

H.264における動き補償ブロックサイズ決定手法に関する一検討

常松 祐一[†] 後藤 崇行[†] 渡辺 裕[†]

[†] 早稲田大学大学院 国際情報通信研究科

〒367-0035 埼玉県本庄市西富田大久保山 1011

E-mail: †{tune.goto}@tom.comm.waseda.ac.jp, ††hiroshi.watanabe@waseda.jp

あらまし H.264/MPEG-4 AVC は高い符号化効率を実現する動画符号化方式であるが、符号化処理量が多い問題点がある。符号化処理量の多くは Inter 予測に関連する処理が占めており、H.264/MPEG-4 AVC では特にその比率が高い。従来の動画符号化方式も Inter 予測の占める割合は高かったが、H.264/MPEG-4 AVC では可変ブロックサイズ動き補償、2枚以上の複数参照フレームに加え 1/4 画素精度の動き補償が利用できるようになっており、符号化処理量増加の主たる原因となっている。そこで本稿では符号化対象マクロブロックと探索中に参照するマクロブロックの画素値の差分を利用することで動き補償ブロックサイズの探索を効率よく削減する手法を提案する。

キーワード H.264/MPEG-4 AVC, Inter 予測, 動き補償

A Study on Fast Macroblock Partition Decision for H.264/AVC

Yuichi TSUNEMATSU[†], Takayuki GOTO[†], and Hiroshi WATANABE^{††}

[†] Graduate School of Global Information of Telecommunication Studies, Waseda University.

1011 Okuboyama Nishi-Tomida Honjo-shi Saitama 367-0035 Japan

E-mail: †{tune.goto}@tom.comm.waseda.ac.jp, ††hiroshi.watanabe@waseda.jp

Abstract The performance improvement in H.264/MPEG-4 AVC is provided at the expense of higher computational complexity. Most of the complexity is caused by Inter prediction. To improve coding efficiency, some functions are added in H.264/MPEG-4 AVC, such as variable block size motion compensation, multi reference frame and quarter-pel motion compensation. A fast macroblock partition decision method is proposed in this paper. The macroblock size is efficiently determined by using the pixel value difference between encoding and the referred macroblock.

Key words H.264/MPEG-4 AVC, Inter Prediction, Motion Compensation

1. はじめに

2003年にITU-TとMPEGにより動画符号化方式H.264/MPEG-4 AVC(以後H.264と表記)が国際標準化された[1]。H.264はMPEG-2やMPEG-4と比較して1.5倍から2倍の高い圧縮率を実現することが出来るが、符号化処理量が多い問題点がある。符号化効率を高めるために導入された複数参照フレーム、可変ブロックサイズ動き補償、1/4画素精度動き補償といった機能により、Inter予測が占める割合は高くなっている。動き補償の探索自体に関する高速化は既に数多くの検討がなされているがH.264で新たに追加された機能に関しての高速化手法は十分に検討がなされていないものもある。そこで本稿では動き補償時における可変ブロックサイズの決定手法に着目し、符号化対象マクロブロックと探索中に参照するマクロブロックの画素値の差分を利用することで動き補償ブロックサ

イズの探索を効率よく削減する手法を提案する。

2. Inter 予測

2.1 H.264におけるInter予測

H.264で新たに追加されたInter予測に関連する機能を図1に示す。まず参照フレーム数の増加がある。MPEG-2やMPEG-4では予測の際に参照フレームを1枚しか利用することができなかったが、H.264では2枚以上の複数参照フレームを参照できるようになった。単純に全フレームに対して探索処理を行うと参照フレーム数に比例して動き補償に要する時間が増加するようになる。

さらに可変サイズブロックサイズが導入された。これまで動き補償の単位は16x16画素単位しかなかったが、H.264では階層構造をなす16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4の7種のブロックサイズが利用できるようになった。階層構造をなすとい

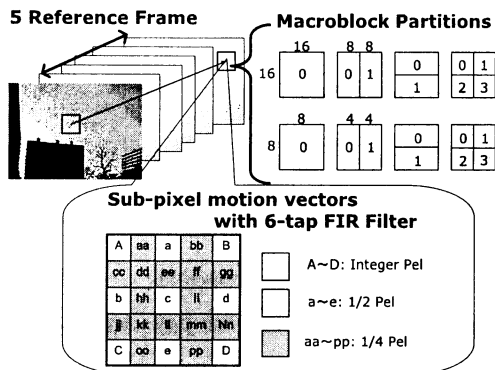


図 1 New Inter Prediction Function in H.264

うのはまず 16x16, 16x8, 8x16, 8x8 のどのサイズを使うか先に決定し、8x8 画素単位の分割を選択したときのみ 8x4, 4x8, 4x4 の小さいブロックサイズが利用できることを意味する。利用できるブロックサイズの種類が増え、全ての組み合わせに対して探索を行うようにすると処理時間の増加を伴う。

最後に動き補償の精度が従来の 1/2 画素精度から 1/4 画素精度へと向上した。1/2 画素精度の画素値は 6 タップの FIR フィルタ^(注1)により計算され、1/4 画素精度の画素値は整数精度の画素値と 1/2 画素精度の画素値を用いて平均値フィルタにより作成される。精度の向上に伴い従来の探索範囲と同じ箇所を探索するにはより広いエリアを探索しなければいけないことになる。

これらの機能追加により、精度の高い動き補償が行えるようになったが、Inter 予測が占める符号化処理量は増加してしまっている。

上記の Inter 予測に追加された機能による符号化効率の違いを確認するために標準動画である QCIF サイズの Foreman, News, CIF サイズの Mobile&Calendar, Tempete を H.264 の参照ソフトウェアである JM [2] で符号化処理を行った。表 1 に実験条件を示す。QP を 28 に設定したときの処理時間の内訳を図 2 に、動き補償ブロックサイズの内訳を図 3 にそれぞれ示す。また利用できるブロックサイズを制限したときの Mobile&Calendar の RD 曲線を図 4 に示す。

処理時間は Intra 予測が全体の 1 割前後を占めるのに対し、16x16~8x8 画素単位までのブロックサイズにおける Inter 予測処理が 2 割前後、8x8 から 4x4 画素単位までのサブブロックサイズにおける Inter 予測処理が 6 割前後を占めている。このことから細かいブロックサイズにおける探索処理を削減するほうが全体に対する効果が高いことがわかる。ブロックサイズの利用割合は動きの少ない News で前のフレームの対応するマクロブロックをコピーして利用するのが目立つが、それ以外の画像では 16x16~8x8 画素単位のブロックサイズが 7 割、8x8~4x4 画素単位のブロックサイズが 1 割前後となっている。小さいブロックサイズを利用することによる符号化効率の違いを図

表 1 Experiment conditions which measure the tendency of Inter Prediction

JM Version	9.6
Profile	Baseline Profile
Input Sequence	Foreman, News (QCIF) Mobile&Calendar, Tempete (CIF)
QP	I,P:22~28
Frame Number	150
Frame Structure	N=15, M=1
Frame Rate	30fps
Reference Frame Number	5
Search Range	16
Optimization	RD-Optimization

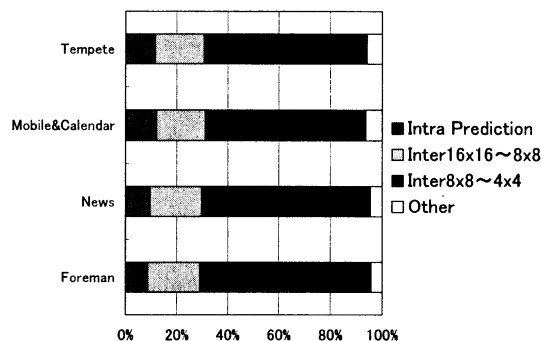


図 2 Percentage of Processing Time

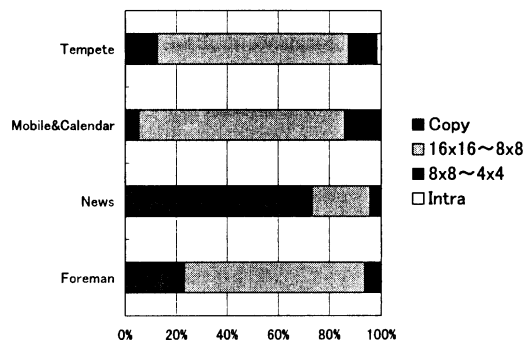


図 3 Percentage of Blocksize

4 に示す。8x8~4x4 画素単位のブロックサイズを利用することによる PSNR の向上は少ないが、16x16~8x8 画素単位に分割することによって 1[dB] 程度 PSNR が向上していることが見て取れる。

以上よりブロックサイズは 8x8 サイズまでの分割は PSNR の観点で非常に有効であり、さらに細かい分割は PSNR が向上するものの全てのマクロブロックで探索を行うと非常に多くの処理量を占めてしまうことがわかった。

2.2 従来手法

Inter 予測の高速化のポイントは大きく 3 つに分けることが

(注 1) : 係数は (1, -5, 20, 20, -5, 1)/32

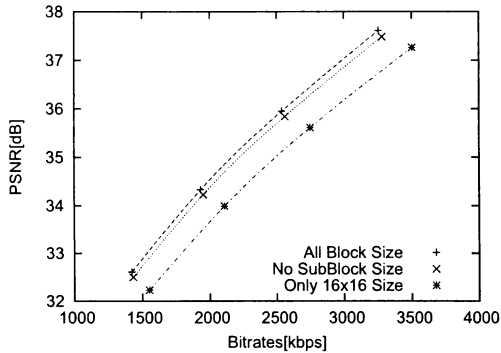


図 4 RD Curve with BlockSize Limit

できる。

- (1) 複数参照フレームの選択
- (2) 動き補償ブロックサイズの決定
- (3) 動きベクトルの探索

動きベクトルの探索は MPEG-2 に対して検討された手法がほぼそのまま利用できるが、複数参照フレームの選択と動き補償ブロックサイズの決定は新たに検討を行う必要がある。動き補償ブロックサイズの決定に関する検討はいくつか行われているが、異なるサイズの予測結果を用いてブロックを統合/分割する手法が多い [3-6]。

文献 [3, 4] は 4x4 や 8x8 といった小さい予測単位で動き補償を行い、動きベクトルの大きさと向きがそろったブロックを統合することで探索処理量を削減する手法を提案している。この手法の問題点として処理量が多い小さいサイズの予測を行わなくてはならず、小さい予測単位で得られる動きベクトルの方が信頼性が低くなってしまう点が挙げられる。

文献 [5, 6] では 8x8 画素単位で始めに動き補償を行い、その後ブロックを統合/分割する手法を提案している。探索処理量の増加、動きベクトルの信頼性という先の 2 つの問題点は解決できるが、ブロックサイズの利用を制限したときに性能が低下する問題が考えられる。

大きい単位から探索を開始し、ブロックサイズの分割を行いながらより小さいブロックの動き補償を行う手法は処理量が最も削減でき、精度の高い動きベクトルが得られることが期待できる。しかし分割を行うかどうかの判断、処理の打ち切りを行うかどうかの判断に用いる閾値によって性能が大きく左右され、閾値を適切に設定しないと処理量が統合するアプローチよりも削減できない問題点がある。大きいサイズから分割し探索を行うアプローチは最も処理量を要する 8x8 サイズ以下の小さいブロックサイズの探索を最も抑ええることができ、また今後新たに登場する動画像符号化方式で動き補償に用いるブロックサイズの種類がさらに増えたときに最も良い性能を発揮すると思われる。よって本稿では大きいサイズから探索を開始し、分割しながら探索を行う手法について検討を行う。

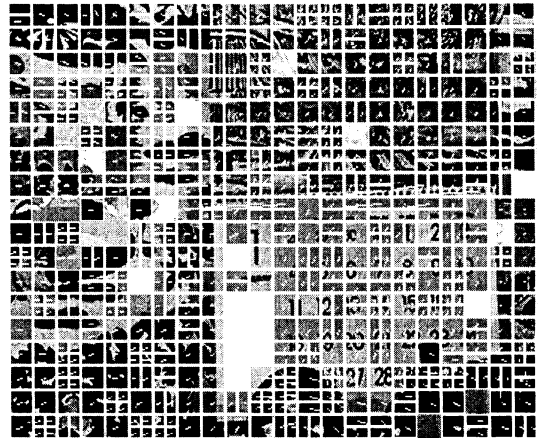


図 5 Ideal Macroblock Partition

3. 提案手法

まず始めに分割形状の判断手法について検討する。動き補償の分割形状を決定するにあたり何らかの情報を必要とするが、例えば分割が動画像中の移動物体の動きに合わせて行われると仮定するとエッジ成分に沿って分割する手法が考えられる。感覚的には動物体の境界に沿ってブロックサイズが分割される方式は自然だが、実際の最適なブロックサイズ分割はエッジ近辺に小さいブロックサイズが集中するもの、異なっているものが一般的である。参考に図 5 に Mobile&Calendar のあるフレームにおける最適なブロックサイズ分割形状を示す。

そこで探索しているブロックサイズを 4 分割した一回り小さい予測単位で SAD^(注2) をとり、判断の基準とする手法を提案する。SAD の大小により、どの位置が上手く動き補償が出来るのかがわかり、実際に動き補償が上手く行っていない箇所に合わせて分割を行うことで良い性能が得られることが期待できる。提案手法を図 6 に示す。大きいサイズから探索を開始し 4 分割した一回り小さいサイズ単位 (16x16 サイズなら 8x8 画素単位, 8x8 サイズなら 4x4 画素単位) で SAD を取りながら、それぞれの箇所において SAD の値と動きベクトルの長さから来るコストの合計値が最小となる位置を記録するようにする。ある程度探索が進んだ後、記録されている位置が揃っているかどうかを見て、打ち切り処理を加えるようにする。4 箇所が同じ位置を指している場合は大きいサイズを採用し、縦横どちらかに 2 箇所ずつ揃っている場合は揃っている向きに合わせて 2 つに分割を行う。大きいサイズでの探索が終了しても処理が打ち切られなかった場合は、3 または 2 つが揃っている場合と 4 つがそれぞれバラバラな位置を指している場合の 3 通りが考えられる。3 または 2 つのブロックが揃っている場合はすべての分割パターンを探索するようにし、指している位置が 4 つバラバラの場合はブロックを 4 つに分割したパターンでのみ探索を

(注2) : Sum of Absolute Difference: 画素値差分の絶対和

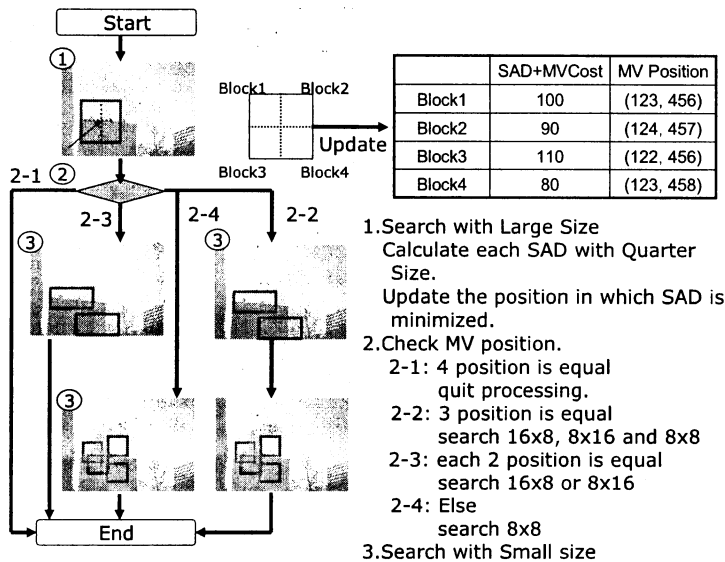


図 6 Proposed Method

行うようにする。16x16 から 8x8 への分割、8x8 から 4x4 への分割は同じ手法を用いるようにする。また分割後の各ブロックにおける探索は分割前の SAD が最小となる位置を中心に行うようにすることで、探索範囲を狭くする。これにより予測サイズが小さくなるほど探索点を削減することが出来る。

次に分割の必要性の判断に用いる閾値について検討を行う。処理を途中で打ち切る手法としては文献 [7] で用いられているような符号化済み周辺マクロブロックから目標 SAD を予測する手法が考えられる。しかしブロックサイズの決定に利用する SAD と動きベクトルを合わせたコスト値はわずかの差で決まることが多く、分割の必要性を判断するのに用いるには向いていないと考えられる。そこで SAD が最小となる位置の揃い方に加え、どれだけの間、値が更新されていないかを判断に用いるようにした。今回は過去 100 回の探索において更新が無かった場合を打ち切り処理の対象とするようにした。

この提案手法は小さいブロックサイズの予測結果から統合する手法と比較して先に挙げた"探索回数が少ない"、"動きベクトルの精度が高い"に加えて、"全体のコストが最小となる探索点が得られる"という利点がある。小さいサイズにおける予測結果から統合を行う場合、大きいサイズが利用されるのは統合前の対応する全ての位置のコストが最小となる時だけである。しかし実際には個々の位置のコストは最小でなくとも、全体のコストが最小となることはあり、統合するアプローチでは対応することが出来ない。文献 [8] のようなブロックサイズ毎に最適な探索手法を適用する手法や Diamond Search [9] などの手法と組み合わせたときに、大きいサイズから探索を行う提案手法では大きいサイズのコストが最小となる点を探索するように探索点が移動するので全体のコストが最小となる点を押さえることが可能となる。

表 2 Experiment Conditions

JM Version	9.6
Profile	Baseline Profile
Input Sequence	Foreman, News(QCIF) Mobile&Calendar, Tempete(CIF)
QP	I, P: 22~28
Frame Number	150
Frame Structure	N=15, M=1
Frame Rate	30fps
MC Block Size	16x16~8x8
Reference Frame Number	1
Search Range	16
Optimization	RD-Optimization
MC Accuracy	Full Pel

4. 実験と考察

4.1 実験

参照ソフトウェアの JM9.6 に提案手法を実装し性能の確認を行った。実験条件を表 2 に示す。ブロックサイズ分割の妥当性を確認するために、参照フレームは 1 とし、利用できるブロックサイズも 8x8 までに制限した。探索ブロックサイズの制限だけを行ったものを Proposed1、分割後の探索範囲を制限したものを Proposed2 として示す。ただし JM の実装の問題から、Proposed2 においては 16x16 の探索により得られた SAD 最小の位置をそのまま小さいブロックにおける動きベクトルとした。

4.2 処理量の削減度合い

提案手法による処理量の削減度合いを表 3 に示す。現在の JM では各マクロブロックごとに探索範囲の全ての位置において SAD をあらかじめ計算しており、打ち切り処理を途中で設

表 3 Search Point Reduction

	Foreman			News		
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	PSNR[dB]	Bitrate[kbps]	Complexity[%]
original	35.29	262.26	-	36.74	144.64	-
16x16	35.15	288.26	-	36.70	150.42	-
Proposed1	35.25	267.04	42.54	36.74	145.59	11.87
Proposed2	35.21	279.57	7.17	36.72	147.50	2.91

	Mobile & Calendar			Tempete		
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	PSNR[dB]	Bitrate[kbps]	Complexity[%]
original	33.48	3914.38	-	34.19	2438.21	-
16x16	33.43	4036.08	-	34.15	2501.36	-
Proposed1	33.48	3934.29	41.67	34.18	2455.12	42.27
Proposed2	33.46	3978.52	6.90	34.17	2475.00	7.00

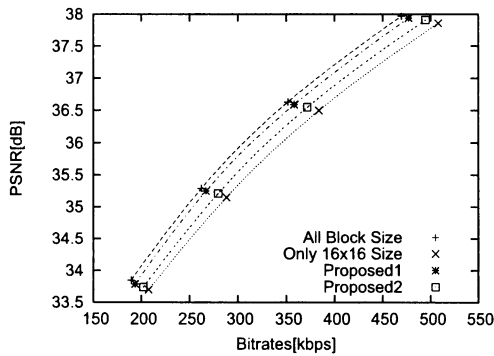


図 7 Foreman RD Curve

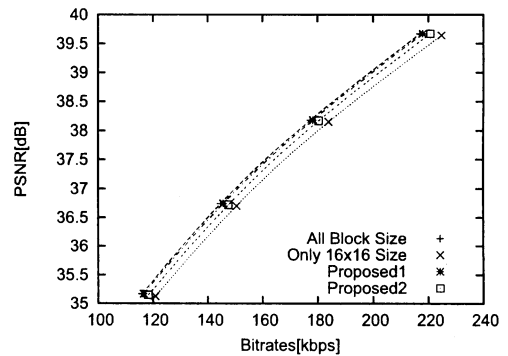


図 8 News RD Curve

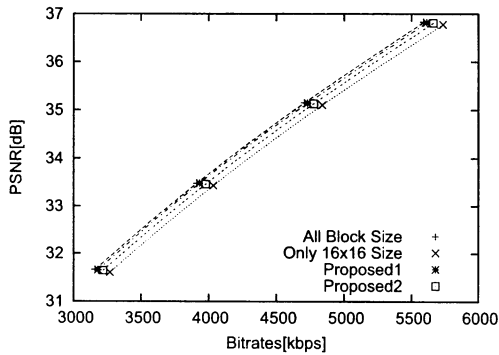


図 9 Mobile RD Curve

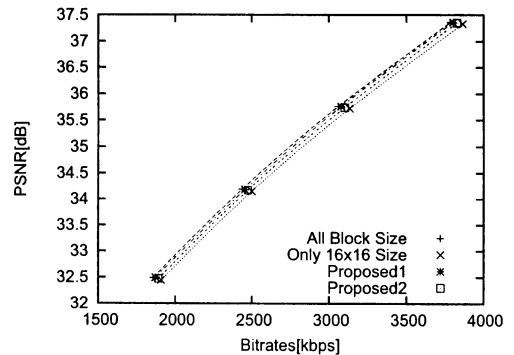


図 10 Tempete RD Curve

けても処理時間にはあまり差がつかないため処理量の削減は探索点の削減度合いで示している。探索は Search Range で指定した範囲全体に対して行われ、Search Range が 16 の場合

$$((16 \times 2) + 1) \times ((16 \times 2) + 1) = 1089[\text{回}]$$

の探索点が 1 つのブロックにおいて調査される。16x8, 8x16 は 16x16 の 2 倍、8x8 は 16x16 の 4 倍の探索処理が行われる。探索ブロックサイズの制限のみで探索を 40%程度まで削減できていることが Proposed1 の結果より見て取れる。この場合の符

号量の増加は少ない。探索範囲を制限した Proposed2 では探索量を 7%まで削減できているが符号量も増加してしまっている。このことから大きいサイズで求めた動きベクトルを小さいブロックサイズにおいて補正する必要があると考えられる。どちらの場合も利用できるブロックサイズを 16x16 だけに制限したときよりも PSNR, 符号量において良い性能が得られていることが見て取れる。

4.3 PSNR と符号量

表 3 では PSNR, 符号量がそれぞれ異なっているため、QP

表 4 Mode Usage Ratio

	Foreman				Mobile & Calendar			
	original	16x16	Proposed1	Proposed2	original	16x16	Proposed1	Proposed2
Copy	3324	3525	3522	3560	2143	2208	2168	2211
16x16	4859	9882	5964	7853	32157	50599	34803	40793
16x8	1965	0	909	620	7137	0	3785	3206
8x16	2418	0	1161	730	6087	0	3443	2679
8x8	1132	0	2086	744	6492	0	9578	4332
Intra16x16	29	41	40	35	98	128	108	120
Intra4x4	133	412	178	318	1326	2505	1555	2099

を変化させて取得したデータによる RD 曲線を図 7 から 10 に示す。これらの図からも提案手法が全ての符号量において 16x16 だけを用いた場合よりも良い性能が得られていることが確認できる。

4.4 符号量増加の原因

最後にモードの利用割合の変化を表 4 に示す。提案手法では符号量の増加が見られるが、表 4 より 16x8, 8x16 といったブロックサイズの利用が減っていることが見て取れる。提案手法では 16x16 と 8x8 の分割判断は上手くできているが、16x8, 8x16 といった分割パターンの判断が上手くできていないと考えられる。これについては今後検討する必要がある。

5. まとめ

H.264 におけるブロックサイズの分割手法について検討し、探索ブロックサイズの 1/4 単位毎に SAD を求めることで分割形状を決定する手法を提案した。提案手法により Inter 予測の探索点を 7% 程度に削減することが出来る。今回の実験では 8x8 サイズまでしか利用していないので、4x4 までブロックサイズを利用できるようにしたときの性能確認と、縦横に分割する場合の判断精度の向上、参照フレームを増やしたときの削減手法が今度の課題といえる。

文 献

- [1] ISO/IEC 14496-10, "Advanced Video Coding," (ITU-T Rec. H.264) 2003.
- [2] <http://iphome.hhi.de/suehring/tml/>, "JVT Reference Software," version 9.6, May 2005.
- [3] Y.K.Tu, J.F.Yang, Y.N.Shen, M.T.Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure With an Adaptive Threshold," Proc. IEEE International Conference on Multimedia & Expo, Baltimore, MD, USA, July, 2003
- [4] M.Kucukgoz, M.T.Sun, "Early-stop and Motion Vector Reusing for MPEG-2 to H.264 Transcoding," SPIE Proc. Visual Comm.Image Proc., San Jose, CA, USA, January, 2004
- [5] Zhi Zhou, Ming-Ting Sun, Yuh-Feng Hsu, "Fast Variable Block-size Motion Estimation Algorithms Based On Merge And Split Procedures For H.264/mpeg-4 Avc," IEEE IS-CAS2004
- [6] X.Li, G.Wu, "Fast Integer Pixel Motion Estimation," JVT-F011, 6th Meeting, Awaji Island, Japan, December, 2002
- [7] Z.Chen, P.Zhou, Y.He, "Fast Motion Estimation for JVT," JVT-G016, 7th Meeting, Pattaya, Thailand, March, 2003
- [8] K.K.Ma, G.Qiu, "An Improved Adaptive Rood Pattern Search For Fast Block-matching Motion Estimation in JVT/H.26L," Proc. IEEE International Symposium on Circuits and Systems, Bangkok, Thailand, May, 2003
- [9] 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 Syst. Video Technol., vol. 8, pp. 369-377, Aug. 1998.