A NOVEL DECODING ALGORITHM FOR REVERSIBLE VARIABLE LENGTH CODES BASED ON THE MASSEY METRIC

M. A. HOSANY and M. Z. BOCUS
Faculty of Engineering, Dept. of Electrical and Electronic Engineering,
University of Mauritius, Reduit, Mauritius.
m.hosany@uom.ac.mu
bocusm@uom.ac.mu

Abstract

The use of Variable Length Codes (VLC) to increase the coding efficiency based on the statistical characteristics of the data has long been employed in digital communication systems. However the catastrophic degradation of VLC encoded data which results from bit errors has led to the development of reversible variable length codes (RVLC) that can be decoded bidirectionally thereby having more efficient error resilient capabilities. Up to this point, no soft decoding algorithm for RVLC has been devised that effectively exploit the bidirectional characteristics of such codes. This paper introduces a novel decoding algorithm for reversible variable length codes that make use of the Massey metric while limiting the complexity of a purely unidirectional sequential decoder.

1. Introduction

Considerable attention has recently been paid to improve performance of practical communication systems. One such effort has been the use of reversible variable length codes at the coding stage. The ability of reversible variable length codes to be decoded in either direction and their relatively good coding efficiency have lead to a lot of research in this area [1-7]. The first paper to study the existence of such codes is by Takishima *et al* [1]. They introduced the main idea of how RVLC, both symmetrical and asymmetrical, could be constructed from the optimal Huffman codes. Later papers mainly tried to improve upon the algorithms that were introduced so as to achieve lower average code length or increasing the error-resilience.

The better error-resilience of RVLC is what leads to its adoption by the H.263+ and the MPEG-4 standard and as a way of encoding data in low-bit rate wireless environment. Even when employing hard-decision decoding, RVLC lead to better error-resilience than VLCs.

To increase performance of common variable length codes such as the Huffman code, several decoding algorithms exist, namely MAP decoding and sequential decoding among others [6, 8-11]. Each of these methods has its respective pros and cons.

Though some papers introduced RVLC decoding methods that could be employed in applications such as the MPEG-4 standard [12], no paper concretely showed how soft-values could be employed with RVLC. The methods proposed by J. Webb ensured memory efficient partitioning of the code table that results in quicker access.

Albeit the direct application of these algorithms to the decoding of RVLC (which is a subset of VLC) is possible, so doing would decrease performance as well as adding unwanted increase in complexity at the decoder side. This paper investigates how the sequential decoding algorithm presented by Massey [8] can be extended to better suit the needs of reversible codewords.

2. Sequential Decoding

The concept of sequential decoding was introduced as a method of maximum likelihood sequence estimation with lower memory requirements and computationally less complex than the Viterbi algorithm by Wozencraft in 1961.

In [7], Fano showed that the branch metric that would lead to maximum likelihood decoding for fixed length codes is

$$\lambda_m(y) = \sum_{i=1}^{l_m} \left[\log \frac{P(y_i \left| x_{m,i} \right.)}{Po(y_i)} \right] - R \qquad (1)$$

where

x transmitted sequence

y received bit sequence

l_m length in bits of each fixed-length codeword

m mth codeword

$Po(yi) = \sum_{x_j} P(x_j) P(y_i \left| x_j \right.)$ - probability distribution of the output given the input symbols averaged over all input symbols.

R bias term equal to code rate

$x_{m,i}$ the i[th] bit in the m[th] codeword

The idea behind Fano algorithm is to choose that codeword that would maximise the metric given in equation (1). The latter algorithm can be stated as follows:
Each code is represented as a tree where each branch corresponds to a codeword. The number of branches stemming from any node is therefore less than or equal to N the total number of codewords in the code. The decoder calculates the branch metric for each node and chooses the best codeword corresponding to the highest branch metric, $\lambda_m(y)$. If the metric is larger than a preset threshold the branch is added to the path and the algorithm continues for new nodes. However if metric is below a certain threshold, the decoder moves back one node, decreases the threshold and retries the best branch. This iterative tracing leads to most likely path.

In the case of Variable Length Codes, the above algorithm cannot be used directly. James Massey showed in [8] that the Fano metric can be extended for the case of VLC. In his paper, the branch metric was showed to be

$$\lambda_m(y) = \sum_{i=N_b+1}^{N_b+l_m} \left[\log \frac{P(y_i | x_{m,(i-N_b)})}{P_o(y_i)} + \frac{1}{l_m} \log P_m \right] \quad (2)$$

where

N_b is the current position in the bit stream, and the remaining notations are similar to those of the Fano metric.

Massey replaced the code rate R by the weighed probability of the codeword occurrence. In the case of variable length codes, the received sequence length limits the number of possible paths that would possibly lead to such a state i.e. only a certain number of transitions on a typical trellis diagram can lead to a given bit length. This limits the number of possible paths followed at the encoder side.
Yet, given large VLC code table with large number of symbols, the number of possible paths that yields a certain state can still be very considerable. If the decoder need to keep track of all sets of paths followed, memory requirements might be quite significant.

The next section introduces the novel algorithm for Reversible Variable Length code decoding based on the Massey metric.

3. Proposed Decoding algorithm

Taking the length of the received bit sequence to limit the possible paths on a trellis is a very attractive approach for variable length code decoding. However this would require the consideration of possible paths prior to decoding and moving back in case the branch metric is lower than threshold, implying certain overhead.
To demonstrate the performance of decoding schemes for VLC, simulations have been performed on the Matlab software. Having this software consider the possible paths at the beginning of the decoding process resulted in a certain delay. We came up with the new scheme as a method of reducing the memory requirements of the system as well as complexity starting with the Massey metric as building stone.

In the proposed method, the decoder does not limit the paths on the trellis; the decoder dynamically chooses the next most probable symbols. If at the end the decoder detects that it has ended up with a wrong path, an error flag is set. A detailed description of the algorithm is given below.

The steps to be followed for decoding the received bit stream bidirectionally, making use of the metric introduced by Massey in [8] are:

1. Making use of the Massey metric, find the best symbol sequence for the received bit stream, or bits within resynchronisation markers, and save to a temporary set, for e.g. *'decoded_forward'*.
2. If an error is detected such as the number of bits left after a certain amount of decoding has been performed is less than the length of the smallest codeword in the symbol set, an error variable is asserted. For example, *'fwd_err'* is set to 1 is case of error otherwise it is set to zero.
3. Perform step (1) and (2) but with the bit stream analysed in the reverse direction. Symbols decoded are stored in the temporary set *'decoded_backward'* and *'bwd_err'* is set to 1 in case of error detection.
4. If the number of symbols in *'decoded_forward'* and *'decoded_backward'* are equal and both *'fwd_err'* and *'bwd_err'* have the same value of nil, the actually decoded symbol sequence is that corresponding symbol with a higher metric value in either temporary set. Otherwise proceed to step (5).
5. Through observation of the Massey decoding algorithm it has been observed that the metric results in values less than a threshold value whenever a bit in the

received sequence differ from the symbol` codeword under investigation. This value is chosen as the threshold. All symbols in *decoded_forward* till the one that has a metric value below the threshold are assigned to the actually decoded sequence. The same process is performed for *decoded_backward* and result stored in a temporary string.

6. After step (5) it results that certain bits in the received sequence remains undecoded. The latter is reanalysed using either step (1) or (2). If the number of bits after a certain period of decoding is less than the length of smallest codeword, i.e. an error has occurred in the process, move back one symbol and find that symbol that satisfy length constraint as well as having the next highest metric value. The latter decoded symbols are appended to the actually decoded bit sequence from (5). The temporary string of symbol from *decoded_backward* is concatenated to end the decoding process.

Note on algorithm:

The proposed algorithm considerably reduces the complexity of the algorithm presented by Massey in that the number of symbols to be back-tracked is reduced to only one. It should be observe that this method need not necessarily yield the correct result.

One important thing that should be noted about the proposed scheme is that it will not always give the best performance. Only one backtracking is performed in case an error is observed. The idea behind the scheme is to reduce the complexity of Massey`s algorithm without causing too much loss in performance.

Ideally, there should be continuous moving back previously decoded symbol (i.e. moving one step back the trellis) to obtain the most likely chosen path in case the metric value falls below a certain threshold. Obviously this would result in large decoding delay.

With the proposed scheme, given that the received bit stream must be decoded in either direction, independently of one another, it is very simple to implement such decoders in practice. Forward and backward decoding can be performed simultaneously (reducing delay) with hardware similar to those presented in [13]. Comparison of metric value would not require too complex hardware either – probably EX-OR and shift-register operations.

A further notice should be made to the reversing of the codebook in case the code is an asymmetrical one before starting the decoding process in the reverse direction.

The next part shows the results obtained through simulations.

4. Simulation Results

To analyse the performance of the proposed decoding scheme, several RVLCs have been considered namely those of [1-3]. Files to be encoded/decoded were taken from the Canterbury corpus [14].

Data were transmitted in packets of 20 symbols each. This would be equivalent to putting resynchronisation after every 20 symbols. The decoder side is not informed of the number of transmitted symbols. If this value is transmitted as further side information, further restriction might be placed on the decoding process at the expense of more overhead.

4.1. Investigating performance with symmetrical RVLC

The performance of the proposed scheme is first considered while symmetrical codewords are employed. Codes for this purpose are generated based on the algorithms of Takishima *et al* [1].

The graphs of error rate against Eb/No values are shown in the figures that follow:

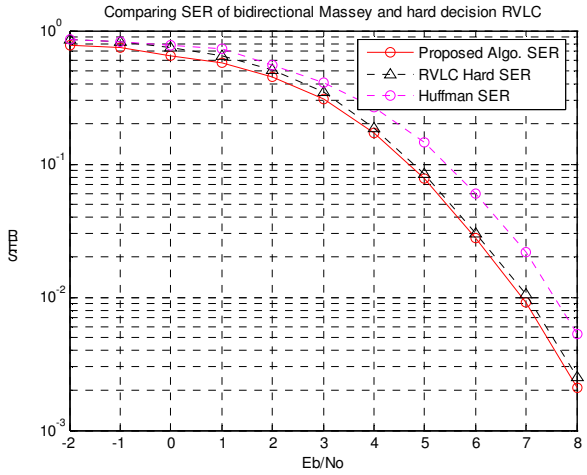

Figure 1: SER comparison of decoding schemes for RVLC and VLC

It is very obvious from the above plot that RVLC lead to far better performance at all signal to noise ratio values compared to the optimal Huffman code, independent of the decoding algorithm used.

To better demonstrate the gain brought about by the proposed decoding scheme, the Bit Error Rate for the two RVLC decoding algorithms is presented in figure 2.

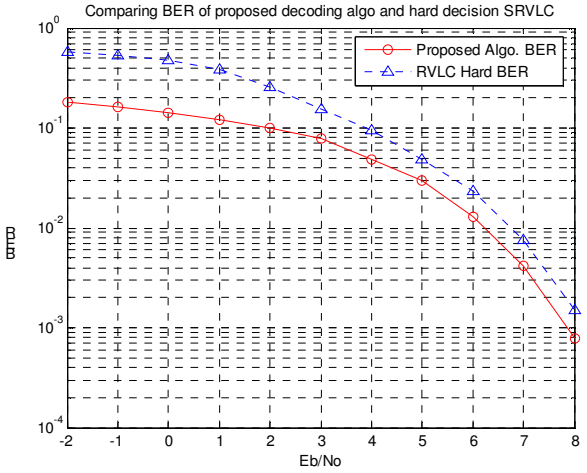

Figure 2: BER comparison of decoding schemes for RVLC

Especially at low signal to noise ratio values, the novel algorithm based on the Massey metric leads to significant gain. As can be observed, for a probability of around 10^{-1}, a gain of approximately 2dB is achieved.

4.2. Investigating performance with asymmetrical RVLC

Though symmetrical codewords have advantage of lower memory requirements- only one table for both forward and backward decoding, they have higher average codeword length as compared to most asymmetrical code. This is one of the main reasons of the widespread use of ARVLC code instead of SRVLC.

Figure 3 and 4 show the SER and BER plots with code generated based on Tsai and Wu`s [2] generic code construction method.

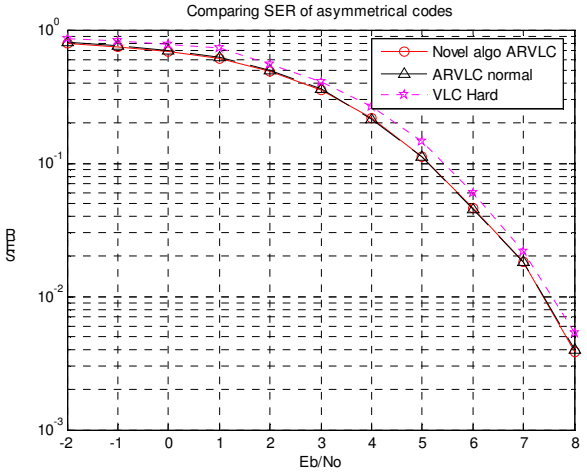

Figure 3: SER plot of asymmetrical code

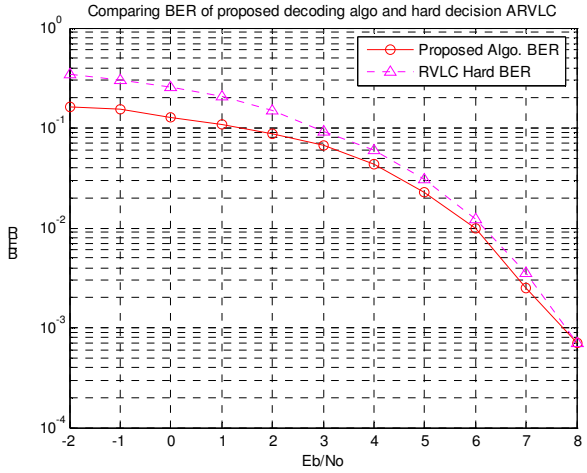

Figure 4: BER plot of asymmetrical code

Interpretation of results

The most striking observation is again the better performance of the RVLC code whichever decoding method has been chosen compared to the Huffman code.

However, as opposed to the larger performance gain of the novel algorithm in the case of the symmetrical code, the above results show only a very minor improvement. The reason for this is the poorer free distance properties of the asymmetrical code generated by Tsai and Wu`s method. The latter has a minimum free distance of only 1 while most symmetrical codes have a value of at least 2. The disadvantage of having a poor free distance property is that a bit in error may change the received bit stream in such a way that what was supposed to be a detectable error turn out to be another valid codeword. In other words the decoder might not notice the error until much later. Even at that point, recovering through continuous backtracking might not lead to adequate gain with respect to effort expended.

Such situations do also arise with symmetrical code with a free distance of 2. However these are more easily spotted and loss of synchronisation minimised.

The following discussion shows the results while employing the fix-free code generated by the procedures of [3]. Though the authors demonstrated that their algorithm leads to highest compression, error recoverability of the efficient fix-free code in case of channel errors is an equally important trait that demands enquiry.

The corresponding plots are shown below.

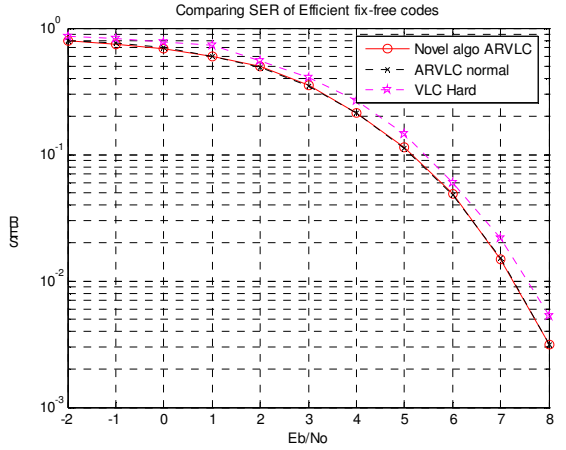

Figure 5: SER plot of fix-free code of Lakovic and Villasenor

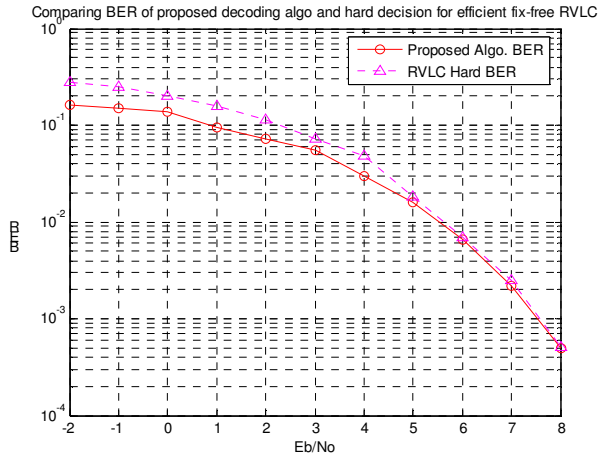

Figure 6: BER plot of fix-free code of Lakovic and Villasenor

The performance gain when using the proposed decoding algorithm is in most cases nil. This is again attributed to poor free distance. The plot also indicate that the free distance property of Tsai and Wu`s ARVLC is better than the one analysed here.

If such a code is to be employed for encoding a sequence, the use of table look-up for decoding would be most appropriate since the added complexity of using the proposed decoding scheme is far too large with respect to the data recovery gain.

4.3. Complexity analysis of decoding algorithm

To examine the complexity of the proposed decoding algorithm, a counter has been introduced in the program to count the required number of computations while encoding and decoding files of different sizes.
The results obtained are shown graphically below:

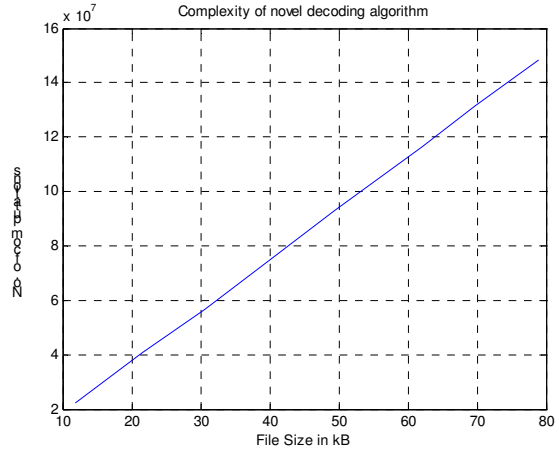

Figure 7: Plot of number of computations against file size for novel algo.

For the purpose of complexity analysis, the signal to noise ratio has been kept constant and only text file size has been varied. The counter in the program has been included for all the decisions and calulations. In practice however, where the system is implemented in hardware and DSPs, the complexity would be probably less than in the simulation program. Certain decisions regarding the decoding process can be performed in parallel, thus reducing delay due to computations.

As is demonstrated in figure 7, the number of computations is directly proportional to the file size. This is an expected result since as the number of symbols transmitted increases, so does the calculations to decide upon the decoded symbol.

In the above case, a signal to noise ratio of 4 dB has been used throughout. Figure 8 shows the graph of number of computations against the channel SNR in dB.

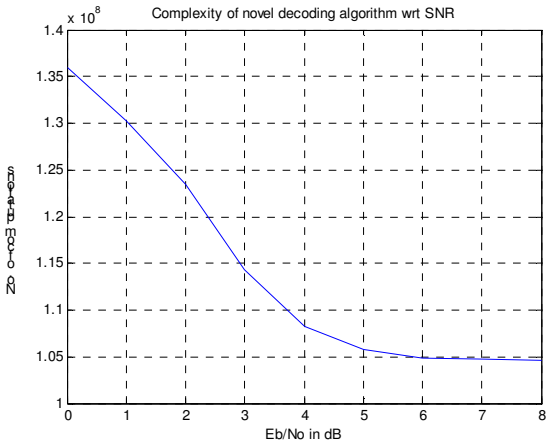

Figure 8: Plot of number of computations against Eb/No for novel algo.

File size for the above case is kept constant to 50 kB. As the Eb/No increases (less noise), the number of computations tend to an almost fixed value.

5. Conclusion

The conclusion to be drawn from the observations is that the proposed decoding method must be avoided in case of code with inappropriate free distances. The increase in computation at the receiver side might not result in sufficient improvement compared to the table look-up operations. This is not the case with the symmetrical code however.

The compression efficiency of asymmetrical RVLC obviously outweighs that of SRVLC. This leads to a serious trade-off for system implementation where both lower bit transmission and better error recoverability are desired. The chosen approach relies mainly on the system designers` definition of an acceptable system. Requirements for the transmission of video data over a wireless channel do not coincide with less time constraint text transmission.
As the free distance increases, the effectiveness of the novel algorithm is guaranteed to increase. The authors of [15, 16] discussed the construction of robust RVLCs with good free-distance properties. These codes are likely to provide high performance when applied to the proposed decoding scheme.

6. References

1. Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes", [J]IEEE Trans. Communication, volume 43, Feb-Apr. 1995: 158-162
2. C.W. Tsai and J.L. Wu, "On constructing the Huffman-code based reversible Variable Length Codes", *IEEE Trans. Commun*, volume 49, pp. 1506-1509, September 2001.
3. K. Lakovic and J. Villasenor, "An algorithm for efficient fix-free codes", *IEEE Comm. Lett., volume 7*, pp 391-393, August 2003.
4. K. Lakovic and J. Villasenor, "On design of error-correcting reversible variable length codes", *IEEE Comm. Lett., volume 6*, August 2002.
5. J. Wen and J. Villasenor, "Reversible Variable Length Codes for efficient and robust image and video coding", in *Data Compression conf.*, 1998, pp. 471-480
6. J. Wen and J. Villasenor, "Utilizing soft information in decoding Variable Length Codes", in *Proc. IEEE Data Compression Conference*, Snowbird, UT, Mar. 1999, pp. 131-139.
7. Robert M. Fano, "A Heuristic Discussion of Probabilistic Decoding", *IEEE Trans. Information Theory*, vol. IT-9, Apr. 1963, pp. 64-73.
8. J.L. Massey, "Variable Length codes and the Fano metric", *IEEE Trans. Inform. Theory,* vol. IT-18, Jan. 1972.
9. M. Bystrom, S. Kaiser, A. Kopansky, "Soft source decoding with applications", *IEEE Transactions on Circuits and systems for video technology*, vol. 11, No. 10, Oct. 2001.
10. J. Wen and Villasenor, "Soft-Input Soft-Output Decoding of variable Length Codes", *IEEE Trans. Communications*, vol. 50, No. 5, May 2002.
11. M. Bystrom and S. Kaiser," Soft decoding of Variable Length Codes".
12. Webb, Jennifer, "Efficient Table Access for reversible Variable Length Decoding", *IEEE Trans*., circuits and systems for video technology 2001.
13. Lingfeng Li, Yun He, "Decoding Algorithms for RVLC/VLC and their LSI architecture", State key Lab. On microwave & Digital Comm., Dept. of Electrical and Electronic Engineering, Tsinghua University.
14. http://corpus.canterburry.ac.nz
15. W. Jeong, Y. Yoon, Y. Ho, "Design of robust reversible variable length codes using the property of free distance", Samsung Electronics Co. LTD
16. W. Jeong, Y. Yoon, Y. Ho, "Design of asymmetrical reversible variable length codes and the comparison of their robustness", Dept. Of Information and Communications, Kwangju Institute of Science and Technology.