Compression of ERTS Multispectral Images within Computer

Morio ONOE* and Masao IWASHITA*

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 anxiliary memories. Although much work has been done on bandwidth compression in image communication, it is not directly applicable to data compression within computer. This paper pointed out major differences and 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. Introduction

Most of image processing facilities are built around minicomputers, because special equipments for input and output of image and interactive displays are usually not availabe at a large computer center. Hence there is a strong demand to compress image data, so that they can be handled by a limited number of magnetic tapes and disks.

For example, one frame of ERTS(LANDSAT I) satellite image consists of 4 spectrum band images, each of which contains 3,216×2,340 pixels of 8 bits gray level. It occupies 4 magnetic tapes, which exceed the capability of simultaneous tape handling in most image processing facilities.

Much work has been done on the bandwidth compression of binary black and white image in facsimile and half-tone image in television. They are not directly applicable to data compression within computer. Major differences are:

(1) Exact restoration is required in data compression, whereas distortions un-

---

This paper first appeared in Japanese in Joho－Shori (Journal of the Information Processing Society of Japan), Vol. 18, No. 8 (1977), pp. 776～780.

* Institute of Industrial Science, University of Tokyo

noticed 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, an algorithm for data compression and restoration is developed based on Huffman code.

The configuration of a system shown in Fig.1 is assumed. The original data in magnetic tapes are compressed in a CPU and then stored in disks. For the processing, required lines or blocks are transfered to the CPU from disks and restored. This form of processing is common in remote sensing by weather or earh resources satellites, of which data is large and needs the compression.

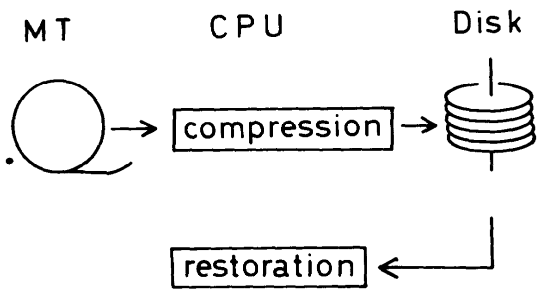

Fig.1 Flow of data compression and restoration

2. Statistics of original image data

For an efficient compression, some knowledge of statistics of original data is essential. As an example, 4 band images of ERTS satellite flying over Tokyo area on November 26, 1972 are used in this paper. A partial image of 804×780 pixels is used.

Since most processing requires line by line access to data, only the compression of DPCM signal along a line is used.

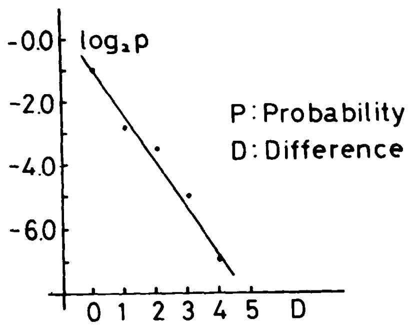

Fig. 2 Statistical distribution of the difference of original signal (Exponential distribution)

The histogram of DPCM signal of the Band 4 image is shown in Fig.2. The same distribution is obtained for both positive and negative differences. The distribution matches with the negative exponential distribution as in the case of most images.[1)] The slope depends on the band number (wavelength). Table 1 summarizes the slope and the entropy for each band.

Band	Slope	Entropy	Average bits/pixel	
			Optimized	4th band
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

Table 1 Exponent of distribution, Entropy and Average bits/pixel of ERTS multispectral images

3. Coding

Huffman code has been known as an optimum code, but rarely used in communication because of the complexity of implementation.[2)] This is not the case of data compression within computer as described in the next section.

The length of code becomes longer for a signal with lower probability. Since the word length of accumulator of a computer is finite, the handling of very long codes is not convenient. Hence an extra category is provided for the difference value of more than 5, which would require the code length of more than 8 bits in Huffman coding. The code for this category consists of a special code followed by the original data of 8 bits. This modification hardly impair the efficiency of Huffman code, because the probability of the extra category is less than a few per cent.

D	F	C	B	T
0	297493	1	1	297493
1	85232	110	3	255696
-1	83909	0 00	3	251727
2	52418	0 010	4	209672
-2	51043	1 010	4	204172
3	18735	00 1 00	5	93675
-3	19377	1 1 00	4	77508
4	5108	1010 1 00	7	35756
-4	5332	0010 1 00	7	37324
E	7693	110 1 00	6	46158

Total 1570725 bits
Average 2.51 bits/pixel

D : Difference F : Frequency C : Code
B : Bits/ 1Code T : Total

Table 2 Code assignment for ERTS 4th band

The first word of a line contains the number of words in the line and the first original data. Then follows variable length codes, which are packed without gap. Any empty space in the last word is arbitrarily filled. These first and last words are overhead.

4. Code assignment

The code assignment for the band 4 is shown in Table 2. The average bits per pixel is 2.51 without overhead and 3.15 with overhead. Compressed values obtained by the optimum code assignment for each band are shown in Table 1, which also includes the compressed values obtained by the code assignment for the band 4. The degradation due to suboptimal assignment is small, because the difference in statistics between bands is small.

5. Algorithm for compression

Fig.3 shows the flow chart of an algorithm for compression. For the sake of simplicity, counters for bits and lines are not shown in the figure. Each difference signal is compared with the class values lined up according to their probability until a matching is obtained and then a corresponding code is issued. Otherwise the extra code and the original value are issued. This process repeats itself until the end of a line.

It should be noted in the flow chart that the higher the probability, the shorter the loop. Hence the Huffman code is also optimum in terms of the execution time.

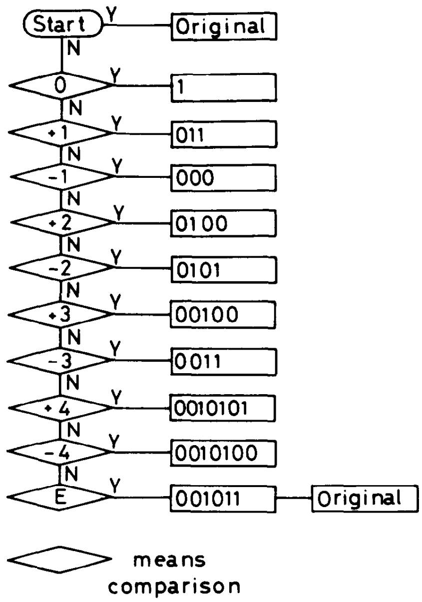

Fig.3 Algorithm for data compression

6. Algorithm for restoration

Fig.4 shows an algorithm for restoration. The first word is the original data.

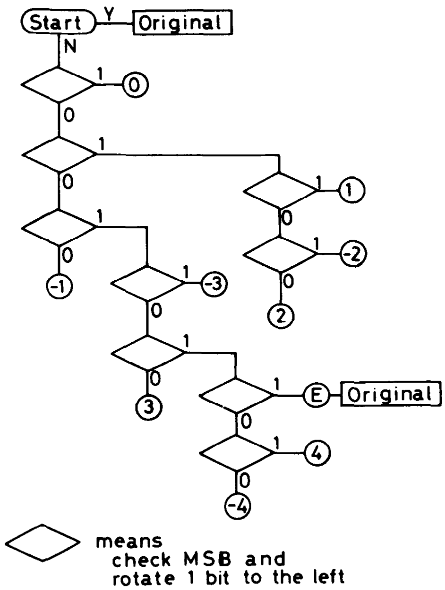

Fig.4 Algorithm for data restoration

The following portion is put into the accumulator. The most significant bit is checked and then the content of the accumulator is shifted to the left. This process is repeated until a corresponding code is found.

7. Adaptive coding

The optimum code assignment depends on the statistics. Hence it is desirable to adaptively switch the assignment, whenever the statistics change.

An examination of the flow chart shown in Fig.4 reveals that the histogram is obtained by merely adding the number of flow in each loop. Hence an adaptive switching is easily done by detecting a significant change in histogram.

8. Two-dimensional prediction coding

Various two-dimensional prediction is tried using two lines. The results reveal only a slight improvement of 20 % at most in comparison with the line by line coding so far discussed.

9. Implementation

The proposed algorithm is implemented for the data transfer between a CPU of a minicomputer (HP-2100) and a disk. The transfer rate is 0.52 msec/128 words and the average disk access time is 16.3 ms. The average transfer time for a line (804 words) under a operating system is 100 msec/line. For the compression or the restoration another 100 msec/line is required. The use of microprograming is being considered to shorten this execution time.

10. Conclusion

The compression of image data within a computer is discussed. Major differences between bandwidth compression in communication and data compression within computer are pointed out. It is shown that the Huffman coding, which is rarely used in communication, is optimum for data compression within a computer in terms of execution time as well as efficiency. An adaptive switching of code assignment can be done by detecting a significant change in histogram, which is easily obtained during the compression.

Statistics of multispectral image from the ERTS satellite are shown. Since there is little difference of statistics between spectral bands, even a fixed code assignment yield the compression ratio of about 2.5 including overhead. The execution time for the compression or the restoration is comparable with the transfer time of data.

Acknowledgement: The authors wish to thank for discussion with Dr. Yasuda and Dr. Takagi.

References: (1) E. P. Kretzmer: Statics of Television Signals, BSTJ, Vol,31, pp.751-763 (1952)

(2) D. A. Huffman: A Method for the Construction of Minimum Redundancy Codes, Proc. IRE, Vol.40, pp.1098-1101 (1952)