HIGHLY EFFICIENT CODING SCHEMES FOR CONTOUR LINE DRAWINGS

Yuk-Hee Chan *and* *Wan-Chi Siu*

Department of Electronic Engineering
The Hong Kong Polytechnic University
Hong Kong

ABSTRACT

In this paper, adaptive coding schemes for contour line drawings based on chain code representation is presented. In this scheme, the chain code or the chain-difference code of a contour is modeled as an n-order Markov sequence and then coded with arithmetic coding scheme adaptively. Experimental result shows that the proposed approach is better than some other conventional approaches.

1. INTRODUCTION

Line drawings can always be described with chain code. Freeman[1] proposed an eight-directional encoding scheme for line drawings. In this coding scheme, we first superimpose a rectangular grid on the curve and then quantize the curve to the nearest grid point. A link is then defined as one of the eight possible straight-line segments between two adjacent quantized points. The ordered sequence of these links forms a chain code. In this case, the chain code is of a 3 bit per link to present a line drawing.

Obviously, successive links are highly correlated. Freeman[2] proposed a chain-difference coding scheme to improve the coding efficiency by making use of this property. In his approach, variable-length codewords are assigned to the difference between two consecutive links. Indeed, this property has been widely used in various coding schemes[3,4] to achieve a better coding efficiency.

Recently, Lu and Dunham[5] have proposed two efficient coding schemes to encode contour line drawings based on Freeman's chain code representation. In these two schemes, the chain-difference code of a curve is first obtained. The chain-difference code is then modeled as a 1st-order or 2nd-order Markov sequence. For each state in the chain-difference code, the conditional probability distributions for the next chain-difference input are measured experimentally. Based on the statistics measured, the chain-difference code is encoded with either Huffman coding or arithmetic coding scheme.

Both Huffman and Arithmetic coding schemes can utilizes the statistics of the message to make the most frequent symbols correspond to the shorter code and the rare symbols correspond to the longer code. However, theoretically speaking, arithmetic coding scheme is superior to Huffman coding scheme since it can be easily adaptive to the local statistics of the message[6]. This property is practically useful especially when the local message statistics varies often. Furthermore, it makes adaptive encoding possible. In this paper, based on Lu and Dunham's coding scheme[5], we introduce two more efficient coding schemes by making use of this property of the arithmetic coding scheme.

2. CODING SCHEMES

In Lu and Dunham's coding schemes[5], the chain-difference code instead of the chain code is modeled as Markov sequence for encoding. Theoretically, either chain-difference code or chain code can be modeled as Markov sequence for encoding. If they are directly encoded without further processing, chain-difference code is much better than chain code to be encoded since it decorrelates most of the correlation between consecutive symbols. However, this is not the case when they are modeled as a Markov sequence for encoding.

Chain-difference code can be obtained from chain code via a simple 1-tag recursive filter which decorrelates the correlation between two consecutive links. Therefore, if an entropy encoder/decoder for non-Markov sequences is exploited, the sequence is better to be a chain-difference code. However, if an entropy encoder/decoder for a high-order Markov sequence is exploited, we found that a similar or even higher compression rate can be achieved when the chain code is used as the input.

In fact, this phenomenon can be explained quite easily. The reason is that the chain-difference code sequence is highly uncorrelated compared with the chain code sequence. This implies that the conditional probability $P(y_{i+n}|y_{i+n-1} \dots y_i)$ is more or less evenly distributed while the conditional probability $P(x_{i+n}|x_{i+n-1} \dots x_i)$ is not, where x_i is the i^{th} symbol of the chain code sequence and $y_i = x_i - x_{i-1}$. Therefore, there will be no gain in applying arithmetic coding to the chain-difference code sequence but to the chain code sequence when they are modeled as a high-order Markov sequence. In other words, the chain code itself is more suitable than the chain-difference code to be modeled as a higher-order Markov sequence.

Lu and Dunham's coding schemes[5] are basically 2-pass algorithms since the statistics of the message has to be obtained before performing the entropy coding. When the

0-8186-7310-9/95 $4.00 © 1995 IEEE

message length is long enough, there will be a considerable processing delay. On the other hand, extra bits for carrying the information of the message statistics are required to be sent or stored. In view of these factors, it would be more efficient to exploit 1-pass algorithm instead. In that case, arithmetic coding is superior to Huffman coding since it can be implemented in a way that it does not require the probability distribution of the symbols to start the encoding process.

However, it doesn't mean that 2-pass algorithms are always inferior. To make the message statistics converge to the real statistics in an 1-pass arithmetic coding scheme, the length of the coding sequence must be long enough. When the sequence is modeled with a higher-order Markov model, the length of the message required for the statistics to converge would be longer. If the input sequence is too short, the compression rate of an 1-pass coding scheme can be much lower than that of a 2-pass scheme. Note also that the overhead of exploiting 2-pass coding schemes is tolerable in this case. Therefore, 2-pass coding schemes would be preferred in such a case.

Adaptive scheme can also be applied in a 2-pass algorithm. Since the real statistics of the whole message is transmitted or stored as a header of the message, the decoder knows the message statistics and, during decoding, it is able to get the statistics of the message segment not yet received based on the symbols already received. In that case, the decoder can always have the real statistics of the following symbol received and therefore can make the best use of the arithmetic coding scheme.

In our proposed approach, the source code sequence is modeled as a Markov sequence. Either chain-difference code or chain code is used as the source code sequence. Which one is used depends on the order of the Markov model exploited. Generally speaking, if the order is higher than 2, chain code is directly modeled. Otherwise, chain-difference code is used instead.

In the first scheme, the message statistics is obtained at the first pass and, at the second pass, the message is encoded with the arithmetic coding scheme based on the conditional probabilities obtained. During the first pass, the number of conditional occurrence of a particular symbol in the message is recorded. This information forms a header and is encoded with variable-length coding technique. Based on this information, the message statistics can be computed. At the second pass, whenever a symbol is transmitted, the encoder decreases the number of this symbol's conditional occurrence by one accordingly. The statistics of the message segment not yet transmitted is also updated correspondingly such that the arithmetic coding encoder can encode the following message symbol with the most updated statistics. At the receiver, the decoder gets the header and calculates the message statistics. The statistics is updated in a similar way as the encoder does and decodes the coming message with the current statistics.

In the second scheme, the message is encoded in a single pass. The message statistics is assumed based on a priori knowledge and is updated as the symbols are encoded/decoded. Note that the updated message statistics will converge to the real one if the message length is long enough. Based on the current message statistics on hand, the input symbol is encoded with the arithmetic coding scheme. Specifically, to encode a particular symbol, its previous symbols and the conditional probabilities of its occurrence are exploited. The number of previous symbols which have to be retained for encoding equals to the order of the Markov model used to describe the message.

3. SIMULATION RESULTS

Simulation has been carried out to evaluate the performance of the proposed coding schemes. A data set which consisted of contour maps of size 2000 $\times$ 2000 each was encoded with chain coding scheme and chain-difference coding scheme. Each output sequence was then modeled as Markov sequences of various orders and encoded with 1-pass and 2-pass arithmetic coding schemes individually. The length of the chain/chain-difference code sequences varied from 6500 to 16000 symbols. Both adaptive and non-adaptive approaches were simulated for comparison. In our simulations, contour maps were divided into 5 groups according to their geographical locations and separately encoded. When 1-pass arithmetic coding scheme was exploited, the average of the statistics of all chain/chain-difference code sequences except those belong to Europe Group was used as the initial message statistics to encode individual sequence.

Tables 1 and 2 show the average bit length per symbol required to encode the chain/chain-difference code sequences when various approaches are used. Overheads for 2-pass approaches are not included in the figures reported. Roughly speaking, when 2-pass algorithms is applied, it takes 0.006 to 0.012 bits per symbol to encode the overhead for an Non-Markov sequence, and, 0.01 to 0.02 bits per symbol for a Markov-I sequence and so on. Based on our simulation results, we have the following observations:

i/. The coding efficiency of adaptive coding scheme is always better than that of non-adaptive coding scheme [5] when 2-pass approach is applied.

ii/. When the length of the sequence being encoded is long enough, the compression rate of the 1-pass adaptive approach is comparable to those of the 2-pass approaches. In fact, by taking the overhead that required in a 2-pass approach into account, sometimes the performance of the 1-pass approach would be even more respectable.

iii/. Higher compression rate can be achieved by encoding chain code instead of chain-difference code if the input sequence is modeled as a high-order Markov sequence during encoding. From our simulation results, we found that it was better to encode chain code directly when the input was modeled as a Markov sequence of order higher than 2.

iv/. Higher compression rate can be achieved by encoding chain-difference code instead of chain code if the input sequence is not modeled as a Markov sequence during encoding.

v/. From the tables shown, it sounds that it is better for us to model the input sequence as a higher-order Markov sequence. However, this is not always true in general. When 1-pass scheme is applied, it takes longer message for the message statistics to converge to the real one if higher-order Markov model is used. One can see the larger difference of the compression rate between 1-pass scheme and 2-pass schemes when higher-order model is used. This is because the message statistics used for encoding in the 1-pass scheme is still quite far from the real. When 2-pass scheme is applied, the overhead is exponentially proportional to the order of the Markov model exploited. Hence, in practice, Markov model of order 2 or 3 is appropriate.

4. CONCLUSIONS

In this paper, two adaptive coding schemes for encoding contour line drawings are proposed. These two schemes model an input sequence, which is either a chain code or a chain-difference code, as a Markov sequence and adaptively encode it with arithmetic coding scheme. Both schemes are adaptive in a way that it updates the message statistics whenever a symbol is encoded/decoded during encoding/decoding the message and encodes/decodes the following symbol with the most updated statistics. Among these two schemes, one is an 1-pass algorithm and another one is a 2-pass algorithm. Both show their advantages over some other non-adaptive coding schemes such as those proposed in ref.[5].

An analysis on the influence of some of the factors which affects the performance of the proposed schemes is also given in this paper. These factors includes the order of the Markov model exploited, the choice between chain code sequence and chain-difference code sequence as the source being modeled, and the choice between 1-pass scheme and 2-pass scheme.

5. ACKNOWLEDGEMENT

The authors would like to thank W. H. Wong, S. O. Choy, S. W. Hong and M. T. Man for their effort in preparing the raw data for simulation.

6. REFERENCES

[1] H. Freeman, "On the encoding of arbitrary geometric configurations," IRE trans. Electron. Comput., Vol.EC-10, Jun 1961, pp.260-268.

[2] H. Freeman, "Application of the generalized chain coding scheme to map data processing," Proc. IEEE Comput. Soc. Conf. Patten Recog. Image Processing, Chicago, IL May 1978, pp.220-226.

[3] T. Kaneko and M. Okudaira, "Encoding of arbitrary curves based on chain code representation," IEEE trans. Commun., Vol.COM-33, Jul 1985, pp.697-707.

[4] Y. Yasuda and T. Kimot, "Efficient differential chain code for line drawings processed by Hilditches thinning method," Proc. Int. Conf. Commun., Jun 1982, paper 2G.1.

[5] C. Lu and J. G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representations," IEEE trans. on Commun., Vol.39, No.10, Oct 1991, pp.1511-1515.

[6] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," Proc. of the IEEE, Vol.82,No.6, Jun 1994, pp.857-865.

Table 1: Average bit rates achieved when various approaches are used to encode the chain code sequences.

	Average bit length per link				
	Australia	Africa	S. America	N. America	Europe
1-pass, Non-Markov, Adaptive	2.9579	2.9459	2.9014	2.9479	2.9260
2-pass, Non-Markov, Non-adaptive	2.9545	2.9429	2.8985	2.9463	2.9244
2-pass, Non-Markov, Adaptive	2.9496	2.9363	2.8924	2.9429	2.9215
1-pass, 1st-Markov, Adaptive	1.5471	1.4926	1.5082	1.4765	1.5026
2-pass, 1st-Markov, Non-adaptive	1.5330	1.4779	1.4904	1.4706	1.4950
2-pass, 1st-Markov, Adaptive	1.5195	1.4631	1.4744	1.4611	1.4866
1-pass, 2nd-Markov, Adaptive	1.4280	1.2504	1.3623	1.3209	1.3499
2-pass, 2nd-Markov, Non-adaptive	1.4048	1.2214	1.3414	1.3107	1.3334
2-pass, 2nd-Markov, Adaptive	1.3724	1.1911	1.3026	1.2895	1.3142
1-pass, 3rd-Markov, Adaptive	1.3988	1.1867	1.3239	1.2859	1.3402
2-pass, 3rd-Markov, Non-adaptive	1.3574	1.1367	1.2839	1.2673	1.2947
2-pass, 3rd-Markov, Adaptive	1.2893	1.0815	1.2115	1.2248	1.2543

Table 2: Average bit rates achieved when various approaches are used to encode the chain-difference code sequences.

	Average bit length per link difference				
	Australia	Africa	S. America	N. America	Europe
1-pass, Non-Markov, Adaptive	1.5751	1.5234	1.5661	1.5228	1.5457
2-pass, Non-Markov, Non-adaptive[5]	1.5743	1.5219	1.5649	1.5218	1.5449
2-pass, Non-Markov, Adaptive	1.5718	1.5187	1.5615	1.5202	1.5436
1-pass, 1st-Markov, Adaptive	1.4593	1.2663	1.4120	1.3623	1.3790
2-pass, 1st-Markov, Non-adaptive[5]	1.4528	1.2573	1.4066	1.3604	1.3775
2-pass, 1st-Markov, Adaptive	1.4453	1.2509	1.3977	1.3557	1.3732
1-pass, 2nd-Markov, Adaptive	1.4355	1.2168	1.3864	1.3314	1.3606
2-pass, 2nd-Markov, Non-adaptive[5]	1.4237	1.1883	1.3745	1.3268	1.3550
2-pass, 2nd-Markov, Adaptive	1.4034	1.1735	1.3525	1.3153	1.3448
1-pass, 3rd-Markov, Adaptive	1.4179	1.1887	1.3626	1.3195	1.3546
2-pass, 3rd-Markov, Non-adaptive	1.3985	1.1532	1.3418	1.3079	1.3364
2-pass, 3rd-Markov, Adaptive	1.3547	1.1234	1.2963	1.2838	1.3133