

算術符号とその中間調画像符号化への応用

Arithmetic Coding and Its Application for Multi-level Image Coding

加藤 茂 夫*

Shigeo KATO

算術符号は、通常のランレングス符号とは異なって、符号語を算術演算により逐次的に構成する非ブロック符号である。この符号は、情報源の拡大を行うことなくシンボルあたりの平均符号長がエントロピーレートに近づく高効率な符号である。本稿では、この算術符号について解説を行うとともに中間調ファクシミリ画像のマルコフモデル符号化に適用する手法について述べる。

1. はじめに

高度情報化社会の進展にともない、ファクシミリはもとより、いよいよ本格的なサービスが始まったキャプテンシステムや、また、今後の発展が期待されているビデオコンファレンス、テレビ電話など、画像情報が人間社会に占める役割りは非常に大きなものとなりつつある。しかし、「百聞は一見にしかず」の故事を持ち出すまでもなく画像情報はきわめて膨大な情報量をもっているため、蓄積コストおよび伝送コストの削減のためにはデータ圧縮が不可欠になる。たとえば、ファクシミリ画像の情報量について考えると、A 4 版の原稿 1 枚は、白黒 2 値の場合でも 0.5 Mbyte ものデータ量をもっているため、効率的な伝送を行うためには何らかの手段で冗長度を抑圧して符号化することが必要である。

2 値ファクシミリ画像については、すでに、国際電信電話諮問委員会 (CCITT) が、その冗長度抑圧符号化方式として、Modified Huffman (MH) 符号および Modified READ (MR) 符号を国際標準として規格化しており、これらの方式は、ファクシミリ実用機に用いられている。この MR 方式を用いれば、CCITT の 2 値ファクシミリ用テスト画像のデータ量を $1/6 \sim 1/16$ に圧縮することが可能である。

しかし、最近になって、J. Rissanen らにより発表された Arithmetic Coding (算術符号)^{1)~3)} は、これら MH, MR 等の国際標準化方式よりもさらに効率のよい符号化法として注目されている。

この符号は、Elias' Code⁴⁾ の概念を、実現可能な形にしたものであり、2 値画像のマルコフモデル符号化に適用することによりマルコフモデル符号化において通常不可欠であった情報源の縮退を行うことなく高効率の符号化が実現可能になった。たとえば、前述 CCITT のテスト用

画像に対してこの符号を用いて 10 画素参照マルコフモデル符号化を行った場合、MR 方式に比べ圧縮率を 20~30% 上げ得ることが報告されている。この方式は、2 値ファクシミリ符号化の標準方式が制定されたあと発表されたため、標準化方式の候補とはなり得なかったが、今後、ファクシミリ画像に限らず、一般の画像通信サービスが、2 値画像から中間調画像へ、また、白黒画像からカラー画像へと高品質化の途をたどることは確実であると考えられるため、この算術符号の原理をこれらの多値画像に対しても拡張して適用することができれば、高効率なデータ圧縮効果が期待できるものと考えられる。

そこで本稿では、まず、算術符号のもととなった Elias' Code の原理について解説し、次に、算術符号を多値画像のマルコフモデル符号化に適用する手法について述べ、シミュレーションにより本手法が多値ファクシミリ画像のデータ圧縮法としても有効であることを示す。

2. Elias' Code

まえがきでも述べたように、算術符号は Elias' Code の原理を用いて、それを実現可能にしたものと考えられることができる。Elias' Code は、情報源からの出力シンボル系列を、次に述べるような方法で、 $(0, 1)$ の範囲の数直線上の 1 点で表し、その点の値を符号系列として伝送するものである。ここで、記号 (a, b) は a 以上 b 未満の範囲であることを示す。

たとえば、2 値情報源 $\{a, b\}$ を考え、その出現確率をおのおの、 $P_a=0.7, P_b=0.3$ であるとする。Elias' Code では、まず、数直線 $(0, 1)$ を、シンボルの出現確率に応じて分割する。すなわち、この例では、 $(0, 0.7)$ と $(0.7, 1.0)$ に分割することになる。情報源からの最初の出力シンボル (第 1 シンボルとする) が、たとえば、 a の場合、シンボルシーケンス $s_1 = \{a\}$ に対する符号語 $c(s_1)$ を図 1 に示されるように、数直線 $(0, 0.7)$ 上の任意の 1 点で表すこ

* 東京大学生産技術研究所 第 3 部

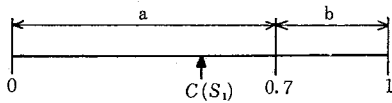


図 1 第 1 シンボルの符号化

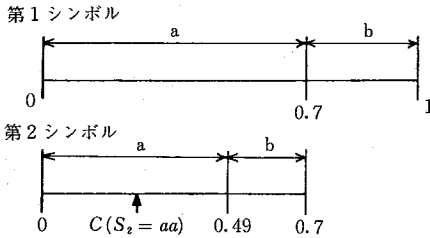


図 2 第 2 シンボルの符号化

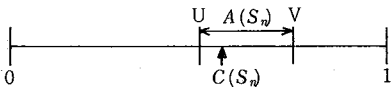


図 3 符号 $C(S_n)$ とスパン $A(S_n)$ との関係

とにする。

たとえば、 $C(S_1)=0.5$ として、これを 2 進小数で表すことにすれば、

$$C(S_1=a)=.1 \tag{1}$$

となる。また、第 1 シンボルが b であれば、 $C(S_1)$ は、数直線 $(0.7, 1)$ 上の点で表されるから、たとえば $C(S_1)=0.75$ とすれば、

$$C(S_1=b)=.11 \tag{2}$$

となる。

さて、第 1 シンボルを a としたとき、第 2 シンボルについて考える。

第 2 シンボルについては、第 1 シンボルが a であるときの符号系列 $C(S_1)$ の取り得る数値範囲 $[0, 0.7)$ を、シンボルの出現確率に応じて図 2 に示すように $[0, 0.7 \times 0.7=0.49)$ 、 $[0.49, 0.7)$ に分割し、 $C(S_1)$ を決定したときと同様にして $C(S_2)$ を決定する。

たとえば、第 1 シンボル、第 2 シンボルともに、 a のとき、 $C(S_2=aa)$ を表す数直線上の点は、第 3 シンボルを考えなければ、 $(0, 0.49)$ 内のどこでもよい。したがって、もっとも少ないビット数で表現できる点をえらべば、効率的であろう。この場合 $(.01)_2=0.25$ であるから

$$C(S_2=aa)=.01 \tag{3}$$

とすればよい。

いま、図 3 に示すように n 個のシンボルシーケンス S_n に対する符号 $C(S_n)$ が、数直線上で取り得る範囲 $[U, V)$ の大きさを $A(S_n)$ とする。すなわち、上記の例では、 $S_1=\{aa\}$ に対して、 $U=0, V=0.49, A(S_2=aa)=0.49$ 、また、 $S_2=\{ab\}$ に対しては、 $U=0.49, V=0.7, A(S_2=ab)=0.21$ となる。この $A(S_n)$ より小さい値を表すこ

とのできるもっとも少ないビット数を B_n, \min とすると、 B_n, \min で表現できる数値群のうち、最低 1 個は必ず区間 $[U, V)$ の中に入る。したがってこの値を $C(S_n)$ とすればよい。

一般に、 a の出現確率を P_a とし、 b の出現確率を P_b とした場合に、任意のシンボル系列

$$S_n = \{abaaaa \dots\}$$

の符号化ビット数を考える。

S_n 中の a の出現個数を n_a 個、 b の出現個数を n_b 個とすれば、

$$\begin{aligned} A(S_n) &= P_a \times P_b \times P_a \times \dots \\ &= P_a^{n_a} \times P_b^{n_b} \end{aligned} \tag{4}$$

$A(S_n)$ を表すことのできる最も少ないビット数 B_n, \min は、記号 $\lceil x \rceil$ を x 以上の最小の整数とすれば、

$$B_n, \min = \lceil -\log_2 A(S_n) \rceil = \lceil -n_a \log_2 P_a - n_b \log_2 P_b \rceil \tag{5}$$

無限長系列では、 $\frac{n_a}{n} = P_a, \frac{n_b}{n} = P_b$ であり、この情報源のエントロピを H とすれば、

$$B_n, \min = \lceil nH \rceil \leq nH + 1 \tag{6}$$

したがって、1 シンボルあたりの平均符号長 B は

$$B = \frac{B_n, \min}{n} = H + \frac{1}{n} \tag{7}$$

となり、この符号は情報源エントロピに極めて近い値で符号化できる効率のよい符号であることがわかる。

しかし、この符号は、すべての出力シンボル系列を知ってからでないと符号が決定されないこと、また、効率を上げるために符号化単位 n を増加させると高い乗算精度が要求されること等から、実用には供しなかった。

この Elias' Code の欠点をなくし、実用化したものが IBM の J. Rissanen らが提案した算術符号である。

算術符号は、簡単に言えばシンボルの出現確率を 2 のべき乗で近似することにより乗算をビットシフト演算に変換し、また、符号は、数直線上でその符号がとり得る範囲の最小値で表すことにしたものである。

次節以降では、算術符号を多値画像の符号化に適用する方法について述べ、あわせてシミュレーション実験による符号化効率等について述べる。

3. 多値算術符号⁵⁾⁻⁷⁾

3.1 符号化の原理

まず、多値算術符号の原理について述べる。算術符号においては、ある任意の出力シンボル系列に対する符号系列は、最左端に小数点が置かれた 2 進小数とみなされる。すなわち、出力シンボル系列のうち、第 i 番目のシンボルを j_i とし、第 n 個目までの出力シンボル系列を $S_n = \{j_1, j_2, \dots, j_i, \dots, j_n\}$ とすれば、 S_n に対する出力符号系列 $C(S_n)$ は、数直線 $(0, 1)$ 上の 1 点の値で表され

る。いま, $0, 1, 2, \dots, (k-1)$ のシンボルからなる k 値情報源を考え, その生起確率を各々 $P(0), P(1), \dots, P(k-1)$ で表す。算術符号化の対象となるシンボル系列は必ずしも予測変換信号である必要はないが, 説明の都合上ここでは予測変換されたシンボル系列を取り扱うことにする。すなわち出現確率の大きいほうから順にシンボルが $0, 1, 2, \dots, (k-1)$ になっているものとする。

まず, 符号系列のとり得る範囲 $[0, 1)$ の大きさを小数点以下 q ビットの小数 $A(\text{NULL})$ で次式のように近似する。

$$A(\text{NULL}) = \underbrace{11 \cdots 1}_{q \text{ ビット}} \quad (8)$$

次に, $A(\text{NULL})$ をシンボルの生起確率に応じて次式のように $A(0), A(1), \dots, A(k-1)$ なる k 個の範囲に分割する。

$$A(j_i) = A(\text{NULL}) \times P(j_i) \quad (9)$$

算術符号においては, この $A(\cdot)$ を Augend と呼んでいる。図 4 にこの分割の様子を示す。

さて, 最初に符号化すべきシンボル j_1 に対する符号 $C(S_1)$ を次式で決定する。

$$\left. \begin{aligned} C(S_1) &= \sum_{l=0}^{j_1-1} A(l) \quad (j_1 \neq 0 \text{ のとき}) \\ C(S_1) &= 0 \quad (j_1 = 0 \text{ のとき}) \end{aligned} \right\} \quad (10)$$

一般に第 n 番目までのシンボル系列 S_n に対する符号系列 $C(S_n)$ はその直前までのシンボル系列 S_{n-1} に対する Augend $A(S_{n-1})$ を用いて決定される。まず, 次に示す繰り返し演算により $A(S_{n-1})$ より $A(S_n)$ を求める。

$$A(S_n) = A(S_{n-1} \cdot j_n) = A(S_{n-1}) \times P(j_n) \quad (11)$$

すなわち $A(S_n)$ は $A(S_{n-1})$ をシンボルの出現確率に応じて分割したものになる。このようにして定めた $A(S_n)$ を用いて, 系列 S_n に対する符号を以下のように定める。すなわち, シンボル系列 S_{n-1} に対する符号系列 $C(S_{n-1})$ はすでに決定されているものとすれば, S_n に対する符号系列 $C(S_n)$ を

$$\left. \begin{aligned} C(S_n) &= C(S_{n-1}) + \sum_{l=0}^{j_n-1} A(S_{n-1} \cdot l) \quad (j_n \neq 0 \text{ のとき}) \\ C(S_n) &= C(S_{n-1}) \quad (j_n = 0 \text{ のとき}) \end{aligned} \right\} \quad (12)$$

によって決定する。

一方, 復号は Augend をシンボル値の小さいものに対応するものから順次加えていき, 受信符号系列 $C(S_n)$ を超える直前のシンボル値をそのときの復号値とすることにより行われる。いま, 受信符号系列 $C(S_n)$ を $C_i(S_n)$ と置き換える。第 1 番目のシンボル j_1 は

$$C_i(S_n) > \sum_{l=0}^i A(l) \quad (13)$$

を満たす最大の i を i_{\max} として

$$j_1 = i_{\max} \quad (14)$$

と復号される。また, 第 2 番目のシンボル j_2 の復号のために残差符号系列 $C_2(S_n)$ を次式により求める。

$$C_2(S_n) = C_1(S_n) - \sum_{l=0}^{j_1} A(l) \quad (15)$$

一般に, 第 m 番目のシンボル j_m は, すでに決定した残差符号系列

$$C_m(S_n) = C_{m-1}(S_n) - \sum_{l=0}^{j_{m-1}} A(S_{m-2} \cdot l) \quad (16)$$

を用いて,

$$C_m(S_n) > \sum_{l=0}^i A(S_{m-1} \cdot l) \quad (17)$$

を満たす最大の i の値に復号される。

3.2 符号化の実現方法および平均符号長

ところで, 式(9)および(11)の演算は 2 進小数乗算であるので計算コストが場大する。そこで, ここでは, Langdon らが二値の場合で行った³⁾ように各シンボルの出現確率を 2 の負のべき乗で近似することにする。こうすることにより式(9)および(11)の乗算はシフト操作のみで済ますことができる。すなわち, Q_i を整数として, $P(l)$ を

$$P(l) \doteq P'(l) = 2^{-Q_l} \quad (l=1, 2, \dots, k-1) \quad (18)$$

で置き換えることにすれば式(9)および(11)は次のように置き換えられる。

$$\left. \begin{aligned} A(j_i) &= A(\text{NULL}) \times 2^{-Q_{j_i}} \\ A(S_n) &= A(S_{n-1} \cdot j_n) = A(S_{n-1}) \times 2^{-Q_{j_n}} \end{aligned} \right\} \quad (j_i=1, 2, \dots, k-1) \quad (19)$$

ここで, Q_i を skew value と呼ぶ。また, $A(0)$ および $A(S_{n-1} \cdot 0)$ に関しては

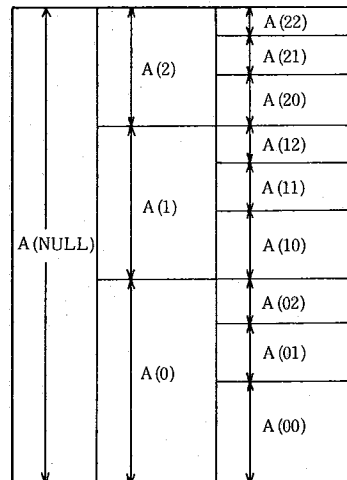


図 4 多値算術符号における Augend の分割 ($k=3$)

$$\left. \begin{aligned} A(0) &= \langle A(\text{NULL}) - \sum_{l=1}^{k-1} A(l) \rangle_q \\ A(S_{n-1} \cdot 0) &= \langle A(S_{n-1}) - \sum_{l=1}^{k-1} A(S_{n-1} \cdot l) \rangle_q \end{aligned} \right\} \quad (20)$$

とすることにより、第1予測変換シンボルの出現確率を2の負のべき乗で近似することから生じる誤差を減少させることができる。なお、記号 $\langle X \rangle_q$ は2進小数 X を有効ビット数 q ビットで打ち切ることを示す。この操作は次式に示すように、 $l \neq 0$ になる l に対して $A(l)$ および $A(S_{n-1} \cdot l)$ を有効ビット数 $(q - Q_l)$ ビットで打ち切り、 $A(0)$ および $A(S_{n-1} \cdot 0)$ を打ち切りなしにすることと等価である。

$$\left. \begin{aligned} A(l) &= \langle A(\text{NULL}) \times 2^{-Q_l} \rangle_{q-Q_l} \\ A(0) &= A(\text{NULL}) - \sum_{l=1}^{k-1} A(l) \end{aligned} \right\} \quad (21)$$

$$\left. \begin{aligned} A(S_{n-1} \cdot l) &= \langle A(S_{n-1}) \times 2^{-Q_l} \rangle_{q-Q_l} \\ A(S_{n-1} \cdot 0) &= A(S_{n-1}) - \sum_{l=1}^{k-1} A(S_{n-1} \cdot l) \end{aligned} \right\} \quad (22)$$

Q_l の値としては、最も出現確率の低いシンボルの近似精度により決められるが、 $2^{-15} \approx 3.1 \times 10^{-5}$ であることより最大 $Q_l = Q_{\max} = 15$ 程度で十分である。したがって $A(\text{NULL})$ として $q = 16$ ビットの精度があればよい。

また、このような多値算術符号化における1シンボルあたり平均符号長 B は、近似的に

$$B = \frac{1}{\sum_{l=1}^{k-1} \{P(l) \times Q_l\} - P(0)} \times \log_2 \left(1 - \sum_{l=1}^{k-1} 2^{-Q_l} \right) \quad (23)$$

で表される。(付録参照)

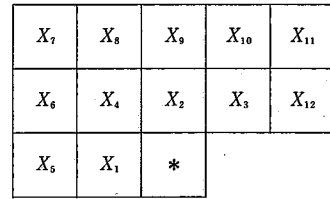
ところで、実際の符号化に当たっては、符号を作成する時点で生じるけた上がりによる符号反転の波及を防ぐ必要がある。そこで、出力符号系列中に1が連続してあるビット数以上続いた場合は、次に強制的に0をそう入する bit stuffing を行う。また、Augend が分割されるにつれ先行する0が増加するため、先行する0の数だけ $A(S_n)$ および $C(S_n)$ を左へシフトする shifting などのテクニックを用いる。いま、1例として、 $k=3, Q_1=2, Q_2=3, q=4$ とし、シンボル系列(012001)を算術符号化する過程を表1に示す。

3.3 多値マルコフモデル符号化への適用

一般に、画像情報源はすでに出力された画素信号値によって状態が定まるマルコフ情報源と見なすことができ、その条件付きエントロピ(マルコフモデルエントロピ)は、画像信号に対する圧縮限界を示すといわれている。マルコフモデルに基づく符号化では、図5に示すように、伝送済みの画素の中から着目画素と強い相関があると思われる画素 X_1, X_2, \dots 等を参照画素として選び、参照画素群のとりパターンにより着目画素を状態分けし、各状態ごとに最適な符号を用いることにより条件付きエン

表1 シンボル系列012001に対する算術符号化の一例

j_n	$A(S_{n-1})$	$A(S_{n-1} \cdot 0)$	$A(S_{n-1} \cdot 1)$	$A(S_{n-1} \cdot 2)$	$C(S_{n-1} \cdot j_n)$
0	.1111	.1011	.0011	.0001	.0000
1	.1011	.1000	.0010	.0001	.1000
2	.0010				
shift	.1000	.0101	.0010	.0001	10.0111
0	.0001				
shift	.1000	.0101	.0010	.0001	10011.1
0	.0101				
shift	.1010	.0111	.0010	.0001	100111.
1	.0111				
shift	.1110	.1010	.0011	.0001	1001110. 1010



* ----- 着目画素

図5 参照画素位置

トロピを達成目標としている。各状態で用いる符号を、従来のようにブロック符号で構成する場合は、符号効率を上げるために、各状態ごとに情報源を拡大して符号化を行う必要がある。したがって参照画素パターンのある状態に属するシンボルを符号化する際、次にいくつかの同一状態が出現するまでそのシンボルに対する符号系列は出力され得ず、シンボルの出現順に瞬時に符号化することは不可能であった。一方、算術符号をマルコフモデル符号化に適用する場合は、符号化すべきシンボルが出現した時点でその状態におけるシンボルの出現確率を用いて Augend を作成することができるので、出現シンボル順に符号を出力できる。すなわち $A(S_{n-1} \cdot l)$ として、シンボル l の参照画素パターン r におけるシンボル l の条件付き確率 $P(l/r)$ で $A(S_{n-1})$ を分割することにすればよく、式(9)および(11)における $P(l)$ は $P(l/r)$ となり、これを

$$P'(l/r) = 2^{-Q_{lr}} \quad (24)$$

で近似することにすれば、1シンボルあたりの平均符号長 B_M は、

$$B_M = \sum_{\text{all } r} P(r) \left\{ \sum_{l=1}^{k-1} P(l/r) \times Q_{lr} - P(0/r) \times \log_2 \left(1 - \sum_{l=1}^{k-1} 2^{-Q_{lr}} \right) \right\} \quad (25)$$

となる。

- なお、本方式実現のための必要メモリ容量としては、
- (i) Q_{lr} をストアしておくためのメモリ
 - (ii) 参照画素パターンを求めるためのラインメモリ

表 2 Q_{ur} 用メモリ容量

レベル数および参照画素数	記憶容量
16値 2 画素	2 kbyte
16値 3 画素	32
16値 4 画素	524

表 3 多値マルコフ算術符号による平均符号長および符号効率

画像	条件付エントロピ	最適 Q_{ur}		共通 Q_{ur}	
		平均符号長	符号効率	平均符号長	符号効率
Mono Girl	bit/pel 1.008	bit/pel 1.021	98.7%	bit/pel 1.078	93.5%
Mono Couple	0.816	0.828	98.6	0.866	94.2
Moon Surface	1.215	1.232	98.7	1.315	92.4
Aerial Scene	1.685	1.712	98.4	1.801	93.6
Girl	1.155	1.172	98.5	1.235	93.5
Lady	0.978	0.992	98.6	1.090	89.7

(iii) 予測変換テーブル

などである。(i) の Q_{ur} メモリとしての必要容量 F は、参照画素数 n 、レベル数 k 、 Q_{ur} の最大値 Q_{max} としたとき

$$F = k^{n+1} \times [\log_2(Q_{max} + 1)] \text{ ビット} \quad (26)$$

となる。ただし、ここで $[x]$ は x 以上の最小の整数を表す。

表 2 に F の具体的な値を示す。表 2 から分かるように、 F の値は 16 値 3 画素参照でも 32 Kbyte 程度であり、この程度のメモリ規模で情報源の縮退なくマルコフ符号化が実現可能である。これは、従来のマルコフ符号化においては参照画素パタンの各状態ごとに符号テーブルが必要であったのに対し算術符号化においては skew value Q_{ur} が符号テーブルに対応しているものと考えられ、 $Q_{max} = 15$ とすれば 1 状態あたり $16 \times 4 = 64$ bit で済むことによる。

4. シミュレーション実験

SIDBA 標準画像 4 枚を含む 6 枚の画像 (256×256 画素) をシミュレーション用原画として算術符号化のシミュレーション実験を行った。MONO-GIRL~AERIAL-SCENE は SIDBA の標準画像、GIRL は SIDBA の 512×512 画素の標準画像を 256×256 にリサンプリングしたものであり、LADY は非 SIDBA 画像である。いずれも 8 bit/画素のものを上位 4 bit をとって 16 レベル画像とした。図 6 にシミュレーションに用いた画像を示す。また、マルコフ符号化における参照画素位置は図 5 に示す X_1, X_2, X_3 の 3 画素を用いている。

まず、表 3 に 16 値原画像の条件付きエントロピの値、

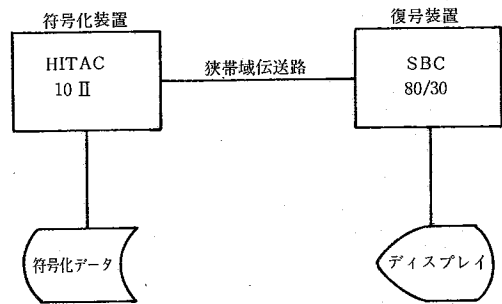


図 7 実験システム

および多値マルコフ算術符号化を行った場合の 1 画素あたり平均符号長および符号効率を示す。skew value Q_{ur} の値としては各画像について別々に統計量を求め、それをもとに各画像について最適な Q_{ur} を用いた場合、および 6 枚の画像を縦に接続したものから求めたものを各画像とも共通に用いた場合の 2 通りについてシミュレーションを行った。表 3 から明らかなように各画像に最適な Q_{ur} を用いた場合約 98%、共通な Q_{ur} を用いた場合でも 90~94% の符号効率を得ている。なお、最適な Q_{ur} を用いる場合、実際上は何らかの方法であらかじめ Q_{ur} を伝送しておく必要があるが、シミュレーションではこのオーバーヘッド分は考慮に入れていない。

5. 実時間伝送復号実験

本方式の実現可能性を検討するため、汎用マイクロコンピュータを用いて伝送復号表示実験を行った。図 7 に実験システムの構成図を示す。

符号化されたデータは、ミニコンピュータ HITAC-10 II に接続されたディスクに格納されており、伝送レート可変の狭帯域伝送路を通じて 8085 マイクロプロセッサ (クロック 2.76 MHz) を主体とする復号表示装置に伝送される。なお、本符号化法では、16 値 3 画素参照時で Q_{ur} メモリとして約 32 Kbyte 程度で済むわけであるが、本復号装置のメモリの制約上、文献 5 で提案した多モード統合法 ($M = 2$) を用いて行っている。

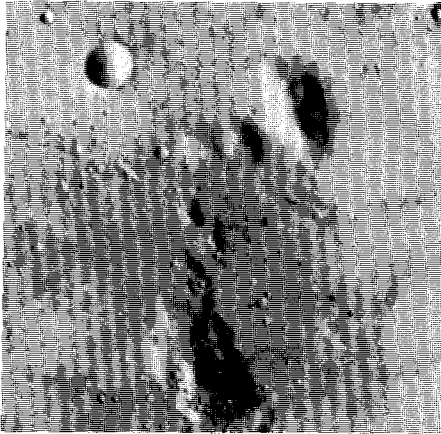
マイクロプロセッサは復号処理中、データを受信すると復号処理を中断し受信割り込み処理を行った後、再び中断された復号処理に戻る。復号プログラムは受信処理ルーチンを含め約 1 kbyte 程度である。ほかに Q_{ur} を蓄積するためのメモリが 1.7 kbyte ($M = 2$ のとき) と、参照画素の値を求めるための 2 ライン分のラインメモリなどが必要となる。なお、実験では復号レベル値をディスプレイする際のアドレッシングを 1 画素ごとに行っているため余分な時間を費やしており、遅滞なく復号表示できるのは伝送レート 1200 baud 程度までで、2400 baud 以上での伝送は現在のところ不可能ではあるが、画像メモリアドレッシングの高速化、高速マイクロプロ



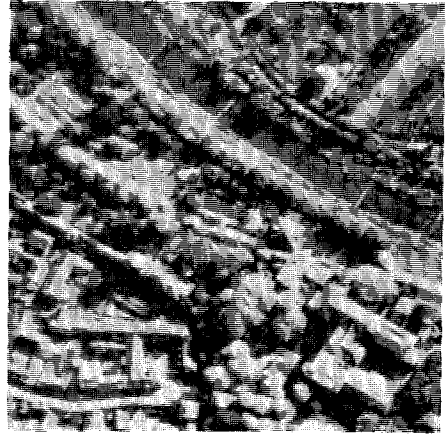
(a) Mono-Girl



(b) Mono-Couple



(c) Moon-Surface



(d) Aerial Scene



(e) Girl



(f) Lady

図 6 シミュレーション原画

セッサの使用, 複数個のマイクロプロセッサによる並列処理の利用等により 2400 baud 以上での伝送も十分可能であると思われる。

6. あとがき

以上, 無記憶情報源に対する高効率符号化法として注目されている算術符号について, その基本原理である Elias' Code より説明し, あわせて中間調画像のマルコフモデル符号化に適用する手法について述べた。シミュレーションを行った結果, 16 値画像については, 0.9~1.8 ビット/画素で符号化が実現可能であることを示した。

また, 汎用マイクロコンピュータを用いて符号化および復号装置を試作し, 実時間伝送実験を行った結果, 本符号化方式が容易に実現可能であることを確認した。

(1984 年 12 月 12 日受理)

付 録

多値算術符号における平均符号長について

多値算術符号における平均符号長は次のようにして求められる。まず, 2 進小数 x の小数点以下のビット数を $B(x)$ で表すことにすれば, $l \neq 0$ の場合に

$$B(A(S_{i-1} \cdot l)) = B(A(S_{i-1})) + Q_l \tag{A-1}$$

となる。さらに, Q_l として非整数をとることを許すことにすれば $l=0$ の場合も (A-1) 式に含めることができる。

次に, 出力符号系列 $C(S_i)$ を次のように書き直すことにする。

$$C(S_i) = \sum_{l=0}^{k-1} \{A(\text{NULL}) \times 2^{-Q_l}\} + \sum_{l=0}^{k-1} \{A(S_1) \times 2^{-Q_l}\} + \dots + \sum_{l=0}^{k-1} \{A(S_{i-1}) \times 2^{-Q_l}\} \tag{A-2}$$

ところで, $C(S_i)$ のビット数を決定するのは右辺各項のうちで最も小さな値の Augend であるので

$$B(C(S_i)) = B(A(S_{i-1}) \times 2^{-Q_{i-1}}) \tag{A-3}$$

となり, 漸化式 (A-1) を用いれば

$$B(C(S_i)) = B(A(\text{NULL})) + Q_{j1} + Q_{j2} + \dots + Q_{j(i-1)} + Q_{ji-1} \tag{A-4}$$

となる。

(A-4) 式において同一の Q_l の出現個数を N_l で表し, 系列全体のシンボル数を N とすれば,

$$N = \sum_{l=0}^{k-1} N_l \tag{A-5}$$

また (A-4) 式は

$$B(C(S_i)) = q + \sum_{l=0}^{k-1} (N_l \times Q_l) + Q_{ji-1} \tag{A-6}$$

となる。無限長系列で考えれば, 1 シンボルあたりの平均ビット数 B は

$$B = \lim_{N \rightarrow \infty} \frac{1}{N} B(C(S_i)) = \lim_{N \rightarrow \infty} \sum_{l=0}^{k-1} \left(\frac{N_l}{N} \times Q_l \right) \tag{A-7}$$

$$\text{ところで, } \lim_{N \rightarrow \infty} (N_l/N) = P(l) \tag{A-8}$$

であるから, (A-7) 式は

$$B = \sum_{l=1}^{k-1} \{P(l) \times Q_l\} + P(0) \times Q_0 \tag{A-9}$$

となる。

残された問題は Q_0 を求めることである。このためにシンボル 0 のみからなる長さ m の系列を S'_m とおき

$$B(A(S \cdot S'_m)) = B(A(S)) + 1 \tag{A-10}$$

となる最小の m を求めることにする。いま, $A(S) = 1$ とすると, (21), (22) 式より

$$A(S \cdot S'_m) = \left(1 - \sum_{l=1}^{k-1} 2^{-Q_l} \right)^m \tag{A-11}$$

となり

$$A(S \cdot S'_m) \leq \frac{1}{2} \tag{A-12}$$

を満たす最小の m は

$$m = \frac{1}{\log_2 \left(1 - \sum_{l=1}^{k-1} 2^{-Q_l} \right)} \tag{A-13}$$

よって,

$$Q_0 = \frac{1}{m} = -\log_2 \left(1 - \sum_{l=1}^{k-1} 2^{-Q_l} \right) \tag{A-14}$$

したがって 1 シンボルあたりの平均符号長 B は

$$B = \sum_{l=1}^{k-1} \{P(l) \times Q_l\} - P(0) \times \log_2 \left(1 - \sum_{l=1}^{k-1} 2^{-Q_l} \right) \tag{A-15}$$

となる。

参 考 文 献

- 1) J. Rissanen and G. G. Langdon, Jr.: "Arithmetic Coding", IBM J. Res. Develop. 23, 2, 149~162, 1979. 3.
- 2) J. J. Rissanen: "Generalized Kraft Inequality and Arithmetic Coding", IBM J. Res. Develop. 20, 198~203, 1976. 5.
- 3) G. G. Langdon, Jr. and J. Rissanen: "Compression of Black-White Images with Arithmetic Coding", IEEE Trans. on. commun. 29, 6, 858~867, 1981. 6.
- 4) N. Abramson: "Information Theory and Coding", 1963 McGraw-Hill Book Co., Inc.
- 5) 加藤茂夫, 安田靖彦: "算術符号による中間調画像の高効率符号化", 画電誌 Vol. 12 No. 3 pp219-226, 1983
- 6) 加藤茂夫, 安田靖彦: "算術符号による多値画像の効率的符号化", 画電全大, No. 2, 1982.
- 7) 加藤茂夫, 安田靖彦: "算術符号による中間調画像の高効率符号化", 電子通信学会技術研究報告, IE 82, 57, 1982. 10.