

高密度時間サンプリングに基づく時間方向適応フィルタ Adaptive temporal filter design based on high density sampling

坂東幸浩[†], 高村誠之[†], 如澤裕尚[†]

Yukihiro BANDO[†], Seishi TAKAMURA[†], and Hirohisa JOZAWA[†]

概要

従来、符号化前処理もしくは符号化器への入力シーケンスは、後段の符号化器とは独立に生成されていた。しかし、連続信号を離散信号に変換する際、符号化処理を考慮して動画像データを生成することができれば、従来の符号化検討の外に、新たに符号化効率向上の余地を見出すことが可能となり、動画像符号化に新たな道を開くことになる。そこで、本稿では、時間方向に高密度にサンプリングされたフレームを用いて、表示系のフレームレートに対応した動画像データを生成する際、フレーム間予測誤差を最小化する動画像データを生成することを目的とする。実験により、提案手法は、平均フィルタと比べて、フレーム間予測画像の PSNR を平均 0.21 [dB] 向上させることが確認できた。

1 はじめに

動画像符号化における符号化効率向上のアプローチとしては、符号化器へ入力された動画像データを効率的に符号化するための符号化処理の高能率化、および符号化器への入力前に動画像データを処理する符号化前処理の高能率化に分類される。符号化処理・符号化前処理の高能率化は、各処理におけるパラメータ選択の最適化に帰着することができる。また、代表的な符号化前処理としては、時空間的な低域通過処理に基づくプレフィルタ処理 [1] [2] をあげることができる。

動画像データは、撮像系において、画素値の量子化・時空間のサンプリングにより、連続信号を離散信号に変換することで生成される。ディスプレイのフレームレート 30~60 [Hz] に対応した動画像データを生成する場合、現在の撮像系は、60[Hz] で撮影された動画像データを出力し、これを符号化処理の入力信号としている。このとき、撮像系において、符号化処理を考慮した処理は行われていない。連続信号を離散信号に変換する際、符号化処理を考慮して動画像データを生成することができれば、従来の符号化検討の外に、新たに符号化効率向上の余地を見出すことが可能となり、動画像符号化に新たな道を拓くことになる。

かつては、60[frame/sec] 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、1000[frame/sec] 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 [3] 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 [4]。大規模カメラアレイを用いた高速撮影システムも開発されている [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]、高速度カメラを用いたリアルタイム追跡システムの開発 [7]、フレームレートとフレーム間予測誤差の関係を表す理論モデルの構築 [8] [9] 等がある。

十分短い時間間隔でサンプリングされたフレームがあれば、そのフレームを用いて、連続信号を入力として所望のフレームレートの動画像データを生成する過程を高い精度で再現可能になる。前述のように、CMOS 技術の発展により、現在では、1000[Hz] を超える高フレームレート映像が取得可能である。一方、現行のディスプレイの表示能力として、フレームレートの上限は 180[Hz] 前後であり、ディスプレイへの入力は、一般的には 60[Hz] とされている。このため、1000[Hz] 程度の高フレームレート映像は、表示系のフレームレートに対して、十分短い時間間隔でサンプリングされた動画像データとみなせる。

本研究の目的は、主観画質を保持しつつ、動画像符号化に適した動画像データを生成することである。動画像符号化では、動き補償フレーム間予測が用いられることから、符号化効率向上のためには、動き補償フレーム間予測の低減が重要となる。そこで、本稿では、時間方向に高密度にサンプリングされたフレームを用いて、表示系のフレームレートに対応した動画像データを生成する際、フレーム間予測誤差を最小化する動画像データを生成することを目的とする。上記は、時間方向のダウンサンプリングフィルタの設計問題として定式化できるため、時間方向の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。

[†]日本電信電話株式会社 NTT サイバースペース研究所

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、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 (以下、単に「フレーム」と略記)。この画像信号 $f(x, t)$ 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。

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

ここで、 i はダウンサンプリング後のフレーム番号である。なお、 Δ は正整数とし、 $2\Delta+1 \leq M$ 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の一例である。

これに対し、本稿では、可変フィルタ係数を用いたフィルタ設計を検討する。ダウンサンプリング後の第 i フレームに対するフィルタ係数を $w_i = (w_i[-\Delta], \dots, w_i[\Delta])$ 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。

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

以下では、ダウンサンプリング後のフレーム番号をステージと呼ぶこととする。式 (2) は第 i ステージのダウンサンプリング処理を示している。なお、フィルタ係数は、以下の正規化条件

$$\sum_{j=-\Delta}^{\Delta} w_i[j] = 1 \quad (3)$$

を満たすものとする。

図1は式 (1) および式 (2) におけるパラメータの物理的な意味を図示している。同図において矩形は、ダウンサンプリング前のフレーム位置を表しており、網掛けされた矩形はフィルタリングにおける参照フレームを示している。また、同図は $M=10, \Delta=1$ の場合を示しており、 $10 (=M)$ フレーム毎に、 $3 (=2\Delta+1)$ フレームを選択し、選択された3フレームを用いてダウンサンプリング後のフレームを生成する場合に対応する。この場合、ダウンサンプリングによりフレームレートは $1/10$ 倍に変換される。

2.2 フィルタ係数の最適化規準

フィルタ設計においては、ダウンサンプリング後の動き補償フレーム間予測誤差をフィルタ設計の規準として導入

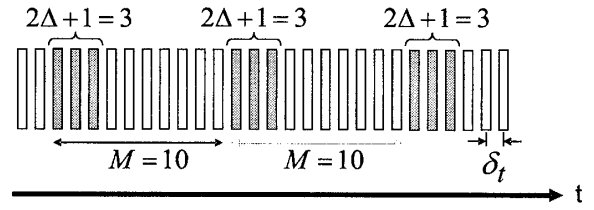


図1: 時間フィルタのパラメータの例 ($\Delta=1, M=10$ の場合)

する。MPEG-2, H.264/AVC に代表される動画像符号化では、動き補償フレーム間予測が用いられており、同予測誤差の低減は符号化効率向上に繋がる。そこで、動き補償フレーム間予測誤差を低減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。

X 画素からなるフレームを K 分割して、分割領域毎に動き補償フレーム間予測を行う場合を考える。フレーム $\hat{f}(x, iM\delta_t, w_i)$ 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 (変位量 $d_i = (d_i[0], \dots, d_i[K-1])$) を行った場合、そのフレーム内の動き補償フレーム間予測誤差 (以下、単に予測誤差と略記) は次のように表現できる。

$$\sigma_i^2(w_i, w_{i-1}, d_i) = \sum_{k=0}^{K-1} \sum_{x \in B[k]} |\hat{f}(x, iM\delta_t, w_i) - \hat{f}(x - d_i[k], (i-1)M\delta_t, w_{i-1})|^2 \quad (4)$$

さらに、第 I ステージまでの動き補償予測誤差の総和 (以下、単に予測誤差和と略記) を次式のように表す。

$$E_I(w_1, \dots, w_0, d_1, \dots, d_1) = \sum_{i=1}^I \sigma_i^2(w_i, w_{i-1}, d_i) \quad (5)$$

本稿では、フィルタ係数の算出法として、次の2種類のアルゴリズムを検証する。2.3節では、式 (4) で示されるステージ毎の予測誤差に対して、ステージ内に閉じた形で各ステージの予測誤差を独立に最小化するアルゴリズムを示す。さらに、2.4節では、同アルゴリズムの拡張として、隣接するステージ間の依存関係を考慮した形で式 (5) で示される全ステージの予測誤差の総和を最小化するアルゴリズムを示す。

2.3 フィルタ係数の局所的最適化アルゴリズム

本節では、ステージ番号の小さなステージから順にフィルタ係数を算出するシーケンシャルな処理を前提とする。つまり、第 $i-1$ ステージにおけるフィルタ係数を求め、同フィルタ係数を求めた値に固定して、第 i ステージのフィルタ係数を求める。そこで、本節に限り、予測誤差 $\sigma_i^2(w_i, w_{i-1}, d_i)$ を $\sigma_i^2(w_i, d_i)$ と略記することとする。

2.3.1 反復処理に基づくアプローチ

予測誤差はフィルタ係数と変移量に依存することが分かる。つまり、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、フィルタ係数と変移量のいずれか一方を固定した状態で他方を最適化するものとし、その最適化の対象を交互に入替えながら、一定の収束条件まで繰り返すアプローチを取る。

まず、第 i ステージのフィルタ係数が初期値 $w_i^{(0)}$ に設定されたものとして、予測誤差 $\sigma_i^2(w_i^{(0)}, d_i)$ を最小化する変位量 $d_i = (d_i[0], \dots, d_i[K-1])$ を求める。ここで求めた変位量 d_i を $d_i^{(1)}$ とする。次に、 $d_i^{(1)}$ を用いた場合の予測誤差 $\sigma_i^2(w_i, d_i^{(1)})$ をフィルタ係数 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 w_i を算出する。具体的な算出方法は 2.3.2 に示す。ここで、求めたフィルタ係数を $w_i^{(1)}$ 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で判定する。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 η_i 以下となれば、繰返し処理を終了する。

$$|\sigma_i^2(w_i^{(n)}, d_i^{(n)}) - \sigma_i^2(w_i^{(n-1)}, d_i^{(n-1)})| \leq \eta_i \quad (6)$$

2.3.2 フィルタ係数の算出方法

動きベクトルが d_i に設定されたものとして、 d_i を用いた場合の予測誤差 $\sigma_i^2(w_i, d_i)$ をフィルタ係数 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 w_i を求める。予測誤差 $\sigma_i^2(w_i, d_i)$ を最小化することから、求めるフィルタ係数は次の連立方程式を満たす。

$$\begin{cases} \frac{\partial \sigma_i^2(w_i, d_i)}{\partial w_i[j]} = 0 \quad (j = -\Delta + 1, \dots, \Delta) \\ \sum_{j=-\Delta}^{\Delta} w_i[j] = 1 \end{cases} \quad (7)$$

$w_i[-\Delta] = 1 - \sum_{j=-\Delta+1}^{\Delta} w_i[j]$ を $\frac{\partial \sigma_i^2(w_i, d_i)}{\partial w_i[j]} = 0$ に代入して、展開すると次式を得る。

$$\begin{aligned} & \sum_{n=-\Delta+1}^{\Delta} w_i[n] \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \epsilon_i(x, n, -\Delta) \\ & + \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \hat{\epsilon}_i(x, -\Delta) = 0 \end{aligned} \quad (8)$$

ここで、 $\epsilon_i(x, n, m)$ 、 $\hat{\epsilon}_i(x, n)$ は次式で定義される。なお、 $n, m = -\Delta, \dots, \Delta$ である。

$$\begin{aligned} \epsilon_i(x, n, m) &= f(x, (iM+n)\delta_i) - f(x, (iM+m)\delta_i) \\ \hat{\epsilon}_i(x, n) &= f(x, (iM+n)\delta_i) - \hat{f}(x - d_i[k], (i-1)M\delta_i, w_{i-1}) \end{aligned}$$

上式を整理すると、式 (7) は次の連立方程式となる。

$$\Phi \tilde{w}_i = \gamma \quad (9)$$

ここで、 \tilde{w}_i は $\tilde{w}_i = (w_i[-\Delta+1], \dots, w_i[\Delta])$ で定義される 2Δ 次元ベクトルである。 \tilde{w}_i の要素として、 $w_i[-\Delta]$ が含まれていないのは、フィルタ係数の正規化条件 (式 (3)) より、 $w_i[-\Delta] = 1 - \sum_{j=-\Delta+1}^{\Delta} w_i[j]$ として求まるためである。また、 γ は、第 $j + \Delta - 1$ 成分 ($j = -\Delta + 1, \dots, \Delta$) を

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \hat{\epsilon}_i(x, -\Delta)$$

とする 2Δ 次元列ベクトルである。さらに、 Φ は、第 $(j + \Delta - 1, n + \Delta - 1)$ 成分 ($j, n = -\Delta + 1, \dots, \Delta$) を

$$\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \epsilon_i(x, n, -\Delta)$$

とする $2\Delta \times 2\Delta$ 行列である。式 (9) を \tilde{w}_i について解き、さらに、その解を用いて $w_i[-\Delta] = 1 - \sum_{j=-\Delta+1}^{\Delta} w_i[j]$ を求めることで、所望のフィルタ係数が求まる。

2.4 フィルタ係数の大域的最適化アルゴリズム

2.3 節のアルゴリズムは、前ステージのフィルタ係数が求まった条件下で当該ステージのフィルタ係数を求めている。このため、ステージ間の依存関係を考慮できておらず、全ステージにおける予測誤差和低減の観点からは改良の余地を残す。そこで、本節では、全ステージにおける予測誤差和の最小化を目的として、2.3 節のアルゴリズムの拡張を検討する。

2.4.1 反復処理に基づくアプローチ

2.3 節と同様に、最適化の対象をフィルタ係数と変移量とで交互に入替えながら、一定の収束条件まで繰り返すアプローチを取る。

フィルタ係数が初期値 $w_i^{(0)}$ ($i = 0, \dots, I$) に設定されたものとして、全ステージのフレームを生成し、生成されたフレーム間の予測誤差を最小化する動きベクトルを求める。ここで求めた変位量 d_i を $d_i^{(1)}$ ($i = 1, \dots, I$) とする。次に、 $d_i^{(1)}$ ($i = 1, \dots, I$) を用いた場合の予測誤差和 $E_I(w_I, \dots, w_0, d_1^{(1)}, \dots, d_1^{(1)})$ をフィルタ係数 w_i ($i = 0, \dots, I$) の関数とみなし、同予測誤差を最小化するフィルタ係数 w_i ($i = 0, \dots, I$) を算出する。具体的な算出方法は 2.4.2 に示す。ここで、求めたフィルタ係数を $w_i^{(1)}$ ($i = 0, \dots, I$) 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で判定する。次式に示すよ

1. フィルタ係数の初期値 $w_i^{(0)}$ ($i = 0, \dots, I$) を読み込む
2. $n = 0$
3. $n++$
4. フィルタ係数を $w_i^{(n-1)}$ ($i = 0, \dots, I$) として、全ステージのフレームを生成し、生成されたフレーム間の予測誤差和を最小化する動きベクトルを算出し、 $d_i^{(n)}$ ($i = 1, \dots, I$) に格納する。
5. 動きベクトルを $d_i^{(n)}$ ($i = 1, \dots, I$) とした場合に、式(5)で示される予測誤差和を最小化するフィルタ係数を算出し、 $w_i^{(n)}$ ($i = 0, \dots, I$) に格納する。具体的な算出方法は、2.4.2 に記載の内容に従う。
6. 反復処理の終了条件 (不等式(10)) 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. $w_i^{(n)}$ を第 i ステージにおけるフィルタ係数 w_i として、式(2)に従い、同ステージのフレームを出力する。

図 2: 2.4 節のアルゴリズムによるフレーム生成処理

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 η_g 以下となれば、繰返し処理を終了する。

$$|E_I(w_I^{(n)}, \dots, w_0^{(n)}, d_I^{(n)}, \dots, d_1^{(n)}) - E_I(w_I^{(n-1)}, \dots, w_0^{(n-1)}, d_I^{(n-1)}, \dots, d_1^{(n-1)})| \leq \eta_g \quad (10)$$

以上の内容を処理フローとして整理し、図 2 に示す。

2.4.2 フィルタ係数の算出方法

動きベクトルが d_I, \dots, d_1 に設定されたものとし、これらの動きベクトルを用いた場合の予測誤差和 $E_I(w_I, \dots, w_0, d_I, \dots, d_1)$ をフィルタ係数 w_I, \dots, w_0 の関数とみなし、同予測誤差和を最小化するフィルタ係数 w_I, \dots, w_0 を求める。求めるフィルタ係数は、全ての $n = -\Delta + 1, \dots, \Delta$, および $i = 0, \dots, I$ に対して、次式を満たす必要がある。

$$\begin{cases} \frac{\partial E_I(w_I, \dots, w_0, d_I, \dots, d_1)}{\partial w_i[n]} = 0 \\ (n = -\Delta + 1, \dots, \Delta; i = 0, \dots, I) \\ \sum_{j=-\Delta}^{\Delta} w_i[j] = 1 \end{cases} \quad (11)$$

なお、 $E_I(w_I, \dots, w_0, d_I, \dots, d_1)$ の定義の基づき、上式における偏微分は、以下の 3 つの式で表される。

$$\begin{cases} \frac{\partial E_I(w_I, \dots, w_0, d_I, \dots, d_1)}{\partial w_0[n]} = \frac{\partial \sigma_1^2(w_1, w_0, d_1)}{\partial w_0[n]} \quad (i = 1) \\ \frac{\partial E_I(w_I, \dots, w_0, d_I, \dots, d_1)}{\partial w_i[n]} = \frac{\partial \sigma_i^2(w_{i+1}, w_i, d_{i+1}) + \sigma_i^2(w_i, w_{i-1}, d_i)}{\partial w_i[n]} \quad (i = 2, \dots, I-1) \\ \frac{\partial E_I(w_I, \dots, w_0, d_I, \dots, d_1)}{\partial w_I[n]} = \frac{\partial \sigma_I^2(w_I, w_{I-1}, d_I)}{\partial w_I[n]} \quad (i = I) \end{cases} \quad (12)$$

さらに、 $w_i[-\Delta] = 1 - \sum_{j=-\Delta+1}^{\Delta} w_i[j]$ を用いて、 $\frac{\partial \sigma_i^2(w_i, w_{i-1}, d_i)}{\partial w_{i-1}[n]}$, $\frac{\partial \sigma_i^2(w_i, w_{i-1}, d_i)}{\partial w_{i-1}[n]}$ は、次のように展開できる。

$$\begin{aligned} \frac{1}{2} \frac{\partial \sigma_i^2(w_i, w_{i-1}, d_i)}{\partial w_i[n]} &= \\ &\sum_{j=-\Delta+1}^{\Delta} w_i[j] \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \epsilon_i(x, n, -\Delta) \\ &- \sum_{j=-\Delta+1}^{\Delta} w_{i-1}[j] \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_{i-1}(x - d_i[k], j, -\Delta) \epsilon_i(x, n, -\Delta) \\ &+ \sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, iM - \Delta) \epsilon_i(x, n, -\Delta) \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{1}{2} \frac{\partial \sigma_i^2(w_i, w_{i-1}, d_i)}{\partial w_{i-1}[n]} &= \\ &- \sum_{j=-\Delta+1}^{\Delta} w_i[j] \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \epsilon_{i-1}(x - d_i[k], n, -\Delta) \\ &+ \sum_{j=-\Delta+1}^{\Delta} w_{i-1}[j] \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_{i-1}(x - d_i[k], j, -\Delta) \epsilon_{i-1}(x - d_i[k], n, -\Delta) \\ &- \sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, iM - \Delta) \epsilon_{i-1}(x - d_i[k], n, -\Delta) \end{aligned} \quad (14)$$

ここで、 $\alpha(x, iM - \Delta)$ は次式で定義される。

$$\alpha(x, iM - \Delta) = f(x, (iM - \Delta)\delta_i) - f(x - d_i[k], ((i-1)M - \Delta)\delta_i)$$

これらを式(11)に代入して整理すれば、次の連立方程式を得る。

$$\Psi W = \beta \quad (15)$$

ここで、 W は第 $2\Delta i + n + \Delta - 1$ 成分を $w_i[n]$ とする $2\Delta I$ 次元列ベクトルである。また、 β は、 $2\Delta I$ 次元列ベクトルであり、第 $2\Delta i + n + \Delta - 1$ 成分は i の値に応じて以下のように定まる。

$i = 0$ の場合：

$$\sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, M - \Delta) \epsilon_0(x - d_1[k], n, -\Delta)$$

$1 \leq i \leq I-1$ の場合：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, iM - \Delta) \epsilon_i(x, n, -\Delta) \\ + \sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, (i+1)M - \Delta) \epsilon_i(x - d_{i+1}[k], n, -\Delta)$$

$i = I$ の場合：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \alpha(x, IM - \Delta) \epsilon_I(x, n, -\Delta)$$

また、 Ψ は、 $2\Delta I \times 2\Delta I$ 行列であり、第 $2\Delta i + n + \Delta - 1$ 行の成分は i および n の値の応じて以下のように定まる。なお、以下においては、 $n, j = -\Delta + 1, \dots, \Delta$ とする。

$i = 0$ の場合：

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta i + (j + \Delta - 1)$ 成分：

$$\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_0(x - d_1[k], j, -\Delta) \epsilon_0(x - d_1[k], n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta(i+1) + (j + \Delta - 1)$ 成分：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_1(x, j, -\Delta) \epsilon_0(x - d_1[k], n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の上記以外の成分：0

$1 \leq i \leq I-1$ の場合：

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta(i-1) + (j + \Delta - 1)$ 成分：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_{i-1}(x - d_i[k], j, -\Delta) \epsilon_i(x, n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta i + (j + \Delta - 1)$ 成分：

$$\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x, j, -\Delta) \epsilon_i(x, n, -\Delta) \\ + \sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_i(x - d_{i+1}[k], j, -\Delta) \epsilon_i(x - d_{i+1}[k], n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta(i+1) + (j + \Delta - 1)$ 成分：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_{i+1}(x, j, -\Delta) \epsilon_i(x - d_{i+1}[k], n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の上記以外の成分：0

$i = I$ の場合：

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta(i-1) + (j + \Delta - 1)$ 成分：

$$-\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_{I-1}(x - d_I[k], j, -\Delta) \epsilon_I(x, n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の $2\Delta i + (j + \Delta - 1)$ 成分：

$$\sum_{k=0}^{K-1} \sum_{x \in B[k]} \epsilon_I(x, j, -\Delta) \epsilon_I(x, n, -\Delta)$$

第 $2\Delta i + n + \Delta - 1$ 行の上記以外の成分：0

3 実験

ダウンサンプルフィルタの対象となる映像は、高速度カメラで撮影した YCbCr カラー映像 (以下の実験では輝度信号のみを使用) であり、フレームレートは 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、映像素材はミニカートの走行シーン (車)、テニスのスイング (テニス)、野球のバッティング (野球 1)、野球のスローイング (野球 2) を撮影したスポーツシーンである。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に関連するパラメータとして、ダウンサンプリング比 M は 32 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。

本手法による予測誤差の低減効果を評価するため、動き補償予測により得られる予測画像の PSNR を表 2 に示す。ここでは、3 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 $M = 32$ 、 $\Delta = 1$ とした場合のフィルタである。つまり、全フィルタ係数を $1/3$ とした場合のフィルタである。適応フィルタ (局所的最適化) は、2.3 節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。適応フィルタ (大域的最適化) は、2.4 節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。表 3 に平均フィルタに対する提案法の予測画像の PSNR の増加量を示す。同表より、提案法は平均フィルタに対して、局所的最適化に基づく適応フィルタにおいて平均 0.16[dB]、さらに、大域的最適化に基づく適応フィルタにおいて平均 0.21[dB]、各々、予測画像の PSNR 向上を確認できた。これは、提案法がダウンサンプリング後の予測誤差を考慮してフィルタを設計したことによる。

なお、パラメータ $\Delta = 1$ の値は、提案法により得られた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の発生しないことが確認された値を用いた。画像の画質差を評価する際、ITU-R 勧告 BT.500 [10] で規定されている二重刺激妨害尺度法 (DSIS 法) を用いた。評価に用いたモニタは 17 インチ CRT ディスプレイであり、解像度は 1024×768 [画素] である。評価映像は、オリジナルのサイズ (640×480 [画素]) で、モニタの画面中央に表示した。評価映像が表示されていないモニタ内の領域は、灰色の単色背景とした。画面の高さを H とし、視距離は $3H$ で観測した。評価者は 20~30 代の男女 6 名であり、平均フィルタにより得られた映像を基準映像として、適応フィルタ (局所的最適化および大域的最適化) により得られた映像に対して、表 1 に示す 5 段階評価を行った。本主観評価の結果、

表 1: 5段階劣化尺度

| 評点 | 評価語 |
|----|------------------|
| 5 | 劣化が分からない |
| 4 | 劣化が分かるが気にならない |
| 3 | 劣化が気になるが、邪魔にならない |
| 2 | 劣化が邪魔になる |
| 1 | 劣化が非常に邪魔になる |

表 2: 予測画像の PSNR[dB]($M = 32, \Delta = 1$)

| | 平均フィルタ | 適応フィルタ (局所的最適化) | 適応フィルタ (大域的最適化) |
|------|--------|--------------------|--------------------|
| テニス | 30.28 | 30.37 | 30.40 |
| 車 | 28.70 | 28.95 | 28.97 |
| 野球 1 | 31.49 | 31.67 | 31.69 |
| 野球 2 | 30.68 | 30.81 | 30.92 |

表 3: 平均フィルタに対する予測画像の PSNR[dB] の増加量 ($M = 32, \Delta = 1$)

| | 適応フィルタ (局所的最適化) | 適応フィルタ (大域的最適化) |
|------|--------------------|--------------------|
| テニス | 0.09 | 0.12 |
| 車 | 0.25 | 0.27 |
| 野球 1 | 0.18 | 0.20 |
| 野球 2 | 0.13 | 0.24 |
| 平均値 | 0.16 | 0.21 |

全評価者が「劣化が分からない」と回答しており、提案法と平均フィルタの生成画像の間に有意な画質差が無いことが確かめられた。

4 おわりに

従来、1000[Hz] 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、スロー再生用途の映像の撮影に用いられてきた。これに対し、本稿では、高速度カメラにより、時間方向に高密度にサンプリングしたフレームを用いることで、符号化効率向上に寄与する映像生成の新たなアプローチを提案した。提案手法を用いることで、主観画質を保持しつつ、フレーム間予測誤差を平均 0.21 [dB] 低減可能である事を実験により確認した。今後は、コーデック

を用いた符号化実験により、符号量削減効果を検証する予定である。

参考文献

- [1] J. Ohm. Advances in scalable video coding. *Proc. of IEEE*, Vol. 93, No. 1, pp. 42–56, 2005.
- [2] A. Golwelkar and J. Woods. Motion-compensated temporal filtering and motion vector coding using biorthogonal filters. *IEEE Trans. Circuits & Systems for Video Technology*, Vol. 17, No. 4, 2007.
- [3] 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.
- [4] 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.
- [5] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. on Graphics*, Vol. 24, No. 3, pp. 765–776, 2005.
- [6] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida. A digital vision chip specialized for high-speed target tracking. *IEEE trans. on Electron Devices*, 2003.
- [7] U. Muehlmann, M. Ribo, P. Lang, and A. Pinz. A new high speed CMOS camera for real-time tracking applications. *IEEE Int. Conf. on Robotics and Automation*, pp. 5195–5200, 2004.
- [8] 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.
- [9] Y. Bandoh, S. Takamura, H. Jozawa, and Y. Yashima. Generalized theoretical modeling of inter-frame prediction error for high frame-rate video signal considering integral phenomenon. *IEICE Trans. Fund. of Elec., Comm. & Comp. Sci.*, Vol. E93-A, No. 8, 2010.
- [10] Methodology for the subjective assessment of the quality of television pictures. *ITU-R Recommendation BT.500-11*, 2002.