実際の特徴点ベクトル間の距離計算や最小値の探索は、ベクトル距離の総和計算が 64 次元の対数オーダでしか性能改善できず、この処理が多数回実行されるため、実際のところは、約 37.9 倍の改善となった。特徴ベクトルマッチング処理では、多くの処理が MX コアで実行しているため、SH-2A が主体的にベクトル距離のしきい値評価に関する処理について、SH-2A の実行率の推定値は約 8% となった。

これらより、全体を通して約 40 倍の高速化は、プロセッサの性能を十分引き出していることが示された。

また, SH-2A と MX コアが共有する SDRAM を利用しているため, 両者の比較ではデータ転送に関する問題は見られなかったが, MX コアの実行率を見たとき, MX コアでの計算が軽い Haar-Wavelet フィルタでは約 23%, BOX フィルタでも約 39% となり, MX コアでの計算が重い特徴点抽出のバイトニックソートでは約 87% となった. これらの結果より, 比較的単純な処理では, データ転送時間に大きな時間がとられていることが分かる. データ転送の効率化が実行時間の短縮に大きな意味をもつことが得られた.

6. む す び

組込み超並列処理プロセッサ MX コアを使用した, U-SURF アルゴリズムの並列化実装手法を提案した. 提案した並列処理手法の評価実験の結果より, MX コアを利用した場合, 同じ動作周波数の汎用 RISC プロセッサのみを使用する場合と比較して, 約 40.6 倍の処理速度を実現した. また, 提案した局所特徴量抽出手法を導入した U-SURF と, オリジナル U-SURF アルゴリズムの実行時間を比較したところ, 約 7.3 倍の高速化に成功し, 提案手法の有効性を示した.

本論文で述べた最適化手法は, 組込み超並列プロセッサを使用した場合, U-SURF に限らず, 他の多くの近傍領域処理を含む画像処理アルゴリズムにおいても効果的であると思われる. 超並列プロセッサに対する, 画像処理アルゴリズムの最適化の指針を以下に示す.

(1) アルゴリズム中に含まれる近傍領域処理(フィルタ処理等)を, 可能な限り画像全領域に対する並列実行する.

(2) 局所領域ごとに必要な必要最小限の演算を, 汎用プロセッサによって実行する.

このような手法により, 様々な画像認識処理の並列プロセッサへの実装が可能となり, これらの処理の組込み用途への更なる応用の拡大が期待される.

文 献

- [1] Y. Tsuduki, H. Fujiyoshi, and T. Kanade, "Mean shift-based point feature tracking using SIFT," J. Information Processing Society, vol.49, no.SIG 6, pp.35-45, 2008.
- [2] O. Kazumasa and N. Hamada, "Bags of Keypoints を用いた道路交通標識の認識," 第 15 回画像センシングシンポジウム (SSII09) 講演論文集, no.IS2-01, June 2009.
- [3] D. Lowe, "Distinctive image features from scale-invariant keypoints, cascade filtering approach," Int. J. Comput. Vis., vol.60, no.2, pp.91-110, Jan. 2004.
- [4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Computer Vision and Image Understanding (CVIU), vol.110, no.3, pp.346-359, 2008.
- [5] H. Noda, M. Nakajima, K. Dosaka, K. Nakata, M. Higashida, O. Yamamoto, K. Mizumoto, T. Tanizaki, T. Gyohten, Y. Okuno, H. Kondo, Y. Shimazu, K. Arimoto, K. Saito, and T. Shimizu, "The design and implementation of the massively parallel processor based on the matrix architecture," IEEE J. Solid-State Circuits, vol.42, no.1, pp.183-192, Jan. 2007.
- [6] P. Viola and M. Jones, "Robust real-time face detection," Int. J. Comput. Vis., vol.57, no.2, pp.137-154, 2004.
- [7] Y. Sato and T. Nagasaki, "MX コアによる高速画像処理の実現," 第 14 回画像センシングシンポジウム (SSII08) 講演論文集, no.IN1-23, June 2008.
- [8] Y. Sato and T. Nagasaki, "U-SURF アルゴリズムの組込み超並列プロセッサへの最適化実装手法の提案," 第 15 回画像センシングシンポジウム (SSII09) 講演論文集, no.IS1-23, June 2009.
- [9] Y. Sato, T. Sugimura, H. Noda, Y. Okuno, K. Arimoto, and T. Nagasaki, "Integral-image based implementation of U-SURF algorithm for embedded super parallel processor," 2009 International Symposium on Intelligent Signal Processing and Communication Systems, no.WA2-B-2 0014, Nov. 2009.

(平成 22 年 4 月 20 日受付, 8 月 8 日再受付)



佐藤 好紀

平 22 立はこだて未来大学大学院システム情報研究科了. 主として並列プロセッサを用いた画像処理に関する研究に従事.



野田 英行

1995 早大・理工・電子通信卒. 1997 同大学院修士課程了. 同年三菱電機 ULSI 開発研究所入社. システム LSI 向け組込みメモリ研究開発に従事. 2003 より(株)ルネサステクノロジに移り, SoC 組込み向けメモリ及び並列プロセッサ IP の開発に従事. 2007 広島大学大学院先端物質科学研究科博士課程了. 2010 年 4 月より現職. 工博.



杉村 武昭

2002 東北大・工・機械知能卒，2004 同
大大学院博士前期課程了，2007 同大大学
院工学研究科博士後期課程了，同年（株）
ルネサステクノロジ入社，SoC 組込み向け
並列プロセッサ IP のソフトウェア・ハー
ドウェアの開発に従事，2010 年 4 月より

現職，IEEE 会員，工博，



長崎 健 （正員）

平 4 北大・工・情報工卒，平 6 同大大学
院修士課程了，平 9 同大学院博士課程単位
取得退学，平 9（株）ピー・ユー・ジー入
社，平 12 よりはこだて未来大講師，平 18
同助教授，平 19 同准教授，コンピュータ
ビジョンの研究に従事，情報処理学会会員，