VARIABLE-TO-BLOCK-WITH-PREFIX SOURCE CODING TECHNIQUE

BACKGROUND OF THE INVENTION

Source coding or data compression techniques are useful in reducing the volume of data generated by a source for the purpose of transmission over a channel or storage. Economies are realized when the volume of data to be transmitted or stored is decreased. There are two kinds of data compression techniques; information preserving and information degrading. A process is information preserving if all the information present before coding can be regenerated after coding. Conversely, a process is information degrading if it is irreversible in the sense that the original information cannot be regenerated exactly after the process is performed. An information degrading process can be used when the user of the information is not interested in all the information generated by the source. If the user is only selectively interested in the information, the information degrading source coding process can reflect that selectivity. Also, if the user is willing to accept a fidelity criterion, the data may be degraded within certain limits and still remain useful. In general, more data compression can be attained with information degrading processes than with information preserving processes.

Source coding systems can be classified as statistical or ad hoc, depending on whether or not the system is in some sense optimal in terms of Shannon's noiseless coding theorum. A final classification of source coding systems is the following: if the system accepts digits from the source in fixed length blocks and then generates a variable length sequence of encoded digits, the system is "block-to-variable"; if the system accepts a variable length sequence of digits from the source and generates a fixed length block of digits the system is "variable-to-block."

The first optimal coding scheme to be developed was Huffman coding. This technique is block-to-variable. Huffman coding specifies the optimal coding for M messages with probabilities $p_1, p_2, \ldots, p_m$, where p is the probability that a "1" is emitted. The least probable message is assigned a code word containing the longest sequence of bits and more probable messages are assigned code words of shorter length. When Huffman coding is applied to a binary memoryless source, the source sequence must be broken up into blocks N bits long. Each block then contains one of $M = 2^N$ possible messages. As N gets large it becomes impractical to calculate the probability of each message and to generate the corresponding code word. However, Huffman coding is only optimal as N approaches infinity. Furthermore, this coding scheme requires knowledge of the source statistics and the code words must be computed a priori. The code words must be stored in the memory of the encoder and a table lock-up performed on each block. The memory requirements grow exponentially with N. In short, complexity and required knowledge of the source statistics detract from the desirability of Huffman coding.

Run length coding is an ad hoc coding scheme that works well when p is either very small or very large. In its simplest implementation the number of consecutive zeros is counted. When the first "1" is encountered, the number of zeros is transmitted as a block of binary digits. Thus, run length coding is variable-to-block and this fact simplifies the hardware considerably. Run length coding is, however, limited by the fact that it is reasonably efficient only when p is very small or very large and even in these cases it is not optimal.

The Schalkwijk algorithm is based on the ranking of binary sequences of length N and weight W. For a binary memoryless source characterized by p, as N approaches infinity, we expect to get a weight, $W = pN$, with a probability equal to 1. The basic idea is to rank all sequences of length N and containing $W = pN$ ones. Thus, the binary encoding of the sequence is its rank in binary form. Schalkwijk has shown that as N approaches infinity this technique is optimal. Since there are $\binom{N}{W}$ possible sequences, i.e.,

$$\frac{N!}{(N-W)!W!},$$

each member can be encoded in a word of $\log_2\binom{N}{W}$ bits. Thus, the compression ratio achieved is

$$\frac{N}{\log_2\binom{N}{W}},$$

which approaches the theoretical limit $H(W/N)^{-1}$ as N approaches infinity. Mathematically, the rank of the sequence is given by,

$$i(t) = \sum_{K=1}^{N} t_K \binom{N-K}{W-K} \text{ where } t = (t_1, t_2, \ldots, t_N) \epsilon T(N, W), \text{ and}$$

where T(N,W) = the set of all binary sequences of length N and weight W, where $t_K \epsilon [0,1] = K^{th}$ member of t, and where

$$W_K = \sum_{i=K}^{N} t_i.$$

An understanding of the above formula can be facilitated by visualizing it as a random walk through Pascal's triangle determined by the bits in the input sequence, starting at the W^{th} position of the N^{th} row, i.e., at the entry corresponding to $\binom{N}{W}$ and terminating at the apex. FIG. **1** illustrates the computation for the sequence 010100.

Given the rank $i(t) = 8$, the original sequence t can be reconstructed by using the following algorithm. Start at $\binom{6}{2} = 15$ in FIG. **1**. The rank **8** of the sequence is less than the number 10, at a single step in the X-direction from the current starting point **15**. Move one step in the X-direction, toward **10** and record a 0. The rank 8 of the sequence t is not less than the number 6, at a single step in the X-direction from the current starting point 10. Move one step in the Y-direction, towards **4**; substract the number 6 used in the comparison from the rank 8 of the sequence t, giving a new rank $8 - 6 = 2$ and a record a 1 giving 01. The current rank 2 of the sequence t is less than the number 3 at a single step in the X-direction from the current starting point **4**. Move one step in the X-direction towards **3** and record a 0, giving 010. The current rank **2** of the sequence is not less than the number 2 at a single step in the X-direction from the current starting point 3. Move one step in the Y-direction towards 1; subtract the number 2 used in the comparison from the current rank 2 of the sequence t, giving a new rank $2 - 2 = 0$ and record a 1, giving 0101. The current rank of the sequence t is now 0. Thus the last