

## 画像局所性を考慮した領域ベース適応補間フィルタ Region-Based Adaptive Interpolation Filter Exploiting Image Locality

松尾 翔平, 坂東 幸浩, 高村 誠之, 如澤 裕尚

Shohei MATSUO, Yukihiro BANDO, Seishi TAKAMURA and Hirohisa JOZAWA

### 1 はじめに

小数画素精度の動き補償の性能改善のため、固定係数を用いた H.264/AVC の補間フィルタに対する改良として、画像に応じて最適な補間フィルタ係数を導出して適用する適応補間フィルタが提案されている。本稿では、従来の適応補間フィルタの問題点について言及し、その改善方法を提案する。

### 2 従来補間フィルタとその問題点

従来の適応補間フィルタ [1] はフレーム全体に対して最適なフィルタ係数を導出するため、画面全体が均一な性質を有している場合は最適な手法と言えるが、画面が複数の特徴的な領域に分割される場合、各領域ごとにフィルタ係数を最適化することで、残差エネルギーをより削減できると考えられる。そこで、著者らは領域ベース適応補間フィルタ [2] を提案した。

上述の提案フィルタは、領域分割手法として動きベクトル (以下、MV) の成分や向きを利用する分割や空間 (左右/上下) 分割を用いていたが、画像の有するテクスチャ情報を利用する分割方法は未検討であった。テクスチャ情報の1つとして、本稿では画像中のエッジに着目した。これは、エッジが画像の構造や模様といった重要な性質を示す指標と考えられるためである。複雑な模様を持つ物体と単純な模様の背景を有する画像の場合、物体と背景で最適な補間フィルタ係数は異なると推測できる。また、フレーム単位で最適な分割手法を選択する機構も導入し、性能改善を試みる。

### 3 領域ベース適応補間フィルタ

フレーム単位で最適な領域分割手法を選択し、分割された領域単位で補間フィルタ係数を最適化する提案手法のアルゴリズムを詳述する。

**Step 1:** 動き補償のブロック単位で最適な MV を導出する。補間には、H.264/AVC の固定係数フィルタを用いる。

**Step 2:** 事前に定義された  $m$  種類の分割手法の中から、1つを選択し、その分割手法に応じて、ブロック単位で領域番号 (1~ $n$ ) を付与する。例えば、本稿で新たに追

加されたエッジ分割手法が選択された場合、**Step 1** で得られた MV が指し示す参照画像のエッジを計算する。対象ブロックに対して、Sobel フィルタを水平および垂直方向にかけ、その両成分から方向を求め、これによりブロックを  $n$  種類に分類し、領域番号を付与する。

**Step 3:** 領域分割後、各領域ごとに最適な水平方向のフィルタ係数を導出し、得られたフィルタ係数を用いて、領域ごとに水平方向の補間処理を行う。導出には、式 (1) に示される予測誤差エネルギー関数  $E_h$  を用い、予測誤差エネルギーが最小化するように連立方程式を解いてフィルタ係数を求める。

$$E_h(\alpha_{m,n}) = \sum_{(x,y) \in \alpha_{m,n}} \left( S_{x,y} - \sum_{c_i} w_{c_i} \cdot P_{\hat{x}+c_i, \hat{y}} \right)^2 \quad (1)$$

なお、 $\alpha_{m,n}$  は各領域を示しており、 $m$  は分割手法番号、 $n$  は領域番号を示す。 $S$  は原画像、 $P$  は復号済参照画像、 $x$  および  $y$  はそれぞれ画像中の水平および垂直方向の座標を示す。また、 $\hat{x} = x + MV_x - L_h/2$  であり、 $MV_x$  は MV の水平方向成分、 $L_h/2$  は位置調整のためのオフセット ( $L_h$  は水平方向フィルタ長) となる。 $w_{c_i}$  は求めるべき水平方向フィルタ係数群  $c_i (0 \leq c_i < L_h)$  を示す。**Step 4:** **Step 3** と同様に、各領域ごとに最適な垂直方向のフィルタ係数を導出して、得られたフィルタ係数から領域ごとに垂直方向の補間処理を行う。具体的には、式 (2) に示される予測誤差エネルギー関数  $E_v$  を最小化するフィルタ係数を求める。

$$E_v(\alpha_{m,n}) = \sum_{(x,y) \in \alpha_{m,n}} \left( S_{x,y} - \sum_{c_j} w_{c_j} \cdot \hat{P}_{\hat{x}, \hat{y}+c_j} \right)^2 \quad (2)$$

なお、 $\alpha_{m,n}$ ,  $m$ ,  $n$ ,  $S$ ,  $x$  および  $y$  は **Step 3** の定義と同じである。 $\hat{P}$  は **Step 3** で得られた水平方向補間処理画像を示す。また、 $\hat{x} = 4(x + MV_x)$  であり、 $MV_x$  は丸められた MV の水平方向成分を示す。 $\hat{y} = y + MV_y - L_v/2$  で表現され、 $MV_y$  は MV の垂直成分、 $L_v/2$  は調整のためのオフセット ( $L_v$  は垂直方向フィルタ長) を示す。 $w_{c_j}$  は求めるべき垂直方向フィルタ係数群  $c_j (0 \leq c_j < L_v)$  を示す。

**Step 5:** 水平および垂直方向の補間処理が終わった画像に対して、再度動きベクトルの探索を行う。

表 1: 領域分割手法の定義

| 分割番号 | 領域 1                        | 領域 2      |
|------|-----------------------------|-----------|
| 0    | 分割なし (従来手法 [1] 相当)          |           |
| 1    | $\alpha_x < MV_x < \beta_x$ | otherwise |
| 2    | $\alpha_y < MV_y < \beta_y$ | otherwise |
| 3    | MV: 第1/第3象限                 | otherwise |
| 4    | $MV_x > 0$                  | otherwise |
| 5    | $MV_y > 0$                  | otherwise |
| 6    | 画面左半分                       | 画面右半分     |
| 7    | 画面上半分                       | 画面下半分     |
| 8    | エッジが垂直                      | エッジが水平    |

**Step 6:** 定義された領域分割手法を全て試行し、その中で最小のレート歪みコストを実現する分割手法番号を取得する。

**Step 7:** 分割手法番号、各種フィルタ係数、MV、テクスチャ情報などの符号化を行い、次フレームに移行する。

領域番号はブロックごとに付与されるが、復号済みの情報から導出可能であるため、符号化する必要はない。また、本提案手法は輝度信号だけでなく、色差信号にも同様に適用可能である。なお、本稿の評価実験では、 $L_h = L_v = 6$ とした。

#### 4 評価実験

次世代映像符号化参照ソフトウェア KTA2.6r1[3] に提案手法を実装し、従来手法 [1] との比較を行った。標準化にて使用されている画像 18 種類を用い、各画像 15 もしくは 30 枚のフレームを IPPP(先頭のみ I) 符号化した。量子化パラメータは  $QP(I) = \{22, 27, 32, 37\}$  ( $QP(P) = QP(I) + 1$ ) の 4 点を採用した。本実験で用いた領域分割手法を表 1 に示す。領域数は 2 で固定し、 $\alpha$  および  $\beta$  は MV のヒストグラムから算出される閾値 (領域 1 と 2 のブロック数がほぼ同等になるように決定される) である。分割手法 1 か 2 が選ばれた場合、 $\alpha$  と  $\beta$  は別途符号化する。

以上の条件により得られた実験結果を表 2 に示す。H.264/AVC の 6 タップ固定フィルタを用いた場合に対する従来手法の符号化利得 BD-rate[4] を導出し、同様に提案手法による BD-rate を求め、それぞれの値を示した。Set ごとに領域分割手法の数を変えており、Set 1 は 4 種 (表 1 の 0, 6, 7, 8)、Set 2 は 8 種 (8 以外全て [2])、Set 3 は 9 種 (表 1 全て) である。従来手法 [1] は、H.264/AVC に対して全画像平均で約 4.61[%] 改善していたが、提案手法 Set 3 の場合、H.264/AVC に対して全画像平均で約 5.04[%] 改善していた。

表 2: H.264/AVC に対する符号化利得 (BD-rate[4] [%])

| 解像度         | 従来 [1] | Set 1 | Set 2 | Set 3 |
|-------------|--------|-------|-------|-------|
| 2560 × 1600 | -4.70  | -4.94 | -5.05 | -5.14 |
| 1920 × 1080 | -5.53  | -6.03 | -6.27 | -6.34 |
| 1280 × 720  | -5.47  | -6.00 | -6.03 | -6.03 |
| 832 × 480   | -3.76  | -4.05 | -4.16 | -4.14 |
| 416 × 240   | -3.63  | -3.59 | -3.55 | -3.54 |
| 平均利得        | -4.61  | -4.92 | -5.02 | -5.04 |

ほとんどの画像で、提案手法による効率改善が確認できたが、最小の解像度においては従来手法が上回っていた。これは低解像度ほど、領域分割時に発生する補間フィルタ係数の符号量、すなわち、オーバーヘッドが支配的になるためと考えられる。領域分割手法の数を増加させることで、ゲインの上昇が確認できた。これは、最適な分割手法の選択肢が増えたためと考えられる。

領域分割手法を増やせば、効率はより向上すると考えられるが、その分、計算処理量が増加する。符号化時間に関して、H.264/AVC に対する従来手法の増加率は約 1.35 倍であったのに対し、提案手法の増加率は約 2.43 倍 (Set 1)、約 3.86 倍 (Set 2)、約 4.22 倍 (Set 3) となっていた。復号時間に関しては、従来手法の増加率は最大で約 1.10 倍程度であったが、提案手法の増加率は最大で約 1.30 倍程度であった。符号化性能と計算量に適した分割手法を設定することが可能となる。

#### 5 まとめ

本稿では、領域分割を行って領域ごとに補間フィルタ係数を最適化する適応補間フィルタについて検討した。新しい領域分割手法を加え、その利得を示し、領域分割手法数を変化させた場合の利得変動に関する考察も行った。今後は、効率的な領域分割手法の導入と高速化に関する検討を行う予定である。

#### 参考文献

- [1] S. Wittmann et al., "Separable adaptive interpolation filter for video coding," in Proc. ICIP2008, Oct. 2008, pp. 2500-2503.
- [2] S. Matsuo et al., "Region-based adaptive interpolation filter for motion compensation," in Proc. IWAIT2010, Jan. 2010.
- [3] "KTA reference software ver. 2.6 revision 1," <http://iphome.hhi.de/suehring/tml/download/KTA/>, Nov. 2009.
- [4] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16 Q.6 VCEG, VCEG-M33, Apr. 2001.