社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－ＳＬＤＭ－121（８）
2005／10／20

超低演算量な動き検出アルゴリズムと専用プロセッサへの実装

平塚 誠一郎† 後藤 敏‡ 池永 剛‡

†福岡県産業・科学技術振興財団 〒814-0001 福岡市早良区百道浜 3-8-33

早稲田大学理工学総合研究センター九州研究所 〒808-0135 北九州市若松区ひびき 2-2

‡早稲田大学大学院情報生産システム研究科 〒808-0135 北九州市若松区ひびき 2-7

E-mail: †hiratsuka@kyushu.rise.waseda.ac.jp, ‡{goto, ikenaga}@waseda.jp

あらまし 本稿では，動画圧縮に必要な動き検出について超低演算量なアルゴリズムと，このアルゴリズムに最適な専用プロセッサの回路設計とソフトウエア実装について述べる．動き検出アルゴリズムはブロック内の画素レベルの変動状況に応じて計算対象になる画素の間引き率を適応的に変える手法を提案し，その他の手法と組み合わせて，全探索に比べて 1000 分の 1 の以下計算量となるアルゴリズムを開発した．また，このアルゴリズムを効率よく実行するために付加演算器をライトバックステージに追加した専用プロセッサを提案し，プロセッサの回路を FPGA で設計し，ソフトウエアを実装した．本プロセッサは従来の動き検出専用プロセッサに比べて 4 分の 1 以下回路規模で実現できるので，コンパクトな動画圧縮システムの開発に寄与が期待できる．

キーワード 動き検出，全探索，SAD，専用プロセッサ

An Ultra-low Complexity Motion Estimation Algorithm and its Implementation of Specific Processor

Seiichiro HIRATSUKA† Satoshi GOTO‡ and Takeshi Ikenaga‡

† Fukuoka Industry, Science & Technology Foundation 3-8-33 Momochihama, Sawara-ku, Fukuoka, 814-0001 Japan

Kyushu Laboratory, Advanced Research Institute for Science & Engineering, Waseda University 2-2 Hibikino, Wakamatsu-ku, Kitakyushu, 808-0135 Japan

‡ Graduate School of Information, Production and Systems, Waseda University 2-7 Hibikino, Wakamatsu-ku, Kitakyushu, 808-0135 Japan

E-mail: †hiratsuka@kyushu.rise.waseda.ac.jp, ‡{goto, ikenaga}@waseda.jp

Abstract Motion estimation (ME) requires huge computation complexity. Many motion estimation algorithms have been proposed to reduce its complexity. But they are still insufficient for embedded video coding systems. So we propose an ultra-low complexity ME algorithm with adaptive block sub-sampling and several techniques. The simulation results show that proposed algorithm has about 1,000 times the speedup than full search (FS) maintaining high image quality. And we also propose the implementation of the application specific instruction-set processor (ASIP). It is based on a reduced instruction set computer (RISC) with sum of absolute difference (SAD) operation circuit. Our ME ASIP is implemented on FPGA. It is required about 3,313 logic elements (LEs) and its hardware scale is about quarter of the previous ME ASIP. This ME ASIP will make a significant contribution to the development of compact video coding systems.

Keyword motion estimation, full search, SAD, application specific instruction-set processor

1. はじめに

動き検出はビデオシーケンスの時間的な冗長性を除去する手法であり，動画圧縮システムにおいて重要な役割を果たしている．動き検出を定義どおりに実行する全探索法は莫大な計算量を必要とするので，計算量を削減した多くの動き検出アルゴリズムが提案されてきた．

動き検出の計算量を削減する手法としていくつか方式が知られている．1 つは探索点を削減する方法[1]-[6]，もう 1 つはマクロブロックの計算画素数を削減する方法[7][8]である．さらに，別の観点から計算量を削減する手法[9]-[12]も提案されている．これらの方法は計算量が全探索法に比べて 1/50 から 1/200 程度でありソフトウエアでの実装を考えた場合，十分なレベルではなかった．

本研究ではまず，マクロブロックの計算画素数を削

減する方法を改善した適応的サンプリング手法を提案する．この方法により固定パターンの間引き手法より計算量を40%削減できることを示す．

また，我々はこの適応的サンプリング手法と動き，検出法動きのないブロックの事前選別，局所的な最小値から脱出するための再探索法，見込みのない探索点での計算を中断する方法などさまざまな手法を組み合わせた方法を考案し，コンピュータ・シミュレーションにより全探索法に比べて約1/1,000の計算量で実現できることを検証する．

さらに，我々は提案する動き検出アルゴリズムをシステムに実装について検討した．動き検出の実装方法として，ASICの設計と汎用プロセッサへの移植がよく行われてきた．しかしながら，ASICには仕様変更やバージョンアップに対応することが難しい．また，汎用プロセッサは効率的に演算処理が行えないことから，我々は専用プロセッサへ実装することにした．

動き検出専用プロセッサは小さなRISCプロセッサをベースに，テンプレート，サーチエリア用メモリ，SAD演算器などを追加して構成し，FPGAで回路を設計した．また，専用プロセッサのソフトウエアをアセンブラで開発し，FPGA上で検証する．

2. 動き検出と計算量削減アルゴリズム

2.1. 動き検出の原理

動き検出は動画データの連続するフレームについて時間方向の冗長性を除去する．動き検出の最も一般的な方法はブロックマッチングである．ブロックマッチングの基本的な演算は下記のとおりである．

1)処理フレームにテンプレートブロックIcを設定
2)参照フレームにサーチウィンドウIrを設定
3)次式のSAD(差分絶対値の総和)を計算

$$SAD(k,l)=\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|Ic(i,j)-Ir(i+k,j+l)\right| \qquad (1)$$

ここで，MNはブロックの縦，横の画素数である．
4) SAD(k',l')がSADの最小値である場合，(k',l')を動きベクトルとする．

全探索法はすべての探索点のSAD値を計算する方法であるため，常に最適な動きベクトルを検出できる．しかしながら，全探索法は計算量が多いという問題があり，次に述べる計算量削減アルゴリズムがよく使われている．

2.2. 計算量削減アルゴリズム

動き検出の計算量を削減する手法としていくつか方式が知られている．探索点を削減する方法，マクロブロックの計算画素数を削減する方法がある．探索点を削減する方法として，3段探索法(TSS)[1]，対数探索[2]および4段探索法[3]は探索の間隔を小さくしていく疎から密のアプローチであるが，計算量の低下がそれほどでもない．また，菱形探索法[4]，ブロック勾配探索法(BBGDS)[5]，1次元勾配降下探索[6]は勾配に基づく方式であり，局所的な最小値に陥るという欠点がある．

テンプレートの計算画素数を削減する方法として，画素を均等に間引くBierlingの方法があるが，動き検出の精度が大きく低下する．これを改善したChanの方法は画像領域により間引きのパターンを適応的に変えるものであるが，画像領域判定や間引きのパターン決定に計算量が多く必要である．

上述した以外の計算量削減として，様々な手法が提案されている．1つはシュミュレーテッド・アニーリング法であり，局所的な最小値から脱出できる効果的な方法であるが，アニーリングに計算量がかかってしまう．また，3段探索法とブロックベース勾配探索法を組み合わせた探索法が提案されている[10][11]が，計算量と動き検出精度の両立が困難である．

3. 動き検出の提案アルゴリズム

3.1. テンプレートの適応的な画素間引き

2.2で述べた計算量削減方法のうち，まずテンプレートの計算画素数の適応的な間引きについて検討した．我々はすでにChanの方法よりも画像領域判定の計算量が少ない方法を開発している[12]．図1(a)(b)は我々の先行技術をしたもので，(a)はオリジナル動画フレーム（"Susie" QCIF 17番目のフレーム）であり，(b)は間引きのパターンを示したものである．間引きのパターンは非エッジ部では4:1に，エッジ部では2:1とし，エッジ判定式は以下のようになる．

$$E_{ij}=\left|T_{i-1,j}-T_{i+1,j}\right|+\left|T_{i,j-1}-T_{i,j+1}\right| \qquad (2)$$

(i, j = 1, 3, ……, 15)

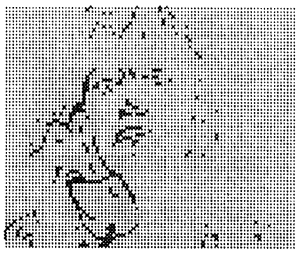
ここで，Tijはテンプレートの内の画素レベルである．もし，Eの値が閾値よりも大きい場合にはエッジ部とし，以下の場合は非エッジ部とする．この方法はChanの方法に比べて計算量が半分以下であり，間引きパターンの偏りも少ない．

図1(c)はさらに画像領域判定の計算量を削減したもの間引きパターンで，8画素×8画素のサブブロックにおいて，次の判別式を1回計算する．

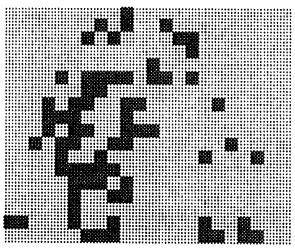
$$F=\sum_{i=0}^{1}\sum_{j=0}^{1}\left|T_{4i,4j+1}-T_{4i+2,4j+1}\right|+\left|T_{4i+1,4j}-T_{4i+1,4j+2}\right| \qquad (3)$$

この方法は我々の先行技術に比べて差分絶対値演算が1/2，判定回数が1/16の計算量で実現できる．

(a) オリジナル ("Susie" 17th



(b) 画素適応間引きパターン



(c) 提案手法

図1 テンプレートの間引きパターン

3.2. 統合アルゴリズムの全体フロー

さらに，我々は3.1で提案したテンプレートの適応的な間引きに加え，さまざまな手法を組み合わせて動き検出全体として計算量の大幅な削減を図った．図2はこの統合アルゴリズムのフローを示したものである．

最初に，本格的に動きベクトルを検出する前に，動きのないマクロブロックのスクリーニングを行う．テンプレートの画素を1:16に間引いてSAD(0,0)を計算する．もし，SAD(0,0)が所定の閾値TH1よりも小さい場合には動きベクトルを(0,0)として計算を打ち切る．

次に，3.1で提案したテンプレートの適応的な間引きを実施する．判別式Fの閾値TH2は経験的に100に設定した．そして，このテンプレートで用いて2.2で述べたBBGDS法により動きベクトルの探索を行う．もし，SADの最小値が所定の閾値TH3よりも小さい場合には動きベクトルを設定として計算を打ち切る．

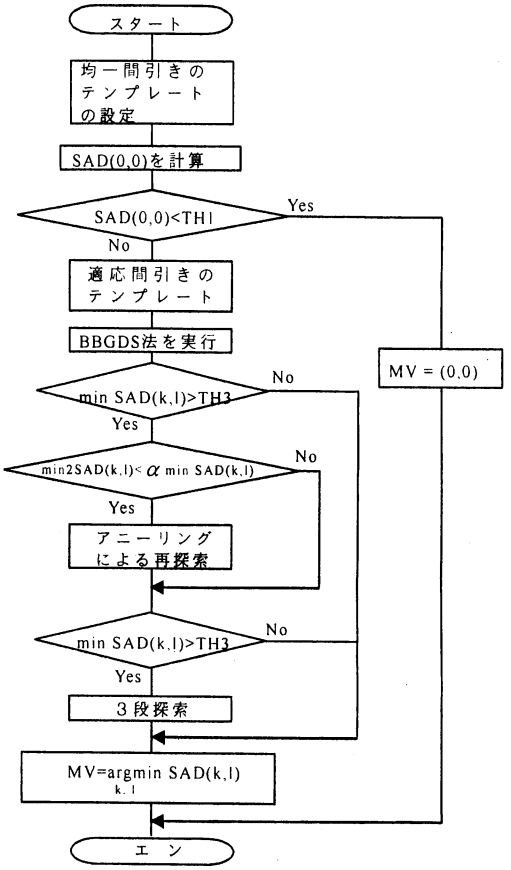


図2 統合アルゴリズムのフローチャート

トして現行の最小値の点をはずしてBBGDS法を行う．もし，SADの最小値が所定の閾値TH3よりも小さい場合には動きベクトルを設定として計算を打ち切る．

4番目に，3段探索法による再探索を行う．これまでのSADの最小値となるベクトルを動きベクトルと設定し，動き検出の処理を完了する．

上記のテクニックに加えて，SSDA(残差逐次検定)法を採用して更なる計算量の削減を図った．この方法は計算中のSAD値が現行の最小値を更新する可能性がほとんどなくなった時点でその探索点での計算を打ち切る方式である．

最初のスクリーニング，テンプレートの適応的間引きとSSDA法は動き検出の性能低下を最小限にして大幅に計算量を削減するためにあり，アニーリングや3段探索は最小限の計算量の追加で局所最小値からの脱出を図って動き検出の性能を向上させる．

3.3. シミュレーション実験結果

我々の提案する動き検出アルゴリズムについてコンピュータ・シミュレーションによる性能評価実験について述べる．我々は全探索法(FS)，3段探索法(TSS)，ブロック勾配探索法(BBGDS)，我々の先行技術(Pixel Edge Detection)および今回の提案について比較検討した．QSIF サイズの 3 つのビデオシーケンス " Susie"，" Foreman"，"Miss America" を用い動き補償後のフレーム画像の品質をオリジナルフレームとの差を PSNR で評価した．また，計算量を示す尺度として，SAD 演算や EDGE パラメータの計算に使われる差分絶対値(AD)の回数を当てた．

表 1 は各ビデオシーケンスについて探索法ごとに AD 回数と PSNR をシミュレーションより求めた結果を示す．提案方法は AD 回数が全探索法に比べて 1/600～1/1800 であり，" Susie"，"Miss America" の PSNR は遜色ない．" Foreman" の PSNR は全探索法よりも劣るが，3 段探索や BBGDS に比べてはるかによい．

表 1　動き検出の計算量と画質評価

探索アルゴリズム	Susie		Foreman		Miss America	
	AD counts	PSNR	AD counts	PSNR	AD counts	PSNR
FS	226819	35.30dB	226819	28.92dB	226819	40.76dB
TSS	7273	34.97dB	7308	28.06dB	7252	40.68dB
BBGDS	2438	35.23dB	3302	28.05dB	2235	40.76dB
画素適応間引き[12]	290	35.28dB	377	28.51dB	172	40.71dB
統合アルゴリズム（提案法）	232	35.28dB	359	28.50dB	123	40.72dB

4. 動き検出専用プロセッサ

4.1. 専用プロセッサのアーキテクチャ

我々は提案する動き検出アルゴリズムの実装について検討した．動き検出の実装方法として，ASIC の設計と汎用プロセッサへの移植がよく行われてきた．しかしながら，ASIC には仕様変更やバージョンアップに対応することが難しい．また，汎用プロセッサは効率的に演算処理が行えないことから，我々は専用プロセッサへ実装することにした．

専用プロセッサにはさまざまなタイプがある[14]-[17]が，我々のアルゴリズムは計算量が少ないので，16 ビットの縮小命令セットコンピュータ(RISC)をベースにする．図 3 はベースプロセッサの構成を示したものである．これは 5 つのステージ(IF, ID, EX, ME, WB)からなっており，PC, IM, PR, RF, ALU, DM などからなっている．

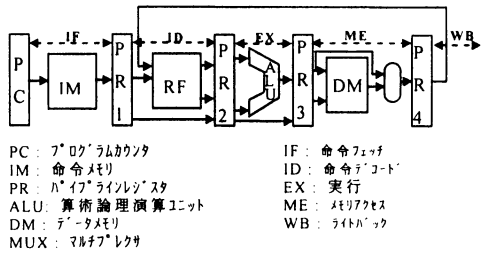


図 3　16 ビット RISC プロセッサ

動き検出専用プロセッサでは SAD 演算や EDGE パラメータの計算を効率よく実行するために，図 4 に示したように，EX ステージに加算器，ME ステージにテンプレートメモリ(TM)とサーチエリアメモリ(SM)，WB ステージに差分絶対値演算器(SAD)，累算レジスタ(ACC)を付加した．

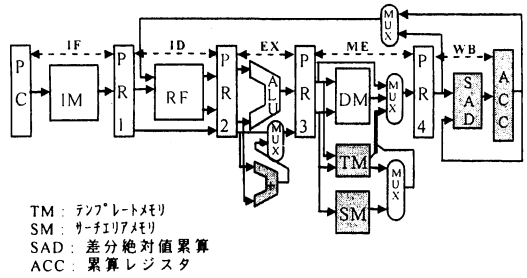


図 4　動き検出専用プロセッサ

図 5 は SAD 命令実行時の TM, SM のアクセスを示したもので，PR3 からアドレスとして TM 用に ADR1 が，SM 用に ADR2 が出力し，TM からはテンプレート，SM からはサーチエリアの画素データが出力される．

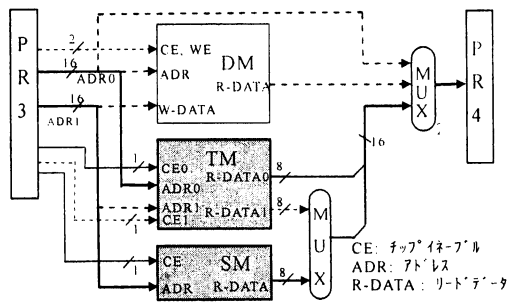


図 5　SAD 演算時のメモリアクセス

図６は，SAD演算やEDGEパラメータの計算するための差分絶対値累算器(SAD)と累算レジスタ(ACC)の構成を示したものである．PR4 からのテンプレートとサーチエリアの画素を差分絶対値を計算して，ACC に加算する．

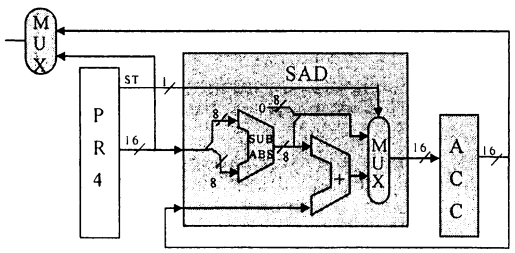


図６　SAD ユニット

表２は動き検出プロセッサの仕様を示したものである．TM, SM, SAD を付加した 16bitRISC で構成した．

表２　動き検出専用プロセッサの仕様

項目	内容
ﾊﾞｽｱｰｷﾃｸﾁｬｰ	ﾊｰﾊﾞｰﾄﾞ・ｱｰｷﾃｸﾁｬｰ
命令ﾒﾓﾘﾊﾞｽ	
ｱﾄﾞﾚｽﾊﾞｽ	12 bits
ﾃﾞｰﾀﾊﾞｽ	16 bits
ﾃﾞｰﾀﾒﾓﾘﾊﾞｽ	(通常)　(SAD, EDGE)
ｱﾄﾞﾚｽﾊﾞｽ	16 bits×1　or　16bit×2
ﾗｲﾄﾃﾞｰﾀﾊﾞｽ	16 bits×1　or　none
ﾘｰﾄﾞﾃﾞｰﾀﾊﾞｽ	16 bits×1　or　8bits×2
汎用ﾚｼﾞｽﾀ	16bits×16
読込ﾎﾟｰﾄ	2
書込ﾎﾟｰﾄ	1
ﾊﾟｲﾌﾟﾗｲﾝ	5ｽﾃｰｼﾞ
ALU	
入力	16bits×2
出力	16bits
演算	加算，減算，ｼﾌﾄ演算 AND, OR, NOT など
命令メモリ(IM)	16bits×4096 words
ﾃﾞｰﾀﾒﾓﾘ(DM)	
RAM	16bits×256 words
ROM	16bits×512 words
ﾃﾝﾌﾟﾚｰﾄﾒﾓﾘ(TM)	8bits×256 words (2 ports)
ｻｰﾁｴﾘｱﾒﾓﾘ(SM)	8bits×2304 words
SADﾕﾆｯﾄ	
入力	8bits×2pixels, 16bits累算値
出力	16bits累算値
その他の機能	Jump, Branch, Subroutine Call CPU Halt など

4.2. 回路設計

我々の動き検出専用プロセッサをハードウエア記述言語Verilogで記述し，Altera社のFPGAデバイスを搭載したボード(図7)を用いて設計・検証を行った．FPGAでの設計結果を表3に示す．回路規模を表すLogic Element数は3,313 LEsであり，Petersらの設計した動き検出専用プロセッサ[17](7,530slices≒15,140LEs)の1/4回路規模で実現できている．

図７　FPGA ボード

表3　FPGAでの設計結果

項目	内容
FPGAﾃﾞﾊﾞｲｽ	Statrix EP1S80F1020C7
Logic Element	3,313 LEs
RAM	98,304bits
最大動作周波数	58.03 MHz

4.3. ソフトウエアの実装

我々が3.2で提案した動き検出の統合アルゴリズムを専用プロセッサに移植した．ソフトウエア開発ツールがないため，Verilogプログラムで命令メモリROMの記述に" `define " を用いてアセンブラ風に記述し，FPGA上で動作検証した．また，実際のビデオシーケンスはデータ量が多くFPGAに載せることができないので，差分絶対値(AD)の演算回数とサイクル数を算出するプログラムを作成し，ビデオシーケンスのAD回数とサイクル数を見積もった．表4に見積もった結果を示す．

表４　AD の演算回数とサイクル数

Video sequence	Susie	Foreman	Miss America
平均			
AD / MB	232	359	123
Cycle / MB	341	516	186
最悪フレーム			
AD / MB	706	1213	286
Cycle / MB	995	1686	420

5. おわりに

本研究ではブロック単位で適応的にテンプレートの画素数を間引き方式と，この方式にいくつかの方式を組み合わせた動き検出アルゴリズムを考案し、全探索の1/1,000の計算量である動き検出アルゴリズム提案した．さらに，このアルゴリズムを効率よく実施できる専用プロセッサを考案し，FPGAで回路設計を行い，ソフトウエアを実装して動き検出の動作を検証した．動き検出専用プロセッサの回路規模は従来の1/4以下で実現できている．

このプロセッサは回路規模が少なく，低い周波数で動き検出の動作が見込めるため，ASICにより LSI開発し，低消費電力を確認していきたい．

謝辞 この研究は，文部科学省の北九州地域と福岡地域の広域知的クラスター創成事業の支援による．

文　献

[1] T. Koga, K. Iimura, A. Hirano, Y. Iijima and T. Ishiguro "Motion-compensated interframe coding for Video Conferencing," Proc. NTC81, pp. C9.6.1-9.6.5, 1981..

[2] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. Communication, vol.29, pages 1799-1808, Dec. 1998.

[3] L.M. Po and W.C. Ma "A novel four step search algorithm for fast block motion estimation". IEEE Trans. Circuit and Systems for video Technology, vol.6, pages 313-317, 1994.

[4] J. Y. Tham, S. Ranganath, M. Ranganath and A. Kassim "A novel unrestricted center-biased diamond search algorithm for block motion estimation". IEEE Trans. Circuit and Systems for video Technology, vol.8, pages 369-377, 1998.

[5] L. K. Liu and E. Feig "A block-based gradient descent search algorithm for block motion estimation in video coding". IEEE Trans. Circuit and Systems for video Technology, vol.6, pages 419-422, 1996

[6] O. T. C. Chen, "Motion estimation using a one-dimensional gradient descent search", IEEE Trans. Circuits Systems Video Technology, vol. 10, pp.608-616, June 2000.

[7] Y. L. Chan, and W.C. Siu, "New adaptive pixel decimation for block motion estimation," IEEE Trans. Circuits Systems Video Technology, vol. 6, pp.113-118, 1996.

[8] C. N. Wang, S. W. Yang, C. M. Liu, and T. Chiang, "A hierarchical N-Queen decimation lattice and hardware architecture for motion estimation," IEEE Trans. Circuits Systems Video Technology, vol.14, pp.429-440, 2004.

[9] M. C. Shie, W. H. Fang, K. J. Hung and F. Lai, "Fast, robust Block Motion Estimation Using Simulated Annealing,". IECIE Trans. Fundamentals, vol.E83-A, No.1, pages 121-127, Jan. 2000.

[10] R. Li, B. Zeng and M. L. Liou "A new three-step search algorithm for block motion estimation" IEEE Trans. Circuit and Systems for video Technology, vol.4, pages 438-442, Apr. 1994.

[11] J. Choi, N. Togawa, M. Yanagisawa and T. Ohtsuki, "An algorithm and a flexible architecture for fast block-matching motion estimation," IECIE Trans. Fundamentals, vol. E83-A, no. 12, pp.2603-2611, 2002

[12] S. Hiratsuka, S. Goto and T. Ikenaga, "A locally adaptive subsampling algorithm for software based motion estimation," Proc. of ISCAS 2005, pp.2891-2894, 2005

[13] D. I. Barnea, and H. F. Silverman, "A class of algorithms for fast digital image registration," IEEE Trans. Computer, vol. C-21, no. 2, pp.179-186, 1972.

[14] H. Harasaki et al., "A single-board video signal processor module employing newly developed LSI devises," IEEE Journal on selected areas in communications, vol. 6, no. 3, pages 513-519, 1988.

[15] A. Hanami et al., "A 165-GOPS motion estimation processor with adaptive dual-array architecture for high quality video-encoding application," Proc. of Custom Integrated Conference, pp.9.1.1-9.1.4, 1998.

[16] A. Beric et al., "A 27mW 1.1mm^2 motion estimator for picture-pate up-converter," Proc. of VLSID 2004, pp.1083-1088, 2004.

[17] H. Peters et al., "Application specific Instruction-set processor template for motion estimation in video applications," IEEE Trans. Circuit and Systems, vol. 15, no. 4, pages 508-527, 2005.