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

2003－ＳＬＤＭ－111　(7)
2003／10／23

高速改良中断法動きベクトル検出アルゴリズムと低消費電力 CMOS 動きベクトル検出回路の開発

渡邊友樹　　榎本忠儀

中央大学 大学院 理工学研究科 情報工学専攻

〒112-8551　東京都文京区春日 1-13-27

あらまし　MPEG-4 対応の動画像符号化 LSI に搭載することを目的に、高速改良中断法（HS-IBOS）動きベクトル検出（ME）アルゴリズムおよび ME 用累算形差分絶対値和回路（累算形 ADA）を開発した．HS-IBOS は、ブロックマッチング（BM）過程で、差分絶対値和（D_a）の減少率が 0 となる回数（換言すると、D_a 値が更新されない回数）が、あらかじめ与えられた回数（d）、に達した時、その時点のマクロブロック（MB）を最適な MB として、BM を終了させる、ME アルゴリズムである．テスト画像 “Carphone”（QCIF、64 kbps、15 fps、探索領域サイズ p = 10 画素）を用い、d を 64 に設定した場合、FS 並の画質を維持したまま、ME 速度を FS、従来の中断法のそれぞれ約 5 倍、約 2 倍に高速化できた．また、HS-IBOS を適用すると 0.13-μm CMOS 累算形 ADA の消費電力は 27.9 μW（0.83 V、220 MHz）となり、FS、従来の中断法のそれぞれ約 1/5、約 1/2.5 に低減できた．

キーワード　MPEG-4、動きベクトル検出、CMOS、差分絶対値和回路、低消費電力

High-Speed Improved Breaking-off-Search Motion Estimation Algorithm and Low-Power CMOS Absolute Difference Accumulator for MPEG-4 Motion Estimation

Tomoki Watanabe and Tadayoshi Enomoto

Graduate School of Science & Engineering, Chuo University

1-13-27 Kasuga, Bunkyo-ku, Tokyo-to 112-8551, Japan

Abstract　To reduce power dissipation of an absolute difference accumulator (ADA) for MPEG-4 motion estimation (ME), a fast ME algorithm called a “high-speed improved breaking-off-search (HS-IBOS)” algorithm was developed. HS-IBOS can improve processing speed of the full-search (FS) method by a factor of 5 to 15, depending on picture types, while maintaining visual quality of the full-search method. At clock frequency of 220 MHz and supply voltage of 0.83 V the power dissipation of a 0.13-μm CMOS ADA using HS-IBOS and a gated-clock pulse was reduced to 27.9 μW which was about 1/5 that of the conventional ADA using FS.

Key words　MPEG-4, motion estimation algorithm, CMOS, absolute difference accumulator

1. はじめに

携帯機器に搭載する動画像符号化 LSI の最重要課題は低消費電力化と小形化である．特に、動き補償のための動きベクトル検出（Motion Estimation；ME）回路は、処理量が極めて大きいため、消費電力ならびに回路規模が非常に大きかった．我々はこれまでに全探索動きベクトル検出法（full-search；FS）に代わる高速 ME アルゴリズム、中断法（Breaking-Off-Search；BOS）[1、2]ならびに 2 ステップ中断法[3]、を開発し、ME 処理を大幅に削減してきた．この結果、ME 回路の主要回路である差分絶対値和回路（Absolute Difference accumulator；ADA）を大規模なアレイ形から小形な累算形へ置換えることができた[4]．本論文では、動きベクトルをさらに高速検出する高速改良中断法（High Speed Improved BOS；HS-IBOS）とその特性を述べる．HS-IBOS は FS 並の高画質を維持し、FS、BOS に比べ、それぞれ 5 倍、2 倍以上の速度で動きベクトルを検出することができる．本 ME アルゴリズムを適用した累算形 ADA の消費電力は 27.9 μW（電源電圧= 0.83 V、クロック周波数 = 220 MHz）であった．これは、FS を適用した累算形 ADA の約 1/5 である．

2. 従来の動きベクトル検出アルゴリズム

2.1. 全探索法と中断法

全探索法（FS）は探索領域内の全てのマクロブロック（MB）に対し差分絶対値和（D_a）を求める動きベクトル検出（ME）アルゴリズムである．FS は探索領域内の全ての MB に対しブロックマッチング（BM）を行うため、高精度に動きベクトルを検出できる．

処理量を大幅に削減できる ME アルゴリズムの一つに中断法（BOS）がある．BOS は BM の途中で D_a があらかじめ設定された閾値（D_t）を下回り（第 1 の中断条件）、かつ、D_a の減少率が 0 となる（第 2 の中断条件）MB を検出した時点で、ME 処理を停止し、以降、未探索領域の BM を省略するアルゴリズムである[1、2]．D_t の設定方法として、前フレームの平均最小 D_a を現フレームの D_t とする、直前の（左隣の）MB の最小 D_a を現 MB の D_t とする、等がある[5]．

2.2. 全探索法と中断法の問題点

“Carphone”と呼ばれるテスト画像を用いて、画像解析を行う．D_t は前フレームの平均最小 D_a とする．画像サイズは QCIF(176 画素×144 ライン)、YUV は 4:2:0、フレーム数（N_f）は 382（I-VOP：1 フレーム、P-VOP：381 フレーム）である．1 画素当たりのデータ数は 8 ビット（b）である．以下では、P-VOP 381 フレームに対して、MPEG-4 に準拠したソフトウェアエンコーダを用いて解析する．データレート（R_d）は 64 kbps、フレームレート（R_f）は 15 fps、に設定する．p を画素数とすると、探索領域サイズは $2p$ 画素×$2p$ ラインで与えられる．以下の解析では p を 10 画素とした．なお、階層探索、多数画素精度探索、半画素精度探索、動きベクトル情報による探索領域のシフト、等は併用しない．また、BOS では渦巻き探索を用いている．

図 2.1、2.2 にそれぞれ FS、BOS の解析結果を示す．いずれの図も (a)は 8 フレーム目の動き補償画像、(b)は灰色の濃淡で表現した MB 毎の D_a の大きさ（D_a が大きい MB は 黒、D_a が小さい MB は白）、(c)は MB の BM 数（N_b）の推移、である．両 ME アルゴリズムの動き補償画像、D_a の大きさに大きな差は見られないが、N_b は大分異なっている．

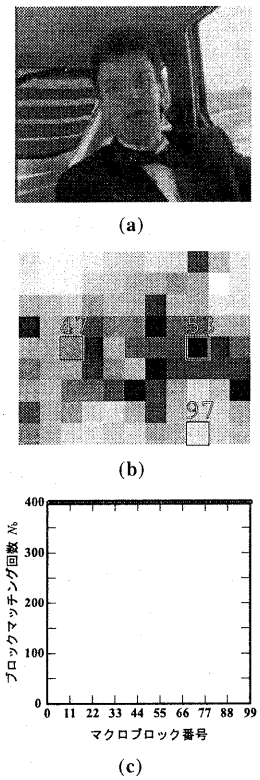


図 2.1 FS を用いた画像解析．(a)動き補償画像．(b) D_a の大きさ．(c) N_b．(“Carphone”、15 fps、64 kbps、15 fps、p = 10 画素、8 フレーム目)

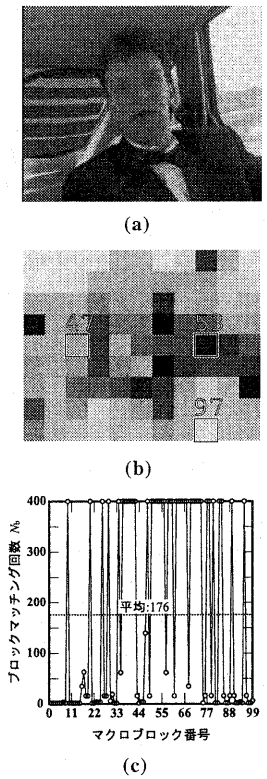


図 2.2 BOS を用いた画像解析．(a) 動き補償画像．(b) D_a の大きさ．(c) N_b．(“Carphone”、15 fps、64 kbps、15 fps、p = 10 画素、8 フレーム目)

FS では、全 MB の N_b は常に $2p \times 2p$ で、大きい．つまり、ここでは 400 である．一方、BOS では、N_b が 400 かあるいはほぼ 0 の MB に分けられ、中間の MB が少ない．

8 フレーム目の 3 個の MB に対して、BOS を用いて BM 処理を施し、D_a の推移を観測した．これを図 2.3 に示す．D_t は 986 である．MB No.47（灰）の MB では D_a が徐々に減少して、N_b が 140 回で、第 1、第 2 の中断条件を満足して、処理を停止する．この場合、同一条件で全探索領域を BM して得られる最小の D_a（=930）に近い値、954 が得られている．

MB No. 53（黒）の MB の場合、17 回目の BM で探索領域内で最小の D_a（=2190）を持つ MB に到達し、第 2 の中断条件を満足している．しかし、第 1 の中断条件を満足できないため、18 回目以後、全 MB に対して無駄な BM を実行している．

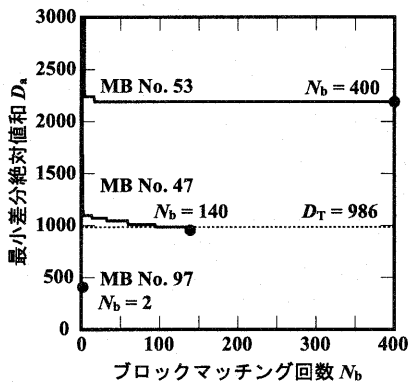


図 2.3 BOS を用いた BM 処理の推移（最小差分絶対値和の推移）.（"Carphone"、64 kbps、15 fps、p = 10 画素、8 フレーム目）

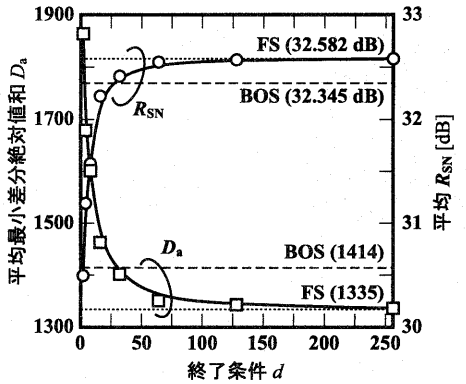


図 3.1　HS-IBOS による画質解析結果（D_a、R_{SN} と d の関係）.（"Carphone"、64 kbps、15 fps、p =10 画素）

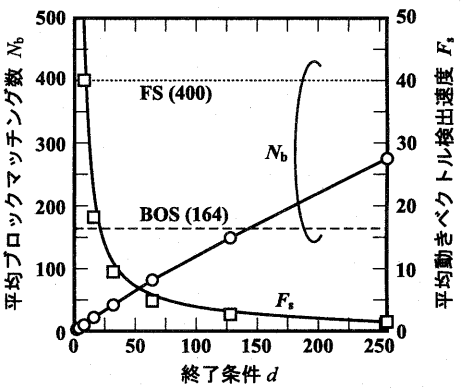


図 3.2 HS-IBOS による処理速度解析結果（N_b、F_s と d の関係）.（"Carphone"、64 kbps、15 fps、p =10 画素）

MB No. 97（白）の MB の場合、1 回目の BM で D_a が 407 となり、第 1 の中断条件を満足する．2 回目の BM で第 2 の中断条件をも満足して、BM を終了する．しかし、実際に BM を継続すると、D_a が 407 となる BM が連続して 4 回続き、探索領域内で最小となる MB は 5 回目に生じる．このように、中断法では最適 MB（D_a が最小の MB）ではない MB を検出する場合がある．

このように BOS は D_t が大きいと、簡単に第 1 の中断条件（$D_a < D_t$）を満足して、早急に探索を停止したり、あるいは D_t が小さいと、この条件を満足できずに全探索領域で BM を実行してしまう．つまり、全ての MB に対して最適な D_t を設定することは難しい．また、第 2 の中断条件（D_a の減少率=0）も、適切に機能しない場合がある．

3. 高速動きベクトル検出アルゴリズム

3.1. 高速改良中断法

ME 速度（FS を 1 とした相対的な速度．以下、F_s）を高速化するため、BOS の第 1 の中断条件（$D_a < D_t$ で、BM を停止）は割愛する．また、D_a を FS 並に小さくし、画質を向上させるため、BOS の第 2 の中断条件（D_a の減少率が 0 で、BM を停止）をさらに厳しくする．つまり、「最小 D_a の減少率が 0 となる BM があらかじめ設定された任意の回数（d）だけ継続した時にのみ、最適な MB を検出したとして、BM を終了させる」ことにする．この条件を満たした ME アルゴリズムを高速改良中断法（HS-IBOS）と呼ぶ．

3.2. 画像解析

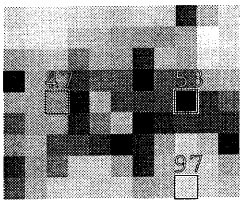
図 3.1 に HS-IBOS による画質解析結果｛D_a、ピーク S/N 比（R_{SN}）と d の関係｝を示す．d を 1 から増加させて行くと、HS-IBOS で得た"Carphone"の平均最小 D_a（以下、HS-IBOS の D_a）は急速に減少する．d が 32 以上で BOS の D_a（=1,414）を下回り、d が 64 以上で FS の D_a（=1,334）とほぼ等しくなる．同様に、HS-IBOS で得た"Carphone"の平均 R_{SN}（以下、HS-IBOS の R_{SN}）は d が 32 以上で BOS の R_{SN}（=32.345 dB）を上回り、d が 64 以上で FS の R_{SN}（=32.852 dB）とほぼ等しくなる．

HS-IBOS による処理速度解析結果（N_b、F_s と d の関係）を図 3.2 に示す．N_b は d に比例して増加し、これに伴い F_s は減少する．HS-IBOS の平均 N_b は、d に関係なく FS より小さく、d が 128 以下で BOS の N_b（=164）より小さくなる．つまり、HS-IBOS は BOS より高速であると言える．なお、BOS と同様に、HS-IBOS も渦巻き探索を用いている．

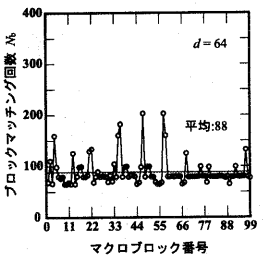
図 3.3 に HS-IBOS の解析結果を示す．(a)は 8 フレーム目の動き補償画像、(b) は灰色の濃淡で表現した MB 毎の D_a の大きさ、(c) は MB の BM 数（N_b）の推移、である．なお、d は 64 である．HS-IBOS の動き補償画像および D_a は FS の動き補償画像および D_a（図 2.1）と比べ、差はほとんど見えない．

BOS では N_b が 400 かあるいはほぼ 0 の MB に 2 分され、中間の MB が少なかった．これに対し、HS-IBOS では N_b が 400 の MB は 1 個も無く、設定した d の値（64）付近に集中している．図 3.3(c) と同様、図 3.4(a)、(b)、

(a)

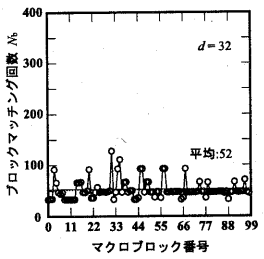


(b)

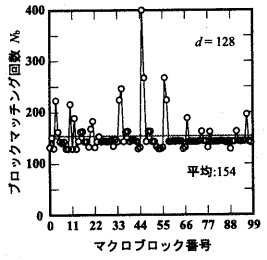


(c)

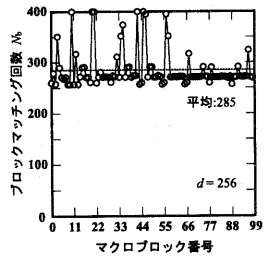
図 3.3 HS-IBOS を用いた画像解析．(a) 動き補償画像．(b) D_a の大きさ．(c) N_b．(“Carphone”、64 kbps、15 fps、p = 10 画素、8 フレーム目)



(a)



(b)



(c)

図 3.4 HS-IBOS を用いた N_b の推移．(a) d=32．(b) d=126．(c) d=256．(“Carphone”、64 kbps、15 fps、p = 10 画素、8 フレーム目)

(c)に d がそれぞれ 32、128、256 に設定した N_b の推移を示す．この場合も、N_b は設定した d の値付近に集中している．

前出の 3 個の MB （MB No.47、53、97）の最小 D_a の推移を図 3.5 に示す．d は 64 である．MB No.47、53、97 はそれぞれ N_b が 203、69、132 回で BM を停止している．また、既に図 3.3 で示されているが、N_b が 400 に達する MB は 1 個も無く、HS-IBOS の BM 停止条件がうまく機能しているのがわかる．

これまでに示した“Carphone”の解析結果より、d を 64 に設定すると、HS-IBOS の D_a は FS と同等であり、かつ、F_s は BOS、FS のそれぞれ 2、4.88 倍である．

HS-IBOS を用いて、テスト画像“Coastguard”、“Mobile & Calendar”の画質解析結果（D_a、R_{SN} と d の関係）を図 3.6 に示す．“Coastguard”（QCIF、YUV が 4:2:0）の解析条件は R_d が 64 kbps、R_f が 15 fps、p が 10 画素、

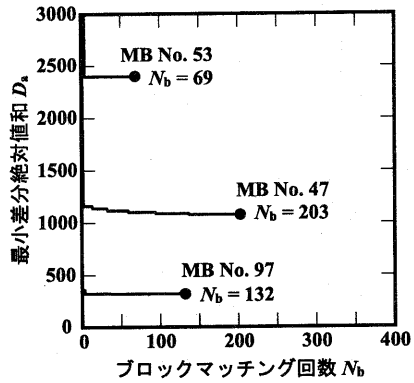


図 3.5 HS-IBOS を用いた BM 処理の推移（最小差分絶対値和の推移）．(“Carphone”、64 kbps、15 fps、p = 10 画素、8 フレーム目、d=64)

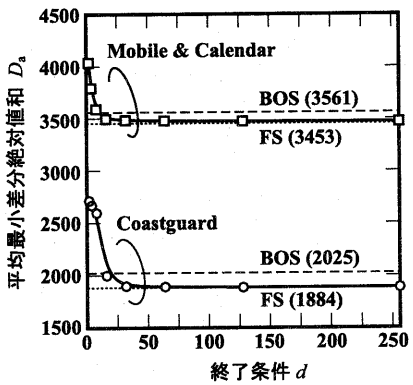


図 3.6 平均最小差分絶対値和 D_a と終了条件 d の関係．“Mobile & Calendar”(384 kbps、15 fps、p =16 画素)．“Coastguard”（64 kbps、15 fps、p = 10 画素）．

“Mobile & Calendar”（CIF、YUV が 4:2:0）の解析条件は R_d が 384 kbps、R_f が 15 fps、p が 10 画素、である．d を 1 から増加させると、“Coastguard”、“Mobile & Calendar”の D_a は減少する．d をさらに増加させると、“Carphone”と同様に、“Coastguard”は d>32 で、“Mobile & Calendar”は d>64 で、FS とほぼ等しい D_a に収束する．すなわち、テスト画像毎に、D_a が収束する d が存在する．

図 3.7、3.8 にそれぞれ“Coastguard”の 70 フレーム目、203 フレーム目の解析結果を示す．両図とも (a)は原画像、(b) FS で得た動きベクトル(x 成分または y 成分のうち大きい方の値)、(c) BOS による MB の BM 数(N_b)の推移、(d) は HS-IBOS による N_b の推移、である．d は 32 である．70 フレーム目の動きベクトルは 3 あるいは 4 が多く、動きが速い．これに対し、203 フレーム目の動きベクトルは 0、1 に集中し、動きが遅い．図 2.1、2.2 に示した BOS による“Carphone”の N_b が 2 分化されたように、BOS による“Coastguard” の N_b は 400、

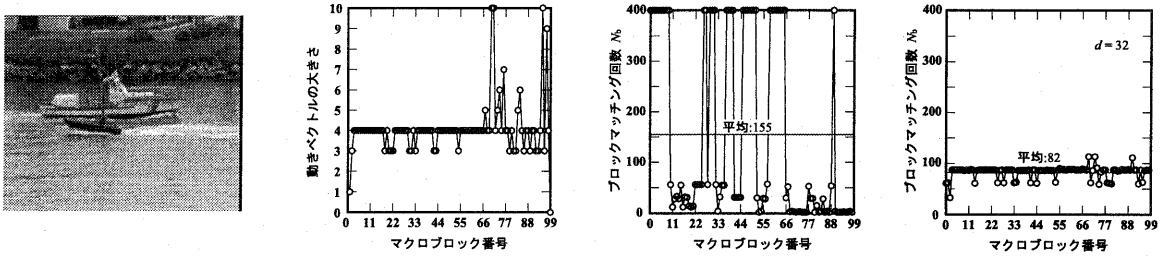


図 3.7 70 フレーム目の解析結果．(a)原画像．(b) FS で求めた動きベクトルの大きさ．
(c) N_b (IBOS)．(d) N_b (HS-IBOS、d=32)．("Coastguard"、64 kbps、15 fps、p=10 画素)

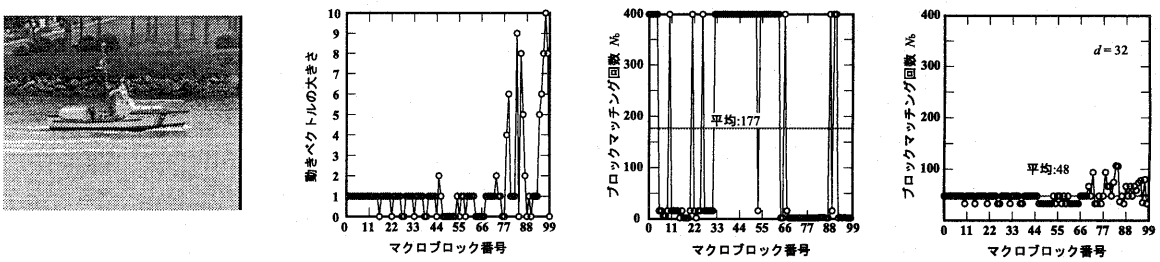


図 3.8 203 フレーム目の解析結果．(a)原画像．(b)) FS で求めた動きベクトルの大きさ．
(c) N_b (IBOS)．(d) N_b (HS-IBOS、d=32)．("Coastguard"、64 kbps、15 fps、p=10 画素)

表 3.1 各種動きベクトル検出アルゴリズムの画像解析結果．(R_f = 15 fps)

テスト画像	画像サイズ	R_d [kbps]	N_f	p	アルゴリズム*	平均 N_b (%)**		平均 F_s	平均 D_a (%)**		平均 R_{SN} [dB]**	
Carphone	QCIF	64	382	10	FS	400	(100.0)	1.00	1,335	(±0.00)	32.582	(±0.000)
					IBOS	164	(41.0)	2.44	1,414	(+5.92)	32.345	(−0.237)
					HS-IBOS (64)	82	(20.5)	4.88	1,351	(+1.20)	32.547	(−0.035)
Coastguard	QCIF	64	300	10	FS	400	(100.0)	1.00	1,884	(±0.00)	29.415	(±0.000)
					IBOS	176	(44.0)	2.27	2,025	(+7.48)	28.919	(−0.496)
					HS-IBOS (32)	44	(11.0)	9.09	1,897	(+0.69)	29.377	(−0.038)
Mobile & Calendar	CIF	384	300	16	FS	1,024	(100.0)	1.00	3,453	(±0.00)	24.331	(±0.000)
					IBOS	469	(45.8)	2.18	3,561	(+3.12)	24.142	(−0.189)
					HS-IBOS (64)	68	(6.6)	15.06	3,478	(+0.72)	24.317	(−0.014)

*) 括弧内は d の値．　**) 括弧内は FS との比較値．

20ー60、ほぼ 0 の MB に 3 分されている．また、BOS による平均 N_b は、動きが速い 70 フレーム目では 155、動きが遅い 203 フレーム目では 177 である．これより、動きの速度（大きさ）と平均 N_b との間に強い相関は見られない．一方、HS-IBOS による平均 N_b は、動きが速い 70 フレーム目が 82、動きが遅い 203 フレーム目が 48、で動きの大小と平均 N_b の大小に相関が見える．つまり、HS-IBOS の平均 N_b は動きベクトルの大小に依存し、かつ、BOS の平均 N_b より小さい．

これまで検討してきた FS、IBOS、HS-IBOS の画像解析結果を表 3.1 に示し、HS-IBOS の画質（平均 D_a）と ME 速度（平均 F_s）を FS、BOS の画質と ME 速度とを以下で比較する．

"Carphone"（d = 64）；HS-IBOS の画質は（D_a=1,351）で、IBOS（D_a=1,414）より 4.7%下回り（向上し）、FS（D_a=1,334）よりその増加（劣化）はわずか 1.2%である．一方、HS-IBOS の ME 速度は、FS の約 5 倍、BOS の約 2 と、高速化されている．

"Coastguard"（d = 32）；HS-IBOS の画質は（D_a=1,897）で、IBOS（D_a=2,025）より 6.8%も向上し、FS（D_a=1,884）のわずか 0.7%の劣化に止まっている．HS-IBOS の ME 速度は、FS の約 9 倍、BOS の約 2 倍と、大幅に高速化されている．

"Mobile & Calendar"（d = 64）；HS-IBOS の画質は（D_a=3,478）で、IBOS（D_a=3,561）より 2.4%も向上し、FS（D_a=3,453）のわずか 0.72%の劣化である．HS-IBOS の ME 速度は、FS の約 15 倍、BOS の約 2 倍と、大変高速化されている．

4. 累算形差分絶対値和回路の低消費電力化

4.1. 累算形差分絶対値和回路の設計

0.13-μm CMOS 技術を用いて設計した 2 段パイプライン構成、16 b 累算形差分絶対値和回路（累算形 ADA）を図 4.1 に示す．本回路は 1 画素分の差分絶対値を得る 8b 差分絶対値回路と 1MB 分の差分絶対値を累算する 16 b 累算回路で構成される．両回路とも回路規模、消費電力が小さい逐次桁上げ加算回路を用いている[3、4、5]．本累算形 ADA は電源電圧（V_D）が 0.83 V 以上で、220 MHz 動作する．この時、8b 差分絶対値回路と上位 8b 累算回路の消費電力（P_A）は 192.2 μW、上位 8b 累算回路の消費電力（P_B）は 64.5 μW、累算形 ADA の総消費電力（P_T=P_A+P_B）は 256.7 μW である．

上位 8b 累算回路の入力は下位 8b 累算回路からの桁上げ信号（C_7= "1"または"0"）である．従って、C_7 が"1"の時のみ、上位 8b 累算回路を動作させれば、回路の消費電力はさらに削減される．また、BM が終了した時点で、ゲーテドクロックパルスで回路を停止しすれば、回路の総消費電力は大幅に削減される[6]．

4.2. 稼働率の算出

上位 8b 累算回路の稼働率（α）は、MB 毎に c_7 が "1" となる回数と画素数（=256）との比を求め、全マクロブロックに対して平均して、算出できる．クロック周波数（f_c）が 220 MHz の時、最大で 578 回の BM が処理できる．従って、各アルゴリズムで得られる平均 N_b と最大 BM 回数との比を、各アルゴリズムを処理する累算形 ADA の稼働率（β）として定義できる．各 ME アルゴリズムの α、β を表 4.1 にまとめて示す（テスト画像は"Carphone"、R_d は 64 kbps、R_f は 15 fps、p は 10 画素）．

4.3. 累算形差分絶対値和回路の消費電力

上位 8b 累算回路の消費電力（P_B'）は αP_B で求められる．さらに、各アルゴリズム毎にゲーテドクロックパルスを適用すると、累算形 ADA の総消費電力（P_T'）は$(P_A+P_B')\beta$ で求められる．FS、BOS、HS-IBOS を適用した P_T'はそれぞれ 137.4 、55.8、27.9 μW となる．HS-IBOS を適用した P_T'は FS の 1/5、BOS の 1/2.5 と、大幅に削減されている．各アルゴリズムを適用した累算形 ADA の P_T'を表 4.1 に示す．

5. おわりに

最小差分絶対値和（D_a）の減少率が連続して 0 となる連続回数（d）をあらかじめ設定し、これをクリアしたとき、最適なマクロブロックを検出したものとして、ブロックマッチングを終了させる高速改良中断法（HS-IBOS）を開発した．QCIF テスト画像、"Carphone"（64 kbps、15 fps、p = 10 画素）を用い、d = 64 としたとき、D_a は 1,351 となり全探索法（FS）の D_a（=1,334）と同等の結果が得られ、動きベクトル検出速度は FS の約 5 倍、中断法の約 2 倍に高速化できた．"Coastguard"、"Mobile & Calendar"では、それぞれ FS の約 9 倍、15 倍であった．また、HS-IBOS を用いると、2 段パイプライン、ゲーテドクロックを適用した累算形差分絶対値和回路の消費電力は 27.9 μW となり、FS を用いた同回路の消費電力（137.4 μW）の約 1/5 に削減できた．今後、d の設定方法、等を検討する．

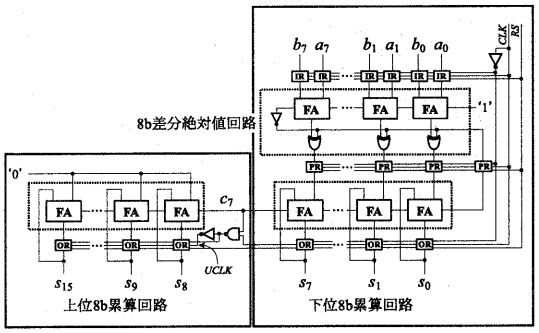


図 4.1 累算形差分絶対値和回路．

表 4.1 0.13-μm CMOS 累算形差分絶対値和回路の特性．("Carphone"、64 kbps、15 fps、p =10 画素)

アルゴリズム	FS	IBOS	HS-IBOS (d=64)
上位 8b 累算回路の稼働率 α [%]	9.90 (100.0)	6.71 (67.8)	6.60 (66.7)
回路全体の稼働率 β [%]	69.2 (100.0)	28.4 (41.0)	14.2 (20.5)
動作時消費電力 [μW]	137.4 (100.0)	55.8 (40.6)	27.9 (20.3)
クロック周波数 f_c [MHz]	220		
電源電圧 V_D [V]	0.83		
論理ゲート数 n_g [個]*	451		
総 FET 数 n_F [個]*	1,468		

*)論理ゲート数および総 FET 数はデザインルールで決まる最小 FET サイズで規格化した値．括弧内は FS との比較値．

謝辞

SPICE 用の 0.13-μm MOSFET のモデルパラメータを快く提供して下さった NEC の関係各位に心より感謝致します．また、ご協力頂いた中央大学教育技術員の原田氏、榎本研究室の関係諸氏に感謝致します．

参考文献

[1] 榎本、笹島、廣部、「中断法動きベクトル検出アルゴリズム」、信学総合大会講演論文集、エレクトロニクス 2、SC-11-6、p.307、1997 年 3 月．

[2] T. Enomoto and A. Kotabe, "Fast Motion Estimation Algorithm and Low-Power CMOS Motion Estimator for MPEG Encoding", IEICE Trans. Electron., vol.E86-C, no.4, pp.535-545, April 2003.

[3] T. Enomoto and A. Kotabe, "Fast Motion Estimation Algorithm and Low Power 0.13 μm CMOS Motion Estimation Circuits", in Proc. of International Symposium on Circuits and Systems (ISCAS'2001), Sidney, Australia, vol-II, pp.449-452, May 2001.

[4] 小田部、榎本、「2 ステップ高速中断法アルゴリズムの開発と低消費電力 CMOS 動きベクトル検出回路の設計」、信学技報、ICD2000-54、pp.57-64、2000 年 8 月 24 日．

[5] 江井、榎本、「MPEG-4 動き補償用 0.13-μm CMOS 差分絶対値和回路の低電力化」、信学技報、ICD2002-58、pp.73-78、2002 年 8 月．

[6] 渡邊、江井、榎本、「稼働率削減による低電力 CMOS 差分絶対値和回路の設計」、信学総合大会講演論文集、C-12.17、2003 年 3 月．