Difference Base-bit Plus Overflow-bit Coding

Qian Shenen, Li Shuqiu and Wang Ruqin

(State Key Laboratory of Applied Optics)

Dai Yisong

(Jilin University of Technology)

Abstract

A new coding methed—Difference Base-bit plus Overflow-bit Coding (DBOC) is proposed Compard with the Huffman coding, this method does not require the statistical properties of coded data, and it can be realized in real-time. Experiments show that the coding efficiency can reach over 90%. With this coding method an excellent result has been obtained in an on-board data compression system for imaging spectrometer.

Key Words: Coding, Real-time processing, Data compression, Imaging spectrometer

1. Introduction

In the research of an on-board data compression system for the High-Resolution Imaging Spectrometer (HIRIS)[1], in order to represent the data compressed by the Two-True-Value Linear Prediction (TIVLP)[2] with the least number of bit and get the maximum compression ratio of the system, it is necessary for the result obtained to code efficiently in real-time. It is known that the Huffman coding is the optimal coding method. But it requires statistical properties of source data, and its algorithm is complicated and can not be realized in real-time[3]. For this reason we have proposed a new coding method—Difference Base-bit plus Overflow-bit Coding. This coding method does not require the statistical properties of the source data. Its algorithm is simple and direct, and is suitable to operate on-line.

2. Difference Base-bit Plus Overflow-bit Coding

After the original data are compressed by the TIVLP, a great deal of redundant information is removed. If the compressed results are dire-

ctly represented, it requires a large number of bits to express compressed data. This is not of benefit to the increment of the total data compression ratio of the system. In fact, there is still a larger redundance in the compressed data. If the compressed data are coded efficiently to remove the redundant information once more, the total data compression ratio of the system can increase further. Below we will discuss the coding method in two steps.

1. Adjacent-Difference for Reducing the Value Range

Generally speaking, the greater the mean μ of the source data to be coded, the larger their value range and the longer the average code length $\overline{L}$ to express these source data; on the contrary, the smaller the mean μ, the smaller the value range and the shorter the average code length $\overline{L}$. So we must first reduce the value range of the source data to be coded in order to decrease the average code length $\overline{L}$.

Table 1 lists 48 groups of source data, which are results of the corn spectral data compressed by the TIVLP. Seeing from the table 1, the values of y1(i) and y2(i) (i = 1, 2, ···48) all are relative larger. The means are as $\mu_1 = 72.75$ and $\mu_2 = 73.06$ respectively. The values of RL(i) (i = 1, 2, ···48) are relative smaller, the mean is $\overline{RL} = 3.35$. Undoubtedly, the total number of bit to express 48 y1(i) or y2(i) must be greater than that to express 48 RL(i), and the average code length of the former must be longer than that of the latter. In fact, the source data y1(i) and y2(i) ompressoed by the TIVLP are two adjacent samples, while y1(i) and y2 (i-1) are the two nearest samples of the two adjacent prediction lines, there is correlation between them. The differences between them is as following:

$$\begin{cases} \Delta y1(i) = y1(i) - x2(i-1), & i = 2, 3, \cdots N; \quad (1) \\ \Delta y2(i) = y2(i) - y1(i), & i = 1, 2, \cdots N; \quad (2) \\ \Delta y1(1) = y1(1) & \quad (3) \end{cases}$$

The values of Δy1(i) or Δy2(i) must be smaller than that of y1(i) or y2 (i), and the mean of Δy1(i) or Δy2(i) also must be snaller than that of the source data.

The difference Δy1(i) and Δy2(i) of the compressed result of corn spectral data are listed in table 2. Their value ranges are greatly reduced. The means of them are decreased to $\mu_1' = 4.35$ and $\mu_2' = 5.77$ respectively. If we code Δy1(i) and Δy2(i) instead of y1(i) and y2(i), it requires a small number of bits to express the source data, and the average code length $\overline{L}$ can be shortened. In reconstruction the source data can be gained according to following:

$$\begin{cases} y1(1) = \Delta y1(1); & (4) \\ y2(i) = \Delta y2(i) + y1(i), \quad i = 1, 2, \cdots N & (5) \\ y1(i) = \Delta y1(i) + y2(i-1), \quad i = 2, 3, \cdots N; & (6) \end{cases}$$

We call this procedure, which replaces the source data with the difference value of its neighbors, "adjacent-difference". This procdeure has generality and can be used for any kind of source data.

2. The Base-bit Plus Overflow-bit Coding

Generally, in source data to be coded there must be a value $R(=2^n)$ then than which most of source data are smaller. In the compressed result of corn spectral data mentioned above, for example, among 48 run lengths RL(i) there are 35 RL (i)s whose values are smaller than $R=4=2^n(n=2)$, this number accounts for 73% of rhe total number. In table 2 among 48 $\Delta y1(i)$s there are 37$\Delta y1(i)$s whose values are smaller than $R=8=2^3(n=3)$, the number accounts for 77% of the total number; among 48$\Delta y2$ (i) s there are 36 $\Delta y2(i)$ s whose values are smaller than $R=8(n=3)$, the number accounts for 75% of the total number. For such a kind of source data whose most of values are smaller than a certain critical value R, we code them with a coding method named Base-bit plus Oveflow-bit Coding (BOC). This coding method divides a coding result into base-bit and overflow-bit two parts. The base-bit is the part whose length is fixed, the length of base-bit is decided by R. The overflow-bit is the part whose length is unfixed, the length of overflow-bit is decided by the values greater than R. Each overflow-bit "1" represents value R $(=2^n)$. How many times greater than R the value of coded data is, how many number of overflow-bits there are. For marking the end of overflow-bit, one bit "0" is used as a comma of overflow-bit. In coding with this method, since most of values of the source data are smaller than R, the length of overflow-bit is equal to zero, n+1 bits is enough to express each coding results, among them one bit is comma. If source data are with signs, an additional sign-bit is needed, total n+2 bits required. while for coding small proportion of source data whose values are greater than R, a coding result involves the base-bit and the overflow-bit two parts.

This coding method is direct and convenient, it does not require statistical properties of source data and can represent the source data with the least number of bit. In coding of the data compressed by the TIVLP, assuming the lengths of base-bit of $\Delta y1(i)$, $\Delta y2(i)$ and RL(i) to be k, n and m respectively, considering RL(i) no sign, only 5+k+n+m bits are required for most of source data. In coding of the compressed data listed in table 1, for example, the lengths of base-bit are $k=n=3, m=2$, it takes only 13 bits to express most of coding results. Comparing with 24 bits (3

bytes) taken when the source data are directly expressed, the number of bit is greatly decreased.

3. Effect of L_b on Coding Efficiency

Compared with Huffman coding, the BOC has only one parameter L_b the length of base-bit. The relation curves between the length of base-bit L_b and the coding efficiency η are showed in Fig.1. The horizontal coordinate is the inverse of μ the mean of source data. If the μ is know, the L_b corresponding to the maximum η can be gotten from the figure. The greater the mean μ, in other words, the smaller the $1/\mu$, the longer the length of base-bit L_b. For instance the mean of the $y2(i)$ $(i=1,2,\cdots 48)$ listed in table 1 $\mu_2=73.06$, its inverse $1/\mu_2=0.0137$, knowing from the figure, when $L_b=6$ the coding efficiency reaches the maximum; by contrast, the mean of $\Delta y2(i)$ $(i=1, 2,\cdots, 48)$ $\mu_2'=5.77$, its inverse $1/\mu_2'=0.17$, when $L_b=2$ the coding efficiency reaches the maximum. Since the $\Delta y2(i)$ is with a sign after difference, an additional sign-bit is needed. The length of base-bit for $\Delta y2(i)$ is actually 3 bits. It is thus obvious that 3 bits can be saved on an average when we code each $\Delta y2(i)$ instead of $y2(i)$.

When we perform coding with this method the mean of the source data can be roughly estimated according to the data to be coded, then select an appropriate L_b. The horizontal coordinate in Fig. 1 is taken the logarithm for the purpose of compressing the proportional scale of the coordinate. It could be seen more clear that the shape of each curve is very plane, if the coordinate were taken linear scale. As for a certain μ, the difference between two efficiencies corresponding to two neighbor curves is not very large. This makes it become easy to select L_b, Even if the mean of the source data is estimated with large error, the coding efficiency does not reduce very much.

Now we take compressed results of corn spectral data as the examples to illustrate the effect of the deviation of L_b on the efficiency. The first example is coding of $\Delta y2(i)$. It has been gotten above that $\mu_2'=5.77$, its inverse $1/\mu_2'=0.17$, and when $L_b=n=2$ the coding efficiency reaches the maximum. When we code $\Delta y2(i)$ $(i=1,2,\cdots 48)$ with $L_b=n=1,2,3$ speparately the three average code lengths corresponding to each L_b obtained are $\overline{L\Delta_{y2}}=5.61, 5.23, 5.54$ respectively. It is evident that when $n=2$ the average code length is the shortest. The second example is coding of $RL(i)$. The mean of RL (i) is $\overline{RL}=3.35$, its inverse is equal to 0.3. Knowing from Fig.1, when $L_b=m=1$ the coding efficiency reaches the maximum. When we code $RL(i)$ $(i=1,2,\cdots,48)$ with $L_b=m=0,1,2$ separately, the three average

code lengths are $\overline{L_{RL}}$ = 3.69c, 3.50, 3.63. It is obvlous that when m = 1 the average code length is the shortest. It can be seen from these two examples that the coding result changes very little, even if the length of base-bit is selected with error. When the L_b is taken the neighboring values of the length corresponding to the mean, the two deviations of the average code length of {Δy2(i)} are 7.2% and 5% respectively, and these of {RL(i)} are 5.4% and 3.7% respectively. The deviations all are not very large. This means that the selection of L_b has a wide available range.

It is known from discussion above that it is of benefit to the increment of coding efficiency to select an appropriate length of base-bit L_b, when coding with the BOC.

4. Experimental Results and Analysis

We perform an experiment with the compressed of each GIFOV's spectrum in the HIRIS. First we convert the {y1(i), y2(i), RL(i)} (i = 1, 2, ···, M) into {Δy1(i), Δy2(i), RL(i)} (i = 1, 2, ···N) by adjacent-difference, then code them with the BOC. Table 3 lists the coding results of compressed data of twnety-one kinds of typical earth resources spectra. For the convenience of analyzing and evaluating the coding efficiencies, table 3 also lists the entropy $H_0(x)$ of each kind of compressed data, the entropy $H_d(x)$ of the data after difference and two coding efficiencies η_1, η_2 corresponding to these two entropies. All the coding results listed in table 3 are obtained under the condition of the L_b taken the value of the maximum coding efficiency. The Shonon's theorem of distortionless coding tells us that the average code length $\overline{L}$ is never smaller than the entropy of the source data. But the average code length L listed in table 3 all are smaller than their entropy $H_0(x)$, and the coding efficiency η_1, which is defined as $H_0(x)$ divided by $\overline{L}$, is greater than 100%. This is because the entropy $H_0(x)$ of N groups of source data is obtained in terms of the generation probability of each value of {y1 (i), y2(i), RL (i)}. Since some correlation between the source data has been removed after the adjacent-difference, the entropy becomes smaller $H_d(x)$. In fact, it is the data after adjacent-difference {Δy1(i), Δy2(i), RL(i)} that the BOC code. The efficiency η_2, which is defined as $H_d(x)$ divided by $\overline{L}$, is the coding efficiency of the BOC. It meets the Shonon's theorem of distortionless coding. In table 3 η_2 all are above 90%, this result is similar with that of the Huffman coding.

The Difference Base-bit plus Overflow-bit Coding is a new coding method, which does not require statistical properties of source data. Its algorithm is direct and simple. It first reduces the value range of sourc

data by the adjacent-difference, then codes the difference with the BOC. The only one parameter of this method is the length of base-bit L_b, which is decided by the mean of the source data. A suitable L_b can make the average code length the shortest. The selection of L_b has a wide available range. The deviation of average code length caused by neighboring values is generally smaller than 5%. The coding efficiency can reach above 90%, which is similar with Huffman coding. Through the experiment of 21 kinds of typical earth resources spectra, the average code length usually is not greater than 5 bits in a 8-bit digital system, the coding efficiency is iecreased by near 1 times.

Table 1 Compressed data of corn spectrum to be coded (N=48)

i	y_1 (i)	y_2 (i)	RL (i)	i	y_1 (i)	y_2 (i)	SL (i)	i	y_1 (i)	y_2 (i)	RL (i)
1	9	8	1	18	163	164	4	35	51	54	10
2	10	11	3	19	165	162	5	36	81	78	0
3	11	11	4	20	150	149	3	37	78	83	0
4	13	15	3	21	149	149	6	38	83	82	9
5	20	19	10	22	147	145	0	39	70	68	2
6	12	12	2	23	145	139	1	40	67	67	3
7	14	17	0	24	134	132	1	41	64	52	1
8	17	24	1	25	127	116	1	42	41	33	1
9	33	66	0	26	102	90	0	43	26	22	4
10	66	79	0	27	90	66	0	44	9	8	3
11	79	111	0	28	66	55	0	45	7	8	17
12	111	131	0	29	55	51	0	46	28	30	2
13	131	143	0	30	51	37	0	47	31	32	3
14	143	147	2	31	37	32	1	48	32	31	27
15	152	153	8	32	30	28	1				
16	158	158	12	33	26	29	7				
17	160	162	2	34	48	48	1	μ	72.75	73.06	3.35

Table 2 Data to be code after difference

i	Δy_1 (i)	Δy_2 (i)	i	Δy_1 (i)	Δy_2 (i)
1	9	−1	40	−1	0
2	2	1	41	−3	−12
3	0	0	42	−11	−8
4	2	2	43	−7	−4
5	5	−1	44	−13	−1
6	−7	0	45	−1	1
7	2	3	46	20	2
8	0	7	47	1	1
9	9	33	48	0	1
10	0	13			
⋮	⋮	⋮			
⋮	⋮	⋮	μ	4.35	5.77

Table 3 Coding results of cmpressed data of 21 typical earth resources spectra

No	Name of Spectrum	Properties of Data		Coding Results		
		$H_0(x)$	$H_d(x)$	$\bar{L}$	$\eta_1(\%)$	$\eta_2(\%)$
1	Organic-dominated Soil	4.59	3.85	4.01	114.3	95.9
2	Iron-affected Soil	4.94	4.37	4.58	107.9	95.4
3	Calcitic Carbonatite	5.16	4.11	4.40	117.2	93.5
4	Kaolinite	5.51	4.63	4.87	113.1	95.0
5	Beans	5.64	4.33	4.46	126.7	97.1
6	Corn	5.64	4.73	5.03	113.0	94.0
7	Red Pine	5.48	4.53	4.96	110.6	91.4
8	Birch	5.52	4.42	4.59	120.3	96.4
9	Water	4.88	4.03	4.33	112.6	92.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
21	Snow	5.51	4.64	4.86	113.2	95.4

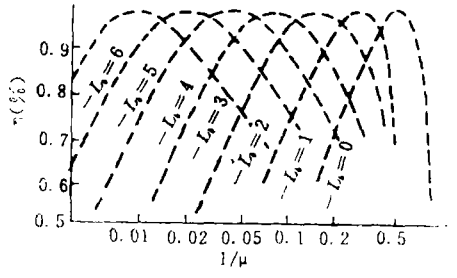

Fig.1 Curves of L_b vs η

References

[1] Qian Shenen, Ph.D thesis, Jilin University of Technology, 1990, (in Chinse)

[2] Qian Shenen, Acta Optical Sinica, 1990, 10(3), 260-266 (in Chinese)

[3] Zhou Jiongpan, Foundation of information theory, Post and Communication Publishers, 1983, 358-378, (in Chinese)

[4] Qian Shenen, Proc.SPIE, 1990, 1244, 331-342

差值“基础比特+溢出比特”编码方法

钱神恩　李树秋　王汝勤　　戴逸松

（应用光学国家重点实验室）　（吉林工业大学）

摘要：提出了差值“基础比特+溢出比特”编码方法。与霍夫曼方法相比，具有无需知道信源的统计特性，算法简便，易于实时处理，编码效率高（可达90%以上）等特点，并在成像光谱仪机上实时数据压缩中取得了良好的应用效果。

关键词：编码；实时处理；数据压缩；成像光谱