

適応的線形予測を用いた動画像可逆符号化

高村 誠之[†] 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 Y-517A
E-mail: †{takamura.seishi,yashima.yoshiyuki}@lab.ntt.co.jp

あらまし H.264 が用いている符号化ツール群を利用し、動画像の可逆符号化を実現する方式について述べる。具体的には直交変換の部分と別の可逆変換にて置換し、量子化部分を省略すること等により可逆復号を実現する。また、この変換を適応的に行うことで、符号化効率を高める方法も提案する。イントラ符号化実験において、提案手法は Motion JPEG2000 の可逆モードに比べ 2% の符号量増加が観察されたが、FRExt 等 他 の方式よりは少ない符号量となった。インター符号化実験においては、提案方式が最も少ない符号量を与えた。さらに、適応的予測の効果も確認した。

キーワード H.264/AVC, FRExt, 可逆動画像符号化

Lossless Video Coding Using Adaptive Linear Prediction

Seishi TAKAMURA[†] and Yoshiyuki YASHIMA[†]

[†] NTT Cyber Space Laboratories, NTT Corporation 1-1 Hikarino-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan
E-mail: †{takamura.seishi,yashima.yoshiyuki}@lab.ntt.co.jp

Abstract In this paper we propose a reversible transform method that is suitable for lossless video coding to be used with H.264 tools. We extensively measure its lossless coding performance with four different reversible transforms and Motion JPEG 2000. Experimental results show that for I picture coding, the proposed method performed slightly worse (2.0% in average) than Motion JPEG 2000 while outperforming the four reversible methods. For B and P pictures, our method offered the best performance. Our method requires minimal modification to H.264 software and provides better performance than other existing methods.

Key words H.264/AVC, FRExt, Lossless Video Coding

1. はじめに

復号画像に符号化歪みが含まれない可逆符号化は、デジタルシネマ、素材伝送、衛星映像、医療映像等、多くの分野において重要である。この動画への適用は今後ますます重要となると思われる。

動画にも利用できる可逆画像符号化方式として Motion JPEG2000 標準 [1] の可逆モードや JPEG-LS [2] がある。いずれも静止画向け方式であるため、フレーム内に閉じた符号化を行っている。そのため動画特有のフレーム間相関を用いた高効率符号化ができないという問題がある。

一方、非可逆な画像符号化方式の一つ H.264 標準 [3] は他の既存非可逆方式に比べ圧倒的に高い符号化効率を有している。この理由の一つは、可変ブロックサイズ動き予測、複数参照フレーム、適応的イントラ予測といった強力な符号化ツールが実現する高い時空間予測能力である。この高い予測能力を可逆符号化にも応用することで、可逆符号化効率の改善が期待できる。

H.264 標準の拡張版 “Fidelity Range Extension (FRExt)” はそ

の一例であり、H.264 の枠組をほぼそのまま用いつつ、フレーム内・フレーム間予測残差信号を (直交変換・量子化せず) そのまま伝送するため、該標準初版との親和性も高く、処理も簡便である。しかしながら、予測残差信号になお残存している相関を除くことができないという問題がある。

このようにフレーム間予測を行うことで効率を高め、かつ H.264 標準との親和性を保ちつつ圧縮効率の高い可逆動画符号化方式は提案されていなかった。

本稿では、既に提案されている可逆な直交変換とそれらの特性を概観すると同時に、H.264 符号化ツールと組み合わせ、可逆動画符号化を実現する。さらに、2 次元線形予測に基づく可逆符号化方式、およびその準最適な予測係数、予測を適応的とする方法を提案する。計算機シミュレーションを通し、本方式と、既存の可逆変換を用いた方式、そして Motion JPEG2000 によるイントラ可逆符号化方式の性能評価および結果の考察を行う。

2. 可逆変換方式

H.264 標準においては、イントラ/インター予測残差信号に対し直交変換、ついで得られた変換係数の量子化を行うが、これらの操作を可逆なものに置換することで、可逆符号化が実現できる。

本章では従来の可逆な 1x4 次元変換を概観する。4x4 信号に対し、1x4 変換を縦横に施すことにより 4x4 係数が得られる。様々な方式が提案されており、必ずしも整数変換に限らない。

2.1 H.264 の整数変換 (量子化なし)

H.264 標準では、以下の行列により表される、DCT 的で比較的単純な整数変換を用いている。

$$\begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = D \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (1)$$

但し

$$D = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix} \quad (2)$$

ここで $(x_0, \dots, x_3)$ はイントラ/インター予測残差 (整数) であり、 $(X_0, \dots, X_3)$ は変換係数である。

本変換は、係数量子化がなければ当然可逆となる。

ここで D の行列式が

$$\det(D) = 40$$

であることに注目する。これは D の変換後の空間が変換前の 40 倍に拡大されることを意味するので、整数画素信号値に対応する変換前の空間の格子点は、変換後の空間の格子点のうち 40 個に 1 個にしか対応していないことになる。4x4 変換においては、本変換を縦横あわせて 8 回繰り返すため、4x4 変換係数の拡大率は

$$40^8 \approx 6.6 \times 10^{12} \text{ (倍)}$$

に拡大されていることになる。これは、可逆符号化を考えると 1 画素あたり

$$\log_2(40^8)/16 \approx 2.66 \text{ (ビット)}$$

の増加に相当し、無視できないオーバーヘッドとなる。

2.2 方式 1: 無変換 (FRExt 方式)

Sun らは、H.264 方式において前述の変換と、量子化処理とともに省く方式を提案している [4]。イントラ/インター予測残差は直接エントロピー符号化器により符号化され、復号後信号は原画像信号に一致する。これは H.264 標準の拡張版 FRExt に採用されている。

この場合変換行列は単位行列に一致し、その行列式は 1 となる。したがって変換前後で空間の拡大は起こらないが、前述の

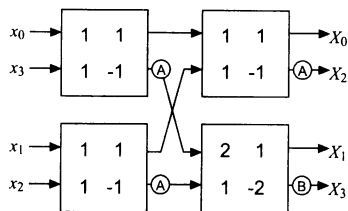


図 1 整数変換のブロックダイアグラム。H.264 の変換ではノードⒶとⒷは無変換。方式 2 の変換ではそれぞれ 2 と 5 で除し四捨五入する。

Fig. 1 Block diagram of integer transform. For H.264 transform, nodes Ⓐ and Ⓑ do nothing. For method 2 transform, they down-scale and round to integers.

ように入力信号間に残る若干の空間相関は当然ながら残存したままとなり、符号化効率改善の余地は残されたままとなっている。

2.3 方式 2: 区分的スケーリング変換

H.264 の変換は、区分的に示すと図 1 のような 4 つの 2x2 変換に分解できる。なおこのときⒶとⒷは何も処理をしない。

各 2x2 変換の出力信号には冗長性がある。例えば $(x_0, x_1) \rightarrow (X_0, X_1)$ の変換を

$$X_0 = x_0 + x_1$$

$$X_1 = x_0 - x_1$$

とする代わりに

$$X_0 = x_0 + x_1$$

$$X_1 = R(x_0 - x_1)$$

$$x'_0 = X_1 + \lfloor X_0/2 \rfloor$$

$$x'_1 = X_0 - x'_0$$

としても原信号値が復元でき、 $(x_0, x_1) = (x'_0, x'_1)$ となる。ここで

$$R(x) = \lfloor x + 0.5 \rfloor$$

である。これは図 1 のノードⒶにおける 1/2 のダウンスケールに相当する。

同様に、

$$X_0 = 2x_0 + x_1$$

$$X_1 = x_0 - 2x_1$$

の 2x2 変換も冗長性を持つ。なんととなれば

$$X_0 = 2x_0 + x_1$$

$$X_1 = R((x_0 - 2x_1)/5)$$

$$x'_0 = X_1 + (2X_0 - \text{offset}[X_0 \bmod 5])/5$$

$$x'_1 = X_0 - 2x'_0,$$

により原信号が復元できるからである。ここで offset は 5 要素 {0, 2, -1, 1, -2} を持つ配列である。これは図 1 のノードⒷにおける 1/5 のダウンスケールに相当する。

$(x_0, \dots, x_3)$ から $(X_0, \dots, X_3)$ への写像は次のようになる:

$$\begin{aligned} X_0 &= x_0 + x_1 + x_2 + x_3 \\ X_1 &= 2R((x_0 - x_3)/2) + R((x_1 - x_2)/2) \\ X_2 &= R((x_0 - x_1 - x_2 + x_3)/2) \\ X_3 &= R((R((x_0 - x_3)/2) - 2R((x_1 - x_2)/2))/5) \end{aligned}$$

丸めを無視するとこの変換行列 D_2 は以下のようになる。

$$D_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & 1 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 1/10 & -1/5 & 1/5 & -1/10 \end{pmatrix} \quad (3)$$

$\det(D_2) = 1$ であるので、変換前後で空間の拡大は生じない。

2.4 方式 3: 可逆 DCT

小松らが提案している可逆 DCT [5] の 1x4 次元版は次のようなものである:

$$\begin{aligned} X_0 &= R((x_0 + x_1 + x_2 + x_3)/4) \\ X_1 &= x_0 - x_3 + R((x_1 - x_2)/C) \\ X_2 &= R((x_0 - x_1 - x_2 + x_3)/2) \\ X_3 &= R((x_0 - x_3)/C + 0.5) - x_1 + x_2, \end{aligned}$$

ここで C は任意の正の実数である。 C によりこの変換の自由度は他の整数変換より高いものとなっている。 $C = 2$ のとき、本変換は H.264 の整数変換と平行 (定数倍の違いのみ) となる。[5] においては C の値として $\cos(\pi/8)/\cos(3\pi/8) = 1 + \sqrt{2}$ を使うことを推奨している。このとき変換は数学的な DCT と平行となる。予備実験では C がこれより大きいほど圧縮率も高まったため、本実験では $C = 1 + \sqrt{2}$, 3, 3.5 の 3 通りを用いた。

この変換行列は丸めを無視すると次のようになる:

$$D_3 = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1 & 1/C & -1/C & 1 \\ 1/2 & -1/2 & -1/2 & 1/2 \\ 1/C & -1 & 1 & -1/C \end{pmatrix} \quad (4)$$

$\det(D_3) = 1 + 1/C^2 > 1$ であるので、変換後の空間は変換前より若干拡大されている。

3. 提案方式

本方式では、直交変換でなく、二次元の DPCM により効率的に相関を除去し、可逆復号を実現する。

3.1 固定予測方式

基本方式として次式のような自己回帰変換を施す。

$$\begin{aligned} X_0 &= x_0 \\ X_1 &= x_1 - R(\alpha x_0) \\ X_2 &= x_2 - R(\alpha x_1) \\ X_3 &= x_3 - R(\alpha x_2) \end{aligned}$$

ここで α は定数 (予測係数) である。 $\alpha = 0$ のとき方式 1 と等価

になる。本方式では隣接ブロックの信号値も用いて (特に X_0 の) 演算することも容易であるが、通常ブロック境界では信号値の変化が大きいため、先に述べた他の方式と同様、ブロック内に閉じた予測を行う。

この変換行列は丸めを無視すると次のようになる:

$$D_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\alpha & 1 & 0 & 0 \\ 0 & -\alpha & 1 & 0 \\ 0 & 0 & -\alpha & 1 \end{pmatrix} \quad (5)$$

このように D_4 は一般に ($\alpha \neq 0$ のとき) 直交変換ではない。しかしながら後の実験結果で示すように、圧縮性能は他の直交変換に劣らない。また、ここで $\det(D_4) = 1$ であるので、信号の空間は変換の前後で拡大しないという好ましい性質があることがわかる。この処理に必要な演算量は方式 1 と同程度であり、方式 2 および 3 や量子化なし方式よりはるかに少ない。

この変換を縦横に施す。その際、 α の値は縦横あるいはピクチャタイプ毎に等しい必要はない。予備実験の結果、I ピクチャについては $\alpha = 0.7$ 、PB ピクチャについては $\alpha = 0.4$ が圧縮率においてよい結果を与えることがわかったので、本実験ではこれらの値を用いた。

3.2 適応的予測方式

予測誤差信号は、隣接したブロック同士では、ほぼ類似の性質をもつことが経験上判明している。そこで、近隣のブロックで最適な予測係数が、処理中のブロックでもよい予測をすると期待できる。

式 (5) において電力集中の意味で最適な α の値は、予測残差信号の自己相関係数により与えられるが、復号前の予測残差信号の値を復号器は知りえないのでこれを用いることはできない。そこで隣接ブロックにおける自己相関係数を用い、 α に反映させることとした。

適応的に α を決定する方針は以下のようである。例えば水平方向の相関が垂直方向のそれよりも大きいと判断されたら、 α の導出において左隣接ブロックの相関係数により大きな重みをつける。垂直方向の相関が大きい場合も同様にする。ブロック間に強い相関がないと判断されたらデフォルト値を用いる。実際の α 決定の流れは以下のようになる。

- (1) $\text{hdif} = |\rho_{II}^{NW} - \rho_{II}^N| + |\rho_{IV}^{NW} - \rho_{IV}^N|$
- (2) $\text{vdif} = |\rho_{II}^{NW} - \rho_{II}^W| + |\rho_{IV}^{NW} - \rho_{IV}^W|$
- (3) if ($\text{hdif} < 0.6 \text{ vdif}$) then // 水平に強い相関
- (4) $\alpha_H = 0.4\rho_{II}^W + 0.3\rho_{II}^N + 0.1\rho_{II}^{NW} + 0.2\alpha_0$
- (5) $\alpha_V = 0.4\rho_{IV}^W + 0.3\rho_{IV}^N + 0.1\rho_{IV}^{NW} + 0.2\alpha_0$
- (6) else if ($\text{vdif} < 0.6 \text{ hdif}$) then // 垂直に強い相関
- (7) $\alpha_H = 0.3\rho_{II}^W + 0.4\rho_{II}^N + 0.1\rho_{II}^{NW} + 0.2\alpha_0$
- (8) $\alpha_V = 0.3\rho_{IV}^W + 0.4\rho_{IV}^N + 0.1\rho_{IV}^{NW} + 0.2\alpha_0$
- (9) else $\alpha_H = \alpha_V = \alpha_0$ // 相関なし

ここで ρ は各位置のブロックに対応する自己相関係数である。上付き添え字 N.W はそれぞれ上、右隣接ブロックを、上付き添え字 H.V はそれぞれ水平、垂直方向の自己相関を表す。デフォルト値 α_0 は、I ピクチャについては 0.7、PB ピクチャについて

表 1 I ピクチャ符号化における可逆圧縮率 (*QCIF,**CIF)

Table 1 Lossless compression ratios for I pictures (*QCIF, **CIF)

画像	量子化なし	方式 1 (FRExt)	方式 2 (区分スケール)	方式 3(可逆 DCT)			提案手法		Motion JPEG2000
				$C=1+\sqrt{2}$	$C=3$	$C=3.5$	固定予測	適応予測	
container*	1.122	1.904	1.902	1.999	2.011	2.015	2.096	2.120	2.111
silent*	1.045	1.691	1.719	1.840	1.846	1.846	1.955	1.948	1.981
news*	1.127	1.846	1.899	1.998	2.008	2.008	2.120	2.124	2.126
stefan*	0.891	1.444	1.440	1.523	1.529	1.531	1.605	1.614	1.675
coast**	1.043	1.738	1.723	1.847	1.852	1.852	1.947	1.963	2.061
foreman**	1.182	2.059	2.032	2.121	2.132	2.134	2.250	2.272	2.265
silent**	1.090	1.841	1.828	1.936	1.946	1.948	2.042	2.050	2.088
paris**	1.007	1.627	1.659	1.734	1.746	1.750	1.825	1.842	1.877
mobile**	0.838	1.327	1.324	1.384	1.387	1.387	1.458	1.463	1.571
Average	1.038	1.720	1.725	1.820	1.829	1.830	1.922	1.933	1.973

は 0.4 とした。

4. 計算機シミュレーション

H.264 リファレンスソフトウェア JM8.2[6] をベースに、以下のような修正を加えた:

- 量子化・逆量子化プロセスの省略
- 4x4 変換部を可逆変換に置換
- デブロッキングフィルタの無効化
- スキップモードは真にロスレスの場合のみ用いるよう制限

4.1 符号化条件

提案手法および比較手法により、4 種の QCIF、5 種の CIF シーケンス (ともに 30fps) を符号化した。

- シンボル符号化: CABAC
- R-D 最適化: 使用
- 参照フレーム数: 5
- 探索範囲: 16

I ピクチャ符号化実験においては、各シーケンス最初の 100 枚をイントラ符号化した。比較のため、同じシーケンスを Motion JPEG2000 により符号化した。使用したコーデックは Kakadu (kdu_v.compress)[7] であり、符号化条件は以下のものである。Clevels は圧縮率を最も高めるよう選択した。

- 色変換なし (Cycce=no)
- 分解レベル (Clevels): CIF で 3、QCIF で 2 (デフォルトは 5)
- 可逆モードの指定 (Creversible=yes)
- 出力フォーマット: Motion JPEG2000 ファイルフォーマット

P ピクチャ符号化実験においては、IPP... 構造で 100 フレームを符号化し、99 枚の P ピクチャの符号量のみを抜きだし、圧縮率を求めた。B ピクチャ符号化実験においては、IBBP... 構造で 298 フレームを符号化し、198 枚の B ピクチャの符号量のみを抜きだし圧縮率を求めた。

4.2 実験結果

様々な画像・符号化方式に対応する圧縮率の一覧を表 1.2.3 に示す。ここで“量子化なし”の方式は 2.1 の、H.264 整数変換

のみを施し量子化を行わない方法である。表内の数字は原画像の大きさに対する圧縮率で、大きいほど効率が高いことを示す。太字の数字は該当の画像について最も圧縮率が高い方式を示している。

4.3 議論

I ピクチャ符号化において、平均的に最も良い圧縮率を与えたのは Motion JPEG2000(イントラ符号化のみ)であり、次いで提案手法、方式 3、方式 2、方式 1 の順となった。“量子化なし”法は最も悪い結果となった。圧縮率が 1 を下回る (逆に伸長する) ケースも見られた。

Motion JPEG 2000 と提案手法の差はわずかであり、container(CIF)、foreman(QCIF)で提案手法 (適応予測) が Motion JPEG2000 を上回っていた。

P、B ピクチャ符号化においては、H.264 の動き補償ツールの相関除去効果の効果により I ピクチャのおよそ半分にまで圧縮されている。この場合、提案手法が最も効率が良く、次いで方式 1、方式 3、方式 2 の順であった。“量子化なし”法は I ピクチャと同様に最も悪かった。

提案手法で適応予測とした場合、全てのピクチャタイプで固定予測よりも平均的に良い結果が得られた。例外は silent(QCIF)であるが、これはブロック間の相関が他の画像よりも特に低かったためと思われる。

方式 3 においては、より大きい C 値 (実験では 3 まで) が、常に高い圧縮率を与える結果となった。画像信号そのものではなくインター/イントラ残差が変換対象であるため、必ずしも数学的 DCT($C=1+\sqrt{2}$)に近い方が高効率となるわけではないためと考えられる。

5. おわりに

本稿では、3 通りの可逆直交変換方式ならびに提案方式について、変換行列の行列式からみた特性等を比較するとともに、これらを H.264 の枠組にあてはめることにより可逆動画画像符号化を実現し、計算機実験によりそれらの符号化効率を比較した。その結果、提案方式が最も高い符号化効率を与え、Motion JPEG2000 とほぼ同等であることを示した。

表 2 P ビクチャの可逆圧縮率 (*QCIF,**CIF)

Table 2 Lossless compression ratios for P pictures (*QCIF, **CIF)

画像	量子化なし	方式 1 (FRExt)	方式 2 (区分スケール)	方式 3(可逆 DCT)			提案手法	
				$C=1+\sqrt{2}$	$C=3$	$C=3.5$	固定予測	適応予測
container*	1.729	4.472	3.635	3.840	3.854	3.861	4.412	4.487
silent*	1.731	4.334	3.588	3.890	3.899	3.920	4.441	4.419
news*	2.455	5.909	4.963	5.267	5.287	5.312	5.978	6.009
stefan*	1.200	2.312	2.151	2.188	2.203	2.215	2.351	2.367
coast**	1.224	2.289	2.145	2.241	2.249	2.286	2.423	2.442
foreman**	1.415	2.857	2.617	2.680	2.698	2.751	2.938	2.964
silent**	1.580	3.542	3.076	3.204	3.219	3.292	3.622	3.652
paris**	1.466	3.171	2.777	2.836	2.855	2.916	3.194	3.249
mobile**	1.148	2.124	1.989	2.010	2.023	2.064	2.178	2.196
Average	1.550	3.446	2.993	3.128	3.143	3.180	3.504	3.532

表 3 B ビクチャの可逆圧縮率 (*QCIF,**CIF)

Table 3 Lossless compression ratios for B pictures (*QCIF, **CIF)

画像	量子化なし	方式 1 (FRExt)	方式 2 (区分スケール)	方式 3(可逆 DCT)			提案手法	
				$C=1+\sqrt{2}$	$C=3$	$C=3.5$	固定予測	適応予測
container*	1.757	4.612	3.736	4.068	4.080	4.155	4.651	4.692
silent*	1.726	4.288	3.593	3.933	3.941	4.035	4.511	4.494
news*	2.520	6.164	5.152	5.536	5.555	5.698	6.343	6.368
stefan*	1.191	2.279	2.128	2.174	2.189	2.229	2.351	2.370
coast**	1.286	2.492	2.320	2.429	2.438	2.475	2.631	2.641
foreman**	1.399	2.841	2.620	2.693	2.711	2.753	2.916	2.952
silent**	1.590	3.632	3.162	3.343	3.358	3.408	3.706	3.735
paris**	1.521	3.465	2.988	3.106	3.125	3.162	3.436	3.501
mobile**	1.188	2.250	2.113	2.139	2.152	2.185	2.296	2.326
Average	1.575	3.558	3.090	3.269	3.283	3.344	3.649	3.675

本実験の可逆符号化においては、直交でない変換が他の直交変換よりも良い性能を示すことを示した。さらに提案方式における実用的な予測係数を求め、さらにその予測を適用的にすることで、符号化効率が改善できることも示した。

今後は、より相関除去能力の高い変換方式による効率の改善を検討する。

文 献

- [1] ISO/IEC 15444-3:2002 *JPEG 2000 image coding system – Part 3: Motion JPEG 2000*, International Standard, Sep. 2002.
- [2] ISO/IEC 14495-1:1999 *Lossless and near-lossless coding of continuous tone still images – Part 1: Baseline*, International Standard, Sep. 2000.
- [3] ISO/IEC SC 29/WG 11 (MPEG) 14496-10:2004, *Coding of audio-visual objects – Part 10: Advanced Video Coding 2nd Ed.*, International Standard, Oct. 2004.
- [4] Shijun Sun and Shawmin Lei, “On study of a lossless video coding algorithm based on H.26L tools,” in *Proceedings of SPIE*, Jan. 2003, vol. 5022.
- [5] 小松邦紀, 瀬崎薫, “濃淡画像の可逆的な変換符号化,” 信学論 A, vol. J79-A, no. 4, pp. 981–990, Apr. 1996.
- [6] <http://bs.hhi.de/~suehring/ttml/>, “JM reference software,” version 8.2, May 2004.
- [7] <http://www.kakadusoftware.com>, “Kakadu software,” version 4.2 linux executable, Feb. 2004.