

探索領域を適応設定する 3 階層サブサンプリング 動きベクトル検出法

小熊浩太郎 木谷光博 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 動きベクトル検出 (ME) のためのブロックマッチング (BM) 処理量を大幅に削減する「探索領域を適応設定する 3 階層サブサンプリング ME 法」を開発した。本アルゴリズムは、まず動きの大小に応じて探索領域サイズの大小を適応的に設定し、次に探索領域のマクロブロックを異なる間引き間隔 (大から小) で階層的に繰り返して BM を行うアルゴリズムである。本アルゴリズムをテスト画像 “Football” (MP@ML, 4Mbps, 30fps) に適用し、画質、BM 処理量を検討した。その結果、全探索法の画質を維持したまま、ME 速度を全探索法、4:1 サブサンプリング法、16:1 サブサンプリング法のそれぞれの約 18 倍、約 4.5 倍、約 1.2 倍に高速化できた。

キーワード MPEG-2, 動きベクトル検出, 探索領域サイズ, サブサンプリング

Fast Motion Vector Estimation Employing Adaptively Assigned Search Area Sizes Followed by Hierarchical Sub-Sampling Block Matching

Koutarou Oguma, Mitsuhiro Kitani and Tadayoshi Enomoto

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

Abstract A “fast motion vector estimation (ME) algorithm employing adaptively assigned search area sizes followed by hierarchical sub-sampling block matching” method was developed for MPEG-2. It can improve processing speed of the full-search method by a factor of about 18, while maintaining visual quality of the full-search method (“Football”, MP@ML, 4 Mbps, 30 frame/sec, search area = 720 pixels × 480 lines).

Keyword MPEG-2, 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 DVD が急速に普及すると予想される。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。同時に、符号化 LSI の低消費電力化が要求され、ME 演算量の削減が求

められる。代表的な ME アルゴリズムの 1 つである、全探索法 (Full Search: FS) は、高精度に動きベクトルを検出できるが、探索領域 (Search Area: SA) 内の全マクロブロック (Macro Block: M-Block) に対して差分絶対値和 (D_0) を求めるため、ブロックマッチング数 (N_b) が極めて多く、HDTV の応用に不適當である。一方、 N_b を大幅に削減する方法としてサブサンプリング (Sub-Sampling: SS) 法 [1] があるが、SA 内の M-Block を縦方向あるいは横方向に間引いてブロックマッ

グ (BM) を行うため、高精度に動きベクトル (Motion Vector: MV) を検出できない。以上の問題を解決するために、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た。

本アルゴリズムを MPEG-2, MP@ML に適用した結果、画質を維持し (平均最小 D_a を FS の約 5% 増以内)、ME 速度を FS の約 18 倍に高速化できた。

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 探索領域を適応設定する 3 階層サブサンプリング ME 法

本アルゴリズムは 2 段階で構成される。画像の動き (速さ、量) は M-Blk 毎に変化する。本アルゴリズムの第 1 段階ではこの特性を利用し、動きの大小に応じて、SA サイズ $[-p, p]$ の大きさを適応的に設定する (小さな SA サイズで ME を実行し、検出された最小 D_a に応じて、次の ME のための SA サイズを決定する) [2]。また、隣接する M-Blk 同士は似ていることが多い。つまり、これらの D_a が互いに近い値または同じ値を持つ可能性が高い。本アルゴリズムの第 2 段階ではこれを利用し、決定された SA サイズを用いて、M-Blk を縦、横共に 4:1 で間引いた SS, 2:1 で間引いた SS, 1 画素精度の BM を階層的に 3 回繰り返して ME を行う [3]。

図 1 に本アルゴリズムの手順を示す。前フレームの平均最小 D_a をしきい値 D_{th} とする。まず、図 2 (a) に示すように、小さな SA サイズ $[-p_1, p_1]$ を参照フレームで設定し、ME を行う。この時、得られる最小の D_a を D_{min1} 、MV を MV_1 とする。 $0 \leq D_{min1} < mD_{th}$ (m は 1 以下の実数) の時、 D_{min1} を最終的な D_a 、 MV_1 を最終的な MV として ME を終了する。 $mD_{th} \leq D_{min1} < nD_{th}$ (n は 1 以上の実数) の時、 MV_1 を中心に SA サイズを $[-p_2, p_2]$ に設定する ($p_2 > p_1$) (図 2 (b))。 $nD_{th} \leq D_{min1}$ の時、SA サイズを $[-p_3, p_3]$ に設定する ($p_3 > p_2$) (図 2 (c))。

は 1 以上の実数) の時、 MV_1 を中心に SA サイズを $[-p_2, p_2]$ に設定する ($p_2 > p_1$) (図 2 (b))。 $nD_{th} \leq D_{min1}$ の時、SA サイズを $[-p_3, p_3]$ に設定する ($p_3 > p_2$) (図 2 (c))。

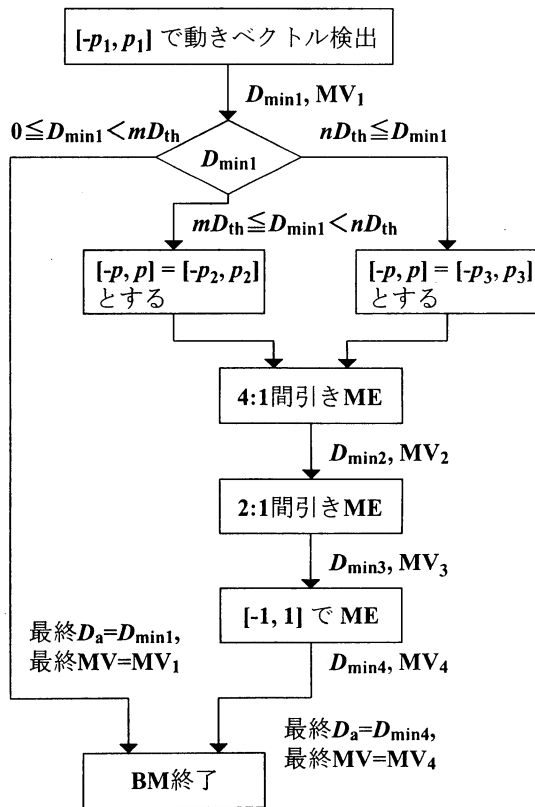


図 1. 本アルゴリズムの流れ図。

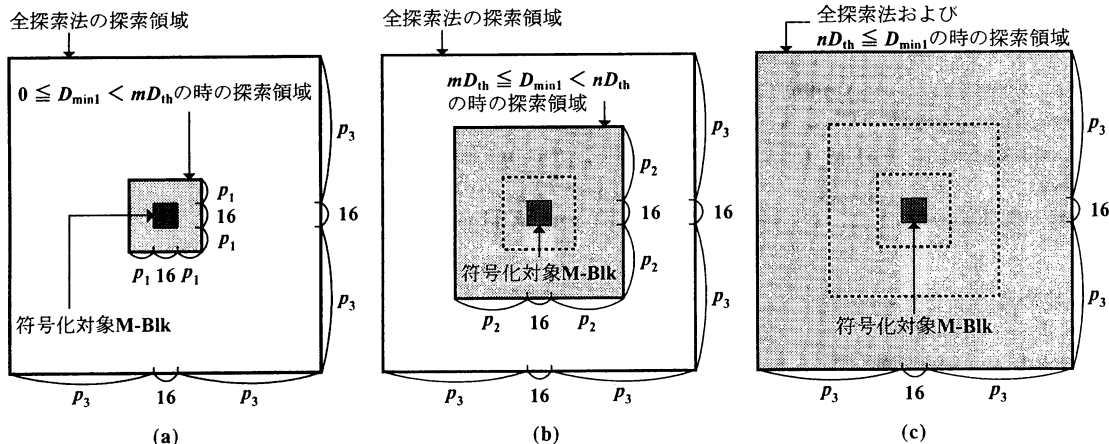


図 2. 探索領域サイズの設定。(a)小さな探索領域サイズ $[-p_1, p_1]$ 。(b)中ぐらいの探索領域サイズ $[-p_2, p_2]$ 。(c)大きな探索領域サイズ $[-p_3, p_3]$ 。

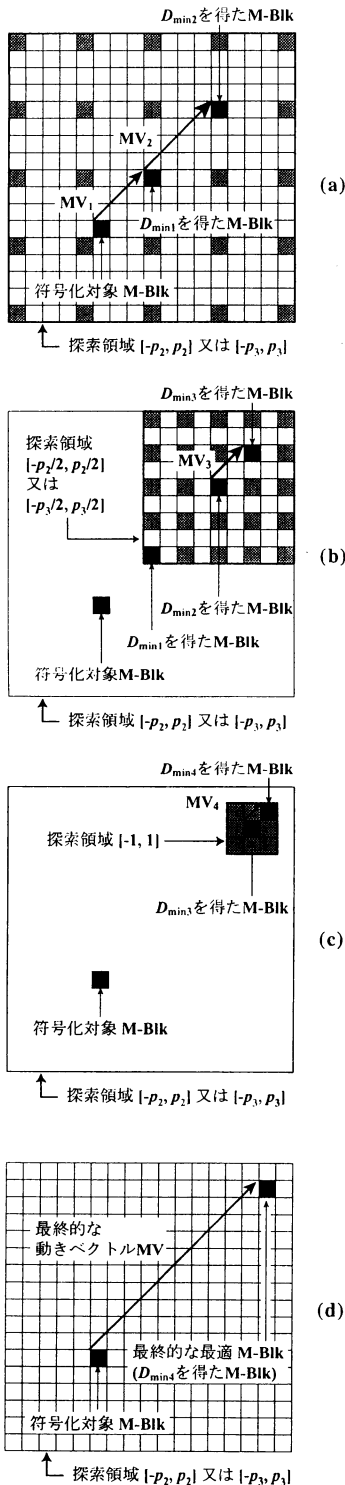


図3. 本アルゴリズムの階層間引き BM 手順
 (a) 1 ステップ目. (b) 2 ステップ目.
 (c) 3 ステップ目. (d) 最終的な MV.

続いて、図 3 (a) に示すように、 D_{min1} , MV_1 を得た M-Bik を中心に、SA 内の M-Bik を水平方向、垂直方向共に 4:1 に間引いて ME を行う。新たに得た D_a を含め、最小 D_a を D_{min2} とし、MV を MV_2 とする。次に、図 3 (b) に示すように、 D_{min2} , MV_2 を得た M-Bik を中心に、SA サイズを $[-p_2/q, p_2/q]$ あるいは $[-p_3/q, p_3/q]$ (q は 1 以上の実数、今回は $q = 2$ とした) とし、SA 内の M-Bik を水平方向、垂直方向共に 2:1 に間引いて ME を行う。新たに得た D_a を含め、最小 D_a を D_{min3} とし、MV を MV_3 とする。最後に、図 3 (c) に示すように、 D_{min3} , MV_3 を得た M-Bik を中心に、SA サイズを $[-1, 1]$ とし、ME を行う。図 3 (d) に示すように、新たに得た D_a を含め、最小 D_a を D_{min4} とし、最終的な D_a とする。符号化対象 M-Bik から D_{min4} を得た M-Bik への MV を最終的な MV_4 とする。なお、BM は左上の M-Bik から右下の M-Bik へ向けた水平スキャン方式を用い、オーバーラップした部分や、設定した SA サイズを超えた部分の ME は行わない。

3 画像解析

3.1 解析画像と解析条件

用いるテスト画像は“Football”と呼ばれる動きの速い動画である。フレームサイズが MP@ML (720 画素 \times 480 ライン) で、フレーム数は 97 (I ピクチャ: 7 フレーム, B ピクチャ: 64 フレーム, P ピクチャ: 26 フレーム) である。以下では P ピクチャに対して MPEG-2 に準拠したソフトウェアエンコーダで処理速度と画質の解析を行う。画素データは符号無し 8 ビット/画素である。データレート R_d は 4Mbps, フレームレート R_f は 30 fps, GOP 構造は $M = 3, N = 15$ である。小さな SA サイズ $[-p_1, p_1]$ は $[-3, 3]$, 中ぐらいの SA サイズは $[-p_2, p_2]$, 大きな SA サイズは $[-p_3, p_3]$, FS の SA サイズ $[-p, p] = [-p_3, p_3]$, m は 0.25~1, n は 1~2 とする。なお、半画素精度は併用しない。

図 4 に原画像 (7 フレーム目) を、図 5, 図 6 にそれぞれ原画像 (7 フレーム目) に対応した水平方向、垂直方向の動きベクトル量を示す (右向き、下向きを正とする。単位は画素数)。SA サイズ $[-p, p]$ は $[-60, 60]$ である。背景部分の各 M-Bik (灰色) の動きベクトルの大きさは 3 画素以下である。これらの M-Bik に対して、 $[-p, p] = [-3, 3]$ 程度を割り当てれば十分であることがわかる。よって、 $p_1 = 3$ とした。一方、動きベクトルの大きさが 3 よりも大きい M-Bik (白色) に対して、 D_{min1} のサイズに応じた SA サイズ、 $[-p_2, p_2]$ 又は $[-p_3, p_3]$ を割り当てる。



図4. “Football”の原画像(7フレーム目).

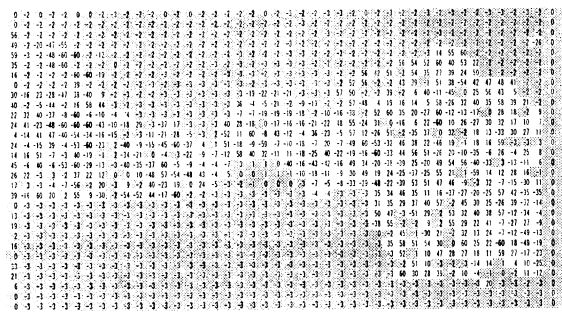


図5. 原画像の水平方向の動きベクトル量.

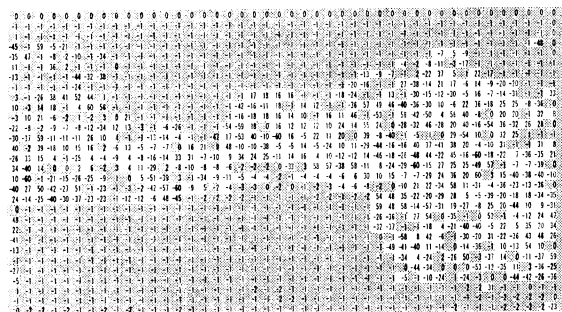


図6. 原画像の垂直方向の動きベクトル量.

3.2 m, n の最適化

図7に平均 N_b (実線), 平均ME速度 (平均 F_s) ((平均 $F_s = \text{FSの平均 } N_b) / (\text{各アルゴリズムの平均 } N_b)$) (破線) とFSのSAサイズ $[-p_3, p_3]$ (図2(c)と同一)の関

係を示す. 本アルゴリズム ($m = 0.5, n = 2$)の平均 N_b は, FSのそれより小さく, $p \geq 40$ で約1/17以下である. つまり, 本アルゴリズムの平均 F_s はFSの17倍以上速い. また, 4:1SS, 16:1SSより, それぞれ約1/4.5, 約1/1.2で, 平均 F_s はそれぞれ約4.5倍, 約1.2倍速い. $p = 60$ の時, 本アルゴリズムの平均 N_b ($= 712$)は, FSの平均 N_b ($= 12,766$), 4:1SSの平均 N_b ($= 3248$), 16:1SSの平均 N_b ($= 841$)のそれぞれ5.58%, 19.86%, 1.01%に減少し, 本アルゴリズムの平均 F_s はFS, 4:1SS, 16:1SSのそれぞれ17.92倍, 4.56倍, 1.18倍に高速化され, SSを超えて最も速いことがわかる.

図8に平均 D_a , 平均ピークS/N比 (平均 R_{sn}) とFSのSAサイズ $[-p_3, p_3]$ (図2(c)と同一)の関係を示す. $p = 60$ の時, 本アルゴリズム ($m = 0.5, n = 2$)の平均最小 D_a ($= 2206.8$)は, FSの平均最小 D_a ($= 2099.9$)の5.09%増 (劣化), 4:1SS, 16:1SSの平均最小 D_a ($= 2290.7, 2580.7$)の, それぞれ4.00%減, 17.81%減 (向上)である. 本アルゴリズムはSSよりも画質が向上している. 同様に, $p = 60$ の時, 本アルゴリズム ($m = 0.5, n = 2$)の平均 R_{sn} ($= 31.5548$ [dB])は, FSの平均 R_{sn} ($= 31.7932$ [dB])の, 0.2384 [dB]減 (劣化), 4:1SS, 16:1SSの平均 R_{sn} ($= 31.4268$ [dB], 31.0160 [dB])の, それぞれ0.128 [dB]増, 0.5388 [dB]増 (向上)である. こちらも本アルゴリズムはSSより画質が向上している.

平均 N_b , 平均 D_a , 平均 R_{sn} と m, n の関係を検討した. 解析結果を図9(a), (b), (c)に示す. m, n の増加に伴い, 平均 N_b は減少している. これは, m, n が大きい時, 小さなSAサイズが割り当てられるためである. $m = 0.25, n = 1$ の時, 平均 N_b は952で最大となり, 平均 F_s はFSの13.4倍である. 一方, $m = 1, n = 2$ の時, 平均 N_b は661で最小となり, 平均 F_s はFSの19.3倍である.

m, n がそれぞれ増加すると, 平均最小 D_a は増加 (劣化)している. これは, m, n が大きい時, 小さなSAサイズが割り当てられ, N_b が減少するためである. 逆に, m, n が小さくなるにつれ, 大きなSAサイズが割り当てられるために, FSと同じ値に近づく. $m = 0.5, n = 1$ の時, 2186でFSの4.1%増であり, $m = 0.5, n = 2$ の時, 2206でFSの5.1%増である. 平均 R_{sn} にも同様の傾向が見られ, m, n のそれぞれ増加に伴い, 平均 R_{sn} は減少 (劣化)している. $m = 0.5, n = 1$ の時, 31.6099 [dB]でFSの0.1833 [dB]減であり, $m = 0.5, n = 2$ の時, 31.5548 [dB]でFSの0.2384 [dB]減である. 図10(a), (b), (c)にFS, 本アルゴリズム ($m = 0.5, n = 1$), 本アルゴリズム ($m = 0.5, n = 2$)の復号画像を示す. 画質の優劣は見られない. 表1にFS, 4:1SS, 16:1SS, 本アルゴリズム ($m = 0.5, n = 2$)の解析結果をまとめる.

表 1. 各種動きベクトル検出アルゴリズムの解析結果.
(Football, MP@ML, 4Mbps, 30fps, $p_1=3, p_2=30, p_3=60$).

アルゴリズム	平均 N_b (%)**	平均 F_s [倍]	平均 最小 D_a (%)**	平均 R_{sn} [dB] (dB)**
FS	12766.4387 (100.00)	1.0000	2099.9078 (± 0.00)	31.7932 (± 0.0000)
従来の4:1SS	3248.3597 (25.44)	3.9301	2290.7142 (+9.09)	31.4268 (-0.3664)
従来の16:1SS	840.8399 (6.59)	15.1830	2580.7142 (+22.9)	31.0160 (-0.7772)
本アルゴリズム*	712.1309 (5.58)	17.9271	2206.7539 (+5.09)	31.5548 (-0.2384)

* : $m = 0.5, n = 2$. ** : 括弧内は FS との比較.

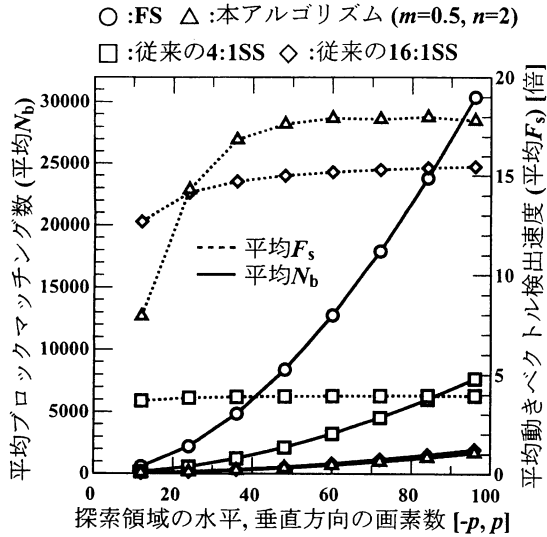


図 7. 平均 N_b , 平均 F_s と探索領域 $[-p, p]$ の関係.

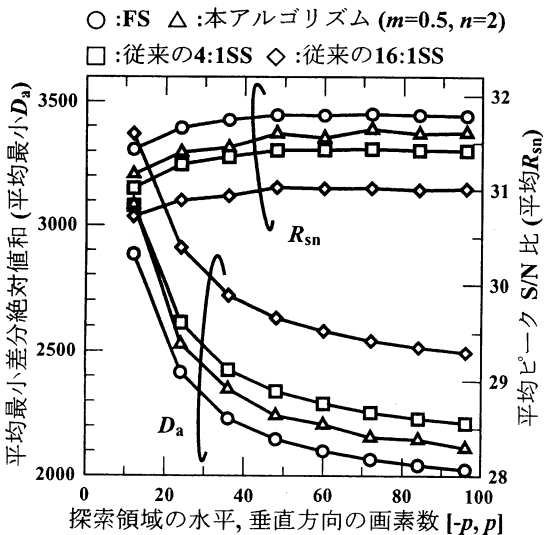


図 8. 平均最小 D_a , 平均 R_{sn} と探索領域 $[-p, p]$ の関係.

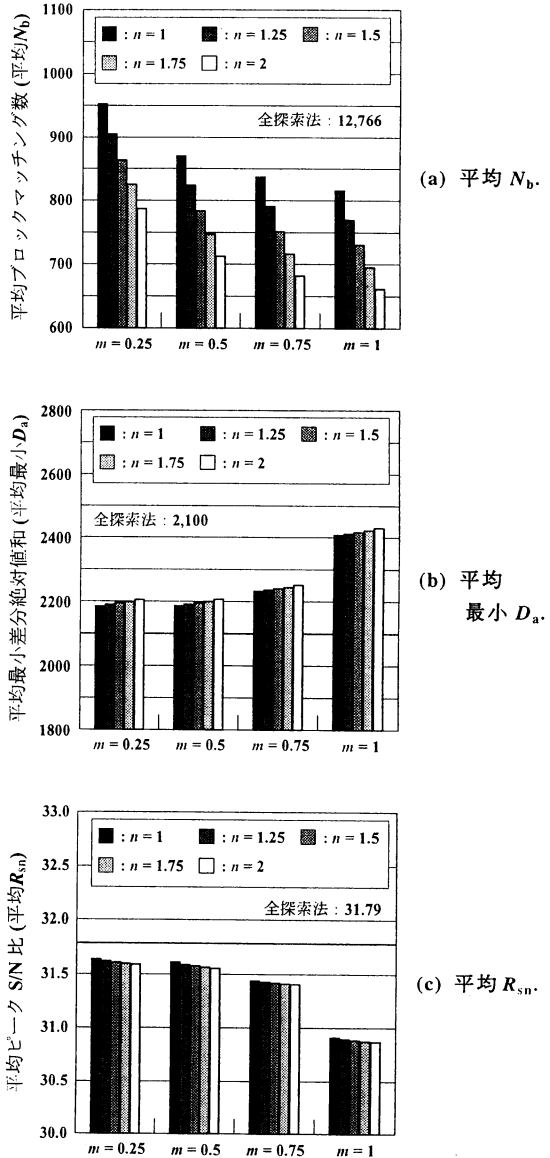


図 9. 平均 N_b , 平均最小 D_a , 平均 R_{sn} と m, n の関係.

4 おわりに

動きの大小に応じて、探索領域 (SA) サイズ $[-p, p]$ の大きさを適応的に設定し、参照フレーム内の SA のマクロブロック (M-Blk) を異なる間引き間隔 (大から小) で階層的に繰り返してブロックマッチング (BM) を行う動きベクトル検出法 (ME)、探索領域を適応設定する 3 階層サブサンプリング動きベクトル検出法を開発した。

テスト画像“Football”(MP@ML, 4Mbps, 30fps, $p_1 = 3$, $p_2 = 30$, $p_3 = 60$) を用いた時、平均最小差分絶対値和 (平均最小 D_a) は 2206.8 となり、全探索法 (FS) の平均 D_a (= 2099.9) の 5.09% の増加となった。4:1, 16:1 サブサンプリング法 (SS) の平均最小 D_a (= 2290.7, 2580.7) に対してはそれぞれ 4.00%, 17.81% の減少となった。平均ピーク S/N 比 (平均 R_{sn}) は 31.5548 [dB] であり、FS の平均 R_{sn} (= 31.7932 [dB]) の 0.2384 [dB] の減少となった。4:1SS, 16:1SS の平均 R_{sn} (= 31.4268 [dB], 31.0160 [dB]) に対しては、それぞれ 0.128 [dB], 0.5388 [dB] の増加となった。本アルゴリズムは画質を維持し (平均最小 D_a を FS の 5.09% 増, 平均 R_{sn} を FS の 0.2384 [dB] 減), SS よりも大幅に向上した。一方、平均動きベクトル検出速度 (平均 F_s) は FS の 17.92 倍に高速化できた。また、4:1SS, 16:1SS の平均 F_s のそれぞれ 4.56 倍, 約 1.18 倍であり、SS よりも速い結果となった。

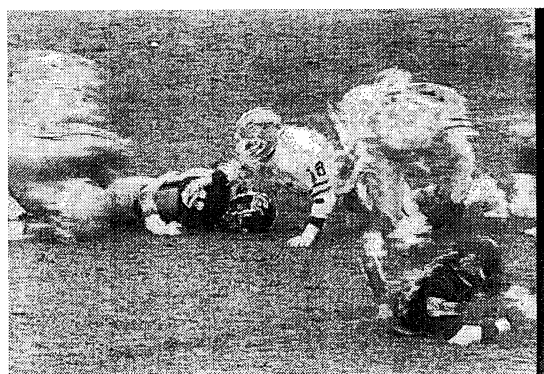
今後の課題として、SA サイズ $[-p_2, p_2]$ の最適化や、 m, n をさらに細かく取る、半画素精度を用いた解析、等を行い、 D_a の削減、さらなる N_0 の減少を目指す。

謝辞

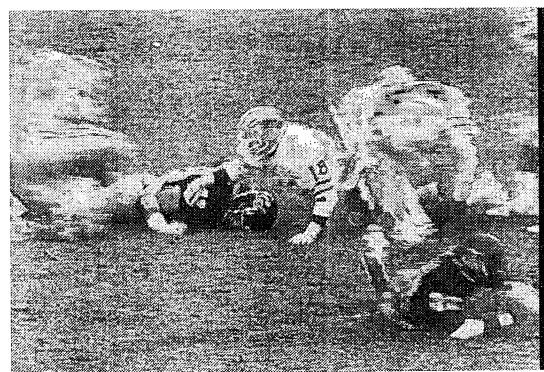
研究にあたり多くの助言を頂きました中央大学榎本研究室の関係諸氏に感謝致します。

参考文献

- [1] 松本, 小田部, 堀口, 榎本, 「階層探索を併用した 4 ビット 2 ステップ中断法と CMOS 差分絶対値和回路」, 信学技報, ICD2000-204, pp.9-14, 2001 年 3 月。
- [2] 木谷, 榎本, 「探索領域サイズを適応的に可変する動きベクトル検出」, 信学総合大会講演論文集, 情報・システム, D-11-61, 2003 年 3 月。
- [3] 小熊, 榎本, 「3 階層ブロックマッチング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, 情報・システム, D-11-39, 2004 年 3 月。



(a)



(b)



(c)

図 10. “Football” の復号画像 (4Mbps, 30fps, $p = 60$).
(a) 全探索法. (b) 本アルゴリズム ($m=0.5, n=1$).
(c) 本アルゴリズム ($m=0.5, n=2$).