

# 探索窓サイズを動きに追従して適応設定する超高速 動きベクトル検出アルゴリズムの開発

榎本忠儀 居坂詩紋 小林伸彰

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 2007年10月、動きの変化に追従して探索窓サイズを決定する高速動きベクトル検出 (ME) アルゴリズムを開発し、MPEG-2 に準拠したソフトウェアエンコーダに実装し、解析結果を報告した。今回、中断法を利用した最終探索をダイヤモンド探索に替え、さらに MPEG-2 に替え H.264 に準拠したソフトウェアエンコーダに実装し、評価した。本アルゴリズムは、まず、対象マクロブロック (M-Blk) の周辺 M-Blk の動きの変化に応じて探索窓サイズ、探索停止条件、探索中心位置を決定する。次に探索中心位置から外側へ向け 8 方向にサブサンプリングし、処理量を大幅に削減する。最後にダイヤモンド探索で高精度に動きベクトルを検出する。本アルゴリズムを SDTV の各種テスト画像、HDTV の "Riverbed"、"Tractor" に適用し、画質、探索処理量をそれぞれ評価した。その結果、全探索法の画質を維持したまま、探索速度を SDTV 画像では全探索法の約 132~272 倍、HDTV 画像では約 500 倍 ("Riverbed")、約 866 倍 ("Tractor") 高速化できた。

キーワード H.264、動きベクトル検出、探索窓サイズ、サブサンプリング

## High Speed, Sub-sampling Motion Estimation Algorithm Employing Adaptively Assigned Search Window Sizes

Tadayoshi Enomoto, Shimon Isaka and Nobuaki Kobayashi

Graduate School of Science and Engineering, Chuo University  
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

**Abstract** A high speed motion estimation (ME) algorithm employing adaptively assigned search window sizes followed by a hierarchical sub-sampling block matching and a diamond search was developed for H.264. This algorithm, first, adaptively assigns a search window size and defines both a search window position and a search stopping condition for each macro-block. Then, an 8-way hierarchical motion estimation from the centre to the outer is carried out followed by the diamond search. ME processing speed was 132 to 272 times faster than that of the full-search (FS) method for SDTV test video sequences and 500 to 866 times faster than that of the full-search method for HDTV test video sequences.

**Keyword** H.264, motion estimation algorithm, search area sizes, sub-sampling

### 1 はじめに

地上デジタル放送が始まり、HDTV (high definition television, MP@HL, 1920 画素×1080 ライン) ならびに HDTV 向け Blue-Ray Disc が急速に普及している。これに伴い、データ量の極めて多い HDTV の符号化、特に動きベクトル検出処理 (Motion Vector Estimation: ME) の高速化技術の開発が急務である。処理量を大幅に削減し、高速化を図る方法としてサブサンプリング (Sub-Sampling: SS) 法、3 ステップ法<sup>[1]</sup>、4 ステップ法<sup>[2]</sup>、テレスコピック法<sup>[3]</sup>等があるが、高精度に動きベクトル (Motion Vector: MV) を検出できな

い、処理が複雑等の問題がある。また、高精度で高速に MV を検出できる方法として unrestricted center-biased diamond search (UCBDS)<sup>[4]</sup>、minimum bounded area search (MIBAS)<sup>[5]</sup>、multilevel successive elimination algorithm (MSEA)<sup>[6]</sup>、unsymmetrical-cross multi-hexagon-grid search (UMHS)<sup>[7]</sup>、simplified UMHS (S-UMHS)<sup>[8]</sup>等があるが、動きの速い画像や複雑な画像では、処理量が比較的多い等の問題がある。

このような問題を解決するため、高速でかつ高精度な動きベクトルを検出するアルゴリズムを開発し、2007年10月に報告した<sup>[9]</sup>。本アルゴリズムを MPEG-2 に準拠したソフトウェアエンコーダに実装し、NTSC 準拠の SDTV 画像に適用した結果、全探索法 (Full Search:

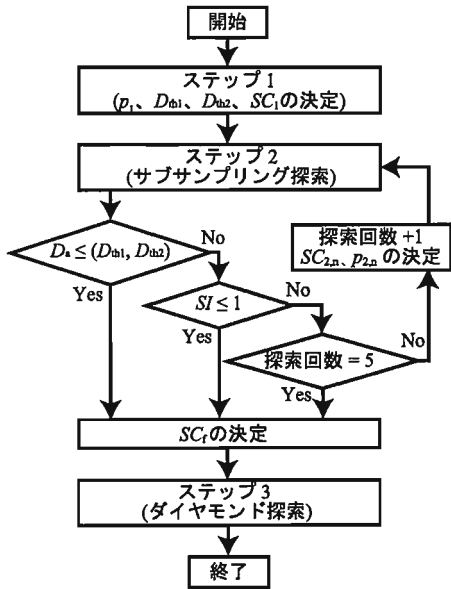


図 2.1 フローチャート

FS) 並の画質を維持し、ME 速度を FS の約 95~300 倍に高速化できた (探索領域サイズ;  $p = \pm 64$  画素)。今回、中断法を利用した最終探索をダイヤモンド探索に替え、新たに MPEG-2 に替え H.264 に準拠したソフトウェアエンコーダに実装し、NTSC 準拠の SDTV 画像、PAL 準拠の SDTV 画像、HDTV 画像に適用した。その結果 FS の画質を維持し、ME 速度を 4 種類の SDTV 画像では FS の約 132~272 倍 (探索領域サイズ;  $p = \pm 64$  画素)、2 種類の HDTV 画像では約 500 倍、約 865 倍 (探索領域サイズ;  $p = \pm 128$  画素) にそれぞれ高速化できた。

以下、第 2 章でアルゴリズムについて述べ、第 3 章で画像解析結果を述べる。

## 2 ME アルゴリズム

本アルゴリズムは図 2.1 に示すように 3 ステップから構成される。まず、ステップ 1 では前処理として、サブサンプリング (SS) 探索で必要となる探索窓 (SW; Search Window) のサイズ ( $p_1$ )、ブロックマッチング (BM) 処理の過程で得られる差分絶対値 ( $D_a$ ) で与えられる探索停止しきい値 ( $D_{th1}$ ,  $D_{th2}$ )、探索の開始位置 ( $SC_1$ ) を、対象マクロブロック (M-Blk) 毎に設定する。次に、ステップ 2 で、SS 法を用いた BM 処理を  $SC_1$  から外側に 8 方向に向けて行い、差分 (Absolute Difference; AD) 演算回数を大幅に削減する。ステップ 2 では対象 M-Blk の最小  $D_a$  が  $D_{th1}$ ,  $D_{th2}$  に達した時点で、BM を終了する。最後に、ステップ 3 で、ダイヤモンド探索を行い、最適な MV が検出された時点で、BM 処理を終了する。以下に本 BM 処理の手順を詳述する。

### 2.1 ステップ 1 (前処理)

対象 M-Blk の周辺にある M-Blk の MV を用いて、SS 探索で必要となる  $p_1$ ,  $D_{th}$ ,  $SC_1$  を M-Blk 毎に最適な値

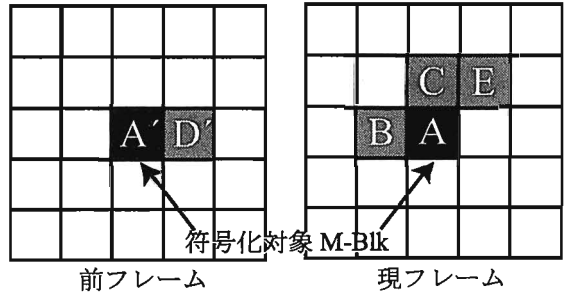


図 2.2 M-Blk の位置関係

に設定する (図 2.2)。A が対象 M-Blk であり、B、C が A の左、真上に位置する M-Blk である。A'、D' は前フレームの M-Blk で、A' は A と同位置、D' は右に位置する。A' の MV の水平方向成分と B の MV の水平方向成分との差 (距離)、A' の MV の垂直方向成分と B の MV の垂直方向成分の MV との差を求める。同様に、A' と C、A' と D' に対しても距離を求める。

#### ・探索窓サイズ ( $p_1$ ) の設定

以下の 2 通りの場合分けを行い、 $p_1$  を設定する。

- (1) 6 個の距離がいずれも予め設定された探索窓サイズ ( $p$ ) の 1/4 以下 (距離  $\leq p/4$ )、 $p_1$  を  $2p/5$  とする。例えば、 $p=64$  の場合、 $p_1$  を 27 とする。
- (2) 上記条件に合致しない場合、つまり  $p/4$  以上の距離が 1 個でもあれば (距離  $> p/4$ )、 $p_1$  を  $3p/5$  とする。

#### ・探索停止しきい値 ( $D_{th}$ ) の設定

以下の 3 通りの場合分けを行い、 $D_{th1}$ ,  $D_{th2}$  を設定する。

- (設定 1)  $p_1$  設定の (1) の場合、つまり 6 個の距離が  $p$  の 1/4 以下 (距離  $\leq p/4$ ) の場合、A' の最小  $D_a$  を  $D_{th1}$  とする。
- (設定 2) 上記 (1) に合致しない場合で、かつ B の最小  $D_a$  と C の最小  $D_a$  が等しければ、B の最小  $D_a$  を  $D_{th2}$  とする。
- (設定 3) 上記 (1)、(2) のいずれの条件にも合致しない場合、 $D_{th1}$ ,  $D_{th2}$  を設定しない。つまり、 $p_1$  内で全ての BM 処理を行う。

#### ・探索開始位置 ( $SC_1$ )

$SC_1$  を決定するために、B、C および対象 M-Blk の右上の M-Blk (E) の MV を用いる (図 2.2)。3 個の MV の水平方向成分、の中央の値 ( $M_h$ )、垂直方向成分、の中央の値 ( $M_v$ ) を求め、A' の中心より水平方向に  $M_h$ 、垂直方向に  $M_v$  だけシフトした位置を探索開始位置 ( $SC_1$ ) とする<sup>[10]</sup>。

### 2.2 ステップ 2 (第 1 ~ 5 回 BM 処理)

ステップ 1 で設定された条件のもとで、SS 法を用い

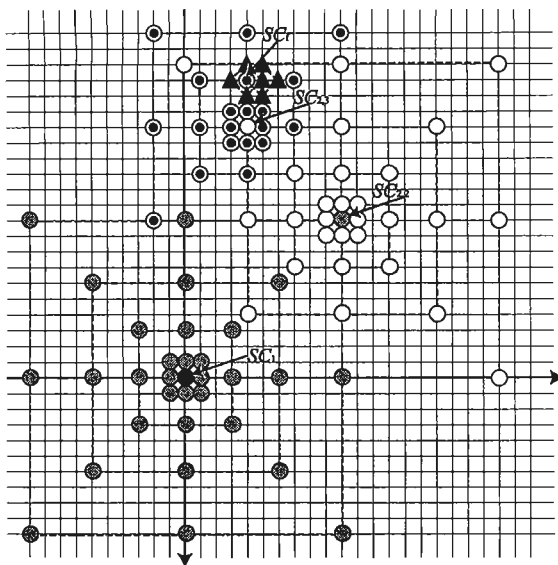


図 2.3 BM 処理

て第 1 回目の BM 処理を行う。探索開始位置は  $SC_1$ 、探索領域は  $p_1$  である。これを図 2.3 に示す。なお、BM 処理位置は  $\odot$  の位置である。つまり  $\odot$  の間隔 (サブサンプリングの間隔) は、探索中心の  $\odot$  を含めて、0、0、1、1、2、2、3、3、...  $k$ 、 $k$  とする。ここで、 $k$  は最大のサブサンプリング間隔で下式に従う自然数である。

$$(k+1)^2 \leq p_{2,n} < (k+2)^2$$

ここで  $p_{2,n}$  ( $n=1, 2, 3, 4, 5$ ) はステップ 2 における第  $n$  回目の探索窓サイズである。従って、 $p_{2,1}$  は  $p_1$  である。

第 1 回 BM 処理の途中で、下記の 2 条件のいずれかに該当する場合、検出された M-Blk の位置をステップ 3 の探索開始位置 ( $SC_i$ ) として、ステップ 3 に移行する。

(条件 1) 最小  $D_n$  の値が  $D_{th1}$ 、 $D_{th2}$  の 1.05 倍を下回る。

(条件 2)  $p_{2,n}$  内の探索が終了した時点で、最小  $D_n$  が探索間隔  $\leq 1$  で見つかる。

上記の 2 つの条件に該当しない場合、最小  $D_n$  の M-Blk の位置を  $SC_{2,2}$  とし、 $SC_{2,2}$  と  $SC_1$  との間の水平あるいは垂直距離のうち、大きい方の距離を探索窓サイズ ( $p_{2,2}$ ) として、ステップ 2 の第 2 回 BM 処理を行う。この様子を図 2.3 に示す。第 2 回目の BM 処理は  $\circ$  の位置で実行される。 $p_{2,2}$  内の  $\odot$  は第 1 回目で探索済みであるから、第 2 回目の BM 処理ではこれを省く。以下同様に、2 条件のいずれかに該当するまで、BM 処理を繰り返し、 $SC_i$  を決定する。但し、ステップ 2 の BM 処理回数は最大で 5 回とする。図 2.3 では一例として、3 回目の BM 処理 ( $\odot$  は探索点) まで示している。

### 2.3 ステップ 3 (ダイヤモンド探索)

ステップ 2 で得た  $SC_i$  を中心として、上下左右の 4 点でステップ 3 の第 1 回 BM 処理を行い、最小  $D_n$  の M-Blk を求める。次に最小  $D_n$  の M-Blk を中心に、再度上下左右の 4 点で BM 処理を行う。この時、既に探索済みの BM 処理は省く。以下、最小  $D_n$  の値が減少しなくなるまで、同様な探索を続ける。最終 MV は探索が停止した時点で得られた MV

表 3.1 解析条件

テスト画像	"Football"、"Mobile&Calendar"、"Tempete" (NTSC)、"F1car" (PAL) "Riverbed"、"Tractor" (HDTV1080p)
エンコーダー	H.264/AVC、JM12.3
プロファイル @レベル	MP@ML (NTSC)、PAL MP@HL (HDTV)
GOP 構造	$N=15$ 、 $M=3$
ターゲットビットレート	4Mbps (NTSC)、PAL 15Mbps (HDTV)
フレームサイズ	720×480 (NTSC)、720×576 (PAL) 1920×1088(1080+8 black lines) (HDTV)
フレームレート	30fps (NTSC)、25fps (PAL、HDTV)
探索窓サイズ	±64 (NTSC、PAL)、±192 (HDTV)
フレーム数	260 (NTSC)、220 (PAL) 150 (HDTV)
1/4 画素探索	使用
参照フレーム数	1
ブロックサイズ	16×16、16×8、8×16、8×8、8×4、4×8、4×4
重み付き予測	off
RD Picture Decision	非使用
RD Optimization	Low complexity mode
レートコントロール	使用
Early Skip	非使用
InitialQP	画像ごとに適応的に設定
UMHex DSR	使用
UMHex Scale	3

とする<sup>[10]</sup>。この様子を図 2.3 に示す。ステップ 3 の探索点を  $\blacktriangle$  で示す。

## 3 画像解析

### 3.1 テスト画像と解析条件

H.264 に準拠したソフトウェアエンコーダ JM<sup>[11]</sup> に本アルゴリズムを実装し、各種テスト画像を用いて画像解析と速度評価を行った。用いたテスト画像は SDTV 画像 (NTSC 方式) として、動きの速い "Football"、および動きの遅い "Mobile & Calendar"、"Tempete"、SDTV 画像 (PAL 方式) として、動きの速い "F1car"、HDTV 画像として動きの速い "Tractor"、動きの複雑な "Riverbed" を用いた。詳細な解析条件を表 3.1 に示す。

### 3.2 SDTV 画像解析結果

#### A. "Football"

図 3.1(a) に本アルゴリズムと全探索法 (FS) の平均ピーク SN 比 (平均  $R_{sn}$ ) の推移を示す。実線が本アルゴリズム、破線が FS、テスト画像は "Football" である。両アルゴリズムの平均  $R_{sn}$  はほぼ一致している。すなわち、本アルゴリズムは FS 並の画質を補償している。図 3.2 に再生画像を示す。(a) が本アルゴリズム、(b) が FS である。両者間に差は見られない。

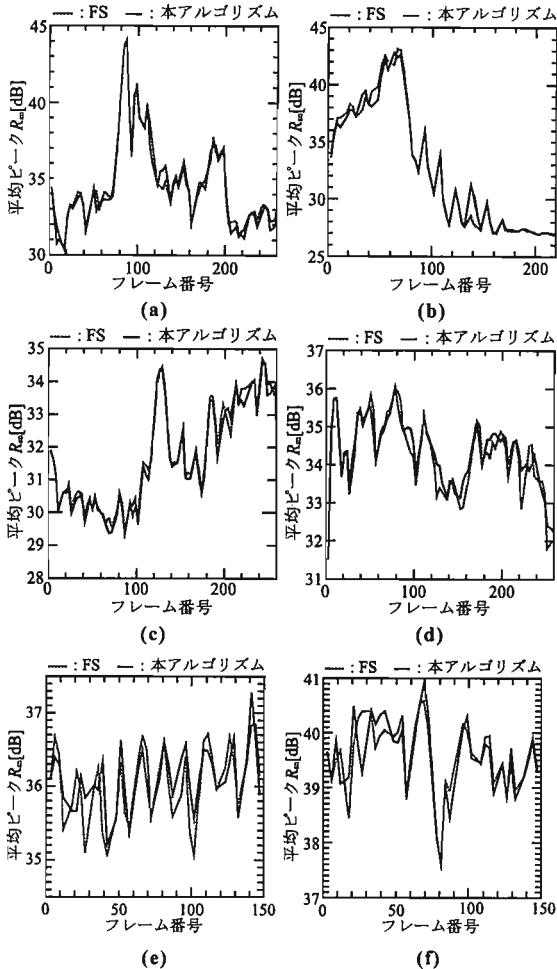


図 3.1 フレーム毎の平均ピーク SN 比。(a) “Football”。(b) “F1car”。(c) “Mobile&Calendar”。(d) “Tempete”。(e) “Riverbed”。(f) “Tractor”。

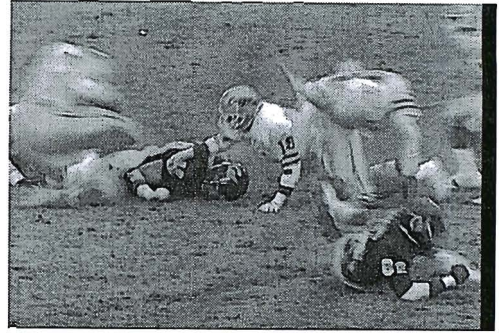
“Football”で評価した本アルゴリズム、FS、UMHS、S-UMHS の解析結果 (画質と速度) を表 3.2 にまとめる。本アルゴリズムの平均  $R_{sn}$  (34.659 dB) は FS の平均  $R_{sn}$  より 0.014 dB 低いだけである。つまり、本アルゴリズムは FS と遜色のない画質を維持できることを示しており、図 3.1(a) の結果と一致する。

平均差分 (AD) 演算回数を比較すると、本アルゴリズムの平均 BM 速度は FS の約 132 倍である。また、本アルゴリズムは、最近開発された高速アルゴリズムである UMHS<sup>[7]</sup>、S-UMHS<sup>[8]</sup> と比べても、それぞれ 2.13 倍、1.76 倍高速化されている。

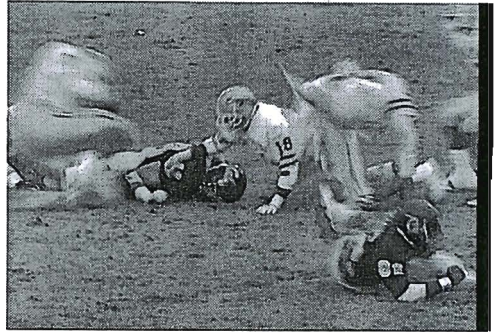
以上、“Football”で評価した結果より、本アルゴリズムは、FS と遜色のない画質を保ちながら、BM 速度は FS、UMHS、S-UMHS のそれぞれ約 132 倍、2.13 倍、1.76 倍に改善されることが分かる。

### B. 他のテスト画像を用いた解析結果

図 3.1(b)~(d) に本アルゴリズムと全探索法 (FS) の平均  $R_{sn}$  の推移を示す。実線が本アルゴリズム、破線が FS である。テスト画像は、(b) が “F1car”、(c) が



(a)



(b)

図 3.2 “Football”の再生画像 (フレーム番号 213)。(a) 本アルゴリズム。(b) 全探索法。

“Mobile&Calendar”、(d) が “Tempete” である。いずれのテスト画像においても、両アルゴリズムの平均  $R_{sn}$  はほぼ一致している。表 3.1 より、本アルゴリズムの平均ピーク  $R_{sn}$  は、FS と比べ、0.049dB の増加 (“F1car”)、0.019dB の増加 (“Tempete”)、0.085dB の減少 (“Mobile&Calendar”) であり、これらの値は極めて小さい。“Football”と同様、本アルゴリズムは FS と遜色のない画質を維持できることを示しており、図 3.1(b)~(d) の結果と一致する。

表 3.2 により、本アルゴリズムの平均 BM 速度は、FS の約 170 倍 (“F1car”)~約 272 倍 (“Mobile&Calendar”) である。また、本アルゴリズムの平均 BM 速度は UMHS の 2.84 倍 (“Tempete”)~3.66 倍 (“Mobile&Calendar”)、S-UMHS の約 1.86 倍 (“F1car”)~3.13 倍 (“Mobile&Calendar”) であり、いずれの高速アルゴリズムより高速である。

これより本アルゴリズムは SDTV 画像において FS と遜色のない画質で、FS、UMHS、S-UMHS と比べ、高速に動きベクトル検出ができる事がわかる。

以上の結果より、本アルゴリズムは、FS と遜色のない画質を保ちながら、高速 BM 処理が可能である。

### 3.3 HDTV 画像解析結果

図 3.1(e)、(f) に本アルゴリズムと全探索法 (FS) の平均  $R_{sn}$  の推移を示す。実線が本アルゴリズム、破線が FS である。テスト画像は、(e) が “Riverbed”、(f) が “Tractor” である。いずれのテスト画像においても、両ア

表 3.2 動画解析結果 (SDTV:4Mbps、30fps、 $p=\pm 64$  画素。HDTV:15Mbps、25fps、 $p=\pm 192$  画素。GOP 構造は  $M=3$ 、 $N=15$ 。)

画像サイズ	テスト画像	アルゴリズム	平均 AD 処理回数 [回]	平均 BM 速度 [倍] 基準:FS	平均 BM 速度 [倍] 基準:UMHS	平均 BM 速度 [倍] 基準:S-UMHS	平均 最小 $D_s$	FS からの増加率 [%]	平均ピーク $R_{sn}$ [dB]	FS との差 [dB]
SDTV	Football	FS	7858264.43	1.00	-	-	2782.51	0.00	34.673	0.000
		UMHS	126595.12	62.07	1.00	-	2953.85	6.16	34.699	0.026
		S-UMHS	105066.16	74.79	1.20	1.00	2984.16	7.25	34.729	0.056
		本アルゴリズム	59572.63	131.91	2.13	1.76	3160.55	13.59	34.659	-0.014
	Flcar	FS	7670011.49	1.00	-	-	3628.28	0.00	32.537	0.000
		UMHS	159807.70	48.00	1.00	-	3760.41	3.64	32.554	0.017
		S-UMHS	83618.20	91.73	1.91	1.00	3824.34	5.40	32.590	0.052
		本アルゴリズム	44986.76	170.49	3.55	1.86	3954.22	8.98	32.587	0.049
	Mobile&Calendar	FS	4785129.43	1.00	-	-	3310.39	0.00	31.801	0.000
		UMHS	64407.07	74.30	1.00	-	3360.11	1.50	31.763	-0.038
		S-UMHS	55102.74	86.84	1.17	1.00	3373.98	1.92	31.733	-0.069
		本アルゴリズム	17592.56	272.00	3.66	3.13	3426.43	3.51	31.716	-0.085
Tempete	FS	5024863.51	1.00	-	-	2574.66	0.00	34.255	0.000	
	UMHS	57743.04	87.02	1.00	-	2624.62	1.94	34.219	-0.036	
	S-UMHS	51076.90	98.38	1.13	1.00	2632.05	2.23	34.288	0.033	
	本アルゴリズム	20308.98	247.42	2.84	2.51	2673.30	3.83	34.273	0.019	
HDTV	Riverbed	FS	46375384.56	1.00	-	-	2123.71	0.00	36.061	0.000
		UMHS	157723.94	294.03	1.00	-	2380.09	12.07	36.054	-0.007
		S-UMHS	201627.83	230.00	0.78	1.00	2351.37	10.72	36.048	-0.013
		本アルゴリズム	92664.75	500.46	1.70	2.18	2391.51	12.61	36.117	0.056
	Tractor	FS	30490034.34	1.00	-	-	1284.52	0.00	39.589	0.000
		UMHS	43093.28	707.54	1.00	-	1356.40	5.60	39.514	-0.075
		S-UMHS	75420.55	404.27	0.57	1.00	1366.45	6.38	39.602	0.013
		本アルゴリズム	35217.78	865.76	1.22	2.14	1380.89	7.50	39.578	-0.011



(a)



(b)

図 3.3 “Tractor” の再生画像 (フレーム番号 81)。 (a) 本アルゴリズム。 (b) 全探索法。

ルゴリズムの平均  $R_{sn}$  はほぼ一致している。

図 3.3(a)、(b) に “Tractor” の再生画像を示す。(a) が本アルゴリズム、(b) が FS である。両者間に差は見られない。表 3.2 より、本アルゴリズムの平均ピーク  $R_{sn}$  は、FS と比べ、0.056 dB の増加 (“Riverbed”)、0.011 dB の減少 (“Tractor”) であり、これらの値は極めて小さい。“Football”と同様、本アルゴリズムは FS と遜色のない画質を維持できることを示しており、図 3.1(e)、(f)の結果と一致する。

本アルゴリズムの平均 BM 速度は、FS の約 500 倍 (“Riverbed”)、約 866 倍 (“Tractor”)である。また、本アルゴリズムの平均 BM 速度は UMHS の 1.7 倍 (“Riverbed”)、1.22 倍 (“Tractor”)、S-UMHS の約 2.18 倍 (“Riverbed”)、2.14 倍 (“Tractor”) であり、いずれの高速アルゴリズムより高速である。以上より、本アルゴリズムは HDTV 画像においても、FS と遜色ない画質を保ちながら、FS、UMHS、S-UMHS より高速に BM 処理できることが分かる。

#### 4 おわりに

動きの変化に追従して探索領域を適応的に決定する超高速サブサンプリング動きベクトル検出アルゴリズムを開発し、4種類のSDTVテスト画像、2種類のHDTVテスト画像を用いて評価した。その結果、動きが速く、絵柄が単純な “Tempete” (SDTV画像) では、平均ピーク SN比 ( $R_{sn}$ ) が全探索法 (FS) より約 0.02dB 向上し、BM処理速度は 247倍高速化できた。また、高速アルゴリズムの UMHS、S-UMHS と比べても、BM速度はそれぞれ 2.84倍、2.51倍改善できた。一方、動きが速く、絵柄が複雑な “Football” でも、FS、UMHS、S-UMHS と比べ、BM速度を 132倍、2.13倍、1.76倍改善でき、同時に FS の画質を維持することができた。動きの速い HDTV 画像 “Tractor” においても平均ピーク  $R_{sn}$  は FS と比べ、0.01dB の減少に抑え、BM速度を FS の 866倍と、大幅に高速化できた。また、BM速度を UMHS、S-UMHS と比べても、それぞれ 1.22倍、2.14倍改善できた。以

上の結果より、本アルゴリズムは、多様なテスト画像に対応できる高速アルゴリズムと言える。

#### 謝辞

研究にあたり多くの助言を頂きました中央大学榎本研究室の関係諸氏に感謝致します。

#### 参考文献

- [1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing", Proc. Nat. Telecomms. Conf., vol. 4, pp. C9.6.1-9.6.5, New Orleans, LA, Nov. 1981.
- [2] L. Po and W. Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," IEEE Trans. on Video., Vol. 6, No. 3, pp. 313-317, June 1996.
- [3] 榎本、「画像 LSI システム設計技術」、コロナ社、pp.82-84、2003年9月。
- [4] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", IEEE Trans. Circuits and Systems for Video Technology, vol. 8, no. 4, pp. 369-377, Aug. 1998.
- [5] V. Christopoulos and J. Cornelism, "A center-biased adaptive search algorithm for block motion estimation", IEEE Trans. Circuits and Systems for Video Technology, vol. 10, no. 3, pp. 423-426, Apr. 2000.
- [6] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation", IEEE Trans. Circuits and Systems for Image Processing, vol. 9, no. 3, pp. 501-504, Mar. 2000.
- [7] Z. Chen, P. Zhou, Y. He, "Fast Integer Pel and Fractional Pel Motion estimation in for JVT", JVT-F017r1.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, Island, Japan, Dec. 2002.
- [8] X. Yi, J. Zhang, N. Ling, and W. Shang, "Improved and simplified fast motion estimation for JM", JVT-P021.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 16th meeting, Pozan, Poland, Jul. 2005.
- [9] 居坂、榎本、「動きの変化に応じて探索領域を適応的に決定する超高速サブサンプリング動きベクトル検出アルゴリズム」、信学技報、ICD2007-102、pp.17-22、2007年10月25日。
- [10] Z. Chen, P. Zhou, Y. He, "Fast Motion Estimation for JVT", JVT-G016.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 7th meeting, Pattaya II, Thailand, Mar. 2003.
- [11] JM Reference Software. [Available Online]. <http://bs.hhi.de/suehring/tml/download/>