

高フレームレート映像に対する 時間方向フィルタ設計の基礎検討

坂東幸浩[†], 高村誠之[†], 上倉一人[†], 八島由幸[†]

[†] 日本電信電話株式会社 NTT サイバースペース研究所
〒 239-0847 神奈川県横須賀市光の丘 1-1

あらまし:

近年、撮像系のフレームレートの増加は目覚しく、1000 Hz を超える高フレームレート映像が撮影可能となっている。一方、現行のディスプレイの上限は 120 Hz 程度であり、映像の入力・出力システムのフレームレートの上限は非対称である。高速度カメラで撮影された映像を実時間再生で表示する場合には、フレームレートのダウンサンプリングが必要となる。本稿では、ダウンサンプリング後のシーケンスに対する符号化効率の観点から最適なダウンサンプリング法を検討する。具体的には、ダウンサンプリング時にフレーム間予測誤差電力を最小化するようにダウンサンプリングフィルタを設計する。平均フィルタと比べて、提案手法はフレーム間予測画像の PSNR を平均 0.26 dB 向上させることが確認できた。

A design of temporal down-sampling filter for high frame-rate video

Yukihiro BANDO[†], Seishi TAKAMURA[†], Kazuto KAMIKURA[†],
and Yoshiyuki YASHIMA[†]

[†]NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikarino-oka, Yokosuka, Kanagawa 239-0847, JAPAN

Over the past decade video acquisition rate has broken through over 1000 Hz. Current display systems, unfortunately, have a maximum frame-rate about 120 Hz. The display systems are definitely slower than the camera system. When a high frame-rate video (over 1000 Hz) is played in real time, however, it is necessary to down-sample the frame-rate of the video. In this paper, we propose a temporal down-sampling filter for high frame-rate video. The proposed method sets filter coefficients so as to minimize inter-frame prediction error. Our method can improve the PSNR of prediction signal by 0.26 dB on average compared to mean filter.

1 はじめに

次世代映像システムの基盤技術として、映像の高画質化に向けた検討が進められている。高精細化 [1]・高ダイナミックレンジ化 [2]・マルチスペクトル化 [3] 等のアプローチに加え、動画像の高画質化に固有の検討事項として、高フレームレート化がある。かつては、60Hz 程度であった撮像系のフレームレートだが、半導体技術の進歩により、1000Hz を超える高フレームレート映像を取得可能な高速度カメラが、利用可能となっている。例えば、文献 [4] には、超高速度カメラ用の撮像素子に関する最近の開発動向がまとめられている。また、時空間解像度の高解像度化の例としては、ハイビジョンの順次走査で毎秒 300 フレームの撮影が可能な高速度カメラが開発されている [5]。

このように高フレームレート映像が取得可能になった状況を受け、高速映像処理技術に関する興味深い検討が進められている。代表的な例としては、VLSI 技術を利用した毎秒 1000 フレームの画像処理が可能な視覚システム（ビジョンチップシステム）の開発 [6]、フレームレートとフレーム間予測誤差電力の理論モデル [7] 等がある。

一方、現行のディスプレイの上限は 120Hz 程度であり、映像の入力・出力システムのフレームレートの上限は非対称である。1000Hz 超の高フレームレート映像をスローモーション再生での観察等に用いる場合には、こうした入出力システムのフレームレートの非対称性は問題にならない。

これに対し、高速度カメラで撮影された映像を実時間再生で表示する場合には、フレームレートのダウンサンプリングが必要となる。フレームレートのダウンサンプリング方法としては、フレーム間引きによるサブサンプリング [8] が代表例である。しかし、単純なフレーム間引き処理では、時間方向のエイリアシングに起因する画質劣化が問題となる。こうした問題を回避するためには、時間軸方向の帯域制限フィルタリングが必要である。このため、撮像系では、フレームレートに応じてシャッターの開口時間を変化させる方法がとられている。これは時間方向の平均フィルタとみなすことができる。しかし、時間方向のエイリアシングの低減は、予測誤差の低減とは直接的には結びつかない。つまり、従来の時間方向のダウンサンプルフィルタは、符号化効率の観点から改善の余地を残す。

そこで、本稿では、ダウンサンプリング後のシーケンスに対する符号化効率の観点から最適なダウンサンプリングフィルタの設計法を検討する。具体的には、ダウンサンプリング時にフレーム間予測誤差電力を最小化するようなフィルタ係数を算出するアルゴリズムを検討する。

2 フレーム間予測誤差を低減する時間方向フィルタの設計

2.1 時間方向フィルタ

フレーム間隔を δ_t として、時刻 $t = \tau\delta_t$ ($\tau = 0, 1, \dots$) のフレームにおける位置 x の画素値を $f(x, t)$ ($x = 0, \dots, X - 1$) と表す。この画素信号 $f(x, t)$ をダウンサンプリングにより、フレーム数を $1/M$ に変換する場合を考える。つまり、フレームレートを $\frac{1}{\delta_t}$ から $\frac{1}{M\delta_t}$ へ変換する場合を想定している。以下では、ダウンサンプリング前後のフレームレートの比 M をダウンサンプリング比と呼ぶ。

タップ長 $2\Delta + 1$ の平均フィルタの場合、ダウンサンプリング後のフレームは次式となる。

$$\frac{1}{2\Delta + 1} \sum_{j=-\Delta}^{\Delta} f(x, (iM + j)\delta_t) \quad (1)$$

なお、 $2\Delta + 1 \leq M$ とする。上記のフィルタは、定数のフィルタ係数をもつフィルタである。

これに対し、提案する手法では、フレーム可変のフィルタ係数を用いたフィルタリングを検討する。第 i フレームに対するフィルタ係数を $\mathbf{W}_i = (w_i[-\Delta], \dots, w_i[\Delta])$ とし、ダウンサンプリング後のフレームを次式により定める。

$$\hat{f}(x, iM\delta_t, \mathbf{W}_i) = \sum_{j=-\Delta}^{\Delta} w_i[j] f(x, (iM + j)\delta_t) \quad (2)$$

なお、フィルタ係数は、

$$\sum_{j=-\Delta}^{\Delta} w_i[j] = 1$$

を満たすものとする。

2.2 フィルタ係数の算出方法

動画像符号化では、動き補償フレーム間予測が用いられていることから、フィルタ設計の基準として動き

補償予測誤差を用いることとする。つまり、動き補償予測誤差を最小化するように、ダウンサンプリング時のフィルタ設計を行う。フレーム $\hat{f}(x, iM\delta_t, \mathbf{W}_i)$ に対して、サイズ $\frac{X}{K}$ の区間 $B[k]$ ($k = 0, 1, \dots, K-1$) に分割し、各区間 $B[k]$ ($k = 0, 1, \dots, K-1$) を単位として動き補償 (推定変位量 $\mathbf{d}_i = (d_i[0], \dots, d_i[K-1])$) を行った場合、その区間内の動き補償予測誤差は次のように表現できる。

$$\sigma_i^2(\mathbf{W}_i, \mathbf{d}_i) = \sum_{k=0}^{K-1} \sum_{x \in B[k]} |\hat{f}(x, iM\delta_t, \mathbf{W}_i) - \hat{f}(x - d_i[k], (i-1)M\delta_t, \mathbf{W}_{i-1})|^2 \quad (3)$$

上式から、予測誤差はフィルタ係数 \mathbf{W}_i と変移量 \mathbf{d}_i に依存することが分かる。従って、動き補償予測誤差を最小化するフィルタ係数は変移量に応じて変化する。そこで、以下に示す繰り返し処理によりフィルタ係数を算出する。

まず、動き補償予測誤差 $\sigma_i^2(\mathbf{W}_i^{(0)}, \mathbf{d}_i)$ を最小化するように、変位量 $\mathbf{d} = (d[0], \dots, d[K-1])$ を設定する。ここで求めた変位量 \mathbf{d} を $\mathbf{d}^{(1)}$ とする。

次に、 $\mathbf{d}^{(1)}$ を用いた場合の予測誤差 $\sigma_i^2(\mathbf{W}_i, \mathbf{d}_i^{(1)})$ をフィルタ係数 \mathbf{W}_i の関数とみなし、同予測誤差を最小化するフィルタ係数 \mathbf{W}_i を求める。具体的には、以下の連立方程式を \mathbf{W}_i について解く。

$$\begin{cases} \frac{\partial \sigma_i^2(\mathbf{W}_i, \mathbf{d}_i^{(1)})}{\partial w_i[j]} = 0 \quad (j = -\Delta \dots, \Delta - 1) \\ \sum_{j=-\Delta}^{\Delta} w_i[j] = 1 \end{cases} \quad (4)$$

上式の解として求めたフィルタ係数を $\mathbf{W}_i^{(1)}$ とする。

上述の変移量の算出、および、フィルタ係数の算出を終了条件を満たすまで繰り返し行う。終了条件は、繰り返し処理による予測誤差の変化量が判定する。次式に示すように、 n 回目の予測誤差と $n-1$ 回目の予測誤差との差が閾値 ϵ_e 以下となれば、繰り返し処理を終了する。

$$|\sigma_i^2(\mathbf{W}_i^{(n)}, \mathbf{d}_i^{(n)}) - \sigma_i^2(\mathbf{W}_i^{(n-1)}, \mathbf{d}_i^{(n-1)})| \leq \epsilon_e \quad (5)$$

ダウンサンプリング後の第 i フレームを生成する処理は、図 1 のようになる。

3 実験

フレーム間引きの基準となる映像は、高速度カメラで撮影した RGB カラー映像であり、フレームレー

1. フィルタ係数の初期値 $\mathbf{W}_i^{(0)}$ を読み込む
2. $n = 0$
3. $n++$
4. 式 (3) で示される予測誤差電力に対して、フィルタ係数を $\mathbf{W}_i^{(n-1)}$ とした場合に同予測誤差電力を最小化する動きベクトルを算出し、 $\mathbf{d}_i^{(n)}$ に格納する。
5. 式 (3) で示される予測誤差電力に対して、動きベクトルを $\mathbf{d}_i^{(n)}$ とした場合に同予測誤差電力を最小化するフィルタ係数を算出し、 $\mathbf{W}_i^{(n)}$ に格納する。具体的な算出方法は、連立方程式 (4) を \mathbf{W}_i について解き、その解を求めるフィルタ係数とする。
6. 不等式 (5) を満たすならば、次のステップに進む。そうでなければ、3. に戻る。
7. $\mathbf{W}_i^{(n)}$ を第 i フレームに対するフィルタ係数 \mathbf{W}_i として、式 (2) に従いダウンサンプリング後のフレームを出力する。

図 1: ダウンサンプリング後の第 i フレームを生成する処理手順

トは 1000Hz、総フレーム数は 1200 フレーム、解像度は 640×480 画素である。また、映像素材はゴルフおよび、野球のスイングを撮影したスポーツシーン (golf, baseball) である。なお、動き補償時のブロックサイズは 16×16 画素とした。また、フレーム間予測は片方向予測とし、参照フレームは直前フレームとした。動き推定アルゴリズムとしては全探索を用いた。ダウンサンプリング比 M は 16 とした。また、パラメータ Δ は 1 とした。この場合、フィルタのタップ長は 3 タップとなる。なお、同パラメータの値は、本手法により得られたダウンサンプリング後の画像と平均フィルタにより得られたそれとの間に有意な画質差は発生しないことが確認された値を用いた。

動き補償予測誤差電力の比較結果を表 1 に示す。ここでは、2 種類のダウンサンプリングフィルタについて比較している。平均フィルタは、式 (1) において $M = 16, \Delta = 1$ とした場合のフィルタである。提案法は、2.2 にて示したアルゴリズムに基づき算出されたフィルタ係数を用いたフィルタである。提

表 1: 予測画像の PSNR[dB]($M = 16, \Delta = 1$)

	平均フィルタ	提案法
tennis	34.37	34.66
golf	38.64	38.86

表 2: 平均フィルタとの類似度

シーケンス	式 (6) の値
tennis	0.797
golf	0.846

案法は平均フィルタと比べて、PSNR が 0.22 ~ 0.29 dB 向上することを確認できた。これは、提案法がダウンサンプリング後のフレーム間予測誤差を考慮してフィルタ係数を設計したことによる。

さらに、次式に示す尺度を用いて、本手法により算出されたフィルタ係数と平均フィルタのフィルタ係数の差を評価した。

$$\frac{1}{3}\bar{w}[-1] + \frac{1}{3}\bar{w}[0] + \frac{1}{3}\bar{w}[1] \quad (6)$$

ここで、 $(\bar{w}[-1], \bar{w}[0], \bar{w}[1])$ は、フレーム毎に算出されたフィルタ係数の平均値である。ダウンサンプリング後のフレーム数を T とすると、次のように表せる。

$$\bar{w}[j] = \frac{1}{T} \sum_{i=0}^{T-1} w_i[j] \quad (j = -1, 0, 1)$$

式 (6) は、2 つのベクトル $(1/3, 1/3, 1/3)$ と $(\bar{w}[-1], \bar{w}[0], \bar{w}[1])$ の内積を表しており、式 (6) の値が 1 に近いほど、 $(\bar{w}[-1], \bar{w}[0], \bar{w}[1])$ は $(1/3, 1/3, 1/3)$ に類似したベクトルといえる。そこで、式 (6) の値を平均フィルタとの類似度と呼ぶことにする。この類似度の値を表 2 に示す。表 1 および表 2 によると、表 2 の類似度が小さいほど、つまり、本手法のフィルタ係数が平均フィルタのそれと異なる程度が大きいほど、平均フィルタに対する予測画像の PSNR のゲインも大きくなることが確認できる。

4 おわりに

本稿では、高フレームレート映像に対する適応的なダウンサンプリングフィルタの設計方法を検討し

た。提案するフィルタ設計方法は、高フレームレート映像から低フレームレート映像へフレームレート変換を実行する際、ダウンサンプリング後の低フレームレート映像の予測誤差を考慮した形で、フィルタ係数を算出する点に特徴がある。本適応処理の導入により平均フィルタと比べて予測画像の PSNR を平均 0.26 dB(最大 0.29 dB) 向上させることが確認できた。さらに、この PSNR のゲインは、算出されたフィルタ係数と平均フィルタのフィルタ係数との類似度に関係があることも確認できた。

参考文献

- [1] “Parameter values for an expanded hierarchy of lsdI image formats for production and international programme exchange,” ITU-R Recommendation BT.1769 (2006), 2006.
- [2] 須川成利, “広ダイナミックレンジ CMOS イメージセンサ技術,” 映メ学技報, vol.30, no.ME2006-114, pp.25-32, 2007.
- [3] 羽石秀昭, “マルチスペクトルに基づく映像技術,” PCSJ, pp.1-6, 2003.
- [4] 江藤剛治, “小特集: 撮像技術の最新動向, 5. 高速高感度イメージセンサの開発,” 映メ誌, vol.60, no.3, pp.303-306, 2007.
- [5] 小笠原俊英, 山内正仁, 戸村義男, 山崎順一, 後藤正勝, 橋本洋二, 長秀雄, 高地栄一, 金山茂弘, “毎秒 300 フレーム順次走査 HDTV 高速度カメラ,” 映メ誌, vol.60, no.3, pp.358-365, 2007.
- [6] 小室孝, 石井抱, 石川正俊, 吉田淳, “高速対象追跡ビジョンチップ,” 信学論, vol.J84-D-II, 2001.
- [7] Y. Bandoh, K. Hayase, S. Takamura, K. Kamikura, and Y. Yashima, “Theoretical modeling of inter-frame prediction error for high frame-rate video signal,” IEICE Trans. Fund. of Elec., Comm. & Comp. Sci., vol.E91-A, no.3, pp.730-739, 2008.
- [8] J. Ohm, “Advances in scalable video coding,” Proc. of IEEE, vol.93, no.1, pp.42-56, 2005.