

Lookup Table Algorithm for Error Correction in Color Images

Ruaa A. Abdul-Sattar[#], Nada Hussein M. Ali[#]

[#] Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq
E-mail: ruaaalqassab@yahoo.com, nada_husn@yahoo.com

Abstract— Error correction and error detection techniques are often used in wireless transmission systems. A color image of type BMP is considered as an application of developed lookup table algorithms to detect and correct errors in these images. Decimal Matrix Code (DMC) and Hamming code (HC) techniques were integrated to compose Hybrid Matrix Code (HMC) to maximize the error detection and correction. The results obtained from HMC still have some error not corrected because the redundant bits added by Hamming codes to the data are considered inadequate, and it is suitable when the error rate is low for detection and correction processes. Besides, a Hamming code could not detect large burst error period, in addition, the ΔH and ΔV have same values sometimes which lead to not detect the error and consequently increase the error ratio. The proposed algorithm LUT_CORR is presented to detect and correct errors in color images over noisy channels, the proposed algorithm depends on the parallel Cyclic Redundancy Code (CRC) method that's based on two algorithms: Sarwate and slicing By N algorithms. The LUT-CORR and the aforementioned algorithms were merged to correct errors in color images, the output results correct the corrupted images with a 100 % ratio almost. The above high correction ratio due to some unique values that the LUT-CORR algorithm have. The HMC and the proposed algorithm applied to different BMP images, the obtained results from LUT-CORR are compared to HMC for both Mean Square Error (MSE) and correction ratio. The outcome from the proposed algorithm shows a good performance and has a high correction ratio to retrieve the source BMP image.

Keywords— CRC, DMC, Mean Square Error, BMP images, error correction.

I. INTRODUCTION

The information usually is transmitted physically over a communication channels such as telephone lines in wire type media or the atmosphere in the wireless communication. Through the transferring of these information, a disturbance; or noise; may arise and leading to alter the data, causing a different from the original information sent [1]. Error correction is much more difficult than error detection. In error detection, the receiver must know only that the received codeword is invalid; while in error correction, the receiver needs to find (or guess) the original codeword sent. Thus, the error correction need more redundant bits than for error detection, error correction can be handled in two ways [2]. Error correcting codes are an effective technique for improving data integrity when transmitting information over a channel. The channel encoding procedure involves adding redundancy in the form of parity check-bits and encoding-decoding logic circuits to improve data robustness to noise during transmission. The channel-encoding problem is one of designing an ECC (Error

Correction Code) with sufficient error handling capabilities while maintaining its area and delay penalties to within the system specifications. The most appropriate choice of ECC depends on the system application and the behaviour of the noise in the channel [3].

II. RELATED WORK

Many papers have presented strategies to improve the performance of reliable error detection and correction as follow: [U. K. Kumar, 2007] suggested an improvement method to eliminate the appended redundant bits. The proposed method reduces the overhead of spread the redundant bits at the transmitter side and their removing at the receiver side. The correction method checking a single bit error for a correction of the corrupted data [4] [Indu I and Manu T S , 2012] presented a slicing by 8 algorithm to generate CRC 32 using Multiple Look-up Table(LUT) algorithm .This algorithm can read large amounts of data at a time, while optimizing their memory requirement to meet the limitations of specific computer structures.[5]

[Mukesh Gholase 2013] proposed a method using orthogonal code block for mapping k-bits to n-bits where (n>k), then these coded blocks are transmitted through the communication channel. The structure coding for orthogonal code block modulation systems is realized by the FPGA. The obtained results of the proposed method enhanced the error detection and correction abilities for the orthogonal codes convolution with a detection rate of 99.99% [6].

[V. Singh and N.Sharma, 2015] proposed technique can be used efficiently where retransmission of data is complicated and expensive too. The bit overhead and code rate for this technique are 50% and 66.66% respectively which provide better results for error detection and correction than other EDAC technique[7]

[Maria Antony 2015] proposed a novel Hybrid Matrix Code method to assure the reliability of memory. The proposed work focuses on the design of HMC for error detection and correction, suitable for memory applications, with reduced area utilization and delay overhead. It uses a less number of redundant bits need to be stored for protection when compared to the existing method [8].

[Neepta et al. 2016] generated CRC error is corrected by using Hybrid Matrix Code (HMC) and they are coded in verilog HDL and simulated using Xilinx ISE Design suite 14.2. This proposed method can provide maximum error detection and correction capability with less number of redundant bits [9].

[K. Sandyarani and P. Nirmal Kumar 2016] proposes an efficient error correction for AES algorithm using extended hamming codes of type (n, k) to detect and correct a single error (SEC), in addition, to identify many types of double and triple adjacent errors (DAED) or (TAED). The results obtained from the enhancement method (Hamming SEC-DAED-TAED codes) are better comparing to based hamming SEC-DAED-TAED codes. The improvement ratio in DAED is 45%, while in TAED are 33% [10].

III. FORWARD ERROR CORRECTION TECHNIQUES

In the real-time applications, a Forward Error Correction (FEC) technique is used to recover the data during the transferring over a communication channel. The information on this method and at the receiver side could be corrected without any support from the sender. The source data are divided into a block of k length, these blocks are converted into a group of n coded data; denoted by (n, k) block; and can be used to recreate or erasure the corrupted information.

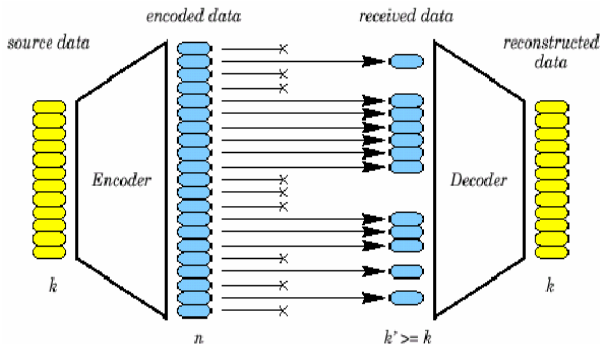


Fig. 1 The FCE method. [11]

Frequently the k data in these groups represent the original source data while the remaining (n – k) are the appended parity bits as shown in Figure 1. Usually, FEC codes are capable to correct both errors and erasures in a block of n symbols [11].

Many existing error correction codes as follows [11]

- a- Hamming Code (HC):
It is a form of linear error correcting code that can detect up to double bit error or correct a single bit error without detection of uncorrected errors.
- b- Reed Muller Code (RMC):
It is considered binary linear codes and represents a subspace of the vector space of all binary n-bit vectors.
- c- Built in Current Sensor (BICS):
This technique is capable of detecting any change in the current block that represents an error. The sensors placed in each column in the memory blocks to identify any change in each bit inside the memory. The BICS can detect two-bit errors and correct one bit error.
- d- Reed Solomon Code (RSC):
Such codes are non-binary codes and it is used to realize a large Hamming space. The decoder of RSC decoder able to adjust the whole character, the correction operation is performed either the errors are caused by a single bit or even by a whole bits in the symbol being corrupted.
- e- Decimal Matrix Code (DMC):
This type of error detection technique is used to preserve the integrity and reliability of the memory contents. The DMC uses decimal algorithm that leads to raise the error detection and correction capability

A. Error correction using Hybrid Matrix Code (HMC) method

Combination of the Decimal Matrix Code and the Hamming Code gives Hybrid Matrix Code(HMC).It utilizes the hamming algorithm to detect the errors so as to provide better error detection and correction capability. The encoder reuse technique is also employed here to reduce the area overhead. In the encoder section, the incoming data stream is arranged in the form of horizontal rows and vertical columns that is the N bit word is divided into k symbols with m bits. Then the horizontal redundant bits are generated by performing the hamming code operation on each symbol per row. The vertical redundant bits are generated by performing the binary operation between vertical columns as shown in Figure 2[9].The HMC for error detection and correction, suitable for memory applications, with reduced area utilization and delay overhead, see [8].

Definition 1 (Hamming distance). For any $x, y \in \Sigma^n$, the Hamming distance between x and y denoted $\Delta(x, y)$, is the number of position i in which x_i and y_i differ: $\Delta(x, y) = |\{i \in [n]: x_i \neq y_i\}|$ [12].

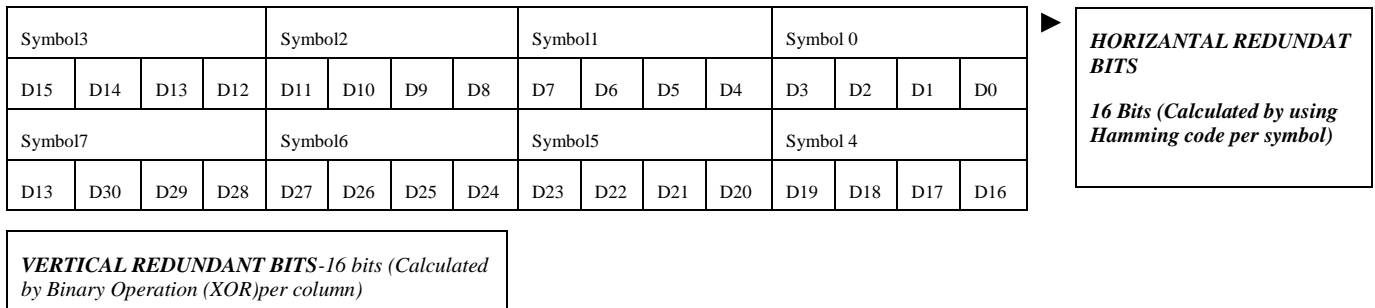


Fig. 2 32bits HMC Logical Organization [11]

B. The Proposed Lookup Table algorithm (LUT-CORR) for image correction

Three BMP image files of different sizes as shown in Figure 3 were used to test the noise effect on these images by implementing both the proposed and HMC algorithms for the encrypted images. Different noise ratios were added to various image sizes to check the strength of these algorithms; Table 1 represents the MSE for three tested images. The noise; 1% and 5% ratio of image size; added randomly on pixels bits to study the detection and correction error approaches.

The computing of the lookup table algorithm is described in Algorithm 1; while the proposed algorithm is demonstrated in Algorithm 2 for error correction and it is tested by applying on color image. The simulated results of the proposed algorithm were compared with HMC [9] to show the improvements obtained by the LUT-CORR algorithm for both time and correction ratio.

Algorithm 1: compute the lookup tables

Input: BMP color image of size[R*C], R=0,1,.....Width-1, C=0,1,.....Height
Output: Two Lookup tables for data corrections:
 Corr_LUT_Div[Max_Size],
 Corr_LUT_Mod[Max_Size]

Step1: Select an Anchor value Anc_Val, such minimum or maximum or median value in the image file
Step2: Repeat steps 3-6
Step3: Divide the image file into blocks each of 16 bytes, and then compute the one-way hash function CRC64 based on GF(2⁸)
Step4: For each block compute two values
 a- Value1: represents the multiples of the current block for Anc_Val value.
 b- Value2: represents the residual of the current block for value Anc_Val value.

Step5: Save the obtained values; Value1 and Value2; in arrays Corr_LUT_Div[Max_Size] and Corr_LUT_Mod[Max_Size]
Step6: Until end of image file

Algorithm 2: the proposed LUT-CORR for image correction

Input: BMP color image of size[R*C], R=0,1,.....Width-1, C=0,1,.....Height with added noise ratio 1% or 5%.
 The redundant bits: The Anchor value Anc_Val, Two Lookup tables for data corrections:
 Corr_LUT_Div[Max_Size], Corr_LUT_Mod[Max_Size].
Output: Correct BMP color image Corr_BMP_Image

Step1: implement Algorithm 1 for the noise image and compute two tables
 Noise_LUT_DIV[Max_Size], Noise_LUT_Mod[Max_Size]
Step2: Repeat steps 3-6
Step3: for i=0,1,.....Max_Size
 Compare both two Block_Image_Source[i] and Block_Image_Noise[i] by compute:
 Corr_LUT_Div[i] and Noise_LUT_DIV[i];
 Corr_LUT_Mod[i] and Noise_LUT_Mod[i];
Step4: if both values in step3 are equals then store the current block in Corr_BMP_Image
 else go to Step5 and Step6
Step5: compute two values: check1= Block_Image_Noise[i]/Anc_Val
 check2=Block_Image_Noise[i]%Anc_Val
Step6: compare the values check1 and check2 with Corr_LUT_Div[Max_Size] and Corr_LUT_Mod[Max_Size] respectively, if equals store check1 in Corr_BMP_Image
Step6: Until end of image file



Fig . 3 The tested images used by the proposed LUT-CORR algorithm

TABLE 1
THE MSE FOR 1% AND 5% ADDED NOISE IN IMAGE

Image name	Size (bytes)	No. of noisy bytes	MSE	
			1%(noise)	5%(noise)
Fruits	786,486	7864	10047.267	39321
Lena	451,142	4510	9059.456	22554
Peppers	76,854	767	10048.684	3839

C. Results analysis and discussions

The goal of the HMC technique in [9] is to maximize error detection and correction capability with reduction in delay time. It has been noticed that the correction capability in HMC algorithm has some error ratio to retrieve the source image transmitted by the sender as in follow:

- 1- The redundant bits added by hamming codes usually used for error detection and correction when the error rate is small. The above technique commonly used in computer memory; (Error Checking & Correction, ECC memory); for error correction where the bits alteration is considered rare.
- 2- In the case of noise in communication channels, a Hamming code could not detect large burst errors period and that leads to misleads of error detection and corrections.
- 3- In some cases, the values of ΔH and ΔV have same which lead to not detect the error and consequently leads to increase the error ratio and not retrieve the source data perfectly.

Figure 4 demonstrates the results obtained by the HMC technique in [9], from this figure one can notice that the image retrieve has some noise which could not corrected by the aforementioned algorithm. Table 2 demonstrates the implementation results obtained by HMC algorithm, the correction ratio is obtained as in Equation 1:

$$\text{correctionratio} = \frac{\text{No. of noise bytes} - \text{No. of corrupted bytes}}{\text{No. of noise bytes}} \quad (1)$$

Some notices could be concluded from the above table:

- ❖ In spite of the implementation of HMC algorithm, some noise remains, and not corrected causing errors in retrieving of decrypted image.
- ❖ Large noise value leads to a reduction in the correct ratio obtained by HMC execution.

Figure 5 demonstrates graphically the results obtained from implementing the HMC algorithm. The latest figure declares the image noise that not corrected as explained above.

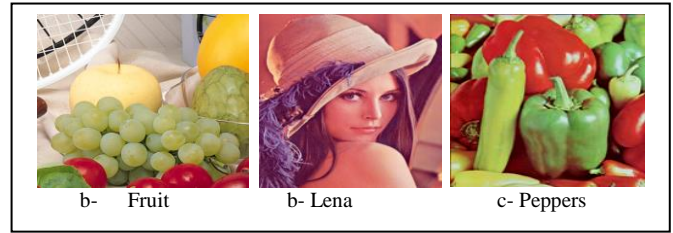


Fig. 4 Images Enhanced by HMC technique to remove noise

TABLE 2
THE HMC FOR THE 1% AND 5% NOISE RATIO OF THE IMAGE SIZE

Image name	Size (bytes)	No. of noisy bytes	No. of Corrupted bytes 1%(noise)	Corrected ratio	Time in SEC	No. of noisy bytes	No. of Corrupt ed bytes 5%(noise)	Corrected ratio	Time in SEC
Fruits	786,416	7864	126	98%	0.32	39320	3692	90%	0.049
Lena	451,072	4510	81	98%	0.021	22554	1960	91%	0.028
Peppers	76,784	767	24	96%	0.003	3839	378	90%	0.004

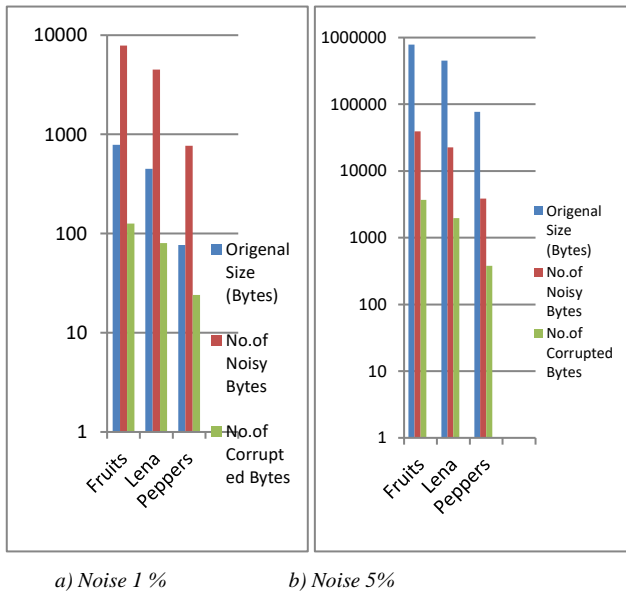


Fig. 5 the HMC algorithm for the 1% and 5% noise ratio of the image size

The implementation results achieved by the proposed Algorithms (1 and 2) for image correction are demonstrated in Fig. 5, one can notice in this figure that the retrieving of the image after correction operation by LUT-CORR algorithm had no difference compared to source images in Figure 3. The corrected images using the LUT-CORR algorithm have been completely corrected where the MSE between the noise and corrected images are zero as shown in Table 3. Figure 6 demonstrates graphically the improvement in error correction between the HMC and LUT-CORR algorithms.

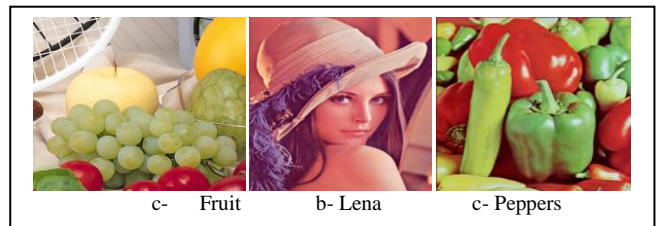


Fig. 6 The proposed LUT-CORR algorithm for image correction.

TABLE 3
PROPOSED LUT FOR IMAGE CORRECTION FOR 1% AND 5% NOISE RATIO

Image name	Size (bytes)	No. of noisy bytes	No. of Corrupted bytes or MSE	Corrected ratio	Time in SEC	1%(noise)		5%(noise)	
						No. of noisy bytes	No. of Corrupted bytes or MSE	Corrected ratio	Time in SEC
Fruits	786,416	7864	0	100%	0.032	39320	0	100%	0.049
Lena	451,072	4510	0	100%	0.021	22553	0	100%	0.023
Peppers	76,784	767	0	100%	0.003	3839	0	100%	0.004

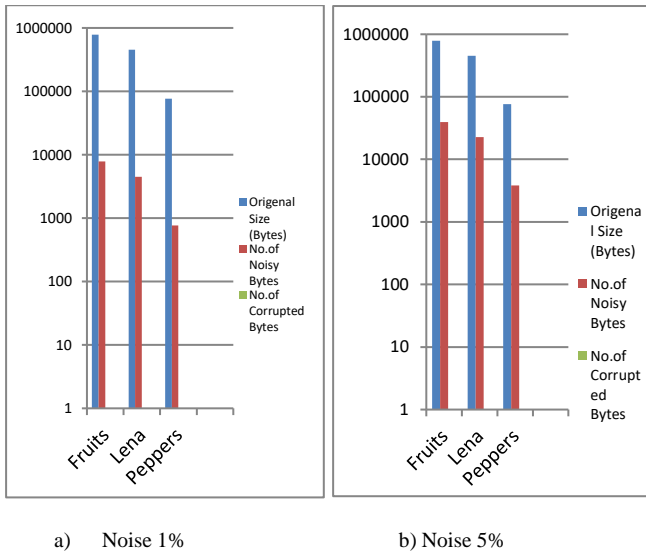


Fig. 7 LUT-CORR algorithm for image correction of 1% and 5% noise ratio

IV. CONCLUSIONS

Hamming codes are reasonably efficient when data transferring is considered small, unfortunately increasing the size of the data sent leads to inaccurate of the receiving information. In addition, Hamming codes do very poorly against bursts of errors caused by fading over a channel. Merging HMC and DMC leads to increase the error correction operation because in real situation errors tend to cluster. The proposed techniques map a k redundant bits data block into $n - bit$ data blocks. The HMC algorithm has 128 redundant bits for 128 data bits, while for each 128 bits data block in the proposed LUT-CORR algorithm there are 160 redundant bits for error correction. In spite the extra bits attached to data block in the proposed algorithm are larger compared to HMC but the correction ratio is maximized. The MSE between the corrected and source images almost near zero that considered a good performance of the proposed algorithm.

REFERENCES

- [1] Rubal C. and Vrinda G., "Error Control Techniques and Their Applications", International Journal of Computer Applications in Engineering Sciences, Vol. 1, Issue 2, pp:187-191,2011.
- [2] Behrouze A., "Data communication and networking", fourth edition, text book.
- [3] Adam N, "Design and Analysis of an Adjacent Multi-bit Error Correcting Code for Nanoscale SRAMs", Ph.D. thesis presented to the University of Waterloo, 2014.
- [4] Indu I and Manu T S, "Cyclic Redundancy Check Generation Using Multiple Lookup Table Algorithms", International Journal of Modern Engineering Research, Vol.2, Issue.4, pp-2445-2451, July-Aug. 2012.
- [5] U. K. Kumar, "Improved Hamming Code for Error Detection and Correction", IEEE 2nd International Symposium on Wireless Pervasive Computing, pp: 498-50, 2007.
- [6] V.Singh and N. Sharma, "Improving Performance Parameters of Error Detection and Correction in HDLC Protocol by using Hamming Method", International Journal of Computer Applications, Vol.126, No.1, 2015
- [7] Mukesh G., L.P.Thakare and A.Y. Deshmukh, "Enhancement of Error Detection and Correction Capability Using Orthogonal Code Convolution", International Journal of Computational Engineering Research, Vol. 3, Issue 4, pp:66-71, 2013.
- [8] Maria Antony.S Sunitha. K, "Hybrid Matrix Codes for Enhanced Memory Reliability against Multiple Cell Upsets", International Journal for Scientific Research and Development| Vol. 3, Issue 1, pp:114-117, 2015.
- [9] Neepa P.Mathew and Anith M., " Matrix Code Based Error Correction For LUT Based Cyclic Redundancy Check", Elsevier Procedia Technology Vol. 25, pp: 590 – 597, 2016.
- [10] K. Sandyarani and P. Nirmal K., " Extended Hamming SEC-DAED-TAED based Fault Detection Technique for AES Encryption and Decryption", Vol. 9, issue 33, issue 33, pp:1-5,2016.
- [11] Abdelhamid Nafaa, Tarik Taleb, Liam Murphy, " Forward Error Correction Strategies for Media Streaming over Wireless Networks", IEEE Communications Magazine Vol. 46 Issue 1, pp:72-79, 2008.
- [12] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson, "Optimal Error Correction for Computationally Bounded Noise", IEEE Transactions on Information Theory, Vol. 56 Issue 11, pp:1-19, 2010