



## 計算機内における画像のデータ圧縮\*

尾上守夫\*\* 岩下正雄\*\*

### Abstract

When large images such as ERTS multispectral images are processed by a system using a minicomputer, data compression is desirable for more efficient uses of auxiliary memories. Although much work has been done on bandwidth compression in image communication, it is not directly applicable to data compression within computer. Major differences are:

- (1) Exact restoration is required in data compression, whereas distortions unnoticed by human eye may be allowed in bandwidth compression.
- (2) Random access to data is required within a computer, whereas serial transmission is sufficient for communication.
- (3) In a computer, complicated processing and abundant use of memories and registers are possible, which are not practical in communication.
- (4) Error control is important in communication, but can be neglected in a computer.

With these points in mind, this paper presents an algorithm for data compression and restoration, which is based on Huffman code.

An implementation in a minicomputer (HP-2100) shows that the compression ratio of 1/2.54 for ERTS data is obtained and the processing time is comparable in comparison with the time for data transfer.

### 1. ま え が き

計算機による画像情報処理は、光学方式、写真方式、ビデオ方式などのアナログ処理に比べて精度、再現性、融通性などの点ではるかにまさるが、膨大な画像データの記憶とそれを直列に演算するためのコストが実用化をはばんでいた。しかし最近の IC, LSI などの進歩によって、デジタル処理のコストは急激に低下し、計算機による画像処理は実用期に入ったといって差支えない。

画像処理の演算そのものは勿論大型計算機で行えるが、画像処理は画像入出力装置との結合、対話型処理が重要なので現在の画像処理システムは中・小型の計算機を中心にして、それに入出力装置、対話型処理装

置をつけて構成している例が多い。この場合、磁気ディスクやテープの台数が少ないために画像データを圧縮する要望が強い。

たとえば地球資源探査衛星 ERTS によるリモートセンシング画像は 1 枚の画像データが 8 ビット/画素、 $3216 \times 2340$  画素、4 バンドからなり、磁気テープ 4 巻に収まっている。したがって少しでもデータ圧縮ができれば、ディスクやテープの交換の回数が減り、人手や処理時間が大幅に短縮される。

画像データの圧縮は 2 値画像の場合はファックスに関連して、また濃淡画像の場合はテレビジョンに関連して帯域圧縮の名のもとに多くの研究がなされている。しかしこれらの通信における圧縮と計算機内における圧縮とでは以下のような相違点がある。

- (1) 通信の場合は視覚特性を考慮して必ずしも正確な再現を行っていない。とくに濃淡画像の場合はそれによって圧縮率を上げている場合が多

\* Date Compression of Large Images within Computer by Morio ONOE and Masao IWASHITA (Institute of Industrial Science, University of Tokyo)

\*\* 東京大学生産技術研究所

- い. 計算機の場合は完全な再現が必要である.
- (2) 通信の場合はデータ全体が直列に送られるが, 計算機の場合は途中からのアクセスが必要である.
- (3) 計算機の場合の方が通信に比べて複雑な演算が可能である. また, 使用できるレジスタ数, 記憶容量も多い.
- (4) 通信の場合は誤り制御が重要であるが, 計算機の場合は計算機の本体側が行っているため, ユーザとしてはほとんど考える必要がない.

以上の点を考慮して計算機内で画像データの圧縮展開に適した一方法を考案し試用した結果好成績を収めたので報告する.

使用の形態は Fig. 1 に示すように MT に収められた原データを CPU で圧縮してディスクに格納する. 処理の際には必要とする任意の行あるいはブロックをディスクから CPU に呼出し, 展開して原データに戻して使用することを想定している.

処理した結果を再び圧縮してディスクにもどすことは一応除外し, 最後に若干の考察を行うにとどめてある. これは圧縮の基礎となる統計が変わってしまうことその他に, 可変長のレコードの管理の問題があるからである.

幸にしてデータ圧縮を最も必要とする気象衛星や資源探査衛星などの画像の処理は前記のような形態のものが多い.

勿論全画面にわたって端から順次にフィルタをかけていくような処理の場合は, Fig. 1 の MT がディスクにかわったと見なせばよく, 処理結果を圧縮できる.

圧縮・展開のための処理時間は CPU, ディスク間の転送時間に比べて短くなければならない.

## 2. 原画像データの統計量

圧縮を能率よく行うためには原画像の統計的性質を知らねばならない. ここでは例として前記の ERTS 画像 (1972. 11. 26. 関東地方) を用いた. 実際には計

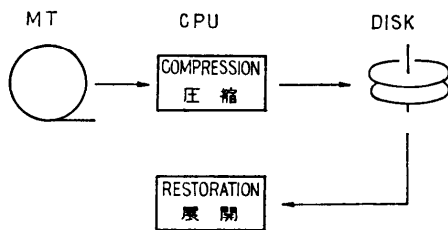


Fig. 1 Flow of Data Compression and Restoration

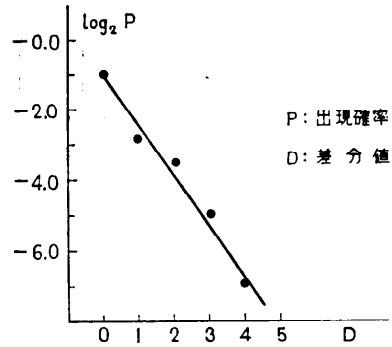


Fig. 2 Statistical Distribution of the Difference of Original Signal (Exponential Distribution)

Table 1 Exponent of Distribution, Entropy and Average Compressed bits/pixel of ERTS Multispectral Images

Band バンド	Slope 傾 斜	Entropy エントロピー	平均所要ビット数/画素	
			最適符号化	第4バンド 符号使用
4	0.96	2.42	2.51	2.51
5	0.63	2.94	3.21	3.26
6	0.50	3.37	3.83	4.12
7	0.77	2.77	2.94	3.02

算に使用したのは 804×780 画素の 4 バンドの部分画像である.

画像は 2 次元的な相関をもっているから圧縮の点からは行方向とともに列方向の相関も利用するのが得策であるが, この種の画像の処理は行単位のアクセスを必要とすることが多いこと, また後述のように 2 次元の相関を利用しても手間の割には圧縮率の向上が僅かなことから, ここでは行単位で圧縮を行うことにし, しかも一番簡単な DPCM を採用することにした.

Fig. 2 は第 4 バンドの差分のヒストグラムである. 一般に濃淡画像の差分は負の指数分布をすればよいと言われるが<sup>1)</sup>, この場合にもよくあてはまっている. 正負の差分はほぼ同一の直線上に並び対称性を有している. ただし, この傾斜はバンドにより若干異なり Table 1 のような値になる. 表にはエントロピーも示してある. これによると理想的な符号化を行えば, オーバーヘッドを考えなければ約 2.4 ビット/画素まで圧縮でき, 原データの 8 ビット/画素に比べて 1/3.3 になる.

## 3. 符号化

符号はハフマン符号を採用した<sup>2)</sup>. これは理論的に最適であるが通信における帯域圧縮では回路が複雑になるため実用になっていない. しかし計算機内では後

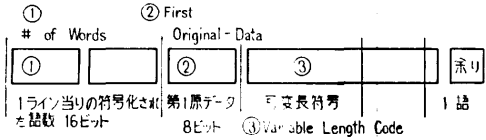


Fig. 3 Code Format for One Line

述のように容易に実現できる。この符号は可変長で出現確率の少ない差分値に対する符号長が長くなる。しかし余り長くなると有限長のアキュミュレータで演算するのに不便なので、ここでは符号長8ビットを目安にして、差分値|5|以上を一括して“その他”という符号を与え、そのうしろに原データ8ビットをそのままおくことにした。こうしても“その他”に属するデータ量は全体の数%程度であるので圧縮率はあまり劣化しない。

行ごとのアクセスを容易にするために、圧縮は行単位で行い、行の先頭には語単位で数えたその行の語数と最初の原データが付加してある。次に可変長の符号をすき間なくつめ、行の最終の語の余りは使用していない。この先頭および末尾のビットがオーバーヘッドになる。以上のフォーマットを Fig. 3 に示す。

4. 符号割当て

第4バンドに対する符号割当てを Table 2 に示す。これによるとオーバーヘッドを含めないで2.51ビット/画素になり、最適値にかなり近い値が得られている。オーバーヘッドを含めると3.15ビット/画素になる。バンドが異なると統計が異なるため最適の符号割当てが変わってくる。しかし、この程度の統計の差なら第4バンドの符号をそのまま使っても最適符号化に対して圧縮率は数%劣化する程度である。その比較を Table 1 に示してある。

Table 2 Code Assignment for ERTS Band 4 Image

D	N	C				B	T	
0	297493				1	1	297493	
1	85232			1	1	0	3	255696
-1	83909		0	0	0	0	3	251727
2	52418		0	0	1	0	4	209672
-2	51043		1	0	1	0	4	204172
3	18735	0	0	1	0	0	5	93675
-3	19377		1	1	0	0	4	77508
4	5108	1	0	1	0	0	7	35756
-4	5332	0	0	1	0	0	7	37324
E	7693	1	1	0	1	0	6	46158

Code Assignment . Total 1570725 b  
 ERTS 4 Band Average 2.51 b/pixel  
 D: 差分値 N: 頻度 C: 符号 (逆順) B: ビット数(1符号語) T: ビット数の小計 E: その他

5. 圧縮のアルゴリズム

Fig. 4 に圧縮のアルゴリズムの流れ図を示す。この他にビットカウンタ、カウンタおよび行番号の管理が必要であるが煩雑になるので図には省略した。

行の最初のデータは原データをそのまま使い、その次からは、前のデータとの差分をとり、差分の出現率が高い順に比較を行い、一致したらそれに対応する符号をテーブルから引き出す。差分の絶対値が5以上のときには、それに対応する可変長コードの後に原データをつける。これを1行が終了するまで繰り返す。この流れ図を見れば判るように出現確率の高い差分が最も速いループをまわる。すなわちハフマン符号は計算機による演算の点からも最適化されているわけである。

6. 展開のアルゴリズム

展開の場合の流れ図を Fig. 5 (次頁参照) に示す。行の最初のデータは原データであり、そのまま使用する。次のデータからは、まず可変長符号の部分で正負判定命令でMSBを比較判定し、次々にシフトを繰り返しながら、順に復号する。復号されたものは、差分であるので、前のデータに加算して、原データが復原される。“その他”という可変長符号が検出されたならば、その後に続く8ビットが原データそのものである。以上の手続きを1行が終るまで繰り返す。

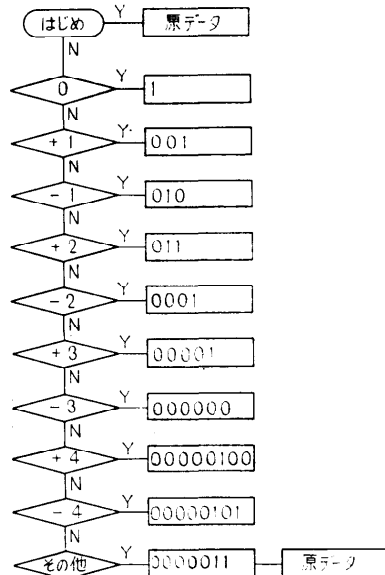


Fig. 4 Algorithm for Data Compression

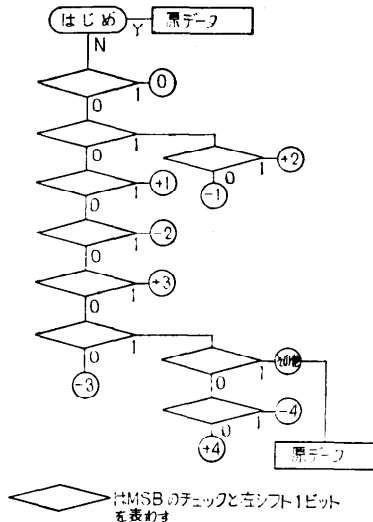


Fig. 5 Algorithm for Data Restoration

7. 適応型符号化

画像の統計が異なれば最適符号が異なる。したがって効率よく符号化するためには画像に応じて適応的に符号を切り替えていくのが望ましい。しかし画像の統計を一々とり直していくのは能率的でない。

Fig. 4 の圧縮の流れ図をよくみると各ループを通る回数をそれぞれ加算していけば、それがヒストグラムになっている。したがってヒストグラムを監視してそれがある値以上ずれたならば符号を切り替えることによって適応型の最適化がはかれる。切り替えるべき符号は予めテーブルにして持っているなり、アルゴリズムで作るなりすればよい。

しかし現在のデータは画像間の統計の差が少ないためこの効果はあまり多くない。Table 1 の比較はその例である。

Table 3 は第4バンドのデータをブロックに分けて各ブロック内で最適化した場合であるが、全体をひと

Table 3 Average Compressed bits/pixel when 1 line is Divided into 5 Blocks  
ERTS 4 Band DPCM

バンド	所要ビット数	平均所要ビット数/画素
1	284380	2.27
2	323113	2.58
3	316483	2.53
4	330689	2.64
5	314097	2.51
	1568762	2.50

Operation	Entropy
Original	3.03
$X-E$	2.42
$X-C$	2.88
$X-(C+E)/2$	2.47
$X-(C+2E)/3$	2.39
$X-(C+10E)/11$	2.41
$X-(D+E)/2$	2.63
$X-(D+2E)/3$	2.56
$X-(2E-D)$	3.06
Bit Plane Run Length	3.03

A	B	C
D	E	X

X: 予測する信号  
A, B, C: 前ラインデータ  
D: 前々置データ  
E: 前置データ

Fig. 6 Comparison of 2-Dimensional Predictive Codings

まとめにした Table 1 の必要ビット数と比較して改善は僅かである。

8. 2次元符号化

2次元の相関を利用するために2行を一括して、種々の予測符号化を行った。その結果を Fig. 6 にまとめてある。現在のデータに対して改善は僅かである。

9. 適用

以上のアルゴリズムをミニコン YHP 2100 とディスクとの間の圧縮展開に適用した。

転送速度は 0.52 msec/128 語、平均アクセス時間は 16.3 msec である。OS のファイル管理などを加えると 100 msec/行かかっている。

命令としては 80 種の内、実行時間の速い正負の符号判定、一致不一致の比較、無条件分岐、ビット回転、ビットシフト等を主に使用した。

その結果、圧縮展開に要する時間はそれぞれ 100 msec/行であり、1行=804 語であるから、転送時間に比し、同程度で済む。

以上では、用いた言語はすべてアセンブラであるが、マイクロプログラミングにより一層の高速化ができる。

10. むすび

中・小型計算機を中心に構成された画像情報処理システムにおいてもリモートセンシング画像のような膨大なデータを計算機内で圧縮し、必要に応じて任意の行単位で CPU に読み出し、展開して原データを復原するような方法を検討した。

通信における帯域圧縮と計算機内におけるデータ圧

縮との相違を指摘し、理論的には最適とされながら、通信では回路が複雑になるために実用されていないハフマン符号が計算機では有効に使える、演算時間の点からも最適化されていることを明らかにした。例として ERTS 衛星画像の種々の統計量を求め、オーバーヘッドを入れても 1/2.54 の圧縮率が得られることを示した。この場合はあるバンドに対する最適符号で他バンドを符号化しても圧縮率はあまり劣化しない。

圧縮のアルゴリズムはまたヒストグラムの監視にも使用でき、適応型の最適化も容易に行えることを示した。

ミニコンに適用した結果、圧縮展開に要する演算時間は転送時間に比してさほど大きくなく、十分実用になることを確めた。

またこの結果からみるとハフマン符号は多くの優れた点があり、マイクロプロセッサなどが容易に使えるようになった現在、通信における使用も考えてよいと思われる。

謝辞 符号についての安田助教授、画像処理についての高木助教授の御教示に厚くお礼申し上げます。

#### 参考文献

- 1) E. P. Kretzmer: Statistics of Television Signals, BSTJ, Vol. 31, pp. 751~763 (1952).
- 2) D. A. Huffman: A Method for the Construction of Minimum Redundancy Codes, Proc. I. R. E., Vol. 40, pp. 1098~1101 (1952).

(昭和51年8月30日受付)

(昭和51年11月20日再受付)