

可逆動画像符号化のためのフレーム内分割とフレーム間統合 に基づいた3次元予測器の設計法

岡崎 修磨[†] 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: †{okazaki,matsuda,itoh}@itohws01.ee.noda.tus.ac.jp

あらまし 筆者らは、動き補償とブロック適応3次元予測に基づいた動画像データの高性能可逆符号化方式を開発している。この方式では、時間方向と空間方向の相関を同時に利用可能な3次元予測器を複数設計し、これらをブロック単位で切り替えることで、画像の局所的な性質に応じた適応予測を実現している。従来は、このブロック適応予測に用いる3次元予測器の設計をフレーム毎に行っていたが、絵柄が空間的な領域によって大きく変化する画像では、それぞれの領域毎に予測器の設計を行った方が有利となる場合があると考えられる。そこで本稿では、画像を空間方向に分割すると共に、時間方向には統合して得られる時空間領域毎に予測器の設計を行うものとし、その適切な条件について考察する。符号化シミュレーションの結果、本手法により符号化レートを0.01-0.06 bits/pel削減できることを確認した。

キーワード 動画像可逆符号化, 動き補償, ブロック適応予測, フレーム内分割, フレーム間統合

A Method of Designing 3D Predictors Based on Intra-Frame Dividing and Inter-Frame Merging for Lossless Video Coding

Shuma OKAZAKI[†], Ichiro MATSUDA[†], and Susumu ITOH[†]

[†] Faculty of Science and Technology, Tokyo University of Science
2641 Yamazaki, Noda-shi, Chiba 278-8510, Japan

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

Abstract We are developing an efficient lossless video coding scheme based on motion compensation and block-adaptive 3D prediction. In this scheme, multiple 3D predictors which use both the current and motion-compensated previous frames are switched block-by-block. In the previous work, these predictors are designed for each frame. However, if textures and/or motions in a sequence vary spatially rather than temporally, designing the predictors for each area given by intra-frame dividing and inter-frame merging would be reasonable. In this paper, therefore, appropriate conditions for such spatial-temporal partitioning are explored. Simulation result indicate that the proposed method can provide bit-rate savings of 0.01-0.06 bits/pel for tested images.

Key words Lossless video coding, Motion compensation, Block-adaptive prediction, Intra-frame dividing, Inter-frame merging.

1. まえがき

動画像の可逆符号化は、文化的価値の高い映像の保存を目的とするデジタルアーカイブや業務用マスタ映像の記録など、高品質な動画像を扱う分野で必要とされる技術であり、近年様々な符号化アルゴリズムが提案されている [1]。このような状況の中、筆者らは、高い圧縮率と高速な再生を両立可能な、動

画像の可逆符号化方式の開発に取り組んでいる [2-5]。この方式では、動き補償された前フレームと当該フレームの符号化済み画素を同時に参照する3次元線形予測器を複数用意し、それらを8×8画素のブロック単位で適応選択することにより、被写体のテクスチャや動きといった局所的な性質に追従した高精度な時空間予測を実現している。文献 [3] では、このブロック適応予測に用いる予測器群の設計を、フレーム毎に行うより複数

フレーム分まとめて実行した方が、全体の符号化効率の点で有利であることを示した。これは、予測係数の更新間隔が広がることでフレーム当たりの付加情報量が減少し、結果的により多くの種類の予測器を利用できるようになるためである。しかし、予測器の設計は多数の反復処理に基づいているため、単純に処理対象となるフレームの枚数や予測器の数を増やすと、符号化に要する演算量が大幅に増加してしまう。これに対し文献 [4] では、空間方向にフレーム分割し、時間方向には統合を施すことで得られる直方体の時空間領域毎に予測器の設計を行うことで、演算量を増加させることなく、フレーム統合のみによる上記の方式と同等な効果が得られることを見出した。特に、背景と前景の被写体のように空間的な位置によって絵柄や動きが大きく異なるシーケンスにおいて、これらの性質が均質となる時空間領域毎に予測器の設計を行うことは、適応予測の効率を高めるという意味で理に通っているといえる。しかし、従来は予測器の数や予測回数といったパラメータの設定値を固定していたため、適応予測に必要な付加情報の符号量と予測性能のバランスを領域の性質に応じて適切に決定することが困難であった。本稿では、予測器の数と個々の予測器における予測回数を予測器の設計手順の中で自動設定する手法 [5] を導入した場合について、上述した時空間分割の適切な条件を調査し、その効果について考察する。

2. 動き補償に基づいたブロック適応 3 次元予測

本方式では、 M 種類の線形予測器を 8×8 画素のブロック単位で適応選択し、得られた予測誤差をコンテキストモデリングに基づいて適応的に算術符号化している [2]。GOP 内の 2 フレーム目以降 (P ピクチャ) については、図 1 のように当該フレームの符号化済み近傍画素 p_k だけではなく、前フレームの画素 q_k も同時に参照する 3 次元予測を適用する。その際、前フレームの参照画素群 $\{q_k | k = 0, 1, \dots, K_q - 1\}$ は、四分木分割に基づいた可変サイズの MC ブロック (32×32 , 16×16 , 8×8 画素の 3 通り) 毎に検出された整数画素精度の動ベクトル v に従って動き補償されている。画素 p_k , q_k の輝度値をそれぞれ $s_t(p_k)$, $s_{t-1}(q_k)$ と表記すると、符号化対象画素 p_0 に対

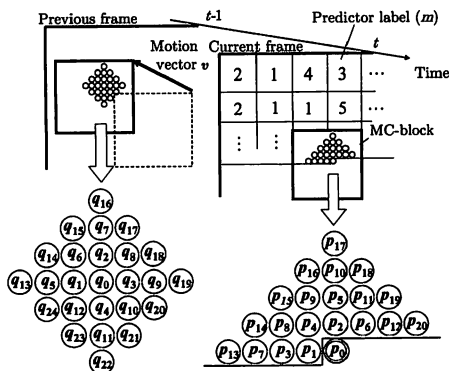


図 1 ブロック適応 3 次元予測

して m 番目の予測器 ($m = 1, 2, \dots, M$) を適用した際の予測値 $\hat{s}_t(p_0)$ は次式により算出される。

$$\hat{s}_t(p_0) = \sum_{k=1}^{K_p} a_m(k) \cdot s_t(p_k) + \sum_{k=0}^{K_q-1} a_m(k + K_p + 1) \cdot s_{t-1}(q_k + v) \quad (1)$$

但し、 K_p , K_q はそれぞれ当該フレームおよび前フレームの参照画素の数、 $a_m(k)$ ($k = 1, 2, \dots, K_p + K_q$) は各参照画素に關する予測の重み (予測係数) である。

3. 予測誤差の符号化

(1) 式によって算出された予測誤差 $e = s_t(p_0) - \hat{s}_t(p_0)$ は、コンテキストモデリングと呼ばれる手法を用いて適応的に算術符号化される。コンテキストモデリングは、符号化済みの周辺画素の状態 (コンテキスト) から予測誤差の確率分布を推定する手法である。提案方式では、符号化済み近傍画素の予測誤差の絶対値和をしきい値 $\{Th_m(1), Th_m(2), \dots, Th_m(15)\}$ を用いて 16 レベルに量子化し、それぞれの量子化レベルに対応したコンテキスト ($n = 1, 2, \dots, 16$) 毎に、予測誤差の条件付確率密度関数を次式の一般化ガウス関数でモデル化している [2]

$$\begin{aligned} \hat{P}(e|n) &= \frac{c_n \eta(c_n, \sigma_n)}{2\Gamma(1/c_n)} \cdot \exp\left\{-|\eta(c_n, \sigma_n) \cdot e|^{c_n}\right\}, \\ \eta(c_n, \sigma_n) &= \frac{1}{\sigma_n} \sqrt{\frac{\Gamma(3/c_n)}{\Gamma(1/c_n)}} \end{aligned} \quad (2)$$

但し、 $\Gamma(\cdot)$ はガンマ関数、 σ_n は一般化ガウス関数の標準偏差、 c_n は尖度を定める形状パラメータである。提案方式では、実際の予測誤差 e の分布に適合した確率密度関数のモデル $\hat{P}(e|n)$ を得るため、同一の予測器を用いる領域毎にしきい値 $\{Th_m(1), Th_m(2), \dots, Th_m(15)\}$ およびコンテキスト毎の形状パラメータ c_n を可変としている。ここで、8 ビット精度の画像では、輝度値が 0-255 の整数値で表現されていることに着目すると、予測値 $\hat{s}_t(p_0)$ が既知の場合の予測誤差 e のとり得る値は以下に示す 256 通りとなる。

$$e \in \{s - \hat{s}_t(p_0) | s = 0, 1, \dots, 255\} \quad (3)$$

従って、予測値 $\hat{s}_t(p_0)$ とコンテキスト n が同時に与えられた場合の予測誤差 e の条件付生起確率を (2) 式に基づいて以下のように表すことができる。

$$\Pr(e | \hat{s}_t(p_0), n) = \frac{\Pr(e|n)}{\sum_{s=0}^{255} \Pr(s - \hat{s}_t(p_0) | n)}, \quad (4)$$

$$\Pr(e|n) = \int_{-h_s}^{h_s} \hat{P}(e + \varepsilon | n) d\varepsilon \quad (5)$$

但し、 h_s は演算誤差の蓄積を防ぐために設定した予測値 $\hat{s}_t(p_0)$ の丸め精度であり、本稿では $h_s = 1/8$ としている。予測誤差 e の値は、画素毎に式 (4) に基づいて動的に更新される確率テーブルを用いて、最終的に多値算術符号化される [2]。

4. フレーム内分割とフレーム間統合に基づいた予測器の設計

本稿では、フレーム内分割とフレーム間統合によって得られる直方体の時空間領域単位で、予測器や動ベクトルといった符号化処理に必要なパラメータの最適化を行うものとする。この最適化手順の対象となる直方体内の画素数が一定となるよう、空間方向の分割数と時間方向の統合枚数を共に D とした場合の分割パターンを例を図 2 に示す。但し、GOP の先頭フレーム (1 ピクチャ) はフレーム単位で予測器を設計し、2 フレーム以降 (全て P ピクチャ) についてのみこの分割パターンが適用されるものとする。従来は、この時空間領域毎に予測器の数 M および予測回数に関するパラメータ K_p, K_q の設定値を変えずに最適化手順を実効していたため [4]、背景部分などテクスチャや動きが単調な領域では過剰な数の予測器が割り当てられ、逆に前景の複雑な領域では予測器が不足する現象が認められた。そこで本稿では、初期値として十分な数 ($M = 100$ 種類) の予測器を用意し、反復最適化手順の中で不要な予測器を徐々に削減する。また、各予測器についても非零となる予測係数の数 (実効予測回数) の増加を抑えつつ参照画素配置の自由度を高める手法 [5] の導入を図る。この手法は、予測誤差の符号量と最適化後のパラメータに関する付加情報のビット数 (B_{side}) の和を表す、以下のコスト関数の下で反復的に実行される。

$$J = - \sum_{\mathbf{p}_0} \log_2 \Pr(e | \hat{s}_t(\mathbf{p}_0), \mathbf{n}) + B_{side} \quad (6)$$

これにより、付加情報も含めた時空間領域全体の符号量が最小となるよう、予測器の数や実効予測回数 (参照画素配置) が自動設定されるものと期待できる。P ピクチャにおける具体的な手順を以下に示す。

(1) MC ブロックサイズが $32 \times 32, 16 \times 16, 8 \times 8$ 画素それぞれの場合について、平均 2 乗誤差最小 (MMSE) 規範に基づいたブロックマッチングを実行し、整数画素精度の動ベクトル \mathbf{v} の初期値を決定する。

(2) MC ブロックのサイズを 16×16 画素に固定した状態で、 $M = 100$ 種類の予測器およびそれらの割り当て状況を初期化する。この処理では、予測回数に関するパラメータを小さな値 ($K_p = 20, K_q = 25$) に設定しておき、予測誤差の確率密度関数がガウス分布に従うことを前提とした簡易な予測器設計手法 [6] を利用する。また、符号化が終了した P ピクチャが存在する $D + 2$ フレーム以降では、時間方向に隣接する時空間領域で使用した予測器を再利用することで、更に演算量の削減を図っている [5]。

(3) 予測回数に関するパラメータを $K_p = 72, K_q = 113$ にそれぞれ拡大すると共に、予測回数 $a_m(k)$ の値を修正する処理を各予測器について一定回数ずつ実行する。この修正処理はランダムに選んだ非零の係数を対象に、その値を微小幅だけ増減させたり、他の係数と値を交換するといった試行の中からコスト関数が最も減少するパターンを探索するものであり、非零の係数が過剰に増加しないよう工夫されている [5]。

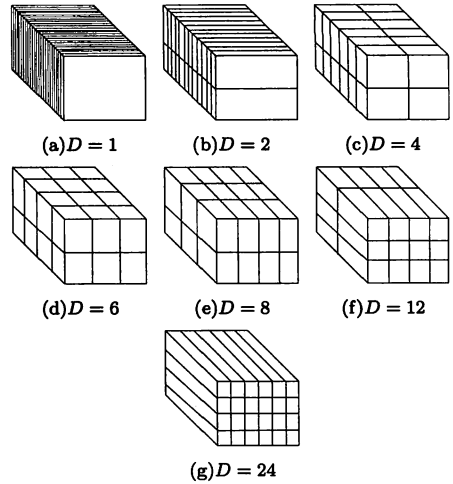


図 2 時空間領域の分割パターン例

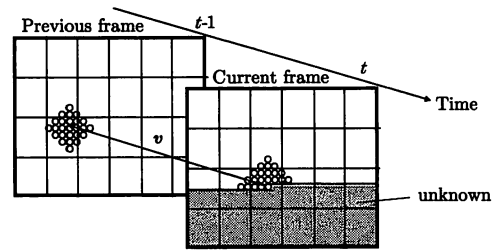


図 3 異なる時空間領域を参照する予測の例 ($D = 24$)

(4) 3. で述べたコンテキストモデリングに用いるしきい値 $\{Th_m(1), Th_m(2), \dots, Th_m(15)\}$ の組み合わせを、同一の予測器 (m) が割り当てられる領域毎に、動的計画法に基づいて決定する [6]。

(5) 確率密度関数モデル $\hat{P}(e | \mathbf{n})$ の形状パラメータを 0.2~3.2 の範囲で 0.2 刻みで変化させ、コスト関数 J が最小となる値に更新する。

(6) 8×8 画素のブロック毎に M 種類の予測器をそれぞれ試し、コスト関数 J が最小となる予測器の番号 m をラベルとして与える。

(7) m 番目の予測器を削除すると仮定し、そのラベル m が付与されたブロックにおいて、2 番目に最適な予測器で代用した場合のコスト J_m を評価する。この時、 J_m の値は m 番目の予測器に関する付加情報を除いて算出し、その最小値が元のコスト J より減少する場合は、対応する予測器を実際に削除した上で M の値を 1 つ減らす。更に、この一連の手順を削除対象となる予測器が存在しなくなるまで繰り返す。

(8) 32×32 画素の MC ブロックを再帰的に 8×8 画素まで分割しながら、対応するサイズの動ベクトルを ± 1 画素の範囲で修正し、コスト関数 J が最小となるブロックサイズと動ベクトルの組み合わせを決定する。

(9) (3)-(8) の手順を時空間ブロック全体のコスト関数 J が減少しなくなるまで繰り返す。

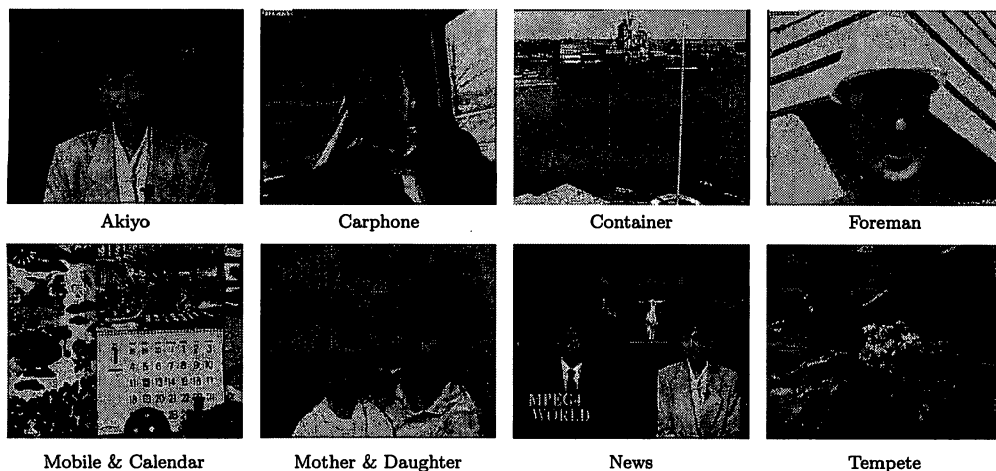


図 4 テスト動画像 (先頭フレーム)

表 1 分割パターンと符号化レートの関係 (bits/pel)

Sequence	$D = 1$	$D = 2$	$D = 4$	$D = 6$	$D = 8$	$D = 12$	$D = 24$
Akiyo	0.899	0.897	0.900	0.894	0.898	0.894	0.893
Carphone	2.682	2.680	2.676	2.672	2.674	2.670	2.663
Container	2.292	2.291	2.285	2.284	2.283	2.280	2.279
Foreman	2.787	2.786	2.782	2.777	2.785	2.786	2.792
Mobile & Calendar	3.601	3.586	3.567	3.557	3.553	3.551	3.539
Mother & Daughter	1.997	1.997	1.993	1.991	1.988	1.987	1.984
News	1.334	1.332	1.328	1.323	1.324	1.324	1.316
Tempete	3.468	3.455	3.445	3.440	3.442	3.435	3.433
Average	2.382	2.378	2.372	2.367	2.368	2.366	2.363

なお、上記の最適化手順が収束した後の実際の符号化処理は、これまで通りフレーム単位で画面全体をラスタ走査しながら行うものとする。これにより、図 3 のように 3 次元予測に用いる画素が異なる時空間領域を参照することが可能となり、空間方向の分割境界付近における予測効率の低下を防止できる。

5. 特性評価

CIF (352×288) サイズのモノクローム動画を対象とし、25 フレーム分を 1 つの GOP (IPPP...) として符号化実験を実施した。実験に使用した画像の一覧 (先頭フレーム) を図 4 に示す。但し、1 ピクチャを除く 24 フレーム分の分割パターンは図 2 に示す通りとした。表 1 に、それぞれの分割パターンについて求めた符号化レート (25 フレーム全体の平均) を示す。なお、表中に太字で示した符号化レートは各画像について最も低い数値であることを示している。これより、多くの画像において、空間方向の分割数 D を増やすことで全体の符号化効率が向上している様子を確認できる。これは、動画像の局所的なテクスチャや動きの性質が、時間方向よりも空間方向に沿って大きく変化する傾向があり、空間方向の分割によって、それぞれの性質に特化した予測器群の設計が可能になったためである

と考えられる。また、画像 Container について、各時空間領域毎に設計された予測器の数および実効予測次数の平均値を調査した結果を図 5、図 6 に示す。これらより、領域毎の絵柄や動きの複雑さに応じて、付加情報の符号量と予測性能のトレードオフに基づいた最適化が行われ、結果として適切な設定値が自動的に得られているものと推察される。なお、図 5 に示した予測器の数 M は、 $D = 1$ の場合に各フレームで設計された数 ($M = 20 - 25$) に比べて小さい値となっている。他の画像でも同様な傾向が認められることから、前節の (2) において初期値として用意する予測器の数を減らすことにより、符号化性能を犠牲にすることなく演算量を削減できる可能性を示唆する結果となっている。

最後に本方式 ($D = 24$) の符号化性能を客観的に評価するため、動画像符号化の国際標準方式である H.264 を拡張した「FRExt」[7] のロスレスモード (JM10.1, 参照フレーム 1 枚)、および静止画像用の可逆符号化方式である JPEG-LS [8] を各フレームに独立に適用する方式との比較を行った。表 2 に示した結果より、「提案方式」は「FRExt」方式に比べて平均で約 19%、JPEG-LS に比べて約 46% それぞれ低い符号化レートを達成している様子を確認できる。

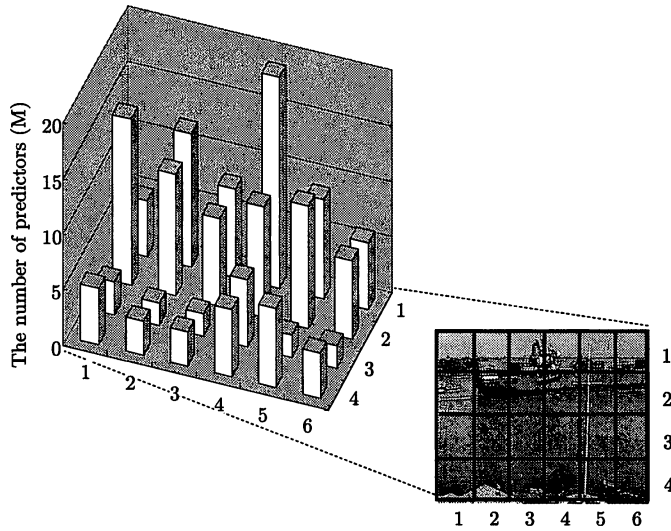


図5 時空間領域毎に設計された予測器の数 (Container)

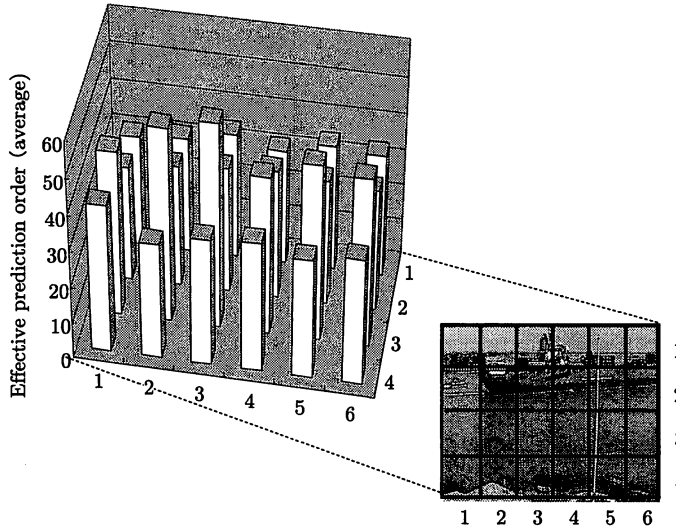


図6 時空間領域毎に設計された予測器の平均実効予測次数 (Container)

表2 他方式との比較 (bits/pel)

Sequence	提案方式	FRExt	JPEG-LS
Akiyo	0.893	1.133	2.956
Carphone	2.663	3.202	3.493
Container	2.279	2.837	4.144
Foreman	2.792	3.347	3.881
Mobile & Calendar	3.539	4.477	5.403
Mother & Daughter	1.984	2.560	3.110
News	1.316	1.702	3.412
Tempete	3.433	4.142	4.848
Average	2.363	2.925	4.369

6. まとめ

筆者らの提案した動画の可逆符号化方式において、フレーム内分割とフレーム間統合によって得られる直方体の時空間領域毎に、予測器や動ベクトルなどの符号化パラメータの最適化を行う手法について検討した。シミュレーションの結果、空間方向の分割数と時間方向のフレーム統合数を共通の値 D とする条件下では、一般に D を大きくした方が全体の符号化効率の点で有利であることが明らかとなった。今後は並列計算機による実装も視野に入れ、符号化効率だけでなく、符号化時間の短縮という観点からも予測器の適切な設計手法について更に検討を加える予定である。

文 献

- [1] 松田一朗, “映像の符号化最前線 5. ロスレスビデオ符号化”, 映像学誌, vol.61, no.4, pp.439-442, Apr. 2007.
- [2] 塩寺太郎, 梅津有司, 前田弘樹, 松田一朗, 森岡一幸, 伊東晋, “フレーム毎に MC と 3 次元予測を最適化する動画像の可逆符号化方式”, 映像学誌, vol.60, no.7, pp.1051-1058, July. 2006.
- [3] 岡崎修磨, 泉俊光, 松田一朗, 伊東晋, “動き補償と適応 3 次元予測に基づいた動画像の可逆符号化～複数フレーム毎の予測器の更新に関する検討～”, 信学総合大, no.D-11-36, Mar. 2007.
- [4] 岡崎修磨, 松田一朗, 伊東晋, 内藤整, 小池淳, “動き補償と適応 3 次元予測に基づいた動画像の可逆符号化～フレーム内分割とフレーム間統合に基づいた予測器の設計～”, PCSJ 2007, no.P-5.19, pp.99-100, Nov. 2007.
- [5] 峯澤彰, 松田一朗, 伊東晋, “動画像の可逆予測符号化のためのパラメータ自動設定”, 信学技報, vol.107, no.40, pp.29-34, May. 2007.
- [6] 松田一朗, 森弘史, 前田謙治, 伊東晋 “可逆符号化のためのレートを最小とする予測器の設計と評価”, 信学論, vol.J85-D2, no.3, pp.448-456, Mar. 2002.
- [7] JVT of ITU-T and ISO/IEC, “Draft Text of H.264/AVC Fidelity Range Extensions Amendment,” JVT-L047, Sep. 2004.
- [8] ISO/IEC, ISO/IEC 14495-1:1999, “Information Technology – Lossless and Near-lossless Compression of Continuous-Tone Still Images: Baseline,” Dec. 1999.