

2 値算術符号による画像情報の圧縮符号化*

半 田 志 郎**

Binary Arithmetic Coding of Images

Shiro HANDA

—The image has a huge amount of data. So a huge amount of memory device is necessary for storage. In this paper, we propose a novel approach to data compression of images. Usually multi-level data such as image is encoded by multi-level encoder which has very complex scheme. We decomposed a multi-level outcome emitted from the source into binary letters. Then the data of binary letters is encoded directly by the binary arithmetic encoder which is very powerful for compression of data. An adaptive scheme for data with unknown statistics and Markoff model coding scheme for removing redundancy by correlation between neighboring pixels is considered. We show the performance for various types of image.

1. ま え が き

画像信号は膨大なデータ量を持っており、そのまま記憶すると膨大な記憶容量を必要とする。更に高精細な画像が求められる今日では、扱うべき画像のデータ量は、記憶装置の大容量化の進歩にもまして増加の一途を辿っている。

一方、画像信号は一般的に隣接する画素の輝度に類似性が高く、周波数スペクトルが低域に集中しているなど、データ量の割合には情報量が少ないという見方もできる。このような統計量を利用して画像を符号化する方法には、大別して2つの方法がある。一方は画像信号を歪みなく再現することが可能な無歪み符号化法であり、他方は人間の視覚効果を利用して目立たない程度の歪みを許す有歪み符号化法である。テレビ会議等の一過性の画像の場合には後者の方法が適用されるが、データベース等に格納する写真データの様に、再生した後じっくり吟味するというような用途には無歪みの符号化法が使われる。

本論文では、無歪み符号化法を2値算術符号を用いて実現する。算術符号は可逆性、漸近的最良性が保証されているなどの特長を持つ強力な符号である。更に、入力データの確率パラメータを求めるモデル化部とそのモデルに基づく符号化部が明確に分離可能であるという大きな特長がある。

画像データを符号化する場合、通常は一つのデータを単位として符号化する⁽¹⁾。ところが算術符号⁽²⁾の使用を前提とした場合、画像のように多値の情報源(通常256値以上程度の値を取る)を扱う算術符号は、符号化過程において多くの演算を必要とし、アルゴリズムも複

* 本研究の一部は、浅間テクノポリス、長野県テクノハイランド開発機構の援助による。

** 電気工学科 助教授

雑となる。また、同一の演算器を用いる場合には、2値算術符号化と比べ、アルファベットの個数分だけ区間の分割が必要となるため、結果として演算精度が低下したことになり、圧縮率の低下をもたらすと考えられる。

以上の認識に立って、本論文では2値算術符号による画像の高能率符号化法を検討し、更に実際の画像データに対してその性能を検討している。まず、多値情報源を2値の成分を持つベクトルとして表現し、それと同時にその確率分布を2値情報源の確率分布を用いて表現する。この分解した確率分布により、多値情報源が効率よく符号化できることを示し、さらに情報源が定常エルゴード的と考えられるならば、分解によるロスなしに2値算術符号で容易に符号化できることを示す。次に画像信号を多重のマルコフ情報源でモデル化して、画素間の相関を取り除く方法を示す。その際、情報源を記述するパラメータが多くなりすぎ、符号化効率が低下してしまうという問題を状態の縮退という考え方を用いて解決している。

2. 算 術 符 号

一般に圧縮符号化とは、生起確率の大きな系列に短い符号語を、生起確率の小さな系列に長い符号語を割り当てることによって、平均の符号長（送るべき符号の平均の長さ）を最小にしようとする方法である。算術符号もこの様な考え方に則った方法であるが、符号化が四則演算と、数の大小比較によって行われるため、その名が付けられている。ここでは簡単のため、2値{0,1}のデータ系列を2値のデータ列に符号化する方法を示すことにする。

一例として、0,1の生起確率が $1/3, 2/3$ のデータ列の符号化を考える。まず区間 $[0,1)$ を図1のように0,1の生起確率に従って2つの区間に分割する（多値情報源の場合には、多値数の数だけの区間に分割する）。初めのデータが0であれば下側、1であれば上側を選ぶ。この時、0に割り当てられた区間 $[0, 1/3)$ は先頭が0のデータ列の累積確率を示し、 $[1/3, 1)$ は先頭が1のデータ列の累積確率を示すことになる。次に、選ばれた区間を再び生起確率に従って分割し、次の入力データに従って区間を選ぶ。以下同様にして区間を選択してゆく。図1は入力データが「110」の場合の例である。この時、最終的にどの区間が選ばれ

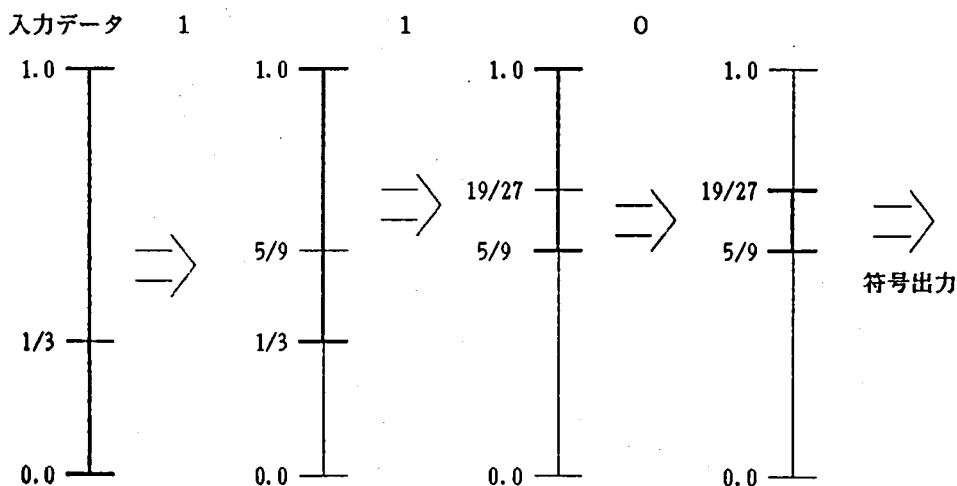


図1 算術符号の原理

たかを伝送すれば、受信側でも同様の分割方法によりどの区間が選ばれているかが分かるので、一意に復号化が可能である。この区間を示すためには、その区間内の数値の代表値を小数点以下 $-\log_2 p(110)$ ビット以上の最小の長さの2進数で表わせばよいことが、シャノンの第一定理から容易に分かる。また、このことから平均符号長が情報源のエントロピーに漸近する事も分かる。ここで、 $p(110)$ は系列「110」の生起確率であり、図1でいえば最終的に選ばれた区間の幅に等しい。容易に分かるように、生起確率の高い系列ほど選ばれる区間の幅が大きくなり、短い符号語が割り当てられる。また、このアルゴリズムにおいて、区間の幅は符号化が進むにつれて狭くなり、精度の高い演算が必要なように思われるが、浮動小数点演算の考え方をいれれば固定精度の演算で十分である。さらに区間の分割は送信側と受信側で同期していれば変更してもかまわないので、適応的符号化は容易である。

3. 多値情報源の符号化

3.1 確率分布の分解表現

多値情報源 X からの出力系列 X_i が 2^n 個の値を取り得るものとする、 X_i は

$$X_i = (x_{i,0}, x_{i,1}, \dots, x_{i,n-1}) \quad (1)$$

とベクトルで表現できる。ここで、 $x_{i,j}$ は0または1のみの値をとるものとする。

この様に表現したビット列は、多値情報源 X が定常的であるならば、明らかにブロック定常となり、各ビットはそれぞれ異なった確率分布を持つことになる。また、文字のコード表現の様な場合には、ビット位置による重要度の様なものはないが、画像情報のように輝度レベルを表した数値の場合には、上位ビットほど重要な意味を持っているので、符号化においてもそのような構造を保持したまま取り扱うことが望ましい。

X_i の出現確率 $p(X_i)$ はこのベクトル表記に対応して、

$$\begin{aligned} p(X_i) &= p(x_{i,0}, x_{i,1}, \dots, x_{i,n-1}) \\ &= p(x_{i,0})p(x_{i,1}|x_{i,0})p(x_{i,2}|x_{i,0}, x_{i,1}) \dots p(x_{i,n-1}|x_{i,0}, x_{i,1}, \dots, x_{i,n-2}) \end{aligned} \quad (2)$$

となる。ここで、 $p(\cdot|\cdot)$ は条件付確率を示す。この表現によると、 X_i の符号化は $p(X_i)$ 自身を用いて多値で行う必要はなく、2値情報の条件付確率分布を用いて、2値算術符号で行えばよい。また、この分解によってエントロピー（符号化の限界を示す）の変化はないので、分解によるロスはない。また、情報源を記述するパラメータの数が不変であることも、容易に確かめられる。例えば、情報源 X_i が256($n=8$)値の値を取り得るものとする、上の分解で、 $p(x_{i,0})$ の記述に1つ、 $p(x_{i,1}|x_{i,0})$ に2つ、以下4, 8, ……、128, 合計で255個のパラメータとなり、256値情報源を記述するのと同一である。

次に情報源をマルコフ情報源によってモデル化する場合には、ある時点での出力は以前の出力系列に依存することになる。すなわち、以前の出力系列によって決まる状態 S_i によって現在の出力 X_i が左右される。従って、出現確率も条件付き確率 $p(X_i|S_i)$ を用いて示される。これも式(2)と同様にして、

$$\begin{aligned}
 p(X_i|S_i) &= p(x_{i,0}, x_{i,1}, \dots, x_{i,n-1} | S_i) \\
 &= p(x_{i,0} | S_i) p(x_{i,1} | x_{i,0}, S_i) p(x_{i,2} | x_{i,0}, x_{i,1}, S_i) \dots p(x_{i,n-1} | x_{i,0}, x_{i,1}, \dots, x_{i,n-2}, S_i) \quad (3)
 \end{aligned}$$

と分解することによって、2値算術符号で容易に符号化できることが分かる。この場合も、情報源を記述するパラメータ数は、多値情報源をそのまま記述するのと同様である。

3.2 適応的符号化

一般に、これから符号化しようとする情報源の確率分布が既知であることは稀である。そのような場合にも効率よく符号化を行うため、次の2つの方法が考えられる。

- (1) 静的符号化法 確率分布を調べ直して、これを受信側へ送った後に符号化を行う方法。
- (2) 適応的符号化法 データが入ってくる度に確率分布を更新して、常にその時点で推定できる最良の確率分布を用いて符号化を行う方法。

情報源が定常エルゴード的であれば、上記の2つの方法は理論的に同一の符号化効率を達成することが知られているが⁽³⁾、(1)の方法は情報源を一度先読みして確率分布を求める必要があるため、現実的には、データの2度読みといった操作が必要となり、符号化速度の点でも不利である。また、設定したモデルのパラメータをすべて受信側へ送らなければならないので⁽⁴⁾、モデル設定の難しさも直接効いてくることになる。すなわち、大きなモデルは情報源をよくモデル化するがそのパラメータを表示するのに多大の情報量が必要になる。これに比べ、(2)の方法はデータの2度読みの必要もなく、また多少大きなモデルを設定しても、パラメータ推定においてモデルの使用しない部分には到達しないといった簡単な自動調整機構が作用するので、それほどモデル設定の難しさが効いてこないという利点がある。そこで、本論文でもこの適応的符号化法を用いることにする。

図2に示す深さnの木を考える。右の枝は入力情報の1、左の枝は0に対応し、各枝にはカウンターが一つずつ付いている。この木を用いて符号化を行う。

符号化法：

- [1] X_i に対して式(1)の分解によって $x_{i,j}$ を得る。
- [2] ポインタをrootに移し($k=0, j=0$)とする。

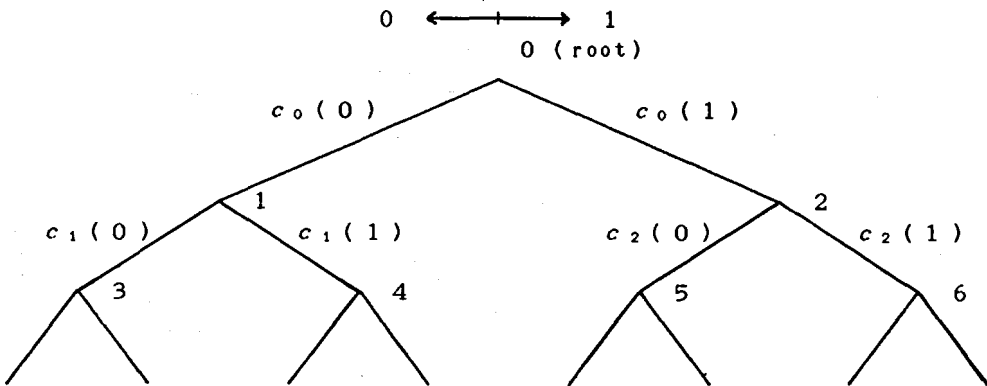


図2 符号化の木

[3] $x_{i,j}$ をポインタ k で示される2つの枝のカウンタ値 $c_k(0), c_k(1)$ によって算術符号化する。

[4] $c_k(x_{i,j}) = c_k(x_{i,j}) + 1$

[5] $x_{i,j}$ が0なら左下, 1なら右下のノードにポインタを移す。

[6] $j = n - 1$ なら終了。そうでなければ, $j = j + 1$ として [3] へ。

以上の符号化において, 木が一段深くなることは, 一つ条件が多く付いたことを意味し, 式(2)の確率分布の分解表現とよく対応しているのが分かる。さらに, 数値情報のように上位ビットと下位ビットで重みが違う場合も, 木構造の根に近い重要な部分ほど多く通過するので, 確率分布の曖昧性がなくなり, 従って真の確率分布に近い確率で符号化できる。また, 各ノード k での算術符号化においては,

$$p_k(0) = \frac{c_k(0) + 1}{c_k(0) + c_k(1) + 2} \quad (4)$$

$$p_k(1) = 1 - p_k(0) \quad (5)$$

として符号化を行っている。これは, 連続の法則 (*The law of succession*) と呼ばれ⁽⁵⁾, $c_k(0) + c_k(1)$ 回の観測の後に, 次の出力が0または1である確率の期待値が上式で与えられることに基づいている。すなわち, 真の確率分布の最良の予測になっている。

4. 画像のモデル

画像は隣接する画素間の相関が強いので, これを利用するとさらに圧縮率の向上が期待できる。このような相関を確率過程としてモデル化するのに, マルコフモデルがよく使われる。ところが, 画像は2次元情報であるため, モデル化の仕方によって圧縮率が大きく異なる。本論文では, 次の3種類のマルコフモデルを検討する。

モデル1: 1次元マルコフモデル

画素 $Y_{i,j}$ が左直前の画素 $Y_{i,j-1}$ に依存している。

表1 実験に用いた画像データ

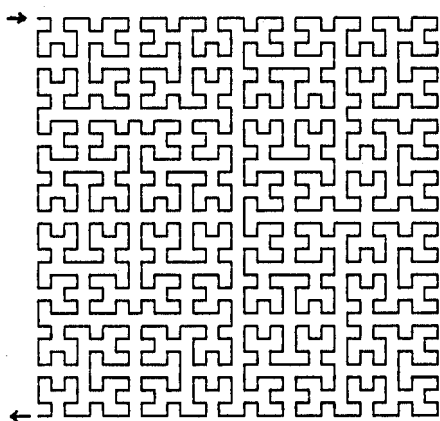


図3 スタンダードサンプリング

ファイル	内 容	compress
girl	女性上半身像	22.35%
couple	男女二人全身像	30.31%
aerial	航空写真	0.00%
moon	月面写真	7.06%
chest	胸部レントゲン写真	32.11%
facts	ファックス原稿写真	43.99%
girl2	女性上半身像	15.42%
area	航空写真	10.65%
barbara	縞模様の服の女性像	1.24%
mandrill	マントヒヒ顔面像	0.00%

モデル2：2次元マルコフモデル

画素 $Y_{i,j}$ が左直前の画素 $Y_{i,j-1}$ と真上の画素 $Y_{i-1,j}$ に依存している。

モデル3：スタンダードサンプリング2重マルコフモデル

画像を図3に示す曲線の順番にサンプリングした場合、ある時点での画素値がその直前の2画素に依存する。

以上のマルコフ情報源モデルを適用する場合には、3.2節で示した符号化の木を状態の数だけ用意し、各状態で別々の木を用いて符号化を行う。ところが、画像のように多値(256)の値を取る情報源をこのようなモデルで符号化すると、符号化の木の数はモデル1で256、モデル2、3で65536と非常に多くなり、その結果各符号化の木におけるカウンタの値は希薄となり過ぎて、確率分布の推定が曖昧になり、圧縮率が低下してしまう。そこで、本論文では状態として参照データの上位数ビットのみを用いることにする。これは、本来のマルコフモデルに対して、状態縮退したモデルを用いていることになる。

5. 符号化実験及び考察

電子計算機上に前章までに述べた符号化器を作成し、符号化シミュレーションを行った。表1に資料として用いたデータの大きさと種類を示している。画像データは、SIDBAとISOの標準画像から選んだ様々の画像である。girl~facsが256×256、girl2以降が512×512の白黒画像である。また、輝度は8ビットで表されている。符号化性能の評価は、次式の圧縮率を用いて行う。

$$\text{圧縮率} = \frac{S-C}{S} \quad (6)$$

ここで、 S は原画像のデータ量、 C は圧縮後のデータ量である。すなわち、符号化によるデータの減少の割合を示しているので、大きいほど良好な圧縮が行われていることを示している。

図4は本論文で提案しているビット単位の符号化とバイト単位(サンプル単位)の符号化の圧縮率を比較したものである。なお、符号化はモデル1である。状態数([2]のように示している)が少ない場合には、それ程大きな違いは見られないが、状態数が増えてくるとビット単位符号化の方が良好な圧縮率が得られている。これは、バイト単位符号化では256値のうちの一つという形で確率分布が取られるが、ビット単位符号化では符号化の木から分かるように、上位ビットから2つのうちのどちらかという形で確率分布が取られるので、特に状態数が多いときに確率分布の曖昧性が少なくなっているからと考えられる。

図5にモデル1における状態のビット数と圧縮率の関係を示している。最大の圧縮率を示す状態のビット数があることに気がつく。これは、状態のビット数があまりに小さいと、モデル自身が的確に表現できなくて圧縮率が低下してしまい、あまり大きいと今度はモデルを記述するパラメータがきちんと表現できなくなってしまうことにある。すなわち、モデル自身の規模があまりにも大きくなると、表現すべきパラメータ数が非常に多くなり、データ数が追付かなくなってしまうためである。モデル1の場合は、直前の画素の上位4ビットから6ビットを状態とみて符号化を行うのが最もよい。

表2にモデル2、3の圧縮率を示している。表2において、[5,2]とあるのは、左直前お

よび真上の画素の参照するビット数を並べて示している。モデル3においても、直前およびその前の参照画素のビット数を並べて示している。

モデル2においては、一部の画像を除いて、左直前の画素のビットを多く参照するモデルの方が高い圧縮率が得られている。これは、画像の入力装置にも依存すると思われるが、ほとんどの画像で横方向の相関が多少大きくなる傾向があるためと考えられる。

モデル1, 2, 3を比較すると、画像によって最適なモデル及びパラメータが異なっている。これは、画像の縦横の相関係数の違い及びその大きさの違い等に影響される為と考えられる。

表1に、UNIXの標準圧縮方式として提供されているcompress(LZW符号化法⁶⁾)コマンドの圧縮率が示されている。LZW法は、文字の連続パターンを符号化する方法であるため、相関の高いデータに対しては非常に強力のはずである。本論文で提案した方法は、数

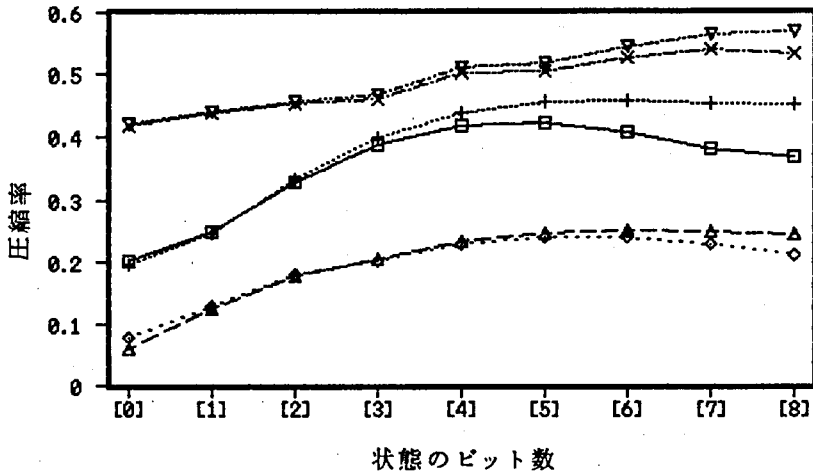


図4 ビット単位符号化とバイト単位符号化の比較

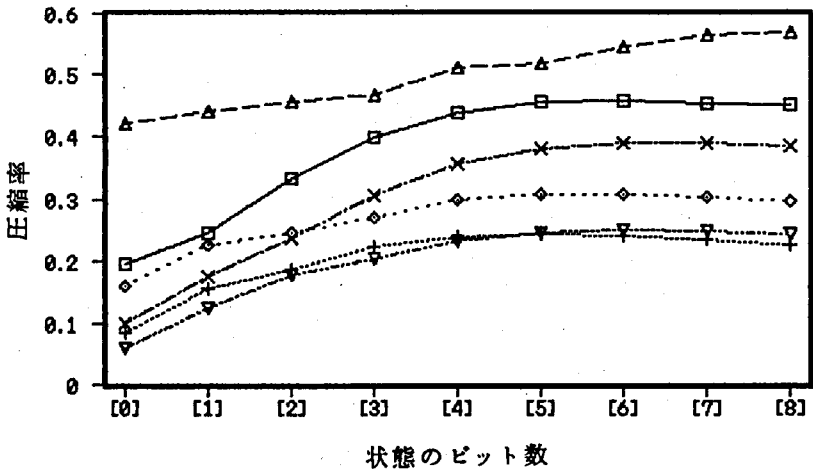


図5 状態のビット数と圧縮率の関係

表2 各モデルに対する圧縮率

モデル2

	[4,2]	[4,4]	[4,6]	[5,2]	[5,4]	[5,6]	[6,2]	[6,4]	[6,6]
girl	43.68%	43.50%	42.02%	45.20%	44.27%	41.64%	45.18%	43.64%	40.06%
couple	42.34%	42.00%	42.55%	45.05%	44.09%	43.07%	46.37%	45.06%	42.63%
aerial	23.80%	23.16%	21.35%	24.07%	23.06%	20.78%	23.62%	22.24%	19.51%
moon	30.70%	31.41%	30.72%	31.28%	31.58%	30.38%	31.12%	31.10%	29.36%
chest	43.93%	41.32%	36.70%	42.82%	39.43%	34.08%	41.22%	37.11%	31.22%
fac	51.28%	52.31%	53.86%	51.76%	52.74%	54.02%	54.37%	55.05%	55.01%
girl2	36.16%	37.60%	38.52%	38.38%	39.06%	39.09%	39.02%	39.39%	38.94%
area	29.28%	29.59%	28.88%	29.95%	29.94%	28.71%	29.82%	29.54%	27.94%
barbara	23.83%	25.12%	26.15%	25.22%	26.29%	26.47%	25.53%	26.41%	26.28%
mandrill	21.48%	21.67%	20.71%	21.69%	21.62%	20.31%	21.43%	21.09%	19.47%

モデル3

	[4,2]	[4,4]	[4,6]	[5,2]	[5,4]	[5,6]	[6,2]	[6,4]	[6,6]
girl	43.60%	43.68%	42.99%	45.15%	44.60%	42.64%	45.22%	44.20%	41.33%
couple	43.05%	43.02%	44.44%	46.01%	45.36%	44.85%	47.49%	46.57%	44.53%
aerial	22.13%	21.68%	20.34%	22.39%	21.63%	19.89%	21.97%	20.98%	18.91%
moon	31.64%	32.11%	31.54%	32.44%	32.49%	31.39%	32.39%	32.16%	30.50%
chest	43.67%	41.43%	37.15%	42.70%	39.70%	34.68%	41.28%	37.63%	31.94%
fac	51.17%	52.15%	54.17%	51.58%	52.50%	54.23%	54.11%	54.78%	54.96%
girl2	36.09%	37.44%	38.29%	38.34%	38.95%	38.92%	38.97%	39.30%	38.81%
area	28.68%	28.85%	28.13%	29.29%	29.16%	27.93%	29.13%	28.76%	27.17%
barbara	26.63%	27.60%	28.35%	28.28%	29.00%	28.87%	28.77%	29.28%	28.76%
mandrill	19.35%	19.54%	18.63%	19.51%	19.45%	18.26%	19.24%	18.94%	17.50%

サンプルのマルコフ性を利用しているにすぎないが、LZW法と比較して10から30%高い圧縮率が得られている。この理由として、本方式はビット単位で圧縮を行うため、バイト単位の情報の内部（ビットの連なり）に存在する冗長性をもよく取り除いていると考えられる。すなわち、バイト単位符号化では、文字は全く別々の記号として扱われ、文字の連なり、確率の偏りの度合いが圧縮率を決めるが、ビット単位符号化では、それらに加えてビットの連なりにおける相関も圧縮率を高める方向に作用している。

6. むすび

画像情報を2値算術符号で効率よく、容易に符号化する方法を示した。まず、一般論とし

て多値情報源をビットのベクトルで表現し、それと同時に多値出力の確率分布を条件付き確率で表現することによって、2 値算術符号での符号化が可能となることを示した。そして、画像を2, 3 のマルコフ情報源でモデル化し、これを用いて実際の画像データを符号化した。その結果、ビット単位符号化法はバイト単位の符号化法と比べ、特にモデルが複雑なところで大きな改善効果があることが明らかとなった。また、高い圧縮率が得られることで知られている UNIX の compress コマンド (LZW 法) と比較しても、本方式は約10から30% 高い圧縮率を示すことが明らかとなった。

今後の課題としては、画像によって圧縮率の最も良い符号化モデルが異なることから、画素単位で自動的にモデルを選択する機構を取り入れ、何時でも最適な圧縮率が達成できるようにする事が考えられる⁽⁷⁾。

参 考 文 献

- (1) 笠原, 田崎, 小倉: 情報理論, 昭晃堂 (昭60-10).
- (2) J. Rissanen and G. G. Langdon, Jr: "Arithmetic Coding," IBM J. Res. DEVELOP, vol.23, pp. 149-162 (March 1979).
- (3) J. Rissanen and G. G. Langdon, Jr: "Universal Modeling and Coding," IEEE Trans. Inform. Theory, vol. IT-27, 1, pp.12-23 (Jan 1981).
- (4) 半田, 田中: "ハフマン符号の符号帳の記述に関する一考察", 電子情報通信学会論文誌 A, Vol. J70-A, No.9, pp.1501-1503 (1987-10)
- (5) A. Papoulis: *Probability, Randon Variables, and Stochastic Processes*, McGraw Hill (1984).
- (6) T. A. Welch: "A Technique for High-Performace Data Compression," IEEE Computer, pp. 8-19 (June 1984).
- (7) 上田, 田中, 半田: "算術符号を用いた適応的データ圧縮の一方式", 電子情報通信学会論文誌 A, Vol. J74-A, No.9, pp.1455-1458 (1991-9)