FAST AND EFFICIENT CODING OF LOW ENTROPY SOURCES WITH TWO-SIDED GEOMETRIC DISTRIBUTION

Alexandre Krivoulets

Abstract We propose a method for efficient coding of sources with two-sided geometric distribution (TSGD) and entropy near to or less than 1 bit. This method is based on the binary tree decomposition of source symbols combined with binary arithmetic coding and allows for fast and efficient implementation. It can be used for coding of transform coefficients in low bit-rate video and image compression systems.

Keywords: Entropy coding, image compression, geometric distribution.

1. Introduction

In this paper, we propose a method for coding of sources with probability distribution defined by

$$\begin{cases} P(i) = 1 - \sqrt{\theta}, & i = 0 \\ P(|i|) = \frac{(1-\theta)\theta^{|i|}}{2\sqrt{\theta}}, & i \neq 0 \end{cases} \tag{1}$$

where $i \in Z$ are the source symbols and θ is the distribution parameter. We also assume, that $0 < \theta < 1/3$, i.e., the entropy of the source $H = -\sum_{i \in Z} P(i) \log_2 P(i) < 2.36$ bits (actually, the proposed algorithm can work for sources with any $0 < \theta < 1$, but, as it will be shown later, in terms of speed, it is most effective under this condition).

For example, in image and video coding systems, the transform coefficients after quantization have such a probability distribution. It has been observed [1], that a probabilistic model of the transform coefficients is well approximated by the continuous two-sided geometric distribution

$$f(x) = \frac{\alpha}{2} \exp^{-\alpha|x-\mu|},$$

where α and μ are the distribution parameters. The transform coefficients are usually quantized using an equidistant quantizer with bin width Δ, centered at zero. After such a quantization, the probability distribution of discrete symbols $i \in Z$ is defined as

$$P(i) = \int_{\frac{\Delta(2i-1)}{2}}^{\frac{\Delta(2i+1)}{2}} f(x)dx.$$

This gives formula (1) (deducing this formula, we denoted $\theta = \exp(-\alpha\Delta)$ and set $\mu = 0$, as in practice $\Delta \gg \mu$ and we may neglect it).

A difficulty encountered when designing a method for coding of such source is that while the entropy is near or less than 1 bit per symbol, the alphabet potentially can be quite large. In this case, direct use of Huffman or even arithmetic coding may be inefficient. A number of methods were proposed for coding of sources with the TSGD (see, e.g. [2, 3]), but all of them work poorly with sources of entropy less than 1 bit. To solve this problem, a binary tree decomposition of the source alphabet combined with binary arithmetic coding was proposed for efficient coding of DC and AC coefficients of the DCT in the JPEG image compression standard [4].

In this paper, we propose a new binary decomposition technique for coding of low-entropy sources with the TSGD and describe the method that requires eleven times less memory than the one used in the JPEG while the compression and speed performance is not worse.

2. Binary decomposition of alphabet for sources with TSGD

Binary decomposition of source symbols combined with binary arithmetic coding is a well known technique for coding of m-ary sources [4, 5, 6]. A general idea of source coding using this method is that any proper and complete binary tree with m leaves can be used to represent symbols from an m-ary source with any probability distribution. Then a source symbol is represented by a sequence of binary decisions when passing through this tree from the root to the leaf, corresponding to this symbol. The sequence of decisions can be regarded as a sequence of binary symbols generated by a Markov source modeled by this tree. Each node of the tree corresponds to a state of the source and the tree defines a corresponding directed graph of state transitions. The initial state is defined by the root node.

A binary tree with m leaves has $K = m - 1$ nodes. To each node η_k, $k = 0, 1, \ldots, K-1$, of the decomposition tree there corresponds a parameter q_k. Each parameter, i.e., the probability of a binary symbol (decision) being '0', assumes a value in $[0, 1]$ and specifies a binary probability distribution at the node. These parameters are uniquely defined by the probability distribution of the source symbols. The sequence of binary decisions can be decomposed into K subsequences of statistically independent binary symbols with probability distributions q_k, corresponding to each node. These subsequences can be effectively encoded using binary arithmetic coding. If the statistics of the source is unknown or changing, one can implement an adaptive coding technique and thereby adjust the coding parameters to the probability distribution of the input source.

Binary arithmetic coding is much simpler for hardware and software implementations than an m-ary arithmetic coding. On the other hand, the coder usually has to encode more than one binary event for each input symbol. However, using an appropriate data structure for the decomposition one can gain better compression and speed performance.

The average number of binary coding operations required to code a symbol a from the source alphabet A is

$$\overline{n} = \sum_{a \in A} P(a)n(a),$$

where $n(a)$ is the number of binary decisions required to code a. The minimum $\overline{n}$ is achieved when the tree is a Huffman tree for this source.

Now let us consider the source generating symbols $i \in Z$ with the probability distribution (1). It can readily be shown, that if $\theta \leq 1/3$, then a Huffman tree for this source is the unary tree in which the path from the root to the leaf i can be defined as

$$\begin{cases} \underbrace{bb\ldots b}_{2i-1}\overline{b} \text{ if } i > 0, \\ \underbrace{bb\ldots b}_{2|i|}\overline{b} \text{ if } i \leq 0, \end{cases} \tag{2}$$

where $b \in \{0,1\}$ and $\overline{b} = 0$, if $b = 1$ and $\overline{b} = 1$, if $b = 0$ or, equivalently, as $P(i) = P(-i)$,

$$\begin{cases} \underbrace{bb\ldots b}_{2|i|-1}\overline{b} \text{ if } i < 0, \\ \underbrace{bb\ldots b}_{2i}\overline{b} \text{ if } i \geq 0. \end{cases}$$

The average number of binary coding operations per one source symbol using such tree is

$$\overline{n} = \sum_{i \in Z} P(i)n(i) = 1 + \sqrt{\theta}\left(\frac{3+\theta}{2-2\theta}\right)$$

and if $\theta \to 0$ (i.e., if $\Delta \to \infty$), then $\overline{n} \to 1$. For $\theta < 1/3$, $\overline{n} < 2.44$. A number of multiplication free binary arithmetic codes was proposed (see, e.g., [7, 8]). Thus, the coding of binary decisions can be performed very efficiently.

If the statistics is stored separately at each node, then one have to store and update as many binary distribution parameters as the number of nodes. This number grows linearly with the alphabet size of the input source. However, if some nodes have the same parameter values, then one can store and update only one parameter for all these nodes. For the source (1), parameters only for tree nodes are sufficient to store and thereby we can design an efficient coding technique.

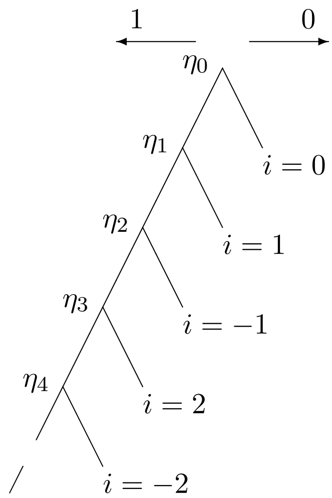

Figure 1. The decomposition tree.

Without loss of generality, assume, that the tree is defined by (2), where $b = 1$ (i.e., $\overline{b} = 0$), see Fig. 1. To code a source symbol i, the encoder sequentially follows through the tree form the node η_0 to the leaf of the node η_{2i-1}, if $i > 0$, or $\eta_{2|i|}$, if $i \leq 0$. It codes '1', if it goes to the descendant node, or '0', if it goes to the leaf. It can readily be verified, that

$$q_k = \begin{cases} 1 - \sqrt{\theta} & k = 0, \\ \frac{1-\theta}{2} & k = 2t - 1, \\ \frac{1-\theta}{1+\theta} & k = 2t, \end{cases}$$

where $t = \{1, 2, \ldots\}$.

Though the decomposition tree has infinite number of nodes, it is sufficient to estimate the parameter at only one node. The parameters for the remaining nodes can be calculated. In order to avoid calculations, it is sufficient to store and update three parameters at the nodes η_0, η_1 and η_2, and use the last two for coding at all odd and even nodes accordingly. If the TSGD source is a Markov source with the number of states S, then the number of storing parameters is only $3S$.

Such a data structure is much simpler and more suitable for the low-entropy TSGD sources than the one proposed for the JPEG compression standard. The decomposition used in the JPEG has 33 parameters for each state S. Hence, the required memory of the proposed method is eleven times less than that of the JPEG for sources with entropy less than 2.36 bits. Also, the average number of binary coding operations per one input symbol is not more than for the JPEG decomposition. This follows from the fact that we use a Huffman decomposition tree.

3. Experimental results

To demonstrate the effectiveness of the proposed method, we employed it for coding of DC and AC quantized coefficients of the DCT using the JPEG scheme and compared it to the JPEG decomposition for compression of nine grayscale images [9]. In our experiments, we used the same QM-binary arithmetic coding as in the JPEG [4] for coding of binary sequences. The results are given in Table 1. The "JPEG" column contains the number of bytes of the JPEG compressed image files excluding header information, markers and stuffed zero bytes. The "New" column contains the file lengths of the coded images using the proposed scheme. Experiments show, that the compression efficiency of the proposed decomposition is not worse and may be even a little better than that of the JPEG, while the complexity and memory requirements are much lower.

4. Conclusions

A new method for coding of low entropy sources with two-sided geometric distribution has been proposed. It requires 11 times less memory than that of the JPEG compression standard while the compression and speed performance are not worse. The method allows for fast and efficient software and hardware implementation and can be used in low bit-rate image and video compression systems for entropy coding of transform coefficients.

Another benefit of the proposed method is that there is no embedded limit on alphabet size, so any value can be encoded and there is no need

Table 1. Comparison of compression performance of the proposed decomposition and the JPEG decomposition for image compression (QP is the quality parameter for the JPEG).

QP	30		50		80	
Method	JPEG	New	JPEG	New	JPEG	New
baloon	**11742**	11769	**16828**	16908	**31956**	32253
barb	30374	**30309**	42418	**42275**	72547	**72220**
barb2	29525	**29513**	41578	**41479**	74226	**73660**
board	**16228**	16246	**23144**	23224	**42035**	42573
boats	21538	**21523**	30303	**30279**	53365	**53339**
girl	22143	**22122**	31224	**31217**	54486	**54377**
gold	23990	**23963**	35116	**35046**	63540	**63122**
hotel	26757	**26629**	37289	**37084**	65358	**65140**
zelda	15297	**15225**	22256	**22173**	**41906**	41911
average	21955	**21922**	31129	**31076**	55491	**55400**

to adjust the algorithm when designing a compression system for images with different number of bits per pixel.

References

[1] R.C. Reininger, J.D. Gibson. Distributions of the two-dimentional DCT coefficients for images. *IEEE Trans. on Comm.* 31(6):835–839, 1983.

[2] R.F. Rice *Some practical universal noisless coding technique.* JPL Publication Tech. Rep. 79-22, Pasadena, California, Mar. 1979.

[3] N. Merhav, G. Seroussi, and M.J. Weinberger. Optimal prefix codes for two-sided geometric distributions. *IEEE Trans. Inform. Theory* 46(1):121-135, Jan. 2000.

[4] W.B. Pennebaker, J.L. Mitchell. *JPEG – Still Image Data Compression Standard.* New York: Van Nostrand Reinhold, 1993.

[5] G.G. Langdon and J.J. Rissanen. A double adaptive file compression algorithm. *IEEE Trans. Comm.* 31(11):1253–1255, 1983.

[6] P.G. Howard. *The design and analysis of efficient lossless data compression systems.* Report CS-93-28, Brown Uneversity, Providence, Rhode Island, 1993.

[7] G.G. Langdon and J.J. Rissanen. A simple general binary source coding. *IEEE Trans. Inf. Theory*, 28(5):800–803, 1982.

[8] W.B. Pennebaker, J.L. Mitchell, G.G. Langdon and R.B. Arps. An overview of the basic principles of the Q-coder adaptive binary arithmetic coder. *IBM J. Res. Develop.*, 32(6):717–726, 1988.

[9] ftp://ftp.csd.uwo.ca/pub/from_wu/images .