

H.264 Inter 予測における動き補償ブロックサイズ高速決定手法

常松 祐一[†] 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@tom.comm.waseda.ac.jp, ††hiroshi.watanabe@waseda.jp

あらまし H.264/MPEG-4 AVC 符号化方式は符号化処理量が多い問題点があるが、その多くは Inter 予測に関連する処理が占めている。符号化効率を向上させるために可変ブロックサイズ、最大 5 枚までの参照フレーム、1/4 画素精度動き補償が利用できるようになっているが、これらが符号化処理量増加の一因となっている。筆者らは以前に符号化対象マクロブロックと参照フレームの画素値差分を利用する効率的な探索ブロックサイズ選択手法を提案しているが複数参照フレームの探索に適用できない問題点がある。そこで本稿では複数参照フレームに対応した探索ブロックサイズ選択手法を提案する。また探索ブロックサイズ選択の判断精度を向上させ、符号化効率を改善する手法についても検討を行う。

キーワード H.264/MPEG-4 AVC, Inter 予測, 動き補償, 可変ブロックサイズ, 複数参照フレーム

Fast MC Block Size Decision for H.264 Inter Prediction

Yuichi TSUNEMATSU[†] 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@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. We have proposed a fast macroblock partition decision method using the pixel value difference between encoding and the referred macroblock. However this method cannot adapt to case using multi reference frame. Therefore we propose a fast macroblock partition decision method adapt to multi reference frame in this paper. We also study precision of division judgment for the improvement of coding efficiency.

Key words H.264/MPEG-4 AVC, Inter Prediction, Motion Compensation, Variable Block Size, Multi Reference Frame

1. はじめに

2003 年に ITU-T と MPEG により動画像符号化方式 H.264/MPEG-4 AVC (以後 H.264 と表記) が標準化された [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できるが、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 予測に関連する処理量が全体に対して占める割合は高くなっている。

符号化処理を効率よく行うことを目的として数多くの検討が行われているが、H.264 で新たに追加された機能に関しての高

速化はまだ十分に検討がなされていないものもある。筆者らは探索ブロックサイズの選択に着目し、以前に符号化対象マクロブロックと参照フレームのマクロブロックの画素値差分を利用することで探索ブロックサイズを効率的に選択する手法を提案している [2]。この手法は複数参照フレームの探索に適用すると参照フレーム間で探索するブロックサイズの種類が異なり、また探索ブロックサイズ選択の際、縦か横の 2 つに分割する場合の判断精度が低く符号化効率が低下する問題点があった。そこで本稿では複数参照フレームに対応し、探索ブロックサイズの判断精度を改善した手法を提案する。

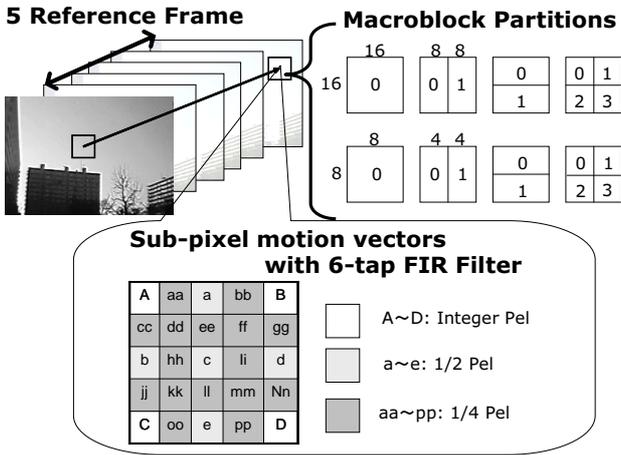


図1 H.264でInter予測に追加された機能

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。

まず利用できる参照フレーム数の増加がある。MPEG-2やMPEG-4では動き補償予測の際に参照フレームを1枚しか利用できなかったが、H.264では複数の参照フレームを利用できるようになった。利用可能な参照フレーム数は利用するプロファイルとレベルにより定義されており、最大で5枚となっている。単純に全フレームに対して動き補償予測処理を行うと参照フレーム数に比例して符号化処理に要する時間が増加する。

さらに可変ブロックサイズ動き補償が導入された。これまで動き補償の単位は16×16画素単位で固定であったが、H.264では16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4画素単位の7種類のブロックサイズが利用できるようになった。ブロックサイズはまず16×16, 16×8, 8×16, 8×8画素単位のうちのどのブロックサイズを利用するか選択し、8×8画素単位の利用を選択した時のみ8×4, 4×8, 4×4画素単位のブロックサイズが利用できるようになっている。動き補償に利用できるブロックサイズの種類が増えたことから、全てのブロックサイズの組み合わせに対して動き補償予測処理を行うと処理時間がやはり増加する。

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。これに伴い従来の探索範囲と同じ範囲を探索する場合4倍の箇所を探索しなければならないことになる。

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り符号化効率が向上した一方、Inter予測が全体に占める符号化処理量は増加している。画像により割合は異なるが、一般的に符号化処理量全体の6割ほどをInter予測に関連する処

理が占めている。特に8×8画素単位以下のブロックサイズの探索処理量が占める割合が多く、Inter予測の処理量のうち7~8割を占める。符号化処理を速く行うには符号化処理量を効率的に削減する必要があり、処理量の多いInter予測処理、特に探索ブロックサイズや参照フレーム、探索範囲を制限することが重要であるといえる。

2.2 従来手法

筆者らが以前に提案した探索ブロックサイズ選択手法を従来手法として図2に示す。大きいブロックサイズから探索を開始し1回り小さいブロックサイズ単位(16×16画素単位の探索を行っている時は8×8画素単位、8×8画素単位の探索を行っている時は4×4画素単位)ごとに、それぞれの箇所において画素値差分の絶対和であるSADの値と動きベクトルの大きさから求めるコストの和が最小となる位置を参照テーブルに記録し、更新する。ある程度の探索点を探索した後、記録されている位置の揃い方を見てその後の探索ブロックサイズを選択する。4箇所が同じ位置を指している場合は大きいブロックサイズの探索のみを行い、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つに分割したサイズの探索を行う。従来手法では4箇所が同じ位置を指している時のみ探索途中の打ち切り処理を加えた。

大きいブロックサイズにおいてすべての範囲の探索が終了した場合以下の基準に従って探索ブロックサイズの選択を行う。3または2つのブロックが指す位置が揃っている場合はすべてのブロックサイズを探索し、指す位置が4つバラバラの場合は4分割である小さいブロックサイズの探索のみを行う。16×16画素単位から8×8画素単位への分割、8×8画素単位から4×4画素単位への分割は同じ手法を用いる。また分割後の各ブロックサイズにおける探索は大きいブロックサイズの探索時に記録した参照テーブルを利用し、コストが最小となる記録された位置を中心に行う。これにより大きいブロックサイズにおける予測処理の結果を再利用することができ、処理が必要な探索点を削減することができる。

探索ブロックサイズを選択することにより符号化処理量を削減する高速化手法として、始めに小さいブロックサイズで探索を行い、動きベクトルの向きと大きさが揃っている時のみ大きいブロックサイズの探索を行う逆のアプローチも提案されている[3, 4]。筆者らが提案した従来手法はこれと比較して以下の利点がある。

- 探索回数が少ない。
- 動きベクトルの精度が高い。
- 任意の動きベクトル探索手法と組み合わせる利用可能。

2.3 従来手法の問題点

従来手法には複数参照フレームの探索において符号化効率が低下する問題点がある。図3に従来手法を複数参照フレームに適用した際の例を示す。探索ブロックサイズの選択を参照フレーム毎に独立に行うと、参照フレーム毎に探索するブロックサイズの種類が異なることになる。H.264の規格ではマクロブロック単位でなく、分割したブロックサイズ毎に参照フレームを選択することができる。図3のように各参照フレーム間で

(注1): 係数は(1, -5, 20, 20, -5, 1)/32

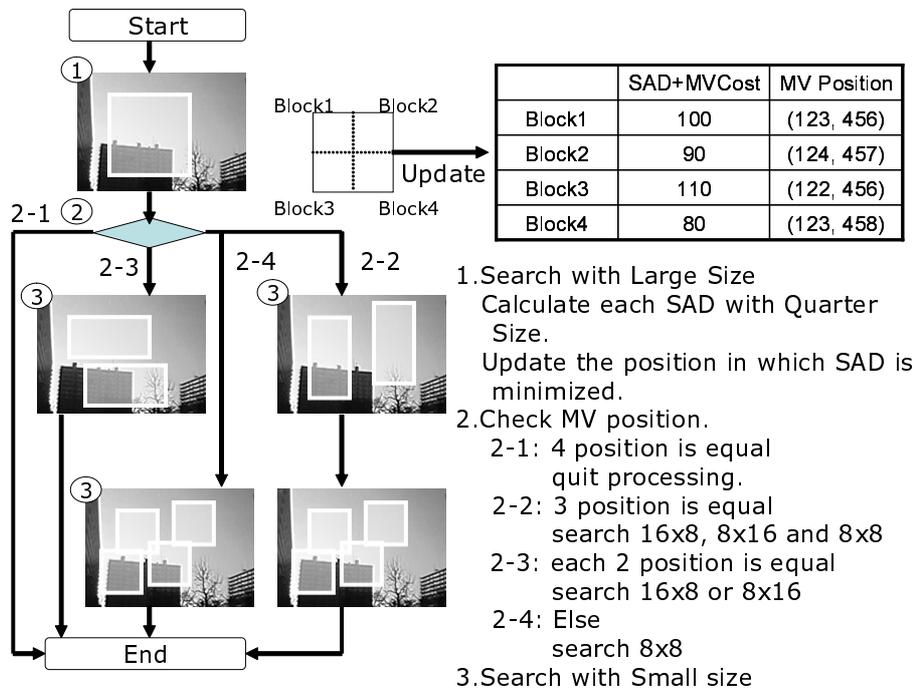


図 2 従来手法

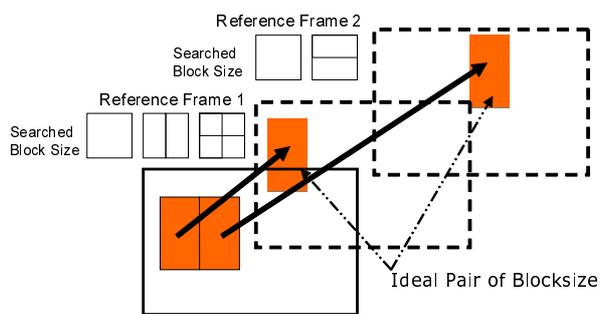


図 3 複数参照フレーム利用時の従来手法の問題点

探索したブロックサイズの種類が異なる場合、もっとも符号量が小さくなるブロックサイズの組み合わせを探し出すことができず、符号量が増加することが考えられる。よって探索するブロックサイズの種類を選択する際に参照フレーム間の整合性を考慮する必要がある。

また従来手法では探索ブロックサイズを選択する際に 2 分割のブロックサイズを探索すべきかの判断精度が低く 16×8, 8×16 画素単位のブロックサイズ選択率が極端に減ることで符号化効率が低下する問題点があった。これはコストが最小となる位置を記録する単位が 1 回り小さいブロックサイズ単位である 4 分割単位であったことが影響していると考えられる。

3. 提案手法

3.1 複数参照フレーム対応

従来手法を複数参照フレーム対応にする方法は 2 通り考えられる。1 つ目は基準フレームを 1 つ選択し探索ブロックサイズの種類を決定、以降探索する参照フレームでは基準フレームで決定した探索ブロックサイズのみを探索する。この手法は実装

が容易であるが基準フレームの選び方によって性能に差が出ることが考えられる。複数参照フレームはオクルージョンが発生したり、動物体が振動する時に効果を発揮するため基準フレームが適切に選択されないと複数参照フレームの利点がうまく生かせない可能性がある。

2 つ目は始めに大きいブロックサイズですべての参照フレームを一度探索し、その後全体の予測結果を利用して探索ブロックサイズを決定する手法である。探索ブロックサイズの選択は 1 つ目の手法よりも精度が向上することが期待できるが、実装が 1 つ目の手法と比較して難しく効率の悪い探索手法になるおそれがある。これは 16×16 画素単位から 4×4 画素単位までのすべてのブロックサイズを利用する場合に顕著に見て取ることができる。

そこで 1 つ目の手法を採用する。ただしすべての参照フレームにおいて同じ種類のブロックサイズを探索すると基準フレームにおいて選択された探索ブロックサイズの種類が多い時に削減できる符号化処理量が少なくなることが考えられる。そこで従来手法で提案した探索ブロックサイズの選択手法を基準フレームに加え、基準フレーム以外の参照フレームでも行い、両方で探索が必要と判断されたブロックサイズのみ探索を行う。これにより不要なブロックサイズの探索処理を減らすことができると考える。

3.2 ブロックサイズ選択精度の向上

探索ブロックサイズ選択精度を向上する案としては 2 分割時のコストを判断に加える手法が考えられる。2 分割時のコストを必要になった時に計算すると SAD 計算のために画素値データをメモリ上に読み込まなければならず、余計な処理が増えることになる。そこで大きいブロックサイズで探索を行っている際に、4 分割単位ごとに対応する 2 分割単位ごとのコストも参

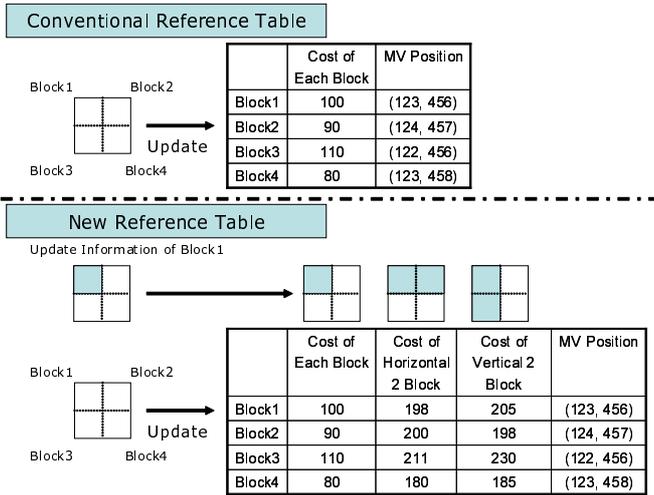


図 4 新たに利用する参照テーブル

参照テーブルに記録する．参照テーブルを更新する処理は大きいブロックサイズの探索時に行うため 2 分割のコストを記録する処理を追加しても使用メモリ量が増加するぐらいで処理量が特別増えることはないと考えられる．図 4 に探索ブロックサイズの選択に利用する参照テーブル構造を示す．従来は参照テーブルに記録された位置の揃い方のみを見て探索ブロックサイズを選択していたが，これに 2 分割単位のコストを判断に加える．具体的には大きいブロックサイズにおける最小のコストと，2 分割単位のコストの和を比較し，2 分割単位のコストが小さい場合，2 分割のブロックサイズを探索する．必然的に探索ブロックサイズの種類は増えることとなり，参照テーブルに記録された位置の揃い方は探索の打ち切りが主な利用用途となる．探索ブロックサイズの種類が増えることで処理量が増加することが考えられるが，これは分割したブロックサイズの探索範囲を制限することで解決できると考えられる．

3.3 提案手法

以上をまとめた提案手法を図 5 に示す．大きいブロックサイズから探索を開始し 1 回り小さいブロックサイズ単位ごとに，それぞれの箇所において画素値差分の絶対和である SAD の値と動きベクトルの大きさから求めるコストの和が最小となる位置を参照テーブルに記録し，更新する．このとき参照テーブルには最小となる位置に対応する 2 分割単位のコストも合わせて記録する．ある程度の探索点を探索した後，記録されている位置の揃い方を見て打ち切り処理を加える．4 箇所が同じ位置を指している場合は大きいブロックサイズの探索のみを行い，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 つに分割したサイズの探索を行う．

記録位置が揃わなかった場合，以下の基準に従って探索ブロックサイズの選択を行う．3 または 2 つのブロックが指す位置が揃っている場合はすべてのブロックサイズを探索し，指す位置が 4 つバラバラの場合は 4 分割である小さいブロックサイズの探索のみを行う．また 2 分割の判断精度を向上させるため，大きいブロックサイズにおける最小コストと，参照テーブルに記録された 2 分割単位のコストの和を比較し，もし 2 分割単位

表 1 実験条件

JM Version	9.8
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	15fps(QCIF) 30fps(CIF)
MC Block Size	16×16 ~ 8×8
Reference Frame Number	5
Search Range	16
Optimization	RD-Optimization
MC Accuracy	Quarter Pel

のコストの和の方が小さければ 2 分割のブロックサイズを探索する．

以降の参照フレームにおいても同様のブロックサイズ選択処理を行うが，最初に探索を行った基準フレームにおいて選択されなかったブロックサイズについては探索を行わない．分割後の探索範囲は従来手法と同様に参照テーブルに記録された位置を中心に行い，探索範囲を制限する．

4. 実験と考察

H.264 の参照ソフトウェアである JM9.8 に提案手法を実装し性能の確認を行った [5]．実験条件を表 1 に示す．

4.1 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 2 に示す．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に設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，整数画素精度の探索は

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

行われる．さらに 1/2 画素精度の修正に 9 回，1/4 画素精度の修正に 9 回の探索点処理が行われる．よって 1 つのブロックに対して 1 枚の参照フレームで 1107 回の探索が探索点が 1 つのブロックにおいて調査される．16×8, 8×16 は 16×16 の 2 倍，8×8 は 16×16 の 4 倍の探索処理が行われる．表 2 の処理量削減度合いは全探索を行った場合の探索点個数 (1107[回] × 5 = 5535[回]) のうち，実際に処理を行った探索点の個数の割合で示している．提案手法は全探索を行った時の JM に近い性能を示しており，従来手法と比較して PSNR, 符号量, 符号化処理量のいずれも改善されていることが見て取れる．符号化処理量の削減は大きいブロックサイズ探索中における処理打ち切り判断基準として，従来の 4 つの箇所が揃っている時に加え，縦横どちらかに 2 箇所ずつ揃っている時を加えたことが影響している．

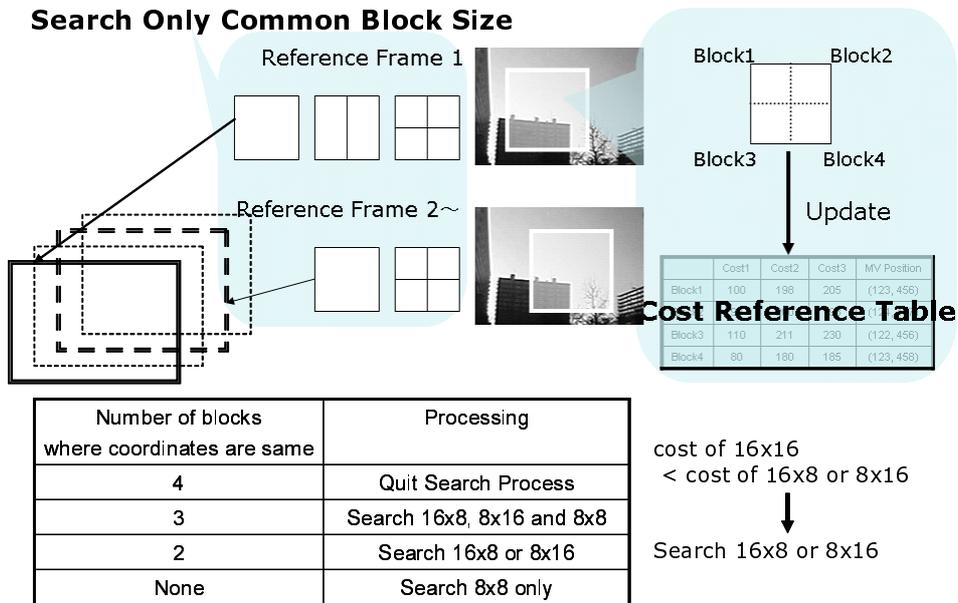


図 5 提案手法

表 2 PSNR, 符号量と処理量の削減

	Foreman			News		
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	35.96	74.94	-	37.11	58.58	-
16x16	35.68	82.79	-	36.92	62.72	-
Conventional	35.82	78.47	9.56	37.00	59.89	4.60
Proposed	35.95	75.82	8.82	37.11	58.74	4.43

	Mobile & Calendar			Tempete		
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	34.16	1988.95	-	34.86	1292.13	-
16x16	33.96	2130.25	-	34.66	1373.41	-
Conventional	34.01	2085.96	8.79	34.72	1351.12	9.02
Proposed	34.13	2019.85	7.94	34.84	1299.88	8.05

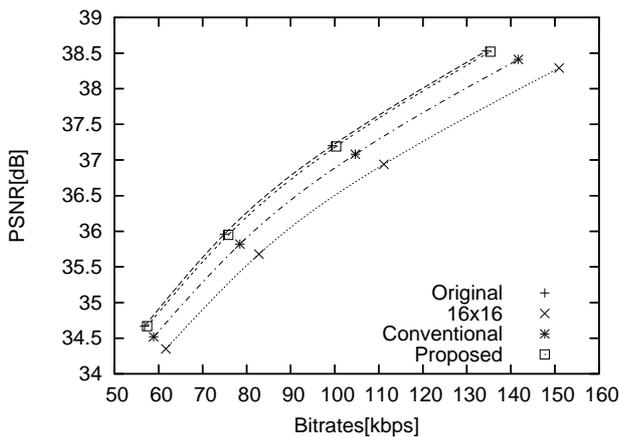


図 6 レート対歪み特性 (Foreman)

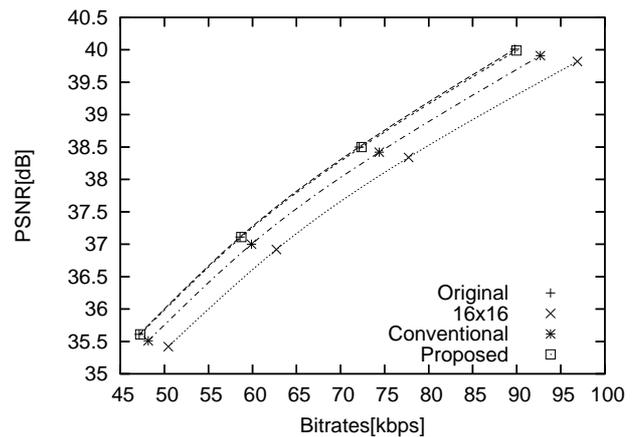


図 7 レート対歪み特性 (News)

4.2 PSNR と符号量

表 2 では PSNR, 符号量がそれぞれ異なっているため, QP を変化させて取得したデータによる RD 曲線を図 6 から図 9 に示す. これらの図からも提案手法がオリジナルの JM に近い性

能を示し, 従来手法と比較しても大きく性能が改善されていることが見て取れる.

符号化効率の大きな改善はモードの利用割合の変化に原因を見ることができる. 各符号化条件におけるモードの利用割合を

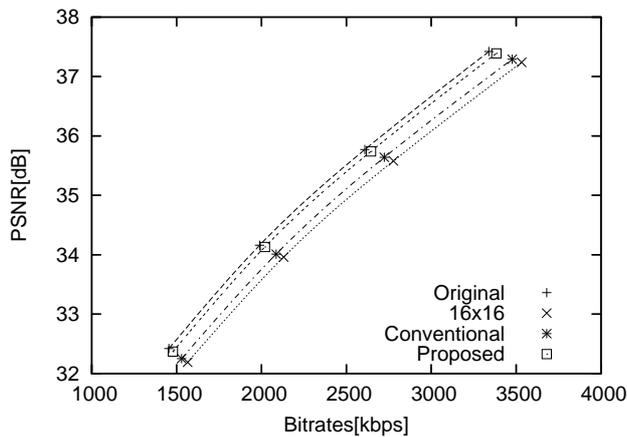


図 8 レート対歪み特性 (Mobile&Calendar)

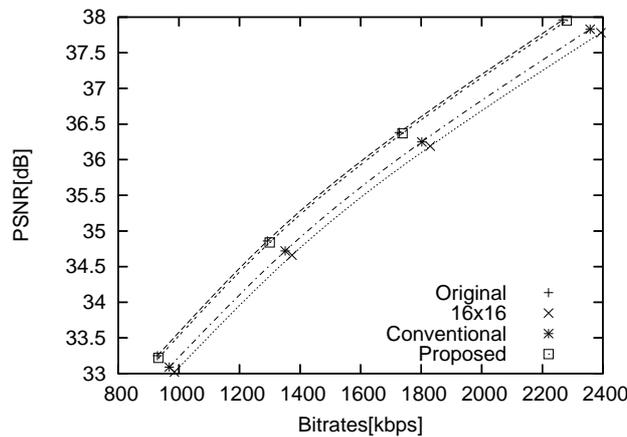


図 9 レート対歪み特性 (Tempete)

表 3 符号化モードの利用割合

	Foreman				Mobile & Calendar			
	Original	16×16	Conventional	Proposed	Original	16×16	Conventional	Proposed
Copy	3280	3486	3535	3202	3062	3981	3785	3180
16×16	4844	10222	7929	5311	19422	51246	44547	25504
16x8	1969	0	299	1802	12183	0	835	9668
8x16	2718	0	438	2573	11893	0	957	10052
8x8	985	0	1571	906	8796	0	5210	6953
Intra16×16	27	42	36	31	64	69	67	54
Intra4x4	37	110	52	35	20	144	39	29

表 3 に示す．従来手法では 16×8, 8×16 といったブロックサイズの利用が大きく減っているが提案手法では 16×16 と 8×8 画素単位のブロックサイズもオリジナルと同様に利用されていることが確認できる．これにより符号化効率が大きく改善されたと考えられる．

5. ま と め

本稿では複数フレームの探索に対応する H.264 におけるブロックサイズ選択手法を提案した．提案手法は筆者らが以前に提案していた従来手法と比較して大きく符号化効率を改善することができ，符号化処理量も同程度に抑えることができる．実験では 8×8 画素単位のブロックサイズまでしか利用しなかったため，8×4, 4×8, 8×4 画素単位のサブブロックサイズを利用したときの性能確認が今後の課題であるといえる．

文 献

- [1] ISO/IEC 14496-10, “Advanced Video Coding,” (ITU-T Rec. H.264) 2003.
- [2] 常松, 後藤, 渡辺, “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,” 情報処理学会 AVM 研究会, June, 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 Re-using for MPEG-2 to H.264 Transcoding,” SPIE Proc. Visual Comm.Image Proc., San Jose, CA, USA, January, 2004
- [5] <http://iphone.hhi.de/suehring/tml/>, “JVT Reference

Software,” version 9.6, May 2005.