Joint Source-Channel Coding Using Variable-Length Error-Correcting Codes

Prof. Maher K. M. AlAzawi
University Of Al – Mustansiriya
College Of Engineering
Electrical Engineering Department

Assist. Lect. Suha Kassim Hadi
University Of Al – Mustansiriya
College Of Engineering
Electrical Engineering Department

Abstract

This paper explores the performance of both source coding and error-correction in a single coding step. Such joint source-channel coding is done here using codewords of varying lengths that have particular distance properties with each other, which allows error detection and correction. Hence, these codes are called Variable-Length Error-Correcting (VLEC) codes. These codes exhibit memory like convolutional codes but spatial memory according to the variable-length nature. These (VLEC) codes are used in image coding and show sub error performance over variable length source coding such as Huffman code.

Keywords *: Variable-Length Error-Correcting(VLEC) Codes, Joint Source-Channel Coding, Greedy algorithm, Majority voting algorithm, Heuristic algorithm, Modified Viterbi algorithm.*

الخلاصة

يعرض هذا البحث أمكانية أنجاز كلاً من تشفير ألمصدر و تصحيح ألخطأ بخطوة واحدة. تُنجز مثل هذه ألعملية هنا باستخدام كلمات مشفرة مختلفة ألأطوال, حيثُ لها خصائص مسافة معينة مع بعضها البعض والتي تسمح باكتشاف وتصحيح الخطأ. نتيجةً لذلك فأن هذه الشفرات تُسمى (شفرات تصحيح الخطأ المختلفة الطول)(VLEC codes) .

ستُبين هذه ألأطروحة أن هذه الشفرات لها ذاكرة تشبه ذاكرة الشفرات المطوية(convolution codes) ولكن لذاكرة معينة تعود إلى طبيعة الطول المختلف.أستخدمت هذه الشفرات في تشفير الصورة وقد اظهرت أداء مختزل للخطا أكثر من شفرات المصدر المختلفة الطول مثل (Huffman code).

1. Introduction

According to the manner in which redundancy is added to amassage, error-correcting code can be divided into two classes: block and convolutional. Block codes process the information on block–by-block basis, treating each block of information bits independently from others. In other words, block coding is a memory less operation, in the sense that codewords are independent from each other. In contrast, the output of convolutional encoder depends not only on the current input information, but also on previous inputs or outputs, either on a block-by-block or on a bit-by-bit basis [1].

In this paper a class of error-correcting codes is examined, which is called Variable-Length Error-Correcting (VLEC) code. As the name implies, the main difference between these codes and the standard block and convolutional codes is the fact that the codewords are of variable length. The codes investigated here are similar to block codes in the respect that each codeword is mapped onto a given set of information symbols irrespective of the previous inputs. However, their main characteristics are very similar to those of convolutional codes. This similarity is brought about by the fact that the position of any codeword within the encoded message depends on the previously occurring codewords and hence VLEC codes exhibit a form of spatial memory.

The paper is organized, as follows: section two presents the fundamentals of VLEC codes with basic concepts of this code, and the most important parameters that may affect the performance of this code. Section three gives the heuristic construction algorithm for VLEC codes. The VLEC codes decoder is described in section four, with a modified version of Viterbi decoding algorithm. In section five, VLEC code is used in a practical application such as image coding, and compared with variable-length source coder such as Huffman code. Finally, section six represents the conclusions.

2. Variable-Length Error-Correcting Code

2.1 Basic Concepts of Variable-Length Error-Correcting Codes

Let X be a code alphabet with cardinality assumed to be equal to 2, that means the code will be binary, although the results may easily be extended to the general case. A finite sequence $x = x_1x_2x_3\ldots x_l$ of code symbols is called a word over X of length $|x| = l$, where $x_i \in$ X, for all i =1,2,….,l. A set C of words is called a code.

Let the code C have s codewords { $c_1, c_2, \ldots, c_s$} and let $l_i = |c_i|$ and $p(c_i)$ denote the length and the probability of occurrence of data source symbol mapped into word $c_i = (c_{i_1} c_{i_2} \ldots . c_{i_{l_i}})$, i=1,2,…..,s. Without loss of generality, assume that, $l_1 \le l_2 \le \ldots\ldots \le l_s$. Further, let σ denote the number of different codeword lengths in the code C and let these lengths be $L_1, L_2, \ldots . L_\sigma$, where $L_1 < L_2 < \ldots\ldots < L_\sigma$.

Let the number of codewords with length L_i be s_i, and the number of codewords with length less than L_i be $\hat{s}_i$, i.e. $s = \sum_{i-1}^{\sigma} s_i$. (s_1 @ L_1, d_{b1}; s_2 @ L_2, d_{b2}; …….. ; s_σ @ L_σ, $d_{b\sigma}$; d_{bmin}, d_{cmin}) should be used to denote such a code [2], [3], where d_{bi}, d_{bmin} and d_{cmin} (are defined later).

Let $f_i = c_{i1}c_{i2}\ldots\ldots\ldots c_{in}$, $c_{ij} \in C \ \forall \ j= 1,2,\ldots\ldots,n$ be a concatenation of n words of VLEC code C. The set $F_N = \{f_i : | f_i | = N\}$ is called the extended code of the variable-length error- correcting code C of order N [4].

Let A be a memory–less data source with s source symbols $\{a_1, a_2, \ldots a_s\}$, each with probability of occurrence $p(a_i)$, $i = 1,2,\ldots\ldots,s$, with $\sum_{i=1}^{s} p(a_i) =1$. Without loss of generality, assume that $p(a_1) \geq p(a_2) \geq \ldots\ldots \geq p(a_s)$.The source A is encoded using code C by mapping symbol a_i to codeword c_i for all $i=1,2,\ldots,s$. In this case, the average codeword length is given by:

$$L_{average} = \sum_{i=1}^{s} l_i \, p(a_i) \qquad (1)$$

2.2 Some Parameters of Variable-Length Error-Correcting Codes

Variable-length error-correcting code will be considered to be trellis codes, their "memory" arising not from some storage element, but from the spatial information. Consequently, the properties of VLEC codes should be similar to those of trellis codes or convolution codes.

Definition1: the hamming weight W(c) of a word c is the number of non- zero symbols in c. The hamming distance $H(c_1, c_2)$ between two words c_1 and c_2 of equal length is the number of position in which c_1 and c_2 differ [2].

Definition2: the minimum block distance b_k associated to the codeword length L_k of a code C is defined as the minimum hamming distance between all distance codewords of C which have the same length (with length L_k) [3], [4]

$$b_k = \min \{h (c_i , c_j) \} \qquad (2)$$

where $(c_i , c_j) \in C, i \neq j$ and $| c_i | = | c_j | = L_k$.

There are σ different minimum block distances, one for each different codeword length. However, if for some length L_k there is only one codeword, i.e. $s_k =1$, then in this case the minimum block distance for length L_k is undefined. The overall minimum block distance, b_{min}, of VLEC code C is defined as the minimum value of block distance b_k over all $k = 1,2,\ldots..\sigma$.

$$b_{min} = \min\ b_k \qquad (3)$$

$$1 \le k \le \sigma$$

Definition3: the diverge distance $d(c_i, c_j)$ between two codewords of unequal lengths $|c_i|$ and $|c_j|$ of a code C is defined as the Hamming distance between the l-length prefixes codewords c_i and c_j, with $l = \min\{|c_i|, |c_j|\}$ [3], [4], [5].

$$d(c_i, c_j) = h(c_{i1}c_{i2}\ldots\ldots c_{il}, c_{j1}c_{j2}\ldots\ldots c_{jl}) \qquad (4)$$

The minimum diverge distance of the variable-length error-correcting code C, d_{min}, is the minimum of all the diverge distances between all possible pairs of unequal length codewords of C, i.e.

$$d_{min} = \min\{d(c_i, c_j)\} \qquad (5)$$

where c_i and $c_j \in C$, $|c_i| \neq |c_j|$.

Definition4: the converge distance between two codewords of unequal lengths $|c_i|$ and $|c_j|$ of a code C is defined as the hamming distance between the l-length suffixes of codewords c_i and c_j, with $l = \min\{|c_i|, |c_j|\}$, where c_i and $c_j \in C$, $|c_i| = l_i$ and $|c_j| = l_j$, with $l_i > l_j$, then $l = l_j$, so [3], [4], [5]

$$C(c_i, c_j) = h(c_{j_1}c_{j_2}\ldots\ldots c_{j_{l_j}}, c_{i_y+1}c_{i_y+2}\ldots\ldots c_{i_{l_i}}) \qquad (6)$$

where $y = l_i - l_j$.

The minimum converge distance of the variable-length error-correcting code C, C_{min}, is the minimum value of all the converge distances between all possible pairs of unequal length codewords of C, i.e.

$$C_{min} = \min\{C(c_i, c_j)\} \qquad (7)$$

where $c_i, c_j \in C$, $|c_i| \neq |c_j|$.

Definition5: The sliding distance $s(c_i, c_j)$ between two codewords c_i and c_j of lengths $|c_i|$ and $|c_j|$ of a code C is defined as the minimum of the hamming distance between the l-length codeword and every sub-word of length l of the other codeword, with $l = \min\{|c_i|, |c_j|\}$.

$$s(c_i, c_j) = \min\{h(c_{i_u+1}c_{i_u+2}\ldots\ldots c_{i_u+l}, c_{j_v+1}c_{j_v+2}\ldots\ldots c_{j_v+l})\} \qquad (8)$$

$$0 \le u \le l_i - l$$

$$0 \le v \le l_j - l$$

where c_i and $c_j \in C$, $l_i = |c_i|$ and $l_j = |c_j|$.

The minimum sliding distance s_{min} of a code C is the minimum value for all sliding distances between every possible pair of codewords in C [4].

$$s_{min} = \min \{s (c_i, c_j)\} \quad (9)$$

where c_i and $c_j \in C$, $i \neq j$.

Definition6: the free distance d_{free} is defined as the minimum Hamming distance between all paths of the same length in bits diverging in some state and converging in the same or another state [6]. In [3], Buttigieg deduced a lower bound on the free distance from the distance between all possible pairs of unequal-length variable-length error-correcting codewords. He said that the free distance, d_{free}, of a VLEC code C is defined as the minimum Hamming distance in the set of all arbitrary long paths that diverge from some common state S_i and converge again in another common state S_j, j > i, so

$$d_{free} = \min \{h(f_i, f_j) : f_i, f_j \in F_N \} \quad (10)$$

where $N = 1, 2, \ldots, \infty$ and F_N is the extended code of order N of C (Note that for certain values of N, the set F_N may be empty).

The free distance of a VLEC code C is bounded by:

$$d_{free} \geq \min (b_{min}, d_{min} + c_{min}) \quad (11)$$

2.3 Levenshtein Distance

When dealing with codes capable of correction deletion and insertion errors, Levenshtein [7] introduced a new distance measure. Similar to the Hamming distance, the Levenshtein distance [8] between words is the minimum number of insertions, deletions and substitutions necessary to get from one word to another. Let a_1 and a_2 be two sequences of source symbols A, not necessary of equal length. Then, the Levenshtein distance between the two sequences is denoted by $L (a_1, a_2)$.

2.4 Symbol Error Probability

The symbol error probability (SEP) of the decoded source message a_r when compared with the transmitted source message a_t, is defined as the ratio of the Levenshtein distance $L(a_t, a_r)$ to the number of source symbols in the transmitted message a_t [2].

$$SEP = \frac{L(a_t, a_r)}{|a_t|} \quad (12)$$

where $| a_t |$ denotes the number of source symbols in a_t.

3. Generation of VLEC Codes

The Heuristic algorithm first step is; C is initially set to a fixed-length code of length L_1 and minimum distance b_{min}. This step, as all the following determination of sets of same length words at a given distance, is carried out either by the Greedy Algorithm (GA) or the Majority Voting Algorithm (MVA) [5]. Therefore, first the flowcharts, of these two algorithms are defined and then, the Heuristic algorithm is explained.

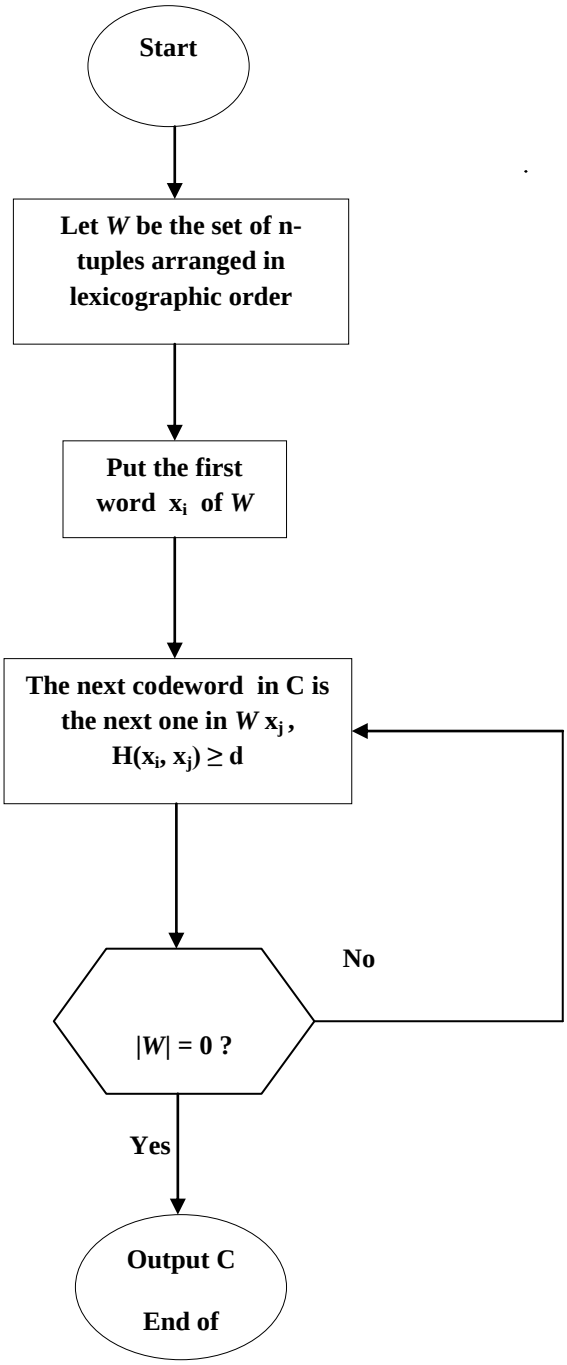


Figure (1): Flowchart of GA

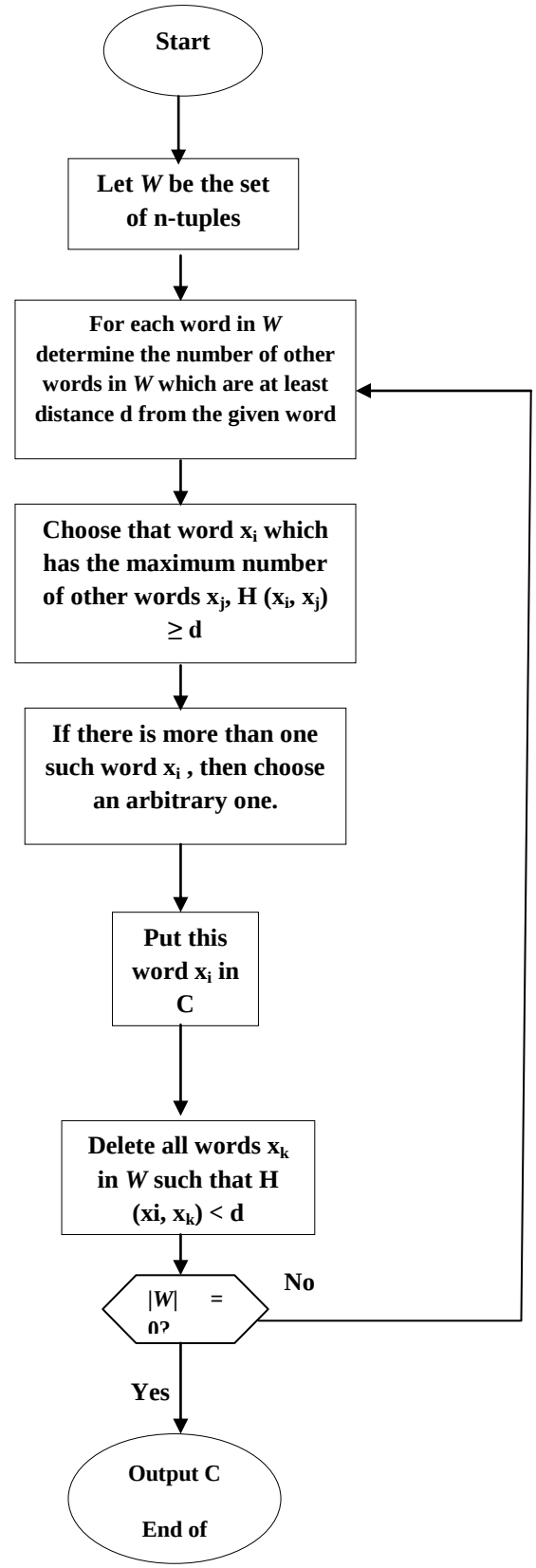


Figure (2): Flowchart of MVA

3.1 The Heuristic Construction Algorithm

The basic construction concepts of Heuristic algorithm are, as follows [5], [9]:-

1. A fixed-length code C with length L_1 and with minimum distance equal to b_{min} with a maximum number of codewords s_1 must be constructed. For a horizontally linear VLEC code, all fixed-length codes constructed with this algorithm must be linear or a coset, then this fixed-length code is constructed by using one of the two algorithms given in section (3).
2. All the possible L_1-tuples, which are at distance d_{min} from the codewords of C, are listed. Let this set of words be *W*.
3. The bound given by expression (11) suggests that it is better to choose $b_{min} = d_{min} + c_{min}$. Therefore, take $b_{min} = \left\lceil dfree/2 \right\rceil$ and $b_{min} = d_{free}$. In which case, $d_{min} < b_{min}$ and such words are possible to find, because if $d_{min} > b_{min}$ then *W* will be empty, so for the moment, *W* will be assumed to be not empty.
4. The number of words in *W* is doubled by increasing the words' length by one bit by affixing first a '0' and then a '1' to the rightmost position of all words in *W*. So now, *W* contains words of length L_1+1.
5. These words are then checked with the codewords in C. Those words in *W* which satisfy the minimum converge distance required with words in C are retained, the others are discarded. At the end of this operation, the words in *W*, when compared with the words in C will satisfy the required minimum diverge and converge distances.
6. The words with the same length must have minimum distance at least equal to b_{min}. So, here again the maximum number of words (s_2) from what is left in *W* must be chosen such that these words form a fixed-length code with minimum distance at least bmin, to achieve this, using GA or MVA.
7. These words are then added to the codewords already in C to form a VLEC code (s_1@ L_1, b_{min}; s_2@ (L_1+1), b_{min}; d_{min}, c_{min}).
8. This whole procedure is then repeated by the next considering all (L_1+1)-tuples which satisfy the minimum diverge distance to all the codewords in C and are also at the minimum distance d_{min} to those codewords in C of the same length, then affixing the extra bit to those words, extracting those words which satisfy the minimum converge distance.
9. This algorithm stops either when there are no farther possible words to be found or else when the required number of codewords is reached.

4. Decoding of VLEC codes.

4.1 Modified Viterbi Algorithm.

The VLEC codes behave very much like convolutional codes. In the case of VLEC codes, the Viterbi algorithm is modified to modified version of Viterbi decoding algorithm [10]. This is a maximum likelihood decoder (assuming all paths are equally probable), in the sense that it finds the closet coded sequence to the received sequence by processing the sequences on an information bit-by-bit. Now a modified version of the Viterbi decoding algorithm will be, as follows:-

Let $y = y_1y_2\ldots y_N$ be the received N-bit sequence, and denote the metric of the surviving path at state S_i by M_i.

1. Assign $M_0 = 0$ and $M_i = \infty$, $\forall\ _i > 0$. Let S_i denote the current state and initially put $i = 0$.
2. For all codewords $c_j \in C$, evaluate the branch metric $m_j = h(c_j, y_{i+1}y_{i+2}\ldots y_{i+lj})$.
3. Flag S_{i+lj} as a visited state. If $m_j + M_i < M_{i+lj}$, then store this codeword for the transition $S_i \rightarrow S_{i+lj}$ (overwriting any other previously stored transitions to state S_{i+lj}), and make $M_{i+lj} = m_j + M_i$.
4. Increment i to the next visited state and until $i > N-l_1$.
5. Decode the message corresponding to the codeword sequence represented by the surviving path to state S_N. The surviving sequence of codewords is decoded to state S_N, hence, the number of bits in the decoded codeword sequence is equal to N, and; therefore, the decoded sequence is a codeword of F_N.

5. Using VLEC Codes and Peak Signal to Noise Ratio (PSNR) in Image Communication.

VLEC codes can be used in image processes, but these are known to be highly susceptible to channel errors. The critical bits need to be protected against channel errors in order to prevent the complete loss of a transmitted image. In this section, it will be shown that, if VLEC codes are used in image processes, the performance of these codes is better than the performance when the variable-length source coding (Huffman code) is used. The image is corrupted by transmitting the corresponding bit stream over an AWGN channel model. The code rate is given by R is [2]:- $\frac{\lceil \log_2 s \rceil}{L_{average}}$, it is assumed that the bit stream is modulated using Binary Phase Shift Keying (BPSK). The block diagram shown in Figure (3) shows the transmitter and the receiver block for image processes. Figure (4) shows the original picture of 'Lena image', Figures (5a, 5b, 5c) show a representative pictures of 'Lena image' decoding using a modified Viterbi decoding algorithm with various SNR values.

The pictures in Figure (5a) are encoded using Huffman code table, whereas the pictures in Figure (5b) are encoded using a VLEC code with free distance = 3, and the pictures in Figure (5c) are encoded using a VLEC code with free distance = 5. The VLEC codes and Huffman code have a total of (256) codewords with code parameters, as shown in Table (1).

Table (1): Types of codes used in image communication

Code type	$L_{average}$ (bits)	Code rate	Maximum length(bits)
VLEC with $d_{free}=3$	10.5415	0.7589	13
VLEC with $d_{free}=5$	13.074	0.6119	17
VLC (Huffman)	7.828	1.022	11

Figure (6) shows the simulated performance curves of 'Lena image' picture with Huffman code and with VLEC codes of d_{free} =3, 5. The performance of VLEC code as it varies with different symbol error probability shows in general a great improvement in the SNR values, at the expense of increased complexity although the code rate of Huffman code is higher than the code rate of VLEC codes of d_{free} = 3, 5.

The improvement in the performance is in the order of (3dB) in terms of SNR values at symbol error probability (SEP=10^{-3}) for VLEC code with dfree = 3 and in order of (5dB) in terms of SNR at symbol error probability (SEP=10^{-3}) for VLEC code with dfree = 5, when compared with the performance of Huffman code.

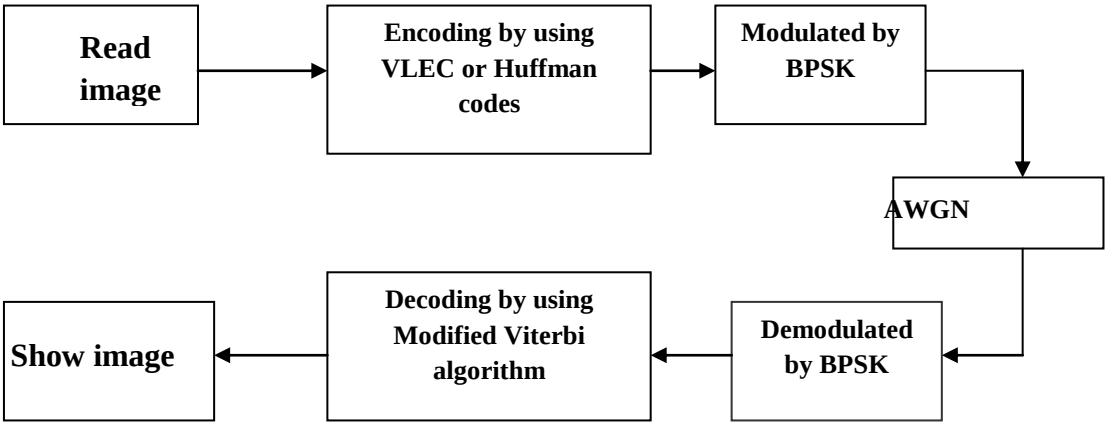


Figure (3): Block diagram of image processes

Figure (4): Original 'Lena image' picture

Image at SNR = 3dB

Image at SNR = 3dB

Image at SNR = 3dB

Image at SNR = 9dB

(a)

Image at SNR = 9dB

(b)

Image at SNR = 9dB

(c)

Figure (5): 'Lena image' picture decoded at different SNR values, (a) Using Huffman code (b) Using VLEC code with d_{free} = 3(c) Using VLEC code with d_{free} = 5, with modified Viterbi decoding.

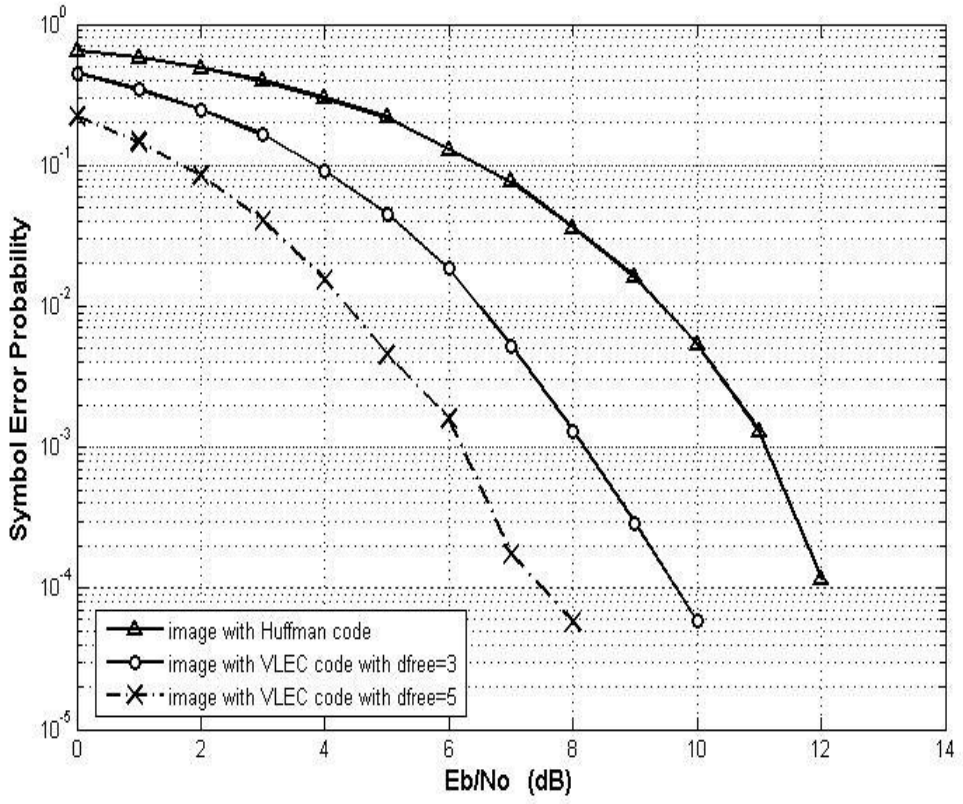


Figure (6): Performance comparison using VLEC codes and VL (Huffman) code in image processing

When the original image is transmitted, the noise on the channel will cause some amount of distortion between the original image and the same received image. The *PSNR* is employed here to show the performance of VLEC codes compared with Huffman code in image process. *PSNR* is usually measured in dB, *PSNR* is given by [11], [12]:-

$$PSNR = 10\log_{10} \frac{M * N * (L-1)^2}{\sum_{r=1}^{M}\sum_{c=1}^{N}\left[T(r,c) - R(r,c)\right]} \qquad (13)$$

where:-

M: height of transmitted image (no. of pixels).

N: width of transmitted image (no. of pixels).

L: the number of gray scales in the transmitted image, in the present case *L*=256 and the gray scale extends from 0-255.

r: row number in image matrix

c: column number in image matrix

T(r,c): transmitted image

R(r,c): received image

Table (2) shows various computed *PSNR* for 'Lena image' (shown in figure 4) when subjected to AWGN. This table shows how much the performance of VLEC codes is better than the performance of Huffman code in image process.

Table (2): Computed PSNR in (dB), using Huffman code and VLEC codes for 'Lena image'

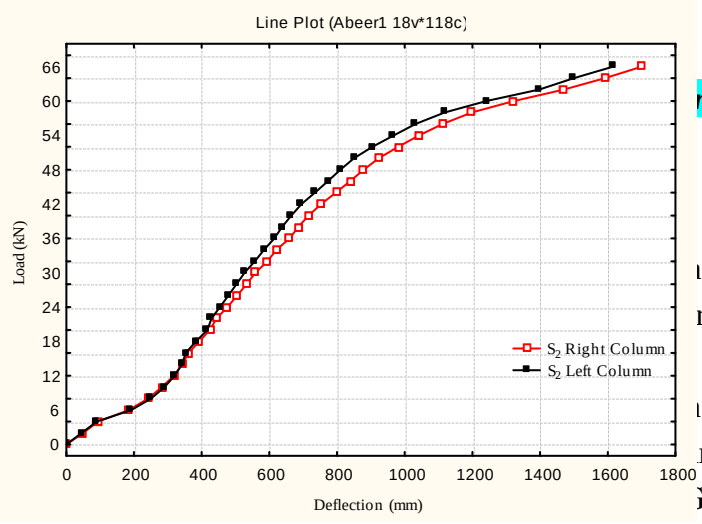
SNR(dB)	PSNR at Huffman code	PSNR at VLEC code with $d_{free} = 3$	PSNR at VLEC code with $d_{free} = 5$
0	13.461	14.9811	18.3300
1	13.9966	15.9992	20.2356
2	14.8063	17.4272	22.7813
3	15.9782	19.2394	25.9981
4	17.0701	21.4827	29.5483
5	18.5721	25.246	34.7446
6	20.3207	28.7642	43.4644
7	23.1503	32.8024	49.9362
8	25.983	41.9453	69.9346
9	29.8497	49.7729	Infinity
10	35.0787	69.8316	Infinity

6. Conclusion

VLEC codes incorporate spatial memory due to their variable-length nature. This leads to derivation of a maximum likelihood decoding algorithm based on the Viterbi algorithm for VLEC code. The three parameters b_{min}, d_{min} and c_{min} are very indicative of the performance of VLEC codes, since in most cases the free distance will be directly determined by these parameters. For the most VLEC codes considered, the bound on d_{free} given in expression (11) is found to be met with equality.

One result of this equality is that whereby the minimum block distance of the shortest length codewords can be the same as that for the longest length codewords, without affecting the performance of the code.

Therefore, there is a large probability of correcting errors in the shorter codewords than in the longer ones. Notice that the shorter length codewords are more probable (since they are mapped to the most probable source symbols). Hence, on average, the error correcting capability of a VLEC codes is improved in this way. Using VLEC code in image processes gives good improvement in terms of Peak Signal to Noise Ratio (*PSNR*) of about 20dB if compared with variable-length source coder such as Huffman code.



ı, "The Art of Error Correcting Coding", Second
rsity, USA.2006 John Wiley and Sons, Ltd. ISBN: 0-

ıgth error-correcting codes" Ph.D. Dissertation,
nchester, United Kingdom, 1995.
J., " Variable-length error-correcting codes", IEEE Proc. Commun., Vol. 147, No. 4, pp. 211-215, Aug. 2000.

[4] Catherine Lamy and Franc¸ois-Xavier Bergot, "Lower bounds on the existence of binary error-correcting variable-length codes", ITW 2003, Paris, France, March 31 – April 4, 2003.

[5] Catherine Lamy', Johann Paccaut, "Optimised constructions for variable- length error correcting codes", ITW2003, Paris. France, March 31-April 4, 2003.

[6] Salma Ben Jamaa, Claudio Weidmann, and Michel Kieffer, "Analytical Tools for Optimizing the Error Correction Performance of Arithmetic Codes", LSS-CNRS-Supélec-Université Paris-Sud. Telecommunications Research , July 7, 2006.

[7] Levenshtein V.I., "Binary codes with correction of deletions, insertions and substitution of symbols", Dokl. Akad. Nank. SSSR, Vol. 163, pp. 845-848, 1965.

[8] Levenshtein V.I., "Binary codes capable of correcting deletions, insertions and reversals", Sov. Phys. Doklady, Vol. 10, p. 707, 1966.

[9] Jin Wang, Lie-Liang Yang, and Lajos Hanzo, Fellow, IEEE, "Iterative Construction of Reversible Variable-Length Codes and Variable-Length Error-Correcting Codes", IEEE Communications Letters, Vol. 8, No. 11, pp. 671-673, November 2004.

[10] Viterbi A.J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Trans. Inform. Theory, Vol. IT-13, pp. 260-269, 1967.

[11] Steven Mann, "Intelligent Image Processing", University of Toronto, Copyright 2002 by John Wiley and Sons.

[12] Suha k. Hadi ," Variable-Length Error-Correcting codes", Msc. Thesis, Al-Mustansiriya University, college of Engineering, 2009.