SELF SYNCHRONISING T-CODES TO REPLACE HUFFMAN CODES

Gavin R Higgie, Electrical and Electronic Engineering, University of Auckland, Private Bag 92019, Auckland, New Zealand

Abstract - This paper describes recent work on the T-Codes, which are a new class of variable length codes with superlative self-synchronizing properties. The T-Code construction algorithm is outlined, and it is shown that in situations where codeword synchronization is important the T-Codes can be used instead of Huffman codes, giving excellent self-synchronizing properties without sacrificing coding efficiency.

Background

When corruption occurs in a stream of data which is coded with variable length codes, the decoder can lose track of where codeword boundaries are located in the data stream, and so the effect of the corruption can extend over a large number of received symbols. Variable length code sets can be chosen such that the receiver is able to determine the correct location of these codeword boundaries relatively quickly after a corruption, but this is usually done by choosing codes in which certain bit sequences occur only at the end of codewords. The receiver must look for these special sequences in the received data stream. In some cases it is possible to choose codes which will self-synchronize as a result of the normal decoding process, but these are relatively difficult to find.

T-Codes

The original discovery of the T-Codes was published in 1984 by Titchener [1]. This work gave an algorithm for generating families of T-Code code sets, and showed that all code sets generated in accordance with this algorithm are self-synchronizing by nature. These properties are not derived from having specific synchronizing bit sequences, but rather the synchronizing information is spread throughout the code as an inherent part of its construction. The construction of the codes is such that when codeword synchronization is lost as a result of a corruption, the receiver will automatically re-synchronize on a subsequent codeword boundary without any special measures being taken. This will happen even when the loss or corruption is not recognized as such. When a data loss or corruption is known to have occurred, an algorithm is available for the receiver to determine the point at which subsequent codeword synchronization is re-established.

T-Code Construction Algorithm

The T-Code construction algorithm is very simple. Code sets are constructed by *augmenting* lower level T-Code code sets, with the lowest level being the code set 0 and 1. The augmentation process consists of writing out a list with two copies of the lower level code set, and then sacrificing a codeword from the first half of the list and using it as a prefix for every codeword in the second half of the list. This produces a new code set which has nearly twice the number of codewords of the lower level code set. An example of this process is given in Table 1.

Level 0	Level 1 Prefix 0	Level 2 Prefix 01
0	--> -	-
1	1	1
	00	00
	01	--> -
		-
		011
		0100
		0101

Table 1
T-Code Construction Algorithm

T-Code Synchronization Properties

Titchener showed that every T-Code will have self-synchronizing properties, with typical synchronizing delays of a few codewords, but it was not until the work by Higgie [2] showed that these self synchronizing properties are available without loss of coding efficiency that the significance of the codes became evident. Any codeword from a code set can be used as the prefix to produce the augmented code set at the next level. The length of the prefix chosen affects the codeword length distribution of the next level code set, so by careful choice of prefix lengths it is possible to produce T-Codes which match the codeword length distribution required for efficiently coding any particular information source (i.e. effectively the same codeword length distribution as a Huffman code designed for the source).

Higgie's work [2] also showed that the T-Codes which give maximum efficiency for any particular information source generally include at least one which has an average synchronization delay of around 1.5 codewords.

Current Research Activity

The work reported in the paper by Higgie [2] used Monte simulation techniques to show that it is possible to choose an efficient and rapidly synchronizing T-Code for any particular application. These simulations also showed that not all T-Codes are equal in their synchronization performance and that the task of choosing the best T-Code for a particular application is not a trivial one. Attempts at justifying why some T-Codes are better than others have recently led to a new technique for theoretically determining the average synchronization delay of T-Codes when they are used efficiently. This technique offers several advantages over the previously used Monte Carlo techniques, and provides insight into how the T-Codes achieve their enviable synchronizing properties.

The theoretical technique is now being used in calculating a database of the T-Codes which have the best synchronizing performance. It is hoped that this database will be useful in enabling a user with a particular information source to choose a T-Code which will be as efficient as a Huffman code designed for the source, but with an average synchronization delay of about 1.5 codewords.

Current research is also focusing on the use of T-Codes in FAX machines and in the JPEG image compression standard, particularly with respect to transmitting images in these formats over mobile radio channels. This is only one of many potential application areas, as T-Codes can be used to advantage in any situation where the probability of data corruption is high enough to make the use of non-synchronizing or poorly synchronizing Huffman codes difficult.

Conclusion

The T-Code generation algorithm has been demonstrated to provide variable length code sets which have both the desirable properties of coding efficiency and rapid self-synchronization. For any particular information source, properly chosen code sets can typically offer average synchronization delays of 1.5 codewords without sacrificing coding efficiency compared to that obtained with a Huffman code designed for the source. This means that it is now possible to use variable length codes in applications where the probability of corruption is high and the problems of codeword synchronization have previous excluded their use.

References

[1] Titchener, M.R.: (1984) 'Digital encoding by means of new T-Codes to provide improved data synchronization and message integrity', Proc IEE, Pt E, Computers & Digital Techniques, Vol 131, (4), pp151-153

[2] Higgie, G.R. and Williamson, A.G.,: (1990) 'Properties of Low Augmentation Level T-codes', Proc. IEE, Pt E, Computers & Digital Techniques., Vol 137 pp129-132

Authorized licensed use limited to: The University of Auckland. Downloaded on November 2, 2008 at 21:41 from IEEE Xplore. Restrictions apply.