

研究速報

2種類の低階調画像を用いた動画の動き推定法

黒崎 正行[†](学生員) 村松 正吾^{††}(正員)
 貴家 仁志[†](正員)

A Motion Estimation Method with Two Images of Less Gray Level

Masayuki KUROSAKI[†], Student Member,
 Shogo MURAMATSU^{††}, and Hitoshi KIYA[†], Members

[†] 東京都立大学大学院工学研究科電気工学専攻, 東京都
 Department of Electrical Eng., Graduate School of Eng.,
 Tokyo Metropolitan Univ., 1-1 Minami-Ohsawa, Hachioji-
 shi, 192-0397 Japan

^{††} 新潟大学工学部電気電子工学科, 新潟市
 Department of Electrical and Electronic Eng., Faculty of
 Eng., Niigata Univ., 8050 Ikarashi 2-no-cho, Niigata-shi,
 950-2181 Japan

あらまし 低階調画像を用いて動きベクトルを推定する場合、動きベクトルの推定精度は、量子化の際のしきい値の値に依存する。本論文では、しきい値を計算するための演算量を低減する二つの方法を提案する。一つは2種類の低階調画像を用いる方法で、他の一つは、マクロブロックを分割し、より細かい単位でしきい値を計算する方法である。これらの二つの方法は組み合わせて使用することも可能である。シミュレーションにより、提案法は、動きベクトルの推定精度を保ち、しきい値決定に要する演算量を低減できることが確認できた。

キーワード 動き推定, MPEG, 低階調画像

1. まえがき

動画データの代表的な国際標準規格に MPEG (Moving Picture Experts Group) がある。MPEGでは離散コサイン変換(DCT)と動き補償を要素技術として採用している。この動き補償を行うために動きベクトルを推定する必要がある。これを動き推定と呼ぶ。この動き推定を行うためには一般にブロックマッチングという方法が用いられている。

ブロックマッチング法は、現在のフレーム内のブロックと誤差の小さなブロックを参照しているフレームから見つける方法で、このブロックの位置の差を動きベクトルと呼ぶ。この探索は一般的には、256階調の画像において実行されている。しかし、動き推定に関する演算量は膨大であり、低コスト化や低消費電力な装置の開発において障害となっている。

そのため、数多くの動きベクトルの推定法が推定に要する演算量や消費電力を低減するために報告されている。それらは、探索する点を低減させる方法[1]~

[5], 探索する点での演算量を低減する方法[6]~[14], の二つに大別される。後者の方法は、画像を表現するために必要なビット数を低減して、演算量の低減を図るのである。

本論文での提案法は、この後者に属する。従来法の主な目的は、どのように低階調画像を生成するかである。線形量子化法[10]は、しきい値を決定する演算量は少なく簡易であるが、低ビットの画像に階調を低減した場合、画像の劣化が激しいことが知られている。画像の劣化を少なくする方法として、平均値法[11], [12]や、メジアンカット量子化法[13], [14]があるが、一般的にこれらの方法はしきい値の計算を複雑にする。

本論文の目的は、推定精度の低下を抑えることのできる平均値法や、メジアンカット量子化法を用いることを前提とし、それらのしきい値の決定に要する演算量を低減する方法を提案することにある。二つの方法が提案される。一つは2種類の低階調を用いる方法で、もう一つはマクロブロックを分割し、より細かい単位でしきい値を推定する方法である。提案法では画像の劣化を極力抑え、演算量を低減できることが確認される。

2. 従来の動き推定法

ここでは、代表的な動きベクトルの推定法であるブロックマッチングについて説明し、次に、それに基づく低階調画像を用いた従来の動きベクトル推定法について説明する。

2.1 ブロックマッチング法

画像をフレーム単位で考え、現フレーム(Current Frame)と参照フレーム(Reference Frame)に着目する(図1参照)。

いま、現フレームをマクロブロック(16×16)に分割し、その中の一つのマクロブロックに対して動きベクトルを推定する。参照フレーム内の対応するマクロ

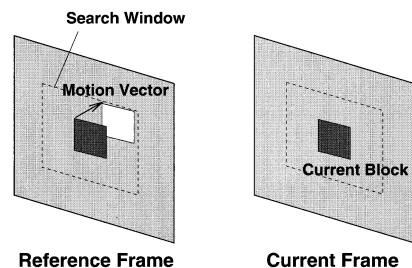


図1 動きベクトルの構成
 Fig.1 A composition of the motion vector.

ブロックと、その周りの値に対して、次の SAD (Sum of Absolute Difference) 値を計算する。

$$SAD(k, l) = \sum_{i=1}^{16} \sum_{j=1}^{16} |p(i, j) - f(i+k, j+l)| \quad (1)$$

ここで、 $p(i, j)$ はマクロブロック内の輝度値、 $f(i, j)$ は参照フレームにおける位置 (i, j) の輝度値である。また、 k, l は、 $-w \leq k, l \leq w$ 内の整数値である。また、この $\pm w$ をサーチウインドサイズという。サーチウインド内において、 $SAD(k, l)$ を計算し、最小の SAD 値を与えるベクトル (k, l) が動きベクトルとして選ばれる。

サーチウインド内のすべての位置において、SAD 値を求め、動きベクトルを推定する方法をフルサーチアルゴリズムと呼ぶ。以下の動き推定はすべてフルサーチアルゴリズムを用いて説明する。

2.2 低階調画像を用いた動き推定法 [6]~[14]

従来の低階調画像を用いた動き推定を簡単に説明する (図 2 参照)。

従来法では、8 ビットの画像を用いてしきい値を決定し、 N ビット ($N \leq 8$) に階調を低減する。 N ビットの画像を用いて式 (1) を計算し動きベクトルを求める。

このように、低階調画像を生成し、ブロックマッチングを行う方法は、8 ビットの画像を用いる場合に比べ、動きベクトルをより少ない演算量で推定できることが知られている [6]~[14]。

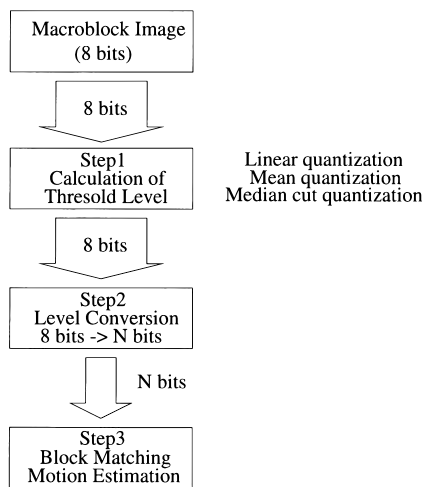


図 2 従来の動き推定法

Fig. 2 The conventional motion estimation using the less-gray-level image.

図 2 の処理を実行するためには、しきい値を求める必要がある。このしきい値の決定法は、いくつか報告されている。以下に、代表的な方法を要約する。

線形量子化法 [10]：しきい値間隔を等間隔に選ぶ方法である。この方法は、しきい値決定が容易である。しかし、しきい値が画像に依存しないため、4 ビット以下 ($N \leq 4$) に階調を低減すると他の方法に比べ、動きベクトルの推定精度は急激に劣化する。

平均値法 [11], [12]：現フレームにあるマクロブロック内の輝度値の平均値をしきい値として使用する方法である。この方法は、画像の性質に依存してしきい値を決定するために、動きベクトルの推定精度が線形量子化に比べ優れている。しかし、しきい値を決定するために、マクロブロックごとに平均操作を実行する必要がある。

メジアンカット量子化法 [13], [14] 現フレームにおいて、マクロブロックごとに対してヒストグラムを求め、しきい値間の画素数が等しくなるよう、しきい値を設定する方法である。この方法は、平均値法と同様に、推定精度が線形量子化に比べ優れている。特に、低ビット ($N=2, 3$) の階調を使用した動きベクトルの推定精度は、平均値法より優れている。しかし、ヒストグラムを求める演算は、他の方法に比べ複雑である。

3. 提案法

平均値法や、メジアンカット量子化法は、しきい値の決定が複雑であるが、高い推定精度を維持することができる。ここでは、このしきい値の決定の複雑さを軽減する二つの方法を提案する。また、これら両方法は組み合わせて用いることもできる。

3.1 2種類の低階調画像を用いる方法

フルサーチアルゴリズムを例にして、提案法の手順を説明する (図 3 参照)。提案法は、四つのステップにより、動きベクトルを推定する。

[ステップ 1] マクロブロックごとに、8 ビットの画像を線形量子化法で L ビット ($L \leq 8$) に階調低減する。

[ステップ 2] マクロブロックごとに、変換された L ビットの画像データからしきい値を決定する (平均値法、メジアンカット量子化法)。

[ステップ 3] L ビットの画像データから、現フレームのマクロブロックを N ビット ($N \leq L$) に階調を低減する。同様に、そのマクロブロックに対応する参照フレーム内も同じしきい値を用いて階調低減する。

[ステップ 4] N ビットの画像を用いて、動き推定を行う。

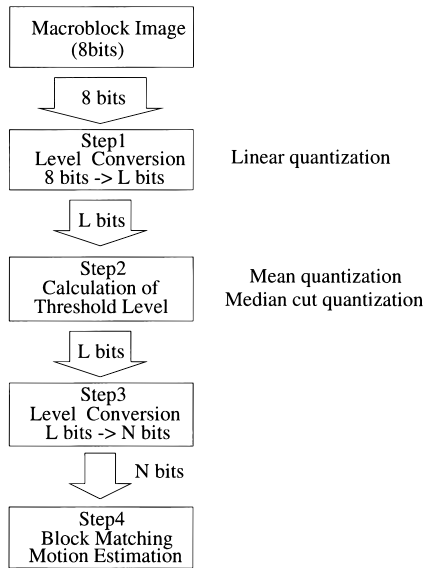


図3 提案法の手順
Fig. 3 A flow of the proposed method.

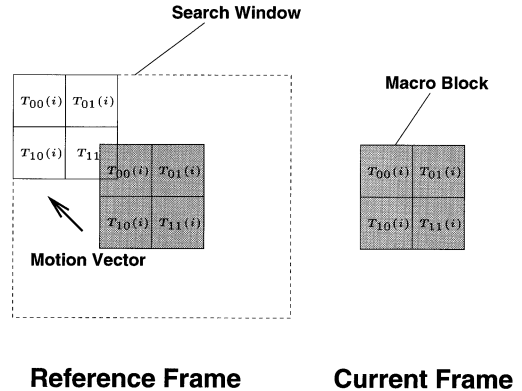


図4 8×8 ブロックを用いた階調低減
Fig. 4 Quantize bits using the 8 × 8 block.

提案法では、ステップ1において、線形量子化を使用しているため、 L ビットの階調変換は容易に実現される。また、ステップ2では複雑な計算を行っているが、 L ビットの計算であるので、従来の方法に比べ、演算量を軽減することができる。

提案法の $L=8$ の場合、すなわち、線形量子化法での前処理を行わない場合は従来法と一致する。

3.2 マクロブロックの分割

MPEG-1 及び 2 では、マクロブロックと呼ばれる 16×16 画素を単位として動きベクトルを推定している。したがって、従来の低階調を用いた動き推定では、このマクロブロックごとにしきい値を決定している。ここでは、 16×16 のブロックを複数の小さなブロックに分割し、それぞれについて、しきい値を決定する方法を提案する。

図4では、マクロブロックを四つの 8×8 画素のブロックに分割する場合を例にして提案法を説明している。現フレームのマクロブロックの四つのブロックごとにしきい値 $T_{00}(i)$, $T_{01}(i)$, $T_{10}(i)$, $T_{11}(i)$ をそれぞれ求め、それらのしきい値を用いて、ブロックごとに階調を低減する。そのマクロブロックに対応する参照フレーム内のブロックも四つのブロックに分割して、それぞれ $T_{00}(i)$, $T_{01}(i)$, $T_{10}(i)$, $T_{11}(i)$ を用いて階調を低減する。

提案法の利点は、しきい値を求めるための画素数を少なくできる点にある。特に、複雑な操作を行うメジアンカット量子化法において、画素数にの増加ともない複雑さが増すので、この方法は有効となる。

3.3 演算量の比較

しきい値の決定に要する演算量を評価し、提案法の有効性を確認する。

3.3.1 2種類の低階調を用いる方法

ここでは、8ビットの加算を1単位とし、ビット数と演算量は比例関係にあると仮定する。すなわち、16ビットの演算量は演算量2とする。また、メジアンカット量子化において、クイックソートアルゴリズムを用いると仮定する。更に、データ交換の操作は、他の演算に等しい演算量であると仮定する。

このような条件のもとで、3.1で述べたように、8ビット画像から一度 L ビットの画像を生成し、その2値化 ($N=1$) に必要なしきい値を求めるための演算量は、表1のようになる。

表1内での $L=8$ ビットは、従来の低階調に基づく推定法に相当する。この表より、メジアンカット量子化法における演算量は L に対して線形で減少することがわかる。また、平均値法も厳密には線形ではないが、 L が減少すると演算量が減少する。

次に、式(1)の計算に要する演算量を考える。8ビットの階調画像 ($N=8$) を用いてサーチウィンドサイズを $\pm w = \pm 15$ [10], [13], [14] とすると、その演算量は 766878 となる。一方、低階調画像 ($N=1$) を用いると、その演算量は 122047 と低減する。この値に、表1のしきい値を求めるための演算量を加えたとしても、

表1 2種類の低階調画像を用いた場合の演算量 ($N = 1$ ビット)Table 1 A number of the calculation using two images of less-gray-level ($N=1$ bit).

L [bits]	Median	Mean
8	2801	337
4	1400	218
3	1050	178
2	700	138
1	350	98

表2 ブロックの分割における演算量

Table 2 A number of the calculation using a divided macro block.

L [bits]	16×16 Block	8×8 Block
8	2801	370

$N=8$ の場合よりも少なく、低階調画像を用いることの有効性が確認できる。

3.3.2 マクロブロックの分割による方法

3.2 で述べた方法の演算量を評価する。ただし、ここではブロック分割の効果を単独に確認するために、8ビット画像 ($L=8$) を直接使用し、 $N=1$ の量子化を仮定する。また、演算量を計算する際の仮定は、上述と同じである。

表2は、メジアンカット量子化法を用いてマクロブロックを8×8の四つのブロックに分割した場合のしきい値を求めるのに必要な1マクロブロック当りの演算量である。演算量は画像に依存するため、0~255までの整数値を疑似ランダムに発生させ、シミュレーションを実行した。表2から、提案法は、メジアンカット量子化において有効であることがわかる。また、表1に示した2種類の低階調画像を用いる方法と組み合わせることが可能であり、より効果的に実行することも可能である。

4. シミュレーション

ここでは、従来法と提案法によって階調を低減し、動きベクトルの推定を行った場合の画素への影響と、ハードウェア構成上の影響について述べる。

4.1 シミュレーション条件と評価法

MPEG-1及び2のエンコード、デコードソフトウェアである文献[15]を用いてMPEG-1に基づき情報圧縮を行う。圧縮を行う画像は白黒のFootball [704画素 × 204画素]とし、ビットレートは10[Mbps]である。また、GOP (Group of Pictures) の構造は周期6のIとPピクチャのみとし、動きベクトルを決定する際の条件は、SAD値(式(1))にのみ従う。つまり、

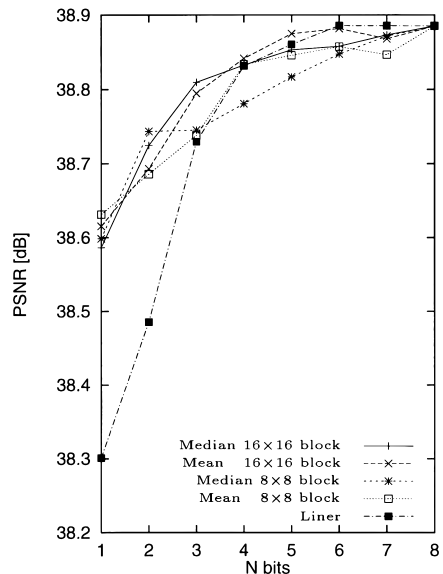


図5 N bits 変換における PSNR の比較 ($L = 8$ ビット)
Fig. 5 Comparison of PSNR ($L=8$ bits).

マクロブロックごとの制御などは行わない。

動きベクトルの推定精度の評価の基準として、伸長された画像の PSNR [dB] (Peak Signal to Noise Ratio), すなわち

$$\text{PSNR} = 10 \log_{10} \frac{\text{ImageSize} \times 255^2}{\sum_{i=1}^{\text{rows}} \sum_{j=1}^{\text{cols}} \{p(i, j) - f(i, j)\}^2} \quad [\text{dB}] \quad (2)$$

を用いる。 $p(i, j)$ は圧縮前のフレームの画素値、 $f(i, j)$ は伸長したフレームの画素値である。また、 rows , cols はそれぞれ画像の横と縦のサイズである。フレームごとの傾向は等しいので、ここでは、Pピクチャ5枚分の PSNR 値の算術平均を求め、結果とする。

しきい値の演算は先に述べた線形量子化法、平均値法、メジアンカット量子化法の三つを用いる。

4.2 ブロック分割の影響

マクロブロックを8×8の四つのブロックに分割し、 $N=1 \sim 7$ ビットの範囲で画像を量子化した。また、ブロック分割を行わない場合についても同様に量子化した(従来法)。ここでは、2種類の低階調画像を使用していない。

結果を図5に示す。横軸は、8 bits から階調低減により、 N bits に変換していることを示し、縦軸はその

ときの PSNR [dB] 値である。図中において、 16×16 block はブロック分割を行わないことを、 8×8 block はマクロブロックを四つに分割することを意味する。また、図 5 はメジアンカット量子化、平均値、線形量子化の三つについて比較を行っている。

この結果より、次のことが確認される。

- 4 ビット階調 (16 階調) の変換までならば、どの階調変換法を使用しても PSNR の差は、 $0.15[\text{dB}]$ 以内である。
- 階調変換を行わない方法 ($N=8$) に比べ 1 ビット画像に変換したとしても、線形量子化変換以外は PSNR の値は $0.3[\text{dB}]$ 以内である。
- 8×8 ブロックを使用した場合は、ビット数が大きいとき ($N=8 \sim 4$) には、他の変換法に比べ PSNR は低くなるが、ビット数が少ないときはその傾向は見られない。

4.3 2 種類の低階調画像の使用

線形量子化により L を 2~7 ビットの範囲で量子化し、メジアンカットと平均値法により、 $N=1, 2$ ビットで量子化を行った。また、ブロック分割を行う場合と行わない場合も合わせて比較した。結果を図 6, 図 7 に示す。この結果から、以下のようなことが確認される。

- $L=7 \sim 4$ ビットの範囲では、2 種類の低階調画像を用いた影響は確認されない。
- $L=1 \sim 4$ ビットの範囲では平均値法が優れているが、 $L=7 \sim 4$ ビットの範囲では、多くの場合メジアンカット量子化法が良好な結果を与える。
- ブロック分割の影響は、 $L=7 \sim 4$ ビットの範囲では、PSNR を向上させ、 $L=1 \sim 4$ ビットの範囲では低下させる傾向にある。

4.4 ハードウェアの構成と合成条件

ここでは、文献 [14] で提案されているハードウェアアーキテクチャに基づいてハードウェア記述言語である VHDL を用いて論理設計し、各種見積りを行う。提案されているアーキテクチャは、しきい値を求めると同時に、階調を低減することが可能であるため、複雑な制御を必要としない。また、このアーキテクチャでは、画像データの処理に依存することなく見積もることが可能である。

見積りには、Synopsys design tools Ver.1998.02 [16] を使用する。LSI 設計用スタンダードセルライブラリとして、VDEC $0.5\mu\text{m}$ CMOS3 層メタルプロセス用九大版 EXDLIB [17] を使用し合成を行う。合成する際の条件を以下に示す。

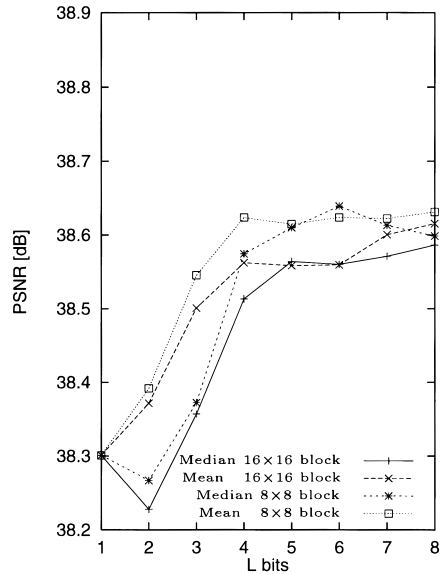


図 6 PSNR の比較 ($N=1$ ビット)
Fig. 6 Comparison of PSNR ($N=1$ bit).

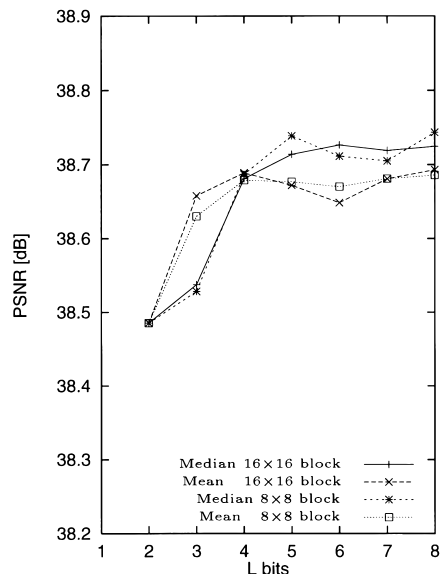


図 7 PSNR の比較 ($N=2$ ビット)
Fig. 7 Comparison of PSNR ($N=2$ bit).

- 動作条件は商用最悪条件 (WCCOM)。
- 配線負荷は " 20×20 "。
- CIF を満たすようクロック制約を設定 [14]。
- クロック以外の全入力信号は、インバータドライブ。

表3 ハードウェアの見積り
Table 3 Estimation of the proposed architecture.

1st quantized bits	$L = 8$			$L = 4$	
	$N = 8$	$N = 1$	$N = 1$	$N = 1$	$N = 1$
2nd quantized bits					
Block size	16×16	16×16	8×8	16×16	8×8
Switching power [mW]	473	342	342	267	248
Total cell area [mm ²]	2.03	1.77	1.60	1.71	1.47

- 負荷として出力信号にインパータを接続。

ここで、CIF とはビデオ信号の共通フォーマットで、 720×480 [pixels]、 30 [frame/s] の 256 階調画像を対象としている [18]。

4.5 ハードウェアの見積もり

2 種類の低階調画像を用いた場合の影響についての調べるため、中間の階調画像を使用しない場合 ($L=8$) と、中間の階調を使用する場合 ($L=4$) について見積りを行った (表 3 参照)。この場合の最終階調はどちらも 1 ビット ($N=1$) である。ここでは、しきい値の決定法はメジアンカット量子化を使用し、入力データには、実画像の統計量を使用する。また、同時に、ブロック分割を行った場合についても示す。

ここで、 $L=N=8$ ビットは低階調画像を使用せず、8 ビットの階調画像を使用して、動きベクトルを推定することを示す。また、 8×8 、 16×16 はブロックの分割の有無を示している。この結果より、以下のようなことが確認される。

- $L=8$ ビットと比較すると、 $L=4$ ビットでは、2 種類の低階調画像を用いることにより、Switching Power を 22[%] (16×16) ~ 27[%] (8×8) 減少させることが可能。
- ブロック分割を行うことにより、Total cell area を 10[%] ($L=8$) ~ 14[%] ($L=4$) 減少させることが可能。
- これらの二つの方法を組み合わせると、従来法に比べ Switching Power, Total cell area とともに効果的に減少させることが可能。

ここで、ブロック分割による面積の減少は、3.2 で述べたように、ブロック分割によってしきい値を求めるための画素数を減少させたことによると考えられる。

5. む す び

本論文では、従来法に比べしきい値の決定のための演算量を低減するために二つのしきい値の決定方法を提案した。一つは、4 ビット付近の階調画像を中間と

して減少する方法で、もう一つは、マクロ・ブロックを細かなブロックに分割し、それぞれにしきい値を計算する方法である。両方は、組み合わせて使用することも可能である。提案した方法は、圧縮時の画像を劣化を抑えて演算量を低減できることが確認された。また、同時にハードウェアの見積りを行い、有用であることを確認した。

文 献

- [1] K. Matsuda, T. Tsuda, T. Ito, and S. Maki, "A new motion compensation coding scheme for video conferences," IEEE Int. Commun. Conf. vol.1, pp.234-237, 1984.
- [2] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol.COM-29, no.12, pp.1799-1808, Dec. 1981.
- [3] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol.38, no.7, pp.950-953, July 1990.
- [4] B. Liu and A. Zaccarin, "New fast algorithm for the estimation of block motion vectors," IEEE Trans. Circuits & Syst. for Video Technol., vol.3, no.2, pp.148-157, April 1993.
- [5] A.M. Tekalp, Digital video processing, Prentice Hall, NJ, 1995.
- [6] J. Feng, K.-T. Lo, H. Mehrpour, and A.E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," IEEE Int. Conf. Image Proc., pp.496-499, Wash. D.C, 1995.
- [7] B. Natarajan, B. Vasudev, and K. Konstantinides, "Low-complexity algorithm and architecture for block-based motion estimation via one-bit transforms," IEEE ICASSP'96, pp.3244-3247, 1996.
- [8] L. Jiang, K. Ito, and H. Kunieda, "Bits truncation adaptive algorithm for motion estimation of MPEG2," IEICE Trans. Fundamentals, vol.E80-A, no.8, pp.1438-1445, Aug. 1997.
- [9] Y. Sasajima and T. Enomoto, "Snapping-off motion vector estimation algorithm using reduced pixel data," Proc. 1998 Information and Systems Society Conference of IEICE, vol.D-11-2, p.117, 1998.
- [10] A. Takagi, S. Muramatsu, and H. Kiya, "Motion estimation with power scalability and its VHDL model," IEICE Trans. Fundamentals, vol.E83-A, no.8, pp.1608-1613, Aug. 2000.
- [11] S. Lee and S.-I. Chae, "Two-step motion estimation algorithm using low-resolution quantization," IEEE ICIP'96, vol.3, pp.795-798, 1996.
- [12] S. Lee, J.-M. Kim, and S.-I. Chae, "New motion estimation algorithm using adaptively quantized low bit-resolution image and its VLSI architecture for MPEG2 video encoding," IEEE Trans. Circuits & Syst. for Video Technol., vol.8, no.6, pp.734-744,

Oct. 1998

- [13] H. Kiya, J. Furukawa, and Y. Noguchi, "Block matching motion estimation based on median cut quantization for MPEG video," *IEICE Trans. Fundamentals*, vol.E82-A, no.6, pp.899-904, June 1999.
 - [14] S. Muramatsu, H. Kiya, and A. Yamada, "A bit-operation algorithm of the median-cut quantization and its hardware architecture," *IEICE Trans. Fundamentals*, vol.E83-A, no.2, pp.320-328, Feb. 2000.
 - [15] MPEG Software Simulation Group, "MPEG-2 encoder/decoder version 1.2," [Online] Available WWW: <http://www.mpeg.org/MPEG/MSSG>
 - [16] Synopsys, Inc., 700 East Middlefield Road, Mountain View, CA 94042 USA, "Online Documentation Version 1998.02," 1998.
 - [17] 塩見謙太郎, 沖野晃一, 川崎隆志, 石原 亨, 安浦寛人, "VDEC 用スタンダードセルライブラリの開発," *信学技報*, CAS97-31, VLD97-31, DSP97-46, 1997.
 - [18] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards, Algorithms and Architectures*, 2nd ed., Kluwer Academic Publishers, 1997.
(平成 12 年 3 月 10 日受付, 6 月 26 日再受付)
-