

オーバーラップ動き補償における窓関数の適応選択

荒木雅裕[†] 松田一朗[†] 伊東晋[†]

[†] 東京理科大学 理工学部

〒278-8510 千葉県 野田市 山崎 2641

E-mail: {araki,matsuda,itoh}@itohws01.ee.noda.tus.ac.jp

ブロックマッチング法によって発生するブロック歪の低減策として、オーバーラップ動き補償方式が提案されている。この方式では、周囲のブロックに対して raised cosine 関数等の固定の窓関数に基づいた重み付け処理を適用し、滑らかな予測画像を生成している。しかし、適切な窓関数の形状は、画像の局所的な性質に応じて変化すると考えられる。本稿では、矩形窓を含む様々な形状の窓関数を用意し、これらをブロック毎に適応選択する方式について検討する。シミュレーションの結果、提案方式によって同一レートにおける再生画像の SN 比を最大で 0.6 dB 改善可能であることが示された。

Adaptive Selection of Window Functions for Overlapped Block Motion Compensation

Masahiro ARAKI[†] Ichiro MATSUDA[†] Susumu ITOH[†]

[†] Faculty of Science and Technology, Tokyo University of Science

2641 Yamazaki, Noda-shi, Chiba 278-8510, Japan

E-mail: {araki,matsuda,itoh}@itohws01.ee.noda.tus.ac.jp

In general, the overlapped block motion compensation technique, which has been developed for reducing blocking artifacts due to block matching based algorithm, uses the fixed window function such as the “raised cosine” function to generate a smoothed predicted image. However, it is expected that an appropriate shape of the window function varies according to local properties of the image to be predicted. This paper proposes an adaptive method which changes the shape of the window function on block-by-block basis. Simulation results indicate that the proposed method provides a coding gain of up to 0.6 dB in SNR compared with the conventional method using the fixed window function.

1 はじめに

オーバーラップ動き補償は、互いにオーバーラップしながら重なり合うブロック単位で動き補償された信号を、連続的な窓関数を用いて合成することで、ブロック歪の無い滑らかな予測画像を生成する手法である^{1) 2)}。通常は、予測画像を生成する際の重みの和が 1 となるように設計された可分型の 2 次元窓関数を、全てのブロックに固定的に適用している。しかし、窓関数にはブロック歪の発生を防ぐだけでなく、信号の加重平均による画素間の内挿や雑音の低減³⁾といった効果も有しており、その適切な形状は画像の局所的なテクスチャや動きの性質、符号化画質などによって変化すると考えられる。

このような考察に基づき、本稿ではフレーム毎に用意した複数の窓関数を、ブロック単位で適応選択する方式について検討する。各窓関数の形状およびそれらのブロックへの割り当て状況は、予測画像に関するレート・歪コストが最小となるよう、フレーム毎に繰り返し最適化している。以下では、具体的な最適化手順を含む本方式のアルゴリズムについて述べると共に、予測誤差の DCT 符号化に基づいたシミュレーション実験により、その有効性を検証する。

2 オーバーラップ動き補償

一般的なオーバーラップ動き補償方式では、マクロブロック (16×16 画素) 単位で動ベクトルを検

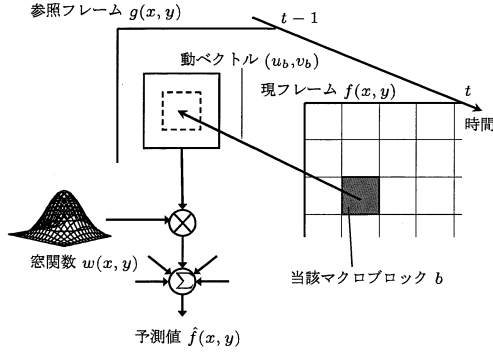


図1 オーバラップ動き補償による予測値の生成手順

出し、更にこの動ベクトルに基づいて動き補償された参照フレーム $g(x, y)$ の領域を、図1のようにマクロブロックより一回り大きいサイズ（例えば 32×32 画素）で切り出している。このようにして得られた信号を、連続的な値を有する窓関数 $w(x, y)$ を乗じて加算することでブロック歪の無い滑らかな予測値 $\hat{f}(x, y)$ を生成することが可能となる²⁾。ここで、マクロブロック b の中心座標を (μ_b, ν_b) 、動ベクトルを (u_b, v_b) とそれぞれおくと、参照フレーム $g(x, y)$ から予測値 $\hat{f}(x, y)$ を生成する手順は以下の式で表せる。

$$\hat{f}(x, y) = \sum_b w(x - \mu_b, y - \nu_b) \cdot g(x + u_b, y + v_b) \quad (1)$$

通常、窓関数 $w(x, y)$ を設計する際には、(1)式における各画素の重みが1となるような拘束条件を課す必要がある。この条件を満たし、かつ連続的な窓関数として、余弦関数の一部を利用した raised cosine 関数や、台形状の trapezoid 関数などが提案されている⁴⁾。

3 窓関数の適応選択

(1)式において、マクロブロック毎に異なる窓関数を適用しようとする、重みの和が1となる条件が満たされず、予測値のレベルが変動して不自然な歪が生じてしまう。そこで本方式では、注目マクロブロック B の内部の画素 $(|x - \mu_B| < 8$ かつ $|y - \nu_B| < 8)$ に関して、常に同種の窓関数 $w_B(x, y)$ を用いて予測値 $\hat{f}_B(x, y)$ の算出を行うも

のとする。

$$\hat{f}_B(x, y) = \sum_{b \in \Omega_B} w_B(x - \mu_b, y - \nu_b) \cdot g(x + u_b, y + v_b) \quad (2)$$

但し、 Ω_B は注目マクロブロックおよびその周囲8近傍に位置するマクロブロックの番号の集合である。本方式では、画像の性質に応じて様々な形状の窓関数を選択できるように、raised cosine 関数⁴⁾を拡張した以下の2次元可分型窓関数 $w_B(x, y)$ を採用している。

$$w(x) = \begin{cases} 1 & (|x| < 8 - d) \\ \cos^2 \left\{ \frac{(|x| + d - 8)\pi}{4d} \right\} & (8 - d \leq |x| < \min(8 + d, 16)) \\ 0 & (\text{otherwise}) \end{cases} \quad (3)$$

$$w_B(x, y) = w(x) \cdot w(y) \quad (4)$$

ここで、 d は窓関数がオーバラップする幅に対応するパラメータである。通常 d の値は8以下となるように設定されているが、本稿ではこれを $0 \sim \infty$ の間で任意の値を許すものとし、図2に示すような様々な形状の窓関数（計19種類）を実現している。但し、 $d > 8$ の場合は、窓関数の影響範囲を隣接8近傍に限定するため、 $|x| \geq 16$ における窓関数 $w(x)$ の値を強制的に0とおく。これにより、窓関数に不連続な部分が生じ、ブロック歪の抑制効果は限定的となるが、ブロックの中心においても重み付け処理が行われるようになるため、雑音低減には有効であると考えられる。また、 $d = 0$ とおくと、 $w(x)$ はマクロブロックサイズに対応した

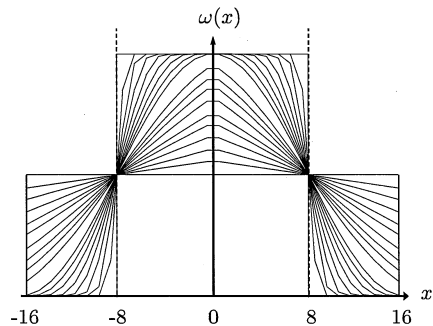


図2 窓関数 $w(x)$

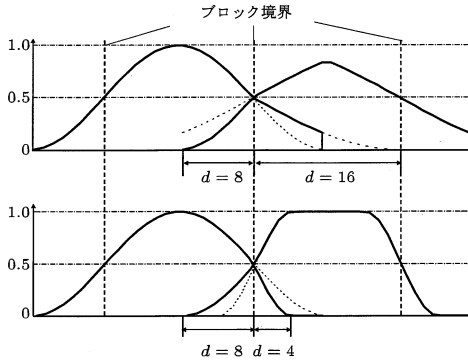


図3 窓関数による重みの例

矩形窓となり、通常ブロックマッチングと同様の処理が可能となる。図3は、隣接ブロックにおいて形状 (d の値) が異なる窓関数が選択された場合の、相互の重みの分布の様子を1次元で表したものである。これより、常に重みの和が1となる条件が満たされている様子を確認できる。

4 パラメータの最適化手順

本方式では、復号側で予測値 $f_B(x, y)$ を生成するため、各マクロブロックの動ベクトル (u_b, v_b) に加え、窓関数の形状に関するパラメータ d の値を送る必要がある。以下にこれらのパラメータの符号化手順を示す。

● 動ベクトル

本方式では、マクロブロック毎に1/4画素精度の動ベクトルを検出している。各動ベクトルの成分 (u_b, v_b) は、符号化が終了した上、左、右上のマクロブロックを参照するメディアン予測に基づいて差分符号化される。なお、1/4画素精度に対応した参照フレーム $g(x, y)$ の値を求めるための内挿処理は、H.264/AVC⁵⁾ と同一の6tapフィルタに基づいて行っている。また、差分動ベクトルの符号化に際しても、H.264/AVCの汎用可変長符号(UVLC)を採用している。

● 窓関数の形状

本方式では、窓関数の種類を図2に示す19通り用意しているため、これらに対応した d の値をマクロブロック毎に直接指定しようとす

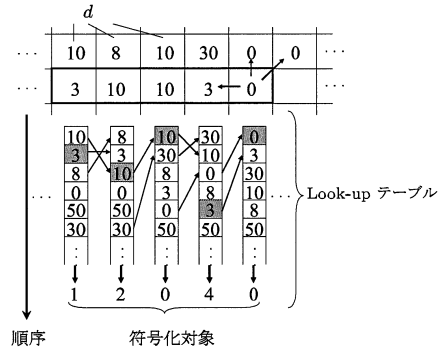


図4 MTFの例

ると、付加情報の増加を招いてしまう。そこで、予めフレーム毎に使用可能な窓関数を M 種類に限定し、対応する d の値のみを Look-up テーブルに格納しておく。更に、マクロブロック毎に MTF (Move-to-Front)⁶⁾ と呼ばれる手法に基づいて、上、左、右上のマクロブロックで選択された d の値が Look-up テーブルの上位に現れるよう並び替えを行う。隣接マクロブロック間で選択された d の値に相関がある場合、図4のように符号化すべき d の値は Look-up テーブルの先頭付近に出現する頻度が高くなるため、その位置(先頭からの順序)を可変長符号化することで、 d の値を効率良く伝送することが可能となる。

本方式では、フレーム毎にこれらの符号化パラメータを、次式で定義されるレート・歪コスト J が最小となるよう繰り返し最適化している。

$$J = D + \lambda \cdot R \quad (5)$$

但し、 D は予測誤差の2乗和 (SSE)、 R は上述した符号化手順によって発生するパラメータの符号量、 λ はラグランジュ乗数をそれぞれ表している⁷⁾。各フレームにおける具体的な最適化手順は以下の通りである。

- (1) 各マクロブロックに対してブロックマッチング法を適用し、1/4画素精度の動ベクトル (u_b, v_b) の初期値を決定する。なお、ブロックマッチングのコストには、窓関数を用いた

重みつき 2 乗誤差と当該動ベクトルの符号量 R_v に基づいた次式を使用している⁸⁾。

$$J' = \sum_{x,y} \left\{ w_b(x,y) \cdot (f(x,y) - g(x+u_b, y+v_b)) \right\}^2 + \lambda R_v \quad (6)$$

但し、窓関数 $w_b(x,y)$ の形状は前フレームと同一（前フレームがイントラ符号化の場合は $d=8$ に固定）とする。また、MTF に用いる Look-up テーブルも前フレームで使用した M 種類の d の値で初期化しておく。

- (2) 各マクロブロックに対して、上記 Look-up テーブルに登録された d の値に対応する窓関数を順次適用し、コスト関数 J の値が最小となるものを一つ選択する。
- (3) Look-up テーブルに登録されている窓関数の更新を行う。具体的には、同一形状の窓関数を使用しているマクロブロックの集合およびその周囲の領域を対象とし、その窓関数を 19 通りのいずれかに変更した場合のコスト関数 J の値をそれぞれ評価する。最終的に J の値を最小とする形状の一つを選び、Look-up テーブルの内容を対応する d の値と置き換える。
- (4) 各マクロブロックおよびその周囲の領域において (2), (3) で選択された窓関数が適用されていることを前提とし、動ベクトルの終点を $\pm 1/4$ 画素の範囲で修正する。その結果、コスト関数 J の値が最小となる状態に動ベクトル (u_b, v_b) の各成分を更新する。
- (5) (2) ~ (4) の手順を、フレーム全体で評価したコスト関数 J が減少しなくなるまで繰り返す。

5 特性評価

図 5 に示す CIF サイズのテスト動画像 (Y 信号のみ, 30Hz, 30 フレーム分) を対象として、符号化実験を実施した。但し、予測誤差信号に対しては、H.263⁹⁾ に準拠した DCT 符号化 (量子化パラメータ Q_p 一定) を適用するものとし、(5) 式および (6) 式に示したラグランジュ乗数の値は、次式により設定した⁷⁾。

$$\lambda = 0.85 \cdot (Q_p/2)^2 \quad (7)$$

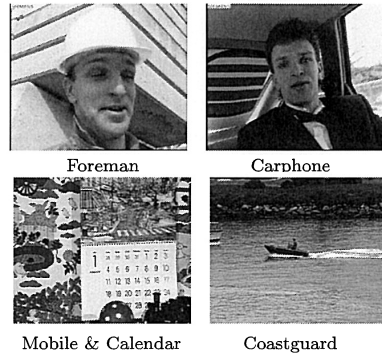


図 5 テスト動画像

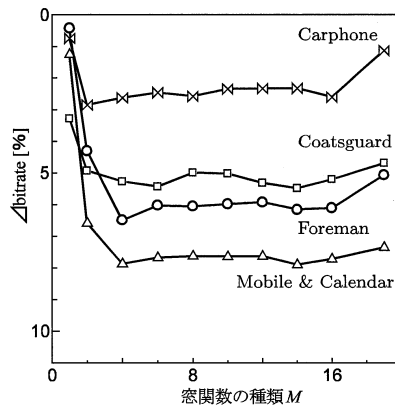


図 6 窓関数の種類 M と Δ bitrate の関係

図 6 は、本方式においてフレーム毎に使用可能な窓関数の種類 M を変化させた際の、符号化効率の推移を調査した結果である。縦軸の Δ bitrate は、従来の固定窓に基づいたオーバーラップ動き補償 (本方式において $d=8$ 一定とした場合に相当) に対して、平均的にどの程度符号化レートが削減されたかを示す指標 (レート削減率) である¹⁰⁾。これより、多くの画像で $M=4$ のときに優れた性能が得られている様子を確認できる。また、そのときの Δ bitrate は、全ての窓関数を使用した $M=19$ の場合と比べて 0.5% 程度良い値を示していることがわかる。これはフレーム毎に使用可能な窓関数の種類を制限したことで付加情報の増加が抑えら

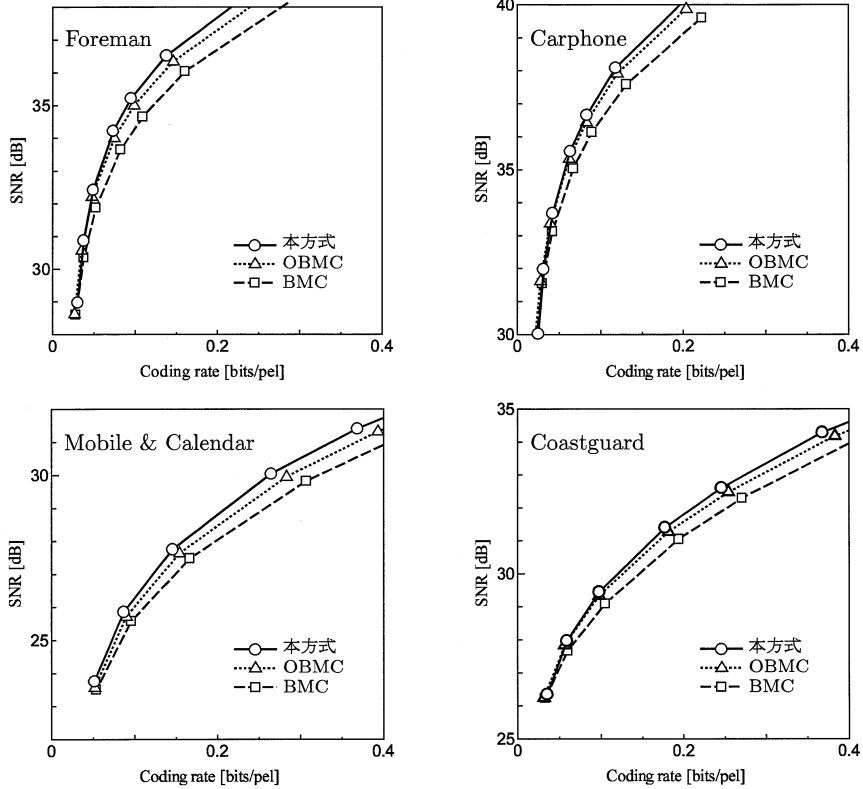


図 7 符号化特性

れ、全体の符号化効率が向上したためである。この結果に基づき、以下本方式では、フレーム毎の窓関数の種類を $M = 4$ と設定するものとする。

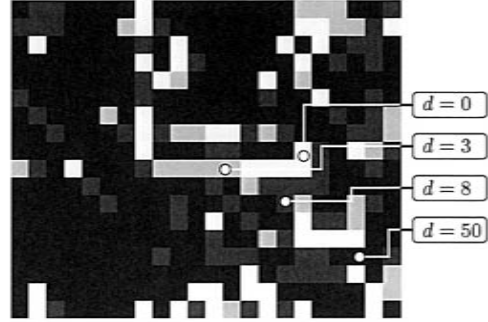
図 7 に、各画像について求めた符号化レートと SN 比の関係を示す。なお、比較方式の OBMC は固定窓による従来のオーバーラップ動き補償、BMC は窓関数を用いないブロックマッチング法に基づいた方式（本方式において、それぞれ $d = 8$, $d = 0$ に固定した場合に相当）を意味している。これより、本方式は OBMC と比較し、同一レートで 0.2 ~ 0.6 dB 程度高い SN 比を達成している様子を確認できる。

図 8 は、画像 Mobile & Calendar（第 2 フレーム）における窓関数の選択状況を示したものである。このシーンでは、カメラがズームアウトしながらカレンダーが上に動き、模型の SL が右から左

へ移動している。カレンダーと壁の境界や SL の周辺では、ブロック間の動きが不連続となるため、従来のオーバーラップ動き補償では参照フレーム上の異なる領域を合成することとなり、エッジなどが 2 重となるゴースト状の歪が発生することがあった。これに対し、本方式では d の値を小さくすることで、周りのブロックの影響を断ち切ることが可能となり、主観的にも良好な再生画像が得られることを確認している。また、壁やカレンダーの絵の部分では d の値が大きな窓関数が選択されている。この種の窓関数は値が不連続な部分を含むが、周囲の動ベクトルが大きく変化しないため、ブロック歪はほとんど発生せず、逆に参照フレームを僅かにずらしながら加重平均することによる雑音低減効果が得られていると考えられる。



原画像 (Mobile & Calendar, 第 2 フレーム)



窓関数の選択状況

図 8 窓関数の選択状況

6 まとめ

本稿では、オーバーラップ動き補償の性能改善策として、窓関数の形状をブロック毎に適応選択する方式について検討した。マクロブロック毎に同種の窓関数を用いて予測画像の生成を行うことにより、各画素における重みの和が 1 となる条件を満たしつつ、矩形窓を含む様々な形状の窓関数を適用することが可能となった。また、フレーム毎に使用可能な窓関数の種類を制限することにより、付加情報の増加を抑えることに成功している。符号化シミュレーションの結果、本方式は固定窓に基づいた従来のオーバーラップ動き補償を用いた場合に対し、再生画像の SN 比を同一レートで 0.2~0.6 dB 程度改善する効果を有していることが確認された。

参考文献

- 1) H. Watanabe *et al.*: “Windowed Motion Compensation”, Proc. of SPIE Visual Communications and Image Processing (VCIP-1991), Vol. 1605, pp.582-589, Nov. 1991.
- 2) C. Auyeung *et al.*: “Overlapped Block Motion Compensation”, Proc. of SPIE Visual Communications and Image Processing (VCIP-1992), Vol. 1818, pp.561-571, Nov. 1992.
- 3) 鈴木 他: “予測信号の平滑化による残差信号の符号量削減”, 第 21 回画像符号化シンポジ

ウム (PCSJ2006), P-5.11, pp.107-108, Nov. 2006.

- 4) J. Katto *et al.*: “A Wavelet Codec With Overlapped Motion Compensation for Very Low Bit-Rate Environment”, IEEE Trans. on Cir. Sys. Video Tech., Vol. 4, No. 3, pp. 328-338, June 1994.
- 5) ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC: “Advanced Video Coding for Generic Audio-visual Services”, 2003.
- 6) 白井 他: “対象画像毎に予測器と算術符号を反復最適化する可逆符号化方式”, 電子情報通信学会技術研究報告, Vol. 102, No.468, pp.19-24, Nov. 2002.
- 7) Gary J. Sullivan *et al.*: “Rate-Distortion Optimization for Video Compression”, IEEE Signal Processing Magazine, pp. 74-90, Nov. 1998.
- 8) 野垣 他: “オーバーラップ動き補償予測に適した動ベクトルの検出方式の検討”, 電子情報通信学会秋季大会, No. D-133, Sep. 1991.
- 9) ITU-T Rec. H.263: “Video Coding for Low Bitrate Communications”, Nov. 1996.
- 10) G. Bjontegaard: “Calculation of average PSNR differences between RD-Curves”, ITU-T SG16, Doc. VCEG-M33, Apr. 2001.