

## 大きさの違う区割りに基づく予測符号化の研究

平松章 † 坂口直秀 †  
淵田孝康 † 村島定行†

この論文では予測符号化の際に原画像の特性に基づいて異なるサイズの区割りを設けることで、データ量の削減を目指したものである。画像の種類によっては JPEG-LS を上回る圧縮率を誇る。

### Study of Predict Coding based on Multi Size Division

AKIRA HIRAMATSU,† NAOHIDE SAKAGUCHI,† TAKAYASU FUCHIDA†  
and SADAYUKI MURASHIMA†

In this paper, we suggest a predict coding based on multi size division based on characteristics of original image, we aim improvements on reduction of data. On some pictures, this method is better than JPEG-LS.

#### 1. ま え が き

現在一般に用いられている画像圧縮技法に多く使用されている予測符号化。その符号化手法は近隣の画素値から予測計算を行い、実際の画素値との差を取ることで 0 付近に集中した値を得ることができエントロピー符号化をするのに都合がよい形にデータを変更できるというものである。予測符号化の際には一画素ごとに予測計算を行うのであるが、ここでは等輝度であるブロックはその中の一画素のみを扱い、残り三画素を切り捨てることでデータ量の削減を図る。

#### 2. 原 理

##### 2.1 離散原画像

サイズが  $M \times N$  である離散原画像  $G$  を以下のよう

$$G = \{g_{i,j} | 0 \leq i \leq M-1, 0 \leq j \leq N-1\} \quad (1)$$

ここでは離散原画像の画素  $g_{i,j}$  は 0~255 の値をとる 8 ビットデータである。

##### 2.2 予測符号化

左側と上側に隣接する画素を用いた予測符号化を式 (2) に示す。

$$l_{i,j} = \left\lfloor \frac{g_{i-1,j} + g_{i,j-1}}{2} \right\rfloor - g_{i,j} \quad (2)$$

尚、画像の左端や上端などの場合には次の (3)~(5) 式を適用する。

$$l_{0,0} = g_{0,0} \quad (3)$$

$$l_{i,0} = g_{i-1,0} - g_{i,0} \quad (4)$$

$$l_{0,j} = g_{0,j-1} - g_{0,j} \quad (5)$$

原画像の画素値  $g_{ij}$  を予測符号化した  $l_{ij}$  は  $g_{ij}$  と比べて 0 付近に集中した値となる。図 1 に示す lisaw 画像に対して予測符号化を行った場合の  $g_{ij}$  と  $l_{ij}$  を比較したものを図 2 に示す。



図 1 離散原画像 lisaw(320 × 200pixels)

一般画像であれば程度の差はあるものの lisaw 画像と似たように 0 付近に集中した値となる。これによって算術符号などのエントロピー符号化の圧縮率を向上することができる。算術符号を用いて  $g_{ij}$  を圧縮した場合と  $l_{ij}$  を圧縮した場合の結果を表 1 に示す。

表 1 で使用した画像はいずれもサイズ 320 × 200 のグレースケール画像である。

† 鹿児島大学工学部情報工学科  
Infomation and Computer Science Faculty of Engineering Kagoshima University

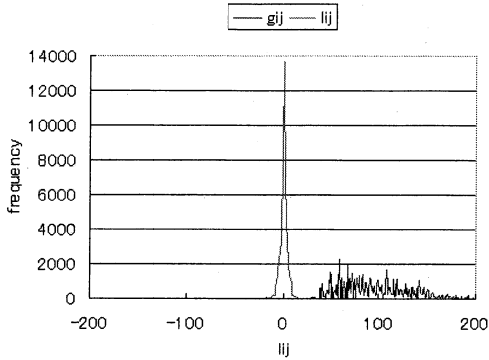


図2  $l_{i,j}$  の分布

表1 圧縮効率の違い

画像	lisaw		mouse		clown	
	$g_{i,j}$	$l_{i,j}$	$g_{i,j}$	$l_{i,j}$	$g_{i,j}$	$l_{i,j}$
データ						
圧縮前	64000	64000	64000	64000	64000	64000
圧縮後	52819	31139	50458	28808	53783	43568

### 2.3 復号化

$l_{i,j}$  から  $g_{i,j}$  を復号するには (2) 式の予測計算を (6) 式のように逆に適用する。

$$g_{i,j} = \left\lfloor \frac{g_{i-1,j} + g_{i,j-1}}{2} \right\rfloor - l_{i,j} \quad (6)$$

画像の左端や上端などの場合も同様に次の (7)~(9) 式を適用する。

$$g_{0,0} = l_{0,0} \quad (7)$$

$$g_{i,0} = g_{i-1,0} - l_{i,0} \quad (8)$$

$$g_{0,j} = g_{0,j-1} - l_{0,j} \quad (9)$$

### 2.4 大きさの違う区割りに基づく予測符号化

前節で述べたように予測符号化は 1 画素ごとに区割りし予測計算を行う符号化であるが、原画像の特性に合わせて部分的に複数画素で区割りをを行い、複数画素の区割りではその中の 1 画素のデータ以外を切り捨てることでデータ量の削減を目指す。ここで言う原画像の特性とは隣接する画素が等輝度である部分がどれだけあるかということである。いくつかの画像に対してこのような区割りがいくつあるかを調べた結果を表 2 に示す。

lenna 画像のようにほとんど等輝度のブロックないものについては圧縮率の向上は期待できないが、rose 画像のように数多くある場合では十分な期待が持てる。

次節以降では  $2 \times 2$  ブロックを扱う場合について述べていく。

表 2 等しい輝度値をもつ  $2 \times 2$ ,  $4 \times 4$  ブロックの個数

画像	pixels	$2 \times 2$	$4 \times 4$
lenna	256×256	22	0
mouse	320×200	2825	407
rose	320×200	5870	1274
cheet	320×200	173	0
clown	320×200	955	58
lisaw	320×200	1614	12

### 2.5 区切り子

まず原画像を  $2 \times 2$  のブロックに分け、そのブロック内の画素が全て等輝度であるかどうかを調べる。この時、各ブロックにおいて内部の画素が全て等輝度であるかどうかを示すものとして区切り子  $f_{i,j}$  を導入する。 $f_{i,j}$  は (10) 式、図 3 に示すようにブロック内が全て等輝度であれば 1、そうでなければ 0 となる。

$$f_{i,j} = \begin{cases} 1 & (\text{if } g_{i,j} = g_{i+1,j} = g_{i,j+1} = g_{i+1,j+1}) \\ 0 & (\text{otherwise}) \end{cases} \quad (10)$$

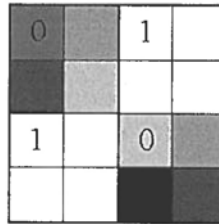


図3  $f_{i,j}$  の配置

### 2.6 区割りごとの予測符号化

予測符号化自体は左側と上側の隣接画素から計算する一般的な手法を用いる。ただし、区切り子  $f_{i,j}$  が 1 である場合は  $l_{i,j}$  だけを記憶し、ブロック内の残りの画素  $l_{i+1,j}$ ,  $l_{i,j+1}$ ,  $l_{i+1,j+1}$  の値は記憶しないのである。しかし、この状態から  $g_{i,j}$  を復号しようとするとき  $l_{i,j}$  の値が抜けているため従来通りの手法ではうまくいかない。そこで  $f_{i,j} = 1$  であったブロックでは  $l_{i+1,j}$ ,  $l_{i,j+1}$ ,  $l_{i+1,j+1}$  に  $l_{i,j}$  の値を代入してから復号を行う。その様子を図 4 に示す。

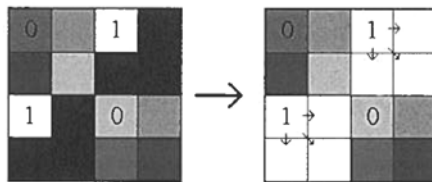


図4  $f_{i,j}$  による  $l_{i,j}$  のコピー

## 2.7 圧縮結果

いくつかの離散原画像を提案手法で符号化してから算術符号で圧縮した結果を表3に示す。尚、従来通り1画素ごとに予測符号化してから算術符号した結果、同様の手法で4×4ブロックまで拡大した場合の結果、予測符号化を用いた可逆圧縮手法の例としてJPEG-LSの結果も併せて表記する。

表3 圧縮結果

画像	pixels	従来手法	2×2	4×4	JPEG-LS
lenna	256×256	66.1 %	66.1 %	66.1 %	53.8 %
mouse	320×200	45.0 %	41.8 %	41.3 %	38.0 %
rose	320×200	57.1 %	47.5 %	45.0 %	46.4 %
cheet	320×200	76.4 %	76.2 %	76.3 %	74.2 %
clown	320×200	68.1 %	66.4 %	66.3 %	62.5 %
lisaw	320×200	48.7 %	46.7 %	46.8 %	46.1 %

隣接画素が等輝度のブロックが多い画像であればブロックサイズを2×2から4×4に拡大したほうが圧縮率が向上しているが、逆に悪くなってしまう場合もある。このことはブロック数を探索した後でその数が一定の数値を超えていればサイズを拡大し、そうでなければ拡大しないというようなやり方で最適なブロックサイズを選択すれば回避できる。JPEG-LSと比較した場合はほとんどの画像で圧縮率が劣ってしまったが、lisaw画像ではほぼ同程度 rose画像ではわずかに勝る結果が得られた。

## 3. 結 論

画像の特性に合わせて予測符号を全て1画素ごとではなく部分的に複数画素を纏めて扱うことでデータ量を削減し圧縮率の向上に努めた。画像の特性によっては複数画素を用いずに従来通りの手法で行ったほうがいい場合や、あまり効果の出ない場合があったものの、複雑な予測符号を含んでいるJPEG-LSに勝る場合もあるなどよい結果も得られた。今後、画像に最適なブロックの大きさや形状を模索し、より多くの画像で既存手法を上回る結果を得られるよう努力する必要がある。

## 参 考 文 献

- 1) M, ネルソン/J.-L. ギェリー著 荻原剛志・山口英訳 “データ圧縮ハンドブック マルチメディアデータ圧縮の実践的プログラミング技法”, Pearson Education Japan.
- 2) 白井規之, 松田一朗, 伊東晋 “レートを最小とする予測器と算術符号を用いた画像の可逆符号化”, FIT2002J-73

- 3) 松田一朗, 森弘史, 前田譲治, 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”, 電子情報通信学会論文誌D-II, Vol.J85-D-II No.3 pp.448-456, 2002