

## H.264/MPEG-4 AVC インター予測フレームにおける高速符号化手法 Fast Inter-frame Coding Method for H.264/MPEG-4 AVC Encoder

清水 智行<sup>†</sup>      高木 幸一<sup>†</sup>      米山 暁夫<sup>†</sup>      中島 康之<sup>†</sup>  
Tomoyuki Shimizu   Koichi Takagi   Akio Yoneyama   Yasuyuki Nakajima

### 1. まえがき

H.264/MPEG-4 AVC(以下、H.264)は、従来の MPEG-2/4 の約 2 倍程度の圧縮符号化効率を達成することが可能である一方、非常に多くの符号化ツールを採用しているために符号化処理には従来方式の約 10 倍以上の計算処理負荷を要する。従って、最適な符号化モードを効率的に決定することが必要となる。

H.264 では、特に動き補償予測とイントラ予測の符号化モード数が非常に多いことが計算量の増加の要因となっている。そこで、本研究では動き補償予測の高速化およびイントラ符号化処理の適応的な削減によって、H.264 の符号化処理を高速化する手法を提案する。

### 2. H.264 におけるイントラ/インター符号化

H.264 のインター符号化に関しては、動き補償の単位ブロックサイズを 16x16, 16x8, 8x16, 8x8 の 4 種類、8x8 ブロックのサブブロックモードとして 8x8, 8x4, 4x8, 4x4 の 4 種類利用可能である。

しかし、全てのブロックサイズについて同様の動き探索を行なうと、16x16 ブロックのみの場合に比べて 7 倍の計算量を要する。従って、探索を行なうブロックサイズの削減や探索範囲の縮小などによって高速化を行なう手法が必要となる。このような高速化手法として、これまでは例えば[2]が提案されている。[2]では、4x4 ブロックでの動き探索を最初に行ない、その結果に応じて上位サイズの探索を行なうが、実際には 8x8 未満のサブブロックが選択される頻度は低いことも多く、さらに計算量を削減できる可能性がある。

イントラ符号化に関しては、16x16 ないし 4x4 ブロックに隣接する画素の補間によって予測画像を生成し、元画像との誤差を符号化する、イントラ予測を採用している。画素の補間パターンには、イントラ 16x16 予測において 4 種類、イントラ 4x4 予測において 9 種類存在するため、従来のイントラ符号化に比べて多くの計算量を必要とする。

しかし、インター符号化を行なうフレームにおいては、必ずしも多くのマクロブロック(以下、MB)においてイントラ符号化が選択されるとは限らないため、イントラ符号化を行なうかどうかを適応的に判断する手法が必要となる。従来の MPEG-1/2 等ではフレーム間予測誤差と元画像のそれぞれのアクティビティを比較することによってイントラ符号化モードを選択すべきかどうかを判定することができたが[3]、H.264 の場合はイントラ符号化においても予測符号化を行なうため、元画像のアクティビティをそのままの状態ですべてのイントラ符号化の符号化効率の推定に用いることは必ずしも適さない。

これらの問題を解決するために、本研究ではフレーム間予測およびイントラ/インター予測モードの判定それぞれに着目してインターフレーム符号化の高速化する手法を提

案する。フレーム間予測については 8x8 ブロックでの動き探索を最初に行なうことによってサブブロックでの動き探索の計算量を削減する。イントラ予測については、ブロック内画素から簡単な演算によって求められる特徴量に基づいてイントラ/インター判定を行ないイントラ予測の計算量を削減する。以下に本手法におけるインター符号化フレームの符号化手順の概要を示す。

1. スキップモードによる動き補償予測を行なう(3 章)。
2. 可変ブロックサイズ動き探索を行なう(3 章)。
3. イントラ/インター判定を行なう(4 章)。インターとして判定された場合は、5.に処理を移す。
4. イントラ予測を行なう。
5. 1.-4.のうち実際に符号化を行なったモードについて、最も符号化効率の高いモードを最終符号化モードとして確定する。

以下、3 章では可変ブロックサイズ動き探索の高速化について説明し、4 章ではイントラ/インター判定について述べ、5 章でシミュレーション実験の結果を示し、6 章で結論を述べる。

### 3. 可変ブロックサイズ動き探索の高速化

本手法における可変ブロックサイズ動き探索の手順を以下に示す。

1. スキップモードで動き補償予測を行なう。SAD が閾値以下であればスキップモードと確定して終了する。
2. 8x8 ブロック動き探索を行なう。
3. 16x16, 16x8, 8x16 ブロック動き探索を行なう。
4. ブロックサイズモードを決定する。
5. 4.において 8x8 ブロックが選択された場合、4x4 ブロック動き探索、続いて 8x4, 4x8 ブロック動き探索を行ない、サブブロックサイズモードを決定する。

最初に、スキップモードでの動き予測を行ない、予測誤差の SAD を求める。ここで、SAD が一定の閾値未満であれば、その MB の符号化モードをスキップモードとして確定する。ここで閾値は[3]と同様に 512 とする。

上記の段階でスキップモードと確定しなかった場合、可変ブロックサイズ動き探索を行なう。一度計算した SAD は 4x4 ブロック単位で保存し、他のブロックサイズでの動き探索において再利用する。

動き探索処理においては、まず 8x8 ブロック単位の動き探索を行なう。探索アルゴリズムは Diamond Search (以下、DS)[5]を用いる。ここで求められた 4 個の動きベクトル(以下、MV)に基づいて、上位ブロックサイズでの動き探索を行なうかどうかを判定する[6]。16x16 については 8x8 ブロックの 4 つの MV について互いの距離が閾値  $Thr1$  を超える場合ないし X 成分・Y 成分の分散の和が閾値  $Thr3$  より大きい場合、16x8, 8x16 については、それぞれ横、縦に隣接する 2 対の 8x8 ブロック MV のうち、片方の対のベクトル間距離が閾値  $Thr2$  より大きい場合に、それぞれのサイ

<sup>†</sup>(株)KDDI 研究所, KDDI R&D Laboratories Inc.

ズでの動き探索を行なう。動き探索の範囲は、先に求めた MV 間距離を半径とする正方形の領域に縮小する。ここでは、 $Thr1=4, Thr2=6, Thr3=24$ とした。

次に、 $16 \times 16$  から  $8 \times 8$  までのサイズのうち、動き探索を行なったサイズの中から、符号化効率のもっとも高いサイズをインター符号化モードとして選択する。さらに、 $8 \times 8$  ブロックが選択された場合は、サブブロックでの動き探索を行なう。まず  $4 \times 4$  ブロック単位での動き探索を行ない、上位サイズと同様の手順によって  $8 \times 4, 4 \times 8$  ブロックでの動き探索を行ない、サブブロックモードを選択する。

#### 4. イントラ/インター判定

インター予測処理に続いて、イントラ/インター判定によってインター予測の予測効率がどうかを判定する。これによって、イントラ予測の効果があまり期待できない MB においてイントラ予測処理を省略し、イントラ予測の処理時間を大きく削減する。

イントラ予測は、平坦部やグラデーションなどの MB において予測誤差が小さくなる傾向がある。従って、MB 内の全ての画素について縦方向および横方向の隣接画素値の差分絶対値和をそれぞれ求めて、小さい方の値をイントラ予測効率推定パラメータ  $D_{Intra}$  として用いる。ここで、イントラ予測処理を行なう MB を、オクルージョンやシーンチェンジ等によって動き補償予測の誤差が大きくなる MB や、平坦部等のイントラ予測の誤差が小さくなる MB 等に限定するために、イントラ予測と動き補償予測のうちどちらの予測効率が高くなるかを比較することによって、モード判定を行なう。本手法では、スキップモードにおける SAD を  $D_{Skip}$  として、 $D_{Intra} < Thr4 \times D_{Skip}$  が成立する場合、イントラ予測処理を行なうことと判定する。予備実験の結果、画像およびビットレートの違いにかかわらず  $Thr4 = 1.0$  が適当であることが確認された。

3 章および 4 章に基づいてインター予測およびイントラ予測を適応的に選択して行ない、実際に予測処理を行なった各モードのうち符号化効率の最も高いモードを最終符号化モードとして確定する。

#### 5. 実験と考察

提案手法を H.264 参照ソフトウェア JM 7.3 に実装してシミュレーション実験を行ない、元の JM 7.3 との性能比較を行なった。符号化条件を表 1 に示す。

符号化効率は同一ビットレートにおける PSNR の増加量 (BDPSNR) および同一 PSNR におけるビットレートの増加量 (BDBR) の計算によって行なった [7]。これらの値は、4 個の等間隔の QP 値におけるビットレートおよび PSNR の値からの補間計算によって求められる。結果を表 2 に示す。

PSNR 劣化は、Foreman を除いて概ね  $0.10 \sim 0.22$  dB 程度に抑えられている。しかし、Foreman では約  $0.55$  dB の PSNR 劣化が見られた。これは、主に動き予測の高速化による劣化と考えられるが [6]、顔や手振り等の細やかな動きの多い動画の場合、動きベクトルの符号量が増加しやすく、最適なブロックサイズの選択も難しくなることによるものと考えられる。動きの複雑な画像における動き探索の最適化に関しては今後の検討課題である。

一方、4 つの QP 値における符号化処理時間の平均値を表 3 に示す。提案手法は元の JM 7.3 に比べて約  $75 \sim 86\%$  程

度の符号化処理時間の削減を実現している。特に、Mobile & Calendar に関しては、他の画像サンプルに比べて極端にイントラ予測モードが選択される MB が少ないため、イントラ/インター判定によるイントラ予測処理省略の効果が非常に高いものと考えられる。

#### 6. 結論

本研究では H.264 におけるインター予測フレームの高速符号化手法を提案した。その結果、PSNR の劣化を  $0.1 \sim 0.5$  dB 程度に抑えつつ、 $75 \sim 86\%$  程度の符号化処理時間の削減を実現した。動きの激しい動画における動き探索の最適化やイントラ予測の高速化等が今後の課題である。

表 1: 符号化条件

プロファイル	Baseline Profile
QP	28, 32, 36, 40 (全フレーム固定)
動き探索範囲	$\pm 32$
最適化処理	R-D 最適化, Hadamard 変換

表 2: 符号化効率の比較

サンプル画像	BDBR [%]	BDPSNR [dB]
Container (QCIF, 10fps)	1.59	-0.08
Foreman (QCIF, 10fps)	11.68	-0.55
News (QCIF, 10fps)	2.45	-0.15
Mobile & Calendar (CIF, 30fps)	2.76	-0.13
Silent Voice (QCIF, 15fps)	2.54	-0.13
Paris (CIF, 15fps)	3.79	-0.22
Tempete (CIF, 30fps)	2.41	-0.10

表 3: 符号化処理時間の比較

サンプル画像	平均符号化処理時間 [sec]	
	JM 7.3	提案手法
Container (QCIF, 10fps)	62.08	14.53 [23.40]
Foreman (QCIF, 10fps)	106.21	25.02 [23.56]
News (QCIF, 10fps)	63.70	15.92 [24.99]
Mobile & Calendar (CIF, 30fps)	1205.20	164.25 [13.63]
Silent Voice (QCIF, 15fps)	100.25	23.48 [23.42]
Paris (CIF, 15fps)	425.63	101.26 [23.79]
Tempete (CIF, 30fps)	932.29	185.10 [19.85]

※ []内は対 JM 7.3 比 [%]

#### 参考文献

- [1] ISO/IEC 14496-10, "Information Technology – Coding of audio-visual objects – Part 10: Advanced video coding," 2003.
- [2] Woong Il Choi, *et al*, "Fast Motion Estimation with Modified Diamond Search for Variable Motion Block Sizes," *IEEE Proc. ICIP*, vol. II, pp.371-374, 2003.
- [3] ISO/IEC JTC1/SC29/WG11, "Test Model 5(TM5)," 1993.
- [4] P. Hosur and K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation," *ICICS '99*, Singapore, 7-10, 1999.
- [5] S. Zhu and K. Ma, "A New Diamond Search Algorithm for Fast Block Matching," *IEEE Trans. Image Processing*, vol. 9, No. 2, pp. 287-290, 2000.
- [6] 清水 智行, 米山 暁夫, 柳原 広昌, 中島 康之, 「H.264 符号化処理における動き予測の高速化に関する一検討」, 情処第 42 回 AVM 研究会, pp. 23-28, 2003.
- [7] G. Bjontegard, "Calculation of Average PSNR Differences between RD-curves," *ITU-T VCEG-M33*, 2001.