

静止画像圧縮における適応型予測木の改良

Improvement of Adaptive Prediction Tree in Still Image Compression

小俣 順* 細谷 剛* 平澤 茂一*
Jun OMATA Gou HOSOYA Shigeichi HIRASAWA

Abstract— Recently, the applications of a digital still image is increasing because of the development of the information technology. It is necessary to choose the suitable encoding method for the various images to compress them to a high compression. APT(Adaptive Prediction Tree) proposed as a good compression technique for various type of image compression. In this paper, we propose a new method to improve a prediction in APT and show its superiority compared to the conventional methods by numerical experiments.

Keywords— APT, Predictive coding, Still-image coding

1 はじめに

近年、情報技術の発展によりデジタル静止画像を用いる機会が急増している。それにより、静止画像を圧縮するには画像の種類や画像の圧縮方法によって最適な圧縮方式が異なる。例えば自然画像の非可逆圧縮には、JPEG(Joint Photographics Experts Group)やJPEG2000などが適しており、可逆圧縮にはJPEG-LS [1]が適している。一方、人工画像の可逆圧縮にはPNGが適しているなど多様な画像に対して高圧縮を行うためにA. Robinson [2]によって多様な画像に適合した符号化方法が提案されている。

様々な画像の種類(人工画像や自然画像など)や可逆圧縮と非可逆圧縮が両方可能な圧縮手法としてAPT(Adaptive Prediction Tree)が提案されている。APTは予測符号化を利用しているが予測方法は人工画像に適しているため自然画像などの画像の変化が激しい画像に対しては予測の精度が低い。

本研究ではAPTの予測方法に改良を加えた手法を提案する。これを用いて、テストデータに対して従来のAPT予測法より圧縮率が向上することを示す。

2 準備

2.1 予測符号化法

予測符号化は画像圧縮において長い歴史があり、ほとんどの圧縮手法に使用されている。例えばJPEGでは、DC成分には予測符号化が使用されておりまた動画圧縮においても動き補償などには予測符号化が利用されている。その中でもJPEG-LS [1]などで採用されている“純

粋な”予測符号化法とは、画素をDCTなどの直交変換を行わずそのままの画素値を1画素ずつ予測する。その際には画素間同士の相関があることを利用して得られた予測値(\hat{x})と実際の画素値(x)の差分 $\varepsilon(x)=x-\hat{x}$ を符号化する方法である。

2.2 バイナリピラミッド [3]

予測を行う時、隣合う画素毎に予測を行うのではなく、まず格子状に予測を行いその誤差を符号化する。次にその中心の画素を予測して誤差を符号化する。それらの画素を組み合わせてまた新たに格子状に予測を行う。その時の予測は以前より細かくに予測していく。

次に図1で例を示す。最初に A_i ($i=1,2,3,4$)と書かれている画素値を符号化する。次のその中心の画素値 B_1 を近傍の A_i から予測し符号化する。その次に C_j ($j=1,2,3,4$)の位置の画素値を A_i と B_1 の位置の画素値から予測し符号化をしていく。その後、より詳細に予測していく。

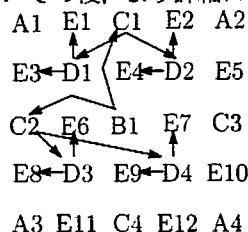


図1: バイナリピラミッドの予測順番と親子関係

次のステップに進むにしたがって予測していく画素の数が2倍ずつ増えていくためこの手順はバイナリピラミッド [3]と呼ばれている。予測していく画素が2倍ずつ増えていくことを利用してバイナリ木構造で符号化する。図1を例にすると C_2 の子供が D_3, D_4 となり孫が $E_6 \sim E_9$ となる。符号化する時 C_2 の子供と孫が全て0の場合 C_2 を符号化した後終了コードを符号化し子や孫を符号化する必要がなくなるため高圧縮が可能となる。なお、後に使用する階層とはバイナリツリーの深さのことである。

3 APT[2]

この節では従来手法であるAPT手法について述べる。その後、問題点と提案の着目点について述べる。

APTでは符号化する時ユーザーがパラメーター Q の値を0(低画質, 高圧縮)から100(可逆圧縮)まで設定する。ユーザーが設定したパラメーター Q から量子化パラメーター S_k は以下の式で計算する。なお k はバイナ

* 〒169-8555 東京都新宿区大久保 3-4-1 早稲田大学理工学部経営システム工学科 School of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan. E-mail: omata@hirasa.mgmt.waseda.ac.jp

リピラミッドの量子化の階層レベル、 M はバイナリピラミッドの総階層数である。

$$S_k = \max(0.8^{M-k} \frac{(100-Q)^2}{100}, 1)$$

以下に APT のアルゴリズムを説明する。

[APT の符号化アルゴリズム]

Step1. 色変換：画像データを 3 つの成分に分ける。(モノクロ画像なら変換を行わない)

Step2. Step1 で分けた成分の内、第一成分をバイナリピラミッドより格子状に予測を行う。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step3. 他の成分の予測を第一成分での予測方法を参照して最も予測誤差が少なかった予測器を利用する。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step4. 木構造を用いてランレングス・ハフマン符号を行う。

□

3.1 第一成分の予測方法

Step2 での第一成分の予測方法について説明する。格子状に読み込んだ第一成分¹ の画素を予測する方法を以下に示す。なお画素の位置関係は次の図 2 に示す。 X は予測する画素、 A, B, C, D は以前の階層までに既に予測済みの画素の中で X に近い画素、 U と V は同じ階層での最も近い画素である。

[第一成分の予測器の選択方法]

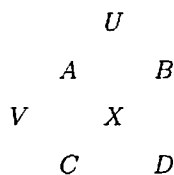


図 2: 予測に使用する参照画素

次の (1) ~ (7) のいずれかを実行する。 \hat{x} は X の位置の画素の予測値である。 a, b, c, d, u, v は位置 A, B, C, D, U, V の画素値である。

- (1) もし $a = b = c = d$ ならば $\hat{x} = a = b = c = d$ とする
- (2) もし $a = b$ かつ $c = d$ ならばもし v が画面内に存在すれば $\hat{x} = v$ それ以外なら $\hat{x} = (a + b + c + d)/4$ とする
- (3) もし $a = c$ かつ $b = d$ ならばもし u が画面内に存在すれば $\hat{x} = u$ それ以外なら $\hat{x} = (a + b + c + d)/4$ とする
- (4) a, b, c, d の最大値と最小値の値が $T = 10 \log S_k$ より小さいならば $\hat{x} = (a + b + c + d)/4$ とする

¹ カラー画像なら RGB に色変換をした第一成分、モノクロ画像なら画素値そのもの

(5) a, b, c, d の内 3 つが同じ値 y 、残り 1 つの値が z ならば $\hat{x} = y$ とする

(6) もし b と c の両方が a と d より大きい、もしくは a と d の両方が b と c より大きいならば次の (a) ~ (d) のいずれかを実行する

(a) もし b と c の両方が a と d より大きいならば $r = (b + c)/2$, $v = (a + d)/2$ とする

(b) それ以外なら $r = (a + d)/2$, $v = (b + c)/2$ とする

(c) もし符号化する値 x が v より r に近い場合シングルビット 1 が符号化され $\hat{x} = r$ とする

(d) それ以外ならシングルビット 0 が符号化され $\hat{x} = v$ とする

(7) それ以外なら $\hat{x} =$ メジアン (a, b, c, d) とする

□

3.2 他の成分の予測

Step3 では第一成分を予測した後、他の成分の予測を次に行う。その時、第一成分の予測方法を利用することでより高精度の予測が可能である。第一成分を予測して符号化した後に一度復号をして、3.1 節の第一成分の予測器の選択方法で決定した予測器より精度が高い予測器があるか調べ、もしあれば他の成分の予測器は第一成分のように周辺画素の大きさによって決めるのではなく、第一成分で最も予測精度が高かった予測器を使用する。予備実験より、他の成分との画素値との相関はないが、最も良い予測器の方向には関係があることが分かっているので高精度の予測が可能になる。

3.3 木構造

Step4 の木構造について説明をする。APT では全ての色成分に対して 1 つの木を用いる。つまり図 1 では第一成分の B1 は全ての成分の C1, C2 の親となる。よって第一成分の親は子が 6 個となる。3.2 節で第一成分の予測方法を用いることにより Step3 で 3 つの成分に分けた第二、第三成分の予測の精度が高くなり予測誤差が 0 になる事が多くなるので高圧縮が可能となる。

3.4 従来手法の問題点

従来手法では画素を予測する時、その周辺画素の大小関係のみに着目して予測器を選択している。その場合、人工画像のように突然変化が起こるような画像に対しては高精度の予測となるが、自然画像のように徐々に画素値が変化している画像に対しては粗い予測しか出来ず、詳細な変化には対応できていないため予測の精度が高くない場合がある。

4 提案手法

4.1 提案手法の着想

前節で述べた従来手法の問題点を解決するために予測器を周辺画素の大小で選択するのではなく周辺画素の状態 (コンテキスト) を計算してそのコンテキストから予

測値を修正する方法を提案する。格子状で予測を行うため粗い状態での予測ではコンテキストにばらつきが生じる。そこで最下位層のレベルのみコンテキストによる修正を加える。最下位層では図3に示すように、 x の予測を行う時に4方全ての画素値を既に予測済みのためそれらを使用することでより高精度の予測が可能になると考えられる。

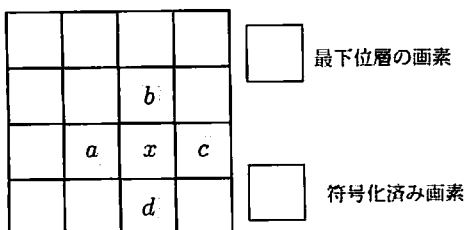


図3: 拡張時の参照画素例

4.2 最下位層でのコンテキスト

最下位層での予測方法ではコンテキストを用いて予測値を修正する。コンテキストは図3で x を予測する場合以下の方法で分ける。 $a \sim d$ は周辺画素の画素値である。

$$g_1 = b - a, g_2 = c - b, g_3 = d - c, g_4 = a - d$$

$g_1 \sim g_4$ を全てまとめた $[g_1, g_2, g_3, g_4]$ は局所的な勾配を表している。 g_j ($j=1, 2, 3, 4$)をそのまま利用すると g_j のとり得る値が $-255 \sim 255$ のため、 511^4 個のコンテキストとなり非常に多すぎるので g_j を量子化した値 q_j をコンテキストとして用いる。その方法を以下に示す。

$$q_j = \text{sign}(g_j) * a \quad ; j=1,2,3,4$$

$$\text{sign}(g_j) = \begin{cases} 1 & \text{if } g_j \geq 0 \\ -1 & \text{if } g_j < 0 \end{cases}$$

ここで $a \in a_i$ ただし a_i は $|g_j|$ を含む集合である。すなわち $|g_j| \in a_i$ 。ここで $a_1 = \{0\}, a_2 = \{1, 2\}, a_3 = \{3, \dots, 6\}, a_4 = \{7, 8, \dots, 20\}, a_5 = \{|g_j| \geq 21\}$ とする。

例えば、 $g_1 = 0, g_2 = -4, g_3 = 9, g_4 = -22$ ならば $[q_1, q_2, q_3, q_4] = [a_1, -a_3, a_4, -a_5]$ とコンテキストが決まり、次の方法で修正値 L を計算する。

$$L = [(N + M/2)/M], \hat{x} := \hat{x} + L$$

なお、 M はコンテキスト $[q_1, q_2, q_3, q_4]$ が検出された回数、 N はコンテキスト $[q_1, q_2, q_3, q_4]$ が検出された時の誤差の累計和である。その後、次に備えてコンテキストを更新する。

$$M := M + 1, N := N + (x - \hat{x})$$

4.3 提案手法のアルゴリズム

以下に提案手法のアルゴリズムを説明する。Step2～5までが従来手法と異なる。

[提案手法のアルゴリズム]

Step1. 色変換：従来手法と同じ

Step2. 第一成分をバイナリピラミッドより格子状に最下位層の1つ前の層まで予測を行う。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step3. 第一成分の最下位層の周辺画素の平均を計算する。 $\hat{x} = (a + b + c + d)/4$
その後、コンテキストを利用して修正を行う。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step4. 他の成分の予測を最下位層の1つ前の層まで行う。その時使用する予測器は第一成分で最も予測誤差が少なかった予測器を利用する。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step5. 他の成分の最下位層の周辺画素の平均を計算する。 $\hat{x} = (a + b + c + d)/4$
その後、コンテキストを利用して修正を行う。実際の画素値から予測値を引いて予測誤差を計算して量子化を行う。

Step6. 木構造を用いてランレングス・ハフマン符号を行う。 □

5 評価実験

5.1 可逆圧縮

評価実験としてSIDBA標準画像[4]などに対して提案手法を適用し、これらの符号化性能を調べた。その際APT1.0[5]に提案手法を適用した。画像の可逆圧縮の原画像に対する圧縮率(bits/pixel)を表1に示す。

表1: 実験結果 [圧縮率 (bits/pixel)]			
画像の名前	APT	提案手法	提案/APT
Greyscale images (8bpp uncompressed)			
lena	4.62256	4.71606	0.980174128
peppers	4.27454	4.36426	0.979442105
aerial	5.83716	5.88538	0.991806816
redhead	4.33691	4.44934	0.974731084
home	4.31726	4.4847	0.962655583
linespr	1.18384	1.17834	1.004667583
Color images (24bpp uncompressed)			
airplane	12.1572	12.4615	0.975580789
girl	13.9518	14.3073	0.975152545

5.2 非可逆圧縮

次に非可逆圧縮で提案手法を適用し、これらの符号化性能を調べた。lena画像におけるbits/pixelとPSNR(Peak Signal-to-Noise Ratio=ピーク信号対雑音比)の関係を表したグラフを図4に示す。なお、PSNRとは画像の信号と混入したノイズとの比率であり、単位はdBで表現される。この値が大きければ大きいほど画質の劣化が少ないことを意味する。

6 考察

6.1 可逆圧縮について

表1から自然画像に対しては従来手法との比較で平均2.7%の圧縮率の向上がみられた。それに対して人工画

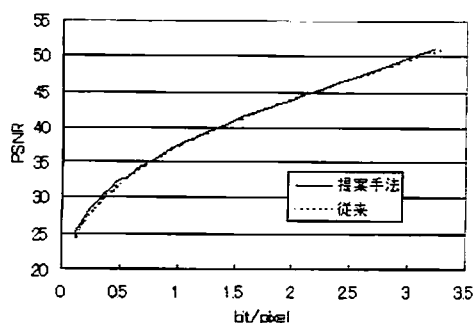


図 4: lena 画像の SN 比

像 linespr に対しては 0.5 % 悪化してしまった。人工画像の圧縮率が悪化してしまった理由として、今回の修正方法では自然画像のように徐々に画像が変化していく場合に周辺画素から特徴を見つけ修正を加えていくに適しているが、人工画像の場合、ある場所が一定で徐々に変化していくのではなく、突然大きく変化が起こりまた平坦になることが多い。そのため急激な場合には対応できず圧縮率が悪化した。しかし、従来手法である APT は元々人工画像圧縮に適しており、人工画像の圧縮率は国際標準である JPEG-LS に比べて圧縮性能が 2 ~ 4 倍程優れているので、圧縮性能の悪化はそれほど大きくないと考えられる。

自然画像の圧縮率が向上しているのは今回の予測方法に修正を加えることにより、予測の精度が向上したためと考えられる。図 5 は lena 画像の最下位層での予測誤差の個数である。図 5 から確認できるように予測誤差が 0 の個数が増えていることが確認できる。画像によって

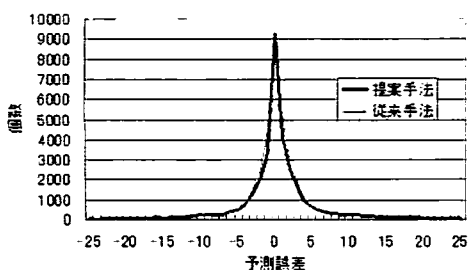


図 5: lena 画像の誤差の個数

圧縮率の向上にバラツキがあるが、変化が激しい画像に対しては予測が難しい。そのために修正を加えてもまだ予測誤差が大きくなってしまいうために圧縮率が向上していない。それに対して変化が緩やかな画像に対しては修正を加えることによって予測精度が向上しやすいために圧縮率が改善している。

6.2 非可逆圧縮について

図 4 から提案手法は従来の APT と同等もしくはそれ以上圧縮性能が向上したことが分かる。圧縮性能がほぼ同じ理由は、画像の予測誤差を量子化するため、ある程度予測誤差が大きく変わらない限り量子化により同じ値になることが多い。通常静止画像の場合、近傍画素との

相関が強く予測誤差が大きくなることはほとんど無く、0 付近に集中するために、予測器を修正しても量子化によってほとんど影響が無くなる。そのため、非可逆圧縮で性能の向上わずかであったと考えられる。他の画像やカラー画像に対しても同じような結果となった。

6.3 計算量について

予測器を変更するため最下位層では 1 画素あたりの計算量は増える。しかし、1 画素あたり最大でも、加算が 11 回、乗算が 3 回増えるのみのため計算量はほとんど変わらないと考えられる。また最下位層の画素の個数は全画素の半分である。計算機上での実行時間もほぼ同じであった。

7 まとめと今後の課題

本論文では、APT での予測方法の中でも最下位層での予測について、周辺画像の大小関係で予測器を選択するのではなく、周辺画素のコンテキストからそのコンテキストでの累計誤差と累計回数から予測値を修正する方法を提案した。これにより自然画像の可逆圧縮では、圧縮率の向上を図ることが可能になり、不可逆圧縮においても同等か向上することを示した。しかし、人工画像に対しわずかながら性能が悪化してしまった。

今後の課題としては、自然画像、人工画像に限らず全ての画像に対してより高精度の予測方法を提案し圧縮性能のさらなる向上をはかりたい。

8 謝辞

著者の一人である小俣は、本研究を行うにあたり数多くのご助言、ご支援を賜りました早稲田大学平澤研究室の各氏に感謝いたします。

文献

- [1] M.J.Weinberger,G.Seroussi,“The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” *IEEE Trans.Image Process.*, vol.9,no.8. 2000. pp.1309-1324.
- [2] John A. Robinson, “Adaptive Prediction Trees for Image Compression,” *IEEE Trans.Image Process.*, vol.15.NO.8. 2131-2145, August 2006.
- [3] J.A.Robinson, “Efficient general-purpose image compression with binary tree predictive coding” *IEEE Trans.Image Process.*,vol.6,no.4.pp.601-607.Apr.1997.
- [4] SIDBA 標準画像 <http://vision.kuee.kyoto-u.ac.jp/lecture/dsp/pnm/index.html>
- [5] APT Online Reference Code.Online.Available: <http://www.intuac.com/userport/john/apt/index.html>