

LJ_010

H.264/MPEG-4 AVCの重み予測を利用したフェード編集に関する一検討 A Study on Fade Editing using Weighted Prediction in H.264/MPEG-4 AVC

加藤 晴久[†]
Haruhisa KATO

米山 暁夫[†]
Akio YONEYAMA

滝嶋 康弘[†]
Yasuhiro TAKISHIMA

1. 序論

映像コンテンツの編集に用いられる様々な特殊効果の中で、場面転換を効果的に表すトランジションとしてフェードが多用されている。しかし、フェードのような輝度変化を伴う映像は、MPEG-1/2/4など従来の映像圧縮方式では効率的に符号化できなかつた。一方で、携帯端末向けの地上波デジタル放送(ワンセグ)や次世代DVDのBlu-ray及びHD DVDでは、映像圧縮方式としてH.264/MPEG-4 AVC [1]を採用している。H.264では輝度変化を補償するために重み予測(Weighted Prediction)が導入され、フェード区間の符号化効率を大幅に向上することができる[2]。本稿では、H.264で符号化された映像コンテンツが与えられたとき、H.264の重み予測を利用して高速にフェードを実現する方式について考察する。

2. 従来のフェード編集技術

フェードアウトを実現する方法としては、画素領域で編集する方式と符号領域で編集する方式がある。ベースバンド変換は、符号化された映像コンテンツを復号してから画素領域でエフェクトをかけた後に再符号化する。しかし、復号と再符号化に膨大な時間が掛かるだけでなく、再符号化による画質劣化が問題となっている。

一方、[3]及び[4]では、符号領域でフェードを実現する。[3]は、DCT係数のDC成分を操作することによって輝度を変化させ、DC値が下限に近づいた場合にAC成分を強制的に0とする。しかし、輝度の濃淡の差が維持されたまま平均値だけが減少するので、フェード区間にあってもコントラストが高く、一様に収束しないという問題がある。[4]ではDCT領域で動き補償を行った後、Intra MBに対してフェードを実現する方式を提案している。しかし、DCT領域での動き補償は計算量が膨大であるため、処理速度に問題がある。また、Intra MBの増加は符号化効率を低下させる恐れがある。

3. 重み予測を利用したフェード編集

3.1 H.264の重み予測

重み予測はH.264のMain以上のプロファイルで利用でき、参照フレームごとの量子化済み重み係数 \hat{w}_i とオフセット係数 \hat{o}_i がスライスヘッダに格納される。参照画素を p_i とすると、片予測による予測値 p' は式(1)で与えられ、双予測の場合は式(2)で与えられる。

$$p' = \text{Clip}(((\hat{w}_1 \cdot p_1 + 2^{\log WD - 1}) \gg \log WD) + \hat{o}_1) \quad (1)$$

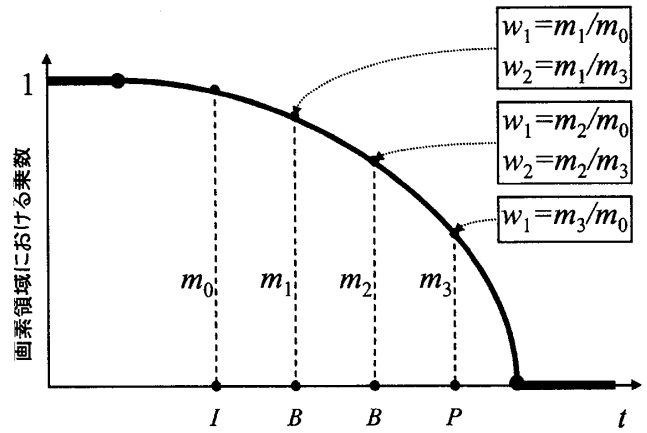


図1 重み係数 w_i の決定

$$p' = \text{Clip}(((\hat{w}_1 \cdot p_1 + \hat{w}_2 \cdot p_2 + 2^{\log WD}) \gg (\log WD + 1)) + ((\hat{o}_1 + \hat{o}_2 + 1) \gg 1)) \quad (2)$$

Clipは0から255に丸め込む処理を表し、 $\log WD$ は量子化済み重み係数 \hat{w}_i を実数に戻す役割を果たす。なお、次節以降では簡単のため量子化前の重みパラメータ w_i 、 o_i を使って議論する。

3.2 重みパラメータの決定

任意のフェードアウトについて、H.264の符号領域でのフェード編集を検討する。まず、フェードアウトはフェード前の画素 q と収束する画素値 c との重み付き平均として定式化できる。

$$p = m \cdot q + (1 - m) \cdot c \quad (0 < m < 1) \quad (3)$$

フェードアウトでは画素を一定の値 c に収束させるため、画素領域における乗数 m_i は単調減少を続け、収束する値に依存しない。図1のようにフェードアウトの区間と遷移過程が設定されたとき、式(3)におけるフェード前の画素 q を介して、参照フレーム(I)の画素 p と被参照フレーム(P)の画素 p' は式(4)の関係にある。

$$p' = \frac{m_3}{m_0} \cdot p + c \cdot \left(1 - \frac{m_3}{m_0}\right) \quad (4)$$

よって、重み予測における重み係数 w_i は、画素領域での参照フレームに対する乗数 m_j と被参照フレームに対する乗数 m_k を使って式(5)のように設定する。

$$w_i = \frac{m_k}{m_j} \quad (5)$$

一方、オフセット係数 o_i は収束する値に依存する。重み係数 w_i が0に収束するとき、コントラストが失われる

[†](株) KDDI 研究所, KDDI R&D Laboratories Inc.
{hkato, yoneyama, takisima}@kddilabs.jp

とともに画素値そのものが0に近づくため、オフセット係数 o_i が収束値を決定する。式(4)では右辺第2項がオフセット係数 o_i に相当する。ただし、重み係数 w_i の量子化誤差を補正するため、オフセット係数 o_i を式(6)で与える。

$$o_i = c \cdot (1 - (\hat{w}_i \gg \log WD)) \quad (6)$$

任意の収束値 c により、白または黒だけでなく中間色へのフェードアウトも可能となる。さらに、重み予測は色差の重みパラメータを輝度とは独立に設定できるので、任意色へのフェードアウトも実現できる。

3.3 整数変換係数へのフェード処理

重み予測はP及びBスライスでのみ用いられるため、図1における乗数 m_0 をIスライスに掛けるには別途処理が必要となる。また、滑らかなフェードを実現するためには、P及びBスライスの残差成分にも重み係数 w_i に応じた変化を与えることが望ましい。

まず、重み予測と同等の処理を整数変換係数に直接反映させることを検討する。4*4整数変換行列および逆変換行列、変換係数をそれぞれ C , D , X で表すとき、ベースバンドで重み予測された係数 X' は式(7)で表される。

$$\begin{aligned} X' &= (C(w_i \cdot D^T X D + o_i \cdot I) C^T) \circ E^{-1} \\ &= w_i \cdot X + o_i \cdot (C I C^T) \circ E^{-1} \end{aligned} \quad (7)$$

ここで \circ および T はそれぞれ Hadamard 積、転置操作を示す。また、 I は全要素が1の4*4行列であり、 E は直交変換のためのスケーリング行列を表す。(($C I C^T$) $\circ E^{-1}$) はDC成分にだけ64が残る。よって、符号領域での重み予測は、変換係数 X に重み係数 w_i を乗じた後、オフセット係数 $64o_i$ をDC成分にだけ加算することになる。

ただし、Iスライスは周囲のブロックからIntra予測で予測値を生成するため、画面左上のマクロブロック(MB)以外の係数 X は、P及びBスライスにおける残差成分に相当する。予測値にはオフセット係数 $64o_i$ がすでに含まれているので、IPBの整数変換係数には重み係数 w_i を掛けるだけでフェードを実現できる。一方、Iスライスの左上のMBだけは固定値に対する残差を符号化する。よって、左上のブロックは重み係数 w_i を掛けるだけでなく、固定値(=128)との差を吸収するためDC成分に $64(c-128)(1-w_i)$ を加算する必要がある。

4. シミュレーション結果と考察

提案法の有効性を検証するため、Pentium4 3.2GHzのWindows PC環境で画質及び変換時間を評価する実験を行った。Mobile & CalendarとFlower Garden(360*240, 60frame)を予めH.264の参照ソフトJM 10.2[5]でH.264($M=1$, $N=15$, Main, CABAC, QP=28固定)に符号化しておき、15~45フレームに線形のフェードアウトを付与した。提案法の比較対象として、ベースバンド変換及び[3]を用いた。ただし、[3]は黒フェードにしか対応していないため、符号領域におけるDC操作を画素領域に

表1 フェード編集処理時間(sec)

ベースバンド変換	従来法[3]	提案法
105.2	0.49	0.51

おける輝度操作で代用した。ベースバンド変換は画素領域でフェードを付与した後に、前述の条件でH.264に再符号化してからPSNRを測定した。

速度比較では、ベースバンド変換、従来法および提案法の処理時間を表1に示す。ベースバンド変換によるフェード処理自体は0.3秒であるが、再符号化に掛かる時間が101.5秒と全体の大部分を占める。一方、提案法は再符号化が必要ないだけでなく、部分的な復号と一部の係数操作で済むためベースバンド変換に対して206倍の高速化を達成した。なお、従来法と提案法では処理速度に大差はない。

画質に関しては、黒および白フェードアウトにおけるPSNRの比較を図2から図5に示す。PSNR算出に用いた原画は、H.264で符号化する前の非圧縮画像に線形のフェードアウトを付与した画像とした。15フレーム以前はフェードを編集していないので、提案法と従来法は同じPSNRである。しかし、ベースバンド変換は再符号化による劣化が生じている。また、45フレーム以降はフェードアウトによって一色となるため、いずれの方式も原画像と完全に一致する。同一の画像同士では定義上PSNRを算出できないので、図2-5では45フレーム以降を省略している。15~45フレームのフェード区間では提案法と従来法に顕著な差が生じる。フェードは画素値に応じて変化するのに対し、従来法は画素値に関わらず画素から一定値ずつ削減する。原画の画素変化と異なるため、従来法はPSNRに大きな劣化が生じる。ただし、フェード区間の後半では全体的に画素値の幅が狭まるため、従来法でも輝度のPSNRは上昇に転じる。一方、提案法は画素値に比例して変化するため、画質が劣化することはない。フェード区間の最初から高いPSNRを保持したまま、フェードアウトで画素値の幅が狭まるにつれPSNRが上昇する。提案法は従来法と比較して、輝度に対して最大22dB以上高いPSNRを実現した。

ベースバンド変換は原画像と同じ方法でフェードを付与するが、再符号化による劣化が発生する。一方、提案法は符号領域で処理するため、再符号化を必要とせず高い画質を維持できる。ベースバンド変換の劣化度合いはビットレートに依存するが、この実験では提案法はベースバンド変換より最大7.4dB以上高いPSNRを実現した。

図6,7は、それぞれ提案法及び従来法を適用したフェード区間の画像を5フレーム置きに表している。従来法は黒フェードアウトで暗い画素ほど速く黒に収束するため、フェードの早い段階から黒潰れが生じる。逆に白フェードアウトでは明るい画素ほど速く白に収束するため、フェードの早い段階から白飛びが発生している。一方、提案法は黒潰れや白飛びを生じることなく、主観的に滑らかなフェードを実現していることが分かる。いずれ

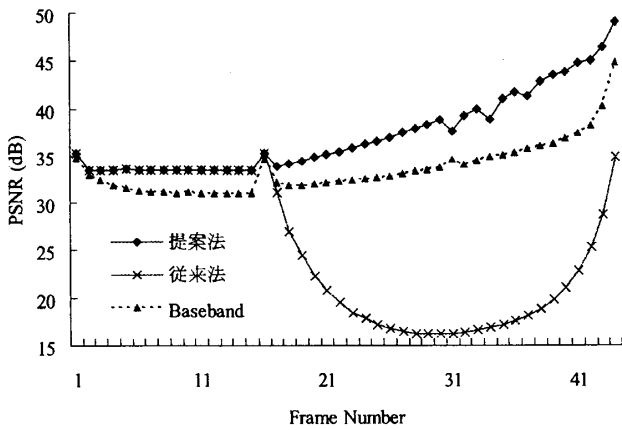


図2 黒フェードアウトにおける PSNR (Mobile & Calendar)

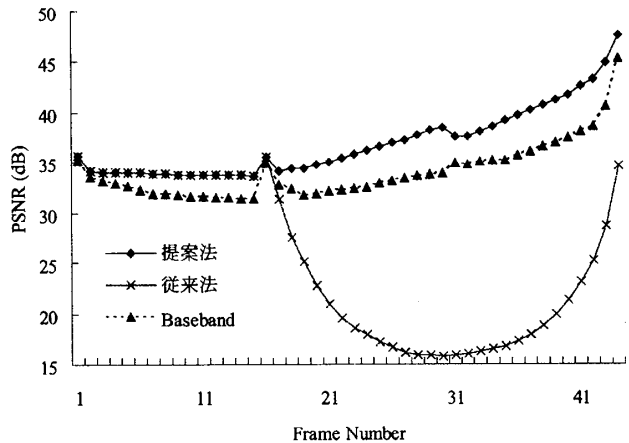


図4 黒フェードアウトにおける PSNR (Flower Garden)

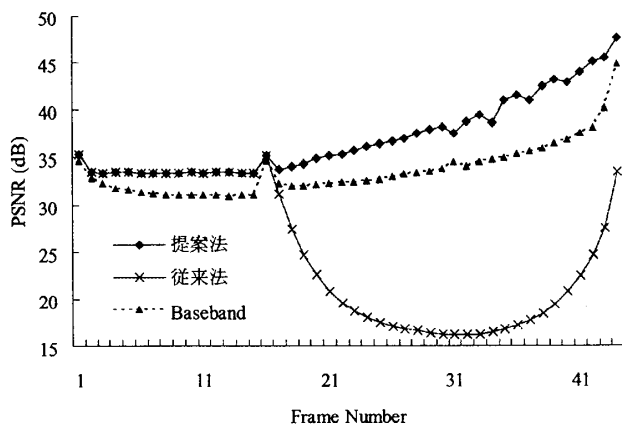


図3 白フェードアウトにおける PSNR (Mobile & Calendar)

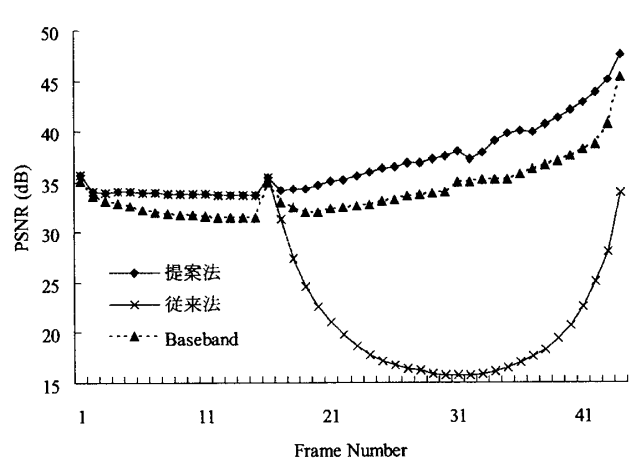


図5 白フェードアウトにおける PSNR (Flower Garden)

も比較のために一般的な線形フェードを適用したが、画素領域における乗数だけから提案法の重み係数を算出できるので、提案法は図1のような非線形フェードにも適用可能である。また、図8、9は、提案法による任意色へのフェードアウトも滑らかに実現されていることを示している。

5. 結論

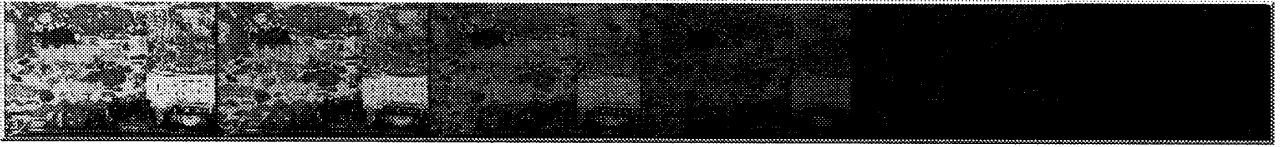
本稿では、符号領域におけるフェード編集方式について提案した。H.264 コンテンツに対して重み予測を利用することで高速かつ高画質なフェード効果を付与した。提案法はベースバンド変換と比較して、画質を保ったまま200倍以上高速になることを実験から確認した。従来法との比較では提案法は画質が最大で22dB改善した。今後はフェードインについて検討する予定である。

参考文献

- [1] ISO/IEC 14496-10: *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding* (2005).
- [2] Boyce, J. M.: Weighted prediction in the H.264/MPEG AVC video coding standard, in *International Symposium on Circuits and Systems (ISCAS)*, Vol. 3, pp. 789–792 (2004).
- [3] Shen, B.: Fast fade-out operation on MPEG video, in *International Conference on Image Processing (ICIP)*, Vol. 1, pp. 852–856 (1998).
- [4] Fernando, W., Canagarajah, C. and Bull, D.: Fade, dissolve and wipe production in MPEG-2 compressed video, *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 3, pp. 717–727 (2000).
- [5] Suehring, K.: H.264 / AVC reference software, <http://bs.hhi.de/suehring/tml>.



(a) 従来法 [3]

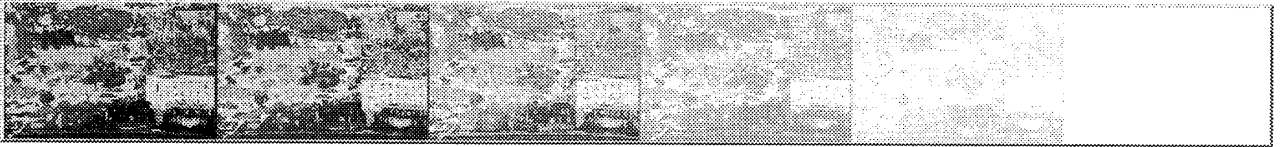


(b) 提案法 ($c_Y = 0, c_U = 128, c_V = 128$)

図6 黒フェードアウト適用画像



(a) 従来法 [3]



(b) 提案法 ($c_Y = 255, c_U = 128, c_V = 128$)

図7 白フェードアウト適用画像

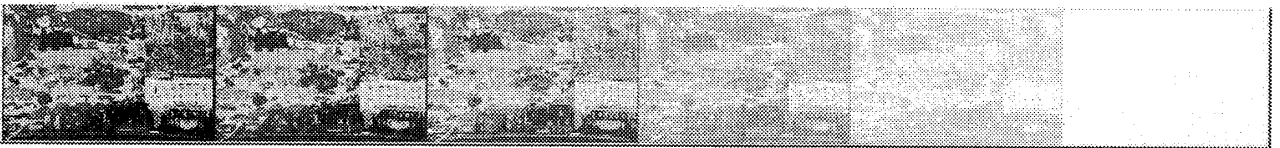


図8 提案法による黄フェードアウト適用画像 ($c_Y = 221, c_U = 3, c_V = 141$)

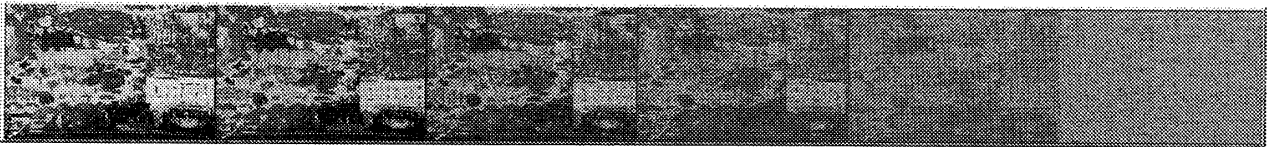


図9 提案法による赤フェードアウト適用画像 ($c_Y = 75, c_U = 86, c_V = 253$)