International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 2 133 – 143

Selective Video Coding based on Bezier Curves

Srividya B V
Dayananda Sagar College Of Engineering,
Bangalore, Karnataka,
India

Dr. Akhila S
BMS College Of Engineering,
Bangalore, Karnataka,
India

Abstract: In this paper the problem of reconstruction of video frames is addressed, when there are missing pixels in each video frame or is corrupted with noise and also the locations of corrupted pixels are not known. The modified data can be corrected using Forward Error Correcting Codes. Forward Error correcting codes detect and correct errors with the help of complex decoders. This work proposes a new approach called Selective encoding for reconstruction of Video Frames from Error. This algorithm combines the Bezier curves over Galois Field GF (p^m) and the Low Density Parity Check Codes for performing encoding and decoding. The proposed decoder is capable of detecting and correcting errors in each video frame, where only selected pixel values are encoded and decoded. This reduces the decoding time significantly. Further, when binary representation of the Galois Field is used, the speed of the decoder is enhanced as there is no carry generation and carry propagation when any modular arithmetic operation is carried out. Further time complexity is improvised by using parallel processing. The coding of the algorithm is carried out using MATLAB.

Keyword: *Selective Encoding, Error Recovery, LDPC codes, Bezier curve, Galois field.*

*****

I. Literature Survey

The work of Daniel Costello J *et al.* in "Applications of Error-Control Coding" mainly discusses the various error control methods in mobile communication, like the different FEC codes like Hamming code, Reed Solomon Code, Golay code, Low Density Parity Check codes (LDPC) and Turbo coding for GSM and CDMA along with the different data rates[1].

M H Azmi in his PhD Thesis on "Design of low-density parity-check codes in relay channels " , discusses about the capacity of LDPC which is the subject of intense interest in research[2].

Sheryl L. Howard in his work on "Error control coding in wireless sensor networks " mainly discusses about the specific error control coding like Checksum, Reed Solomon, Cyclic Redundancy Check and Convolution codes in WSN's. The authors have implemented several decoders like Reed Solomon, Hard decision Viterbi, Soft Decision Viterbi, LDPC codes, for the data in different environments[3].

The works of sanjeev kumar *et al.* mainly discusses the methods of improving the total BER which is obtained through the combination of RS codes for correcting burst errors and convolution codes which are good for correcting random errors, that are caused due to a noisy channel. LDPC codes are known to perform well in the presence of Additive White Gaussian Noise (AWGN) but for very large block lengths[5].

The authors of "Performance study of non-binary LDPC codes over Galois field", V S Ganepola *et al.* Have proposed to define the codes over higher order Galois fields to overcome the limitations of having a large block length[6].

The authors of" Review paper on the decoding of LDPC codes using Advanced Gallagers algorithm" Padmini U Wasule *et al.* mainly discusses on the decoding of Low Density Parity Check Codes using Advanced Gallagers algorithm, which has a fully parallel implementation to reduce the Bit Error Rate and lower the hardware complexity at the cost of increase in area to achieve maximum throughput[7].

Alin Sindhu has proposed "A Galois field based very fast and compact error correcting technique" which discusses on Euclidean Geometry based LDPC where serial one step majority logic decoder is used. The received vector is cyclically shifted and then fed to the shift register circuit to perform the error correction. However, as the number of bits increases, the decoding time increases. Also the hardware complexity enhances, if the information to be encoded increases, as the proposed method uses p-input XOR gates, depending on the size of the parity matrix[8].

M. P. C. Fossorier *et al.* In their work on " Reduced complexity iterative decoding of low density parity check nodes based on belief propagation", discusses on Fast decoding algorithms based on Fast Fourier Transform to reduce the computation complexity of the belief propagation algorithm using higher order Galois field but for moderate code lengths. The algorithm reduces the computational complexity by simplifying the check node computation. But the algorithm is unable to improve the decoding performance of the LDPC codes[9].

J.P chen *et al.* Have discussed on "Density evolution for two improved BP-based decoding algorithm for LDPC codes", which have an improvement in the decoding performance, but decoding performance suffers from degradation when output is near to zero[10].

The work of Meng Xu *et al.* is on Modified Offset min-sum algorithm (MOMS), which has an improvement in the decoding performance and requires P + 2 more addition operations compared, to OMS algorithm[11].

From the Literature Survey, it is observed that a better performance in decoding is required. The proposed algorithm is on Selective Encoding. Here the complexity of the hardware can be simplified and a better performance of the decoder can be achieved even when output is zero. Further the improvement achieved in the proposed work, is reduction in the area as the Hardware used is XOR gates. Further, there is No performance degradation seen when output is zero. Also, K-P modulo-additions are required for decoding, where K is the length of the information digits and P is the number of non zero digits of the Parity matrix. It has been observed that a better improvement in the decoder performance is achieved using Selective Encoding in Galois Field GF (2^m).

II. INTRODUCTION

The proposed work is based on Bezier curves over Galois field GF (p^m), combined with the Low Density Parity check (LDPC) codes for the purpose of encoding and decoding the data. The construction of the Generator matrix G for encoding and the parity check matrix H for decoding is based on Bezier curves over Galois field GF(2^m). The proposed decoder can detect and correct any random errors up to 150 digits in a word of 256 digits. Low Density Parity Check codes, Bezier curve and Galois field are discussed.

2.1 Low Density Parity Check Codes

Low density parity check codes are very widely used for error detection and correction purposes [12]. Low Density Parity Check Codes is a class of codes, which have a small number of 1's compared to zeros. LDPC are defined by a randomly generated parity matrix which can be of type regular or irregular. The Regular parity matrix P is constructed to have a uniform column weight and row weight [12][13]. Such algebraic construction methods ensure that each row has exactly the same number of elements and each column has exactly the same number of elements. These conditions ensure that the parity matrix P has uniform row and column weights forming a Regular LDPC code [14]. The Parity matrix P that does not adhere to the property of having uniform row and column weight forms an Irregular Parity matrix.

In the proposed work the parity matrix P is of type regular, constructed using Bezier curve elements over Galois Field GF (p^m).

2.2 Bezier curve

Bezier curves were widely publicized in 1962 by the French engineer Pierre Bezier. The Bezier curve is a parametric curve. Bernstein polynomial functionally defines the Bezier curve[16]. Bezier curves are a method of designing polynomial curve segments, where in the shape of curves can be controlled using the control points. The Bezier curves have control points from P0 to Pn, where n is the order of the Bezier curve. Based on the value of 'n', the following are the different classes of Bezier curves.

(i) Linear Bezier curves

Figure 1 shows the plot of the Linear Bezier curve B (t) Versus t, which is equivalent to linear interpolation between two points.

The equation for linear Bezier curve is given by $B(t) = (1-t)P_0 + tP_1, t \in [0,1]$ Where P0 and P1 are the control points.

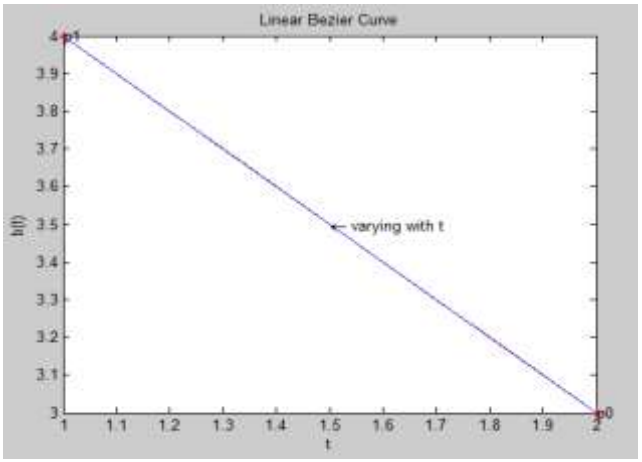

Figure : 1 Linear Bezier Curve

(ii) Quadratic Bezier curve

Figure 2 shows the plot of the Quadratic Bezier curve[16] B (t) versus t, which can be interpreted as the linear interpolate of corresponding points on the linear Bezier curves from P0 to P1 and from P1 to P2 respectively.

A quadratic Bezier curve [11] is defined by the function B(t), with P0, P1, and P2 as control points. The curve equation is

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, t \in [0,1]$$

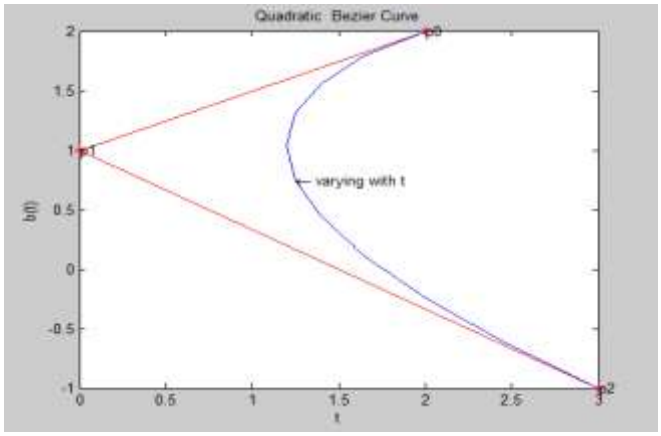

Figure : 2 Quadratic Bezier Curve

(iii) Cubic Bezier curves

Figure 3 shows the plot of the Cubic Bezier curve[16] B (t) versus t.

Four control points P0, P1, P2 and P3 defines a cubic Bezier curve for n=3.

The explicit form of the cubic Bezier curve is given by

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3(1-t)t^2 P_2 + t^3 P_3; t \in (0,1)$$

The first and the last points are on the curve, while the middle two points are not on the curve. The change in the control points, changes the shape of the curve. Connecting the control points by the line segments form a control polygon. The curve is tangent to the control polygon.

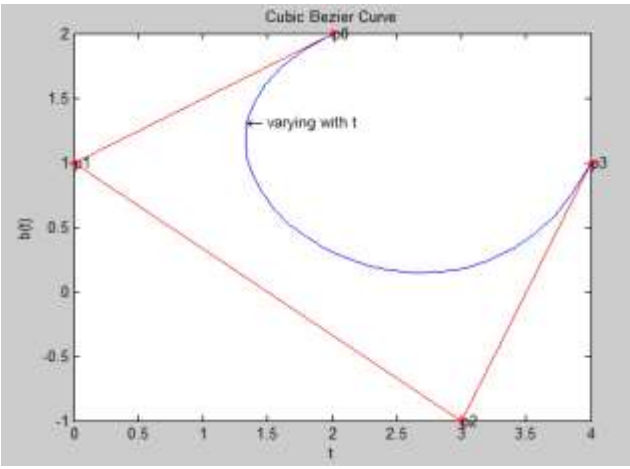

Figure 3 : Cubic Bezier Curve

(iv) Quartic Bezier curves

Figure 4 shows the plot of the Quartic Bezier curve B (t) versus t.

Five points P0, P1, P2, P3and P4 define a Quartic Bezier curve [16] given n=4. The explicit form of the Quartic Bezier curve is given by

$(1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t)P_3 + t^4 P_4$ Where $t \in (0,1)$

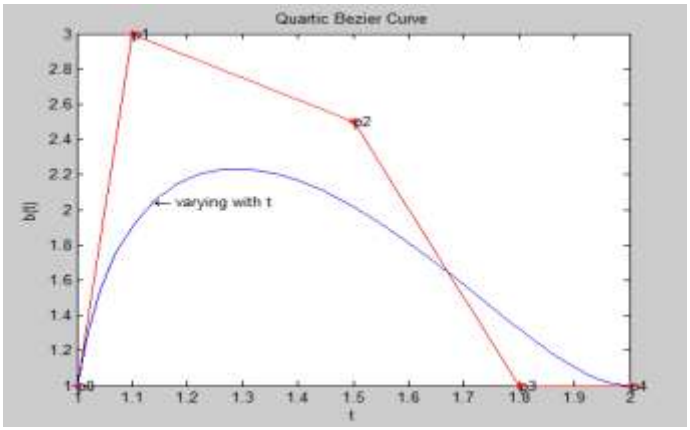

Figure 4 : Quartic Bezier Curve

2.3 Galois Field

Galois field Algebra is named after its inventor Evariste Galois. In Galois field GF (2^m), there are finite number of elements. These finite fields are extensively used in coding theory like BCH Codes, Reed Solomon codes [17]. The order of a finite field is always a prime or power of a prime. Coding theory focuses on finite fields. For any prime integer p and any integer m greater than or equal to 1, there is a unique field with p^m elements denoted as GF (p^m).In case m is equal to 1, the field is just the integers mod p. In coding theory, and in cryptography, normal practice is to almost always take the value of p to be 2, which is called as binary extension and represented as GF (2^m). Let α €GF (2^m) be the root of a primitive polynomial of degree m over GF (p). The elements of GF (2^m) are {0, 1, α α^2, α^3… α^{m-2}}. Each element in GF (2^m) can be represented using m-bits. Arithmetic operations can be performed for the elements of GF (2^m), which is useful in coding theory.

The following section shows the arithmetic operations performed on GF (2^m)

(i) Elements in GF (2^m)

In GF (2^m), modular arithmetic operations are simpler. The need for hardware also reduces since there is no concept of carry generation and carry propagation.

The elements of Galois field GF (2^4) is constructed using the primitive polynomial P(x) = x^4 + x + 1 and is shown in Table 1

Element	Polynomial representation	Binary representation
0	0	(0000)
α^0	1	(1000)
α^1	X	(0100)
α^2	X^2	(0010)
α^3	X^3	(0001)
α^4	X+1	(1100)
α^5	X^2+X	(0110)
α^6	X^2+ X^3	(0011)
α^7	1+X+ X^3	(1101)
α^8	1+X^2	(1010)
α^9	X+ X^3	(0101)
α^{10}	1+X+X^2	(1110)
α^{11}	X+X^2+ X^3	(0111)
α^{12}	1+X+X^2+ X^3	(1111)
α^{13}	1+X^2+X^3	(1011)
α^{14}	1+ X^3	(1001)

Table 1: Elements of GF (2^4)

(ii) *Addition in GF (2^m)*

Illustrating the Galois field addition with an example: The Galois Field GF (2^4) has

P(x) = x^4 + x + 1 as the primitive polynomial. Table 1 show that each element in GF (2^4) can be represented using 4-bits in binary. Bitwise XOR is used while adding the elements of GF (2^4). For example:

$\alpha^5 + \alpha^5$ = (0110) + (0110) = (0000) = 0= α^0

$\alpha^2 + \alpha^5$= (0010) + (0110) = (0100) = 1= α^1

The addition table for GF (2^4) is as shown in Table2

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	110	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 2: Addition in GF (2^4)

(iii) Multiplication in GF (2^m)

In most algorithms the modular product is computed using the polynomial multiplication succeeded by the modular reduction. Let A(x), B(x) be the polynomial represented elements of GF (2^m) and P(x) be the irreducible field generator polynomial.

Example: If $P(x) = X^4+X+1$, $A(x) = X^2+1$, $B(x) = X^2+X$

Then $A(x)* B(x) = (X^2+1)* (X^2+X) = (X^4+ X^3+ X^2+X)$

Modular reduction

$(X^4+ X^3+ X^2+X) \bmod (X^4+X+1) = 1+X^2+X^3$

Using the Galois Field GF (2^4) which is based on $P(x) = x^4+x+1$, the multiplication table for GF (2^4)[10] is as shown in Table 3

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	3	1	7	5	11	9	15	13
3	3	6	5	12	15	10	9	11	8	13	14	7	4	1	2
4	4	8	12	3	7	11	15	6	2	14	10	5	1	13	9
5	5	10	15	7	2	13	8	14	11	4	1	9	12	3	6
6	6	12	10	11	13	7	1	5	3	9	15	14	8	2	4
7	7	14	9	15	8	1	6	13	10	3	4	2	5	12	11
8	8	3	11	6	14	5	13	12	4	15	7	10	2	9	1
9	9	1	8	2	11	3	10	4	13	5	12	6	15	7	14
10	10	7	13	14	4	9	3	15	5	8	2	1	11	6	12
11	11	5	14	10	1	15	4	7	12	2	9	13	6	8	3
12	12	11	7	5	9	14	2	10	6	1	13	15	3	4	8
13	13	9	4	1	12	8	5	2	15	11	6	3	14	10	7
14	14	15	1	13	3	2	12	9	7	6	8	4	10	11	5
15	15	13	2	9	6	4	11	1	14	12	3	8	7	5	10

Table 3: Multiplication in GF (2^4)

Multiplication takes place on the 4-bit binary values and then the modular reduction is performed on the binary product.

The Binary representation of $P(x) = (X^4+X+1) = (1101)$.

The modular multiplication in binary can be performed as illustrated in Table 3.

For example: If A=9 and B=9, then

$AXB = 9 \times 9 = (1001) \times (1001) = (1010001)$

$(1010001) \bmod (1101) = (1011) = 13$

Exponential operation performed using GF (2^m) is shown below

$5^7 = (5 \times 5 \times 5 \times 5 \times 5 \times 5 \times 5) \text{ GF } (2^4)$

$= (2 \times 2 \times 2 \times 5) \text{ GF } (2^4)$

$= (4 \times 10) \text{ GF } (2^4)$

$=14$

Section 2 discusses briefly describes the proposed algorithm. Section 3 discusses the results obtained and Section 4 is the conclusion arrived about the proposed work.

III. PROPOSED RECONSTRUCTION OF VIDEO FRAMES

The proposed approach consists of the following five broad steps: (1) Construction of Parity matrix P, to obtain the Generator matrix G for the purpose of encoding at the sender's end(2) Reading the YUV Video (3) Selecting the pixels for Encoding (4) Decoding (5) Parallelization for reducing computational complexity.
Each of these are discussed in the following sections

3.1 CONSTRUCTION OF PARITY MATRIX

Construction of LDPC codes are based on Cubic Bezier curve over Galois fields GF (2^m). The systematic generator matrix G is defined as [P I_K] that is used for the purpose of encoding. While the parity check matrix H defined as [I_{N-K} P^T] is used for the purpose of decoding.
The coefficients of the Bezier curve over GF (2^m) are used for the construction of Parity matrix. The first row of the Parity matrix is the coefficients of the Cubic Bezier curve over GF (2^m). The Remaining rows are obtained by shifting the coefficients of the Bezier curve to the left using shift register. The Parity matrix so obtained is of size 2^m X2^m.
The parity matrix P has the following important properties.

- Any i^{th} row or j^{th} column is the transpose of the other.
- Also (P.P^T) over GF (2^m) = $I_{2\wedge m}$; where P^T is the transpose of P.

The systematic generator matrix G is obtained by appending $I2^m$ which is 2^mX2^m Identity matrix, to the Parity matrix P, thereby making the generator matrix of size 2^mX2^{m+1}.

3.2 SELECTIVE ENCODING

Let MXN be the size of the Video Frame to be encoded and MXN1 be the encoded Video Frame. In selective encoding, the encoded data is generated by multiplying the selected digits of the message and the generator matrix G over Galois field GF (2^m).
This encoded data is obtained by performing modular multiplication of the selected N digits of data and the constructed generator matrix G over Galois field GF (2^m). The code word C is given by C= [D][G]
These N1 digits of code vector C is of the form C= [C1 C2 C3 …… C512] where the first 2^m digits of codeword C are the checksum produced and the remaining 2^m digits is the information.
Figure 5 shows the flow chart used in Selective encoding. An uncompressed raw video is accepted as an input and is converted to .mpg format using the FFMPEG line command tool. The number of video Frames, of this .mpg video is determined. Further, the size of each video Frame is determined. Processing on each Video Frame is performed parallely. In every i^{th} row of the Video Frame, if two consecutive data digits data(i) and data(i+1) are the same, then the first digit data(i) is replaced with a zero. This process continues till all the 2^m digits of data have been checked for repetition with its adjacent value. This selected data denoted as Ds has zeros when adjacent values are the same. This selected data Ds is encoded, by performing modular multiplication of Ds and the Generator matrix G.

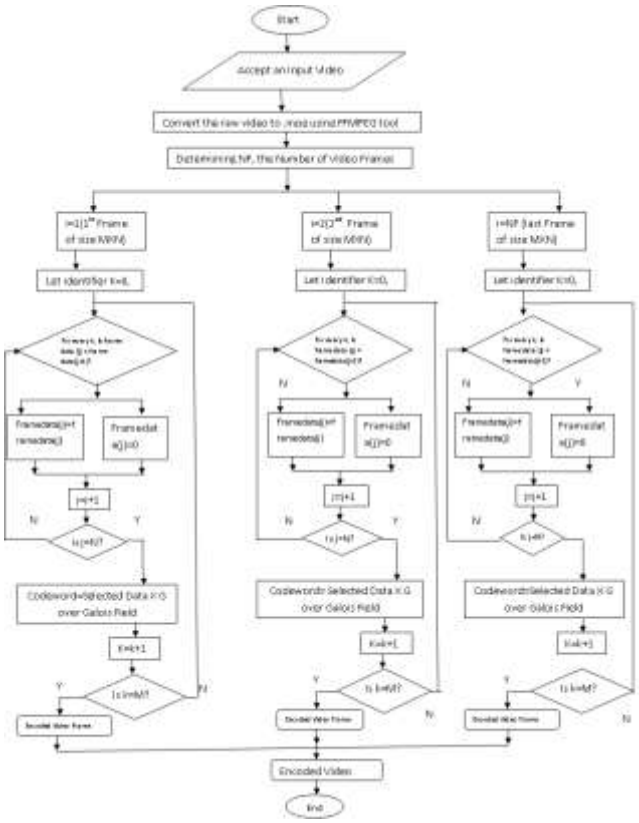

Figure 5: Selective Encoding in Video frames

In Selective encoding, the repeated data is replaced by zero. The following example illustrates the methodology used in selective encoding. Let D be the data that needs to be encoded and the pixel values form 200-215 that needs to be encoded is chosen from one of the rows of a Video Frame. D= [200, 201, 201, 201, 201, 201, 205, 206, 208, 209, 210, 210, 210, 210, 211, 215]. After selecting the data to be encoded, the repeating pixel values are replaced by zeros leading to Ds= [200,0,0,0,0,201,205,206,208,209,0,0,0,210,211,215].
Ds, when multiplied with G over GF (2^m), give the codeword C. It is seen that in selective encoding, replacing the repetitive pixel values by zero, reduces the number of multiplication operation leading to increase in speed of encoding.

3.3 PROPOSED DECODING

The decoder has to recover the original data from the received code vector sent by the transmitter, without requesting for re-transmission. To recover the original data without re-transmission being performed, the FEC Codes are applied. In LDPC, each row of the encoded Video Frame has 2^{m+1} digits of data that is given as input to the decoder which consist of 2^m digits of checksum and 2^m digits of

information. The Received vector R has 2^m digits of checksum and 2^m digits of information.

The decoder calculates the syndrome after obtaining this received vector. This syndrome S is calculated using $S=R.H^T$ by performing modular multiplication over GF (2^8), where R is the received vector and H is the parity check matrix. The syndrome is 2^m digits denoted as S= [S1 S2 S3 S4… $S2^m$] where S € GF (2^m).

If the Syndrome S is Zero, then the received vector is error free else, the decoder determines the location of the error. The error location is determined by referring to the Parity Check matrix H.

To illustrate this with an example, if the Syndrome S1 is Zero and the syndrome digits [S2...S256] is Non-zero, then according to the Parity Check matrix H, the received data has errors in the position 107 to 256. These P digits of errors can be corrected using the checksum equation C256=2RI2+ 6RI3+ 7RI4+10RI5+15RI6+17RI7+ 21RI8+ 23RI9+ 31RI10 + 32RI11 + 36RI12 +…. 252RI105 + 255RI106.

The following Table 4 illustrates some of the Zero Syndrome value and the possible error location, which is determined using the parity check matrix.

Zero Syndrome	Error in position
S1	(RI107- RI256)
S8	(RI100-RI249)
S16	(RI92-RI241)
S32	(RI76-RI225)
S64	(RI64-RI193)
S128	(RI1-RI129) &(RI236-RI256)
S255	(RI1-RI2)&(RI109-RI256)

Table 4: Syndrome values

.

These erroneous digits can be corrected by determining the first zero syndrome digit. The checksum corresponding to the identified zero syndrome digit is used to correct errors.

After correcting the errors, the consecutive zeros will be replaced by the right most non zero pixel value. The following example illustrates the removal of consecutive zeros after the error correction is performed by the decoder.

If the corrected vector Vc is obtained as [200,0,0,0,0,201,205,206,208,209,0,0,0,210,211,215] , then by replacing all zeros with the right most non zero value the final decoded vector V would be, V=[200, 201, 201, 201, 201, 201, 205, 206, 208, 209, 210, 210, 210, 210, 211, 215].

Mean Square Error can be calculated between the Data D that was encoded selectively before transmission and the decoded data V at the receiver, to check for equality.

The speed of error correction is increased, as the repeating consecutive pixel values are replaced by zero.

Figure 6 shows the flowchart of the proposed decoding. Each Video Frame is processed Parallely. The syndrome is calculated for every i^{th} row of every received Video Frame. If the Syndrome S is Zero, then the received vector is error free else, the decoder determines the location of the error, by determining the first zero syndrome digit. The checksum equation corresponding to the zero syndrome digit is used to correct the errors. This corrected data is denoted as V. Thus each received video frame is made error free.

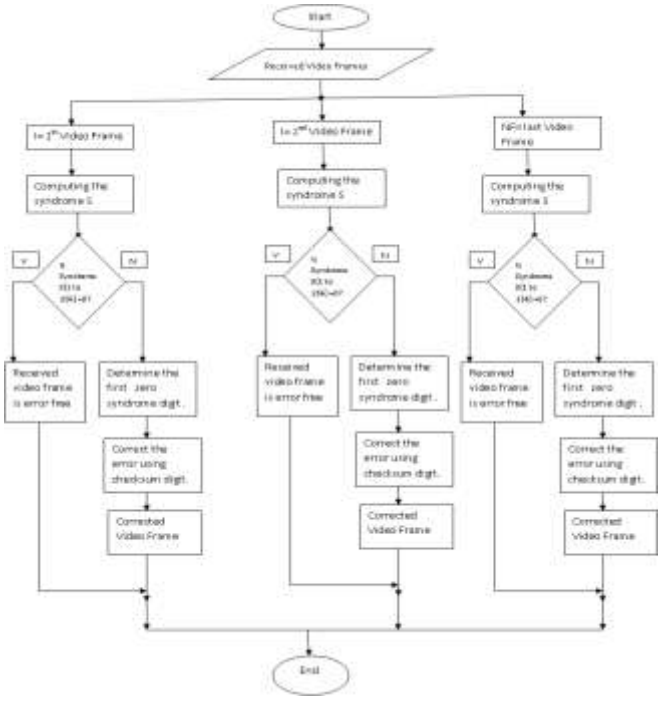

Figure 6: Proposed Decoding

The figure 7 shows the pseudo code for correcting errors in M video frames when any of the Syndrome digits between S4 to S106=0

```
parfor ie=1:M /* using parallel for loop for M video frames
for i=1:103-ki+k2
   tempz2(i+1)= atable(tempz2(i)+1,tempz1(i)+1);
end
 if((s(inew,3+ki)==0)&(tempz2(104-
ki+k2))~=cv{ie}(inew,3+ki-k2))
for i=1:103-ki+k2
   tempz4(i+1)=atable(tempz4(i)+1,tempz3(i)+1);
end
tempz5=atable(tempz4(104-ki+k2)+1,r(inew,3+ki-k2)+1);
tempz6=1;
tempz7=1;
while(tempz7~=0)
   tempz8=ptable(255,tempz6);
   tempz7=atable(tempz5+1,tempz8+1);
   tempz6=tempz6+1;
  if(tempz6==256)
   tempz6=1;
  break;
  else
  continue;
  end
```

 ISSN: 2321-8169
133 – 143

```
  end
r(inew,360-ki+k2)=tempz6-1;
else
r(inew,360-ki+k2)=r(inew,360-ki+k2);
  end
end
```

/*All the additions and multiplications are over Galois field GF(2^8)*/

Figure 7: Pseudo code

The proposed algorithm for encoding and decoding is verified for a Video Frame by taking different cases and is discussed in the following section.
If the Video Frame to be transmitted is denoted as V, then the encoded Video Frame denoted as CV is obtained by performing matrix multiplication of the selected pixel values of every Video Frame and the Generator matrix G, over GF (2^8). The encoded Video Frame is transmitted by the sender. The receiver calculates the syndrome to know whether the received Video Frame is modified. If the Syndrome is zero, then the Video Frame received is error free else the Video Frame has to be recovered from the erroneous Video Frame.
The Mean Square Error is calculated between the Original Video Frame and the decoded Video Frame. The Mean Square Error (MSE) measures the difference between the Original Video Frame and the estimated Video Frame.

The MSE is given $\frac{1}{n}\sum_{i=1}^{n}(\hat{Xi} - Xi)^2$ where $\hat{Xi}$ are the pixel values of the estimated Video Frame and Xi are the pixel values of the Original Video Frame.

The coding for the proposed algorithm is done in MATLAB. BER is considered to draw conclusions about the algorithm. The Bit Error Rate (BER) is the number of bit errors divided by the total number of transferred bits during a time interval.

3.4 PARALLELIZATION

Handling video files is computationally much more intensive as compared to images. When a large dataset is used, the implementation and the experimentations involve considering the time complexity. Parallel computing is a solution to this problem.
In MATLAB parallel processing tool box, the basic parallelization approach uses the parfor construct to execute independent passes through a loop [18]. Each variable in the parfor loop are classified as shown in Fig. 8. In the code snippet shown in Fig. 8, the loop variable represents the loop index. Sliced variables are arrays whose segments are operated on in different iterations of the loop. Broadcast variables are defined before the loop begins, and are required within the loop, but never assigned values inside the loop.
Reduction variables are those which accumulate values across multiple iterations of the loop, regardless of iteration order. Temporary variables are created inside the loop, and never accessed outside the loop.

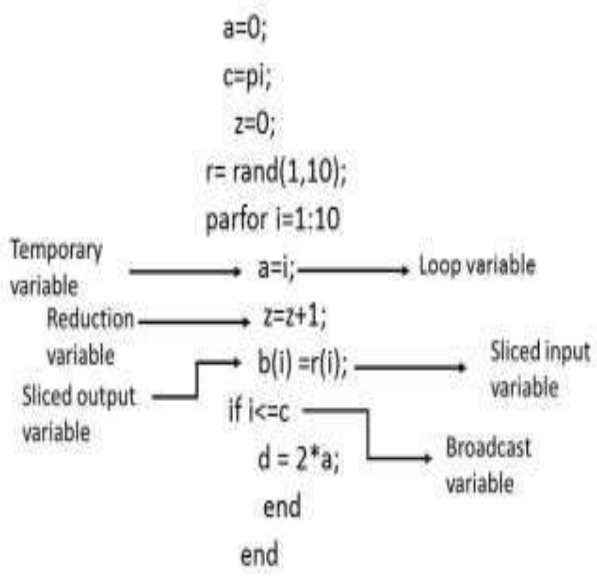

Figure 8: Classification of variables

To improve the time complexity of the proposed method, the MATLAB parfor construct is used, as discussed above, in our test videos, so that multiple Video frames are handled by multiple workers in parallel.

The coding for the proposed algorithm is done in MATLAB. The video considered for experimental purpose is foreman.yuv. This .yuv video is converted to .mpg using line Command tool FFMPEG. This .mpg video has a sequence of 152 frames, with each frame being a JPG image of size 352X288. BER is considered to draw conclusions about the algorithm. The Bit Error Rate (BER) is the number of bit errors divided by the total number of transferred bits during a time interval.

Case 1: Selective encoding
Here for the purpose of experimentation, noise has been introduced randomly. The BER is a parameter used to derive conclusion about the performance of the algorithm. 1st and 8th Video frame has been randomly chosen experimental purpose. The BER is taken as 612/1289 in the case of 1st Video Frame and 663/1408 in the case of 8th Video Frame.
Figure 9a, b, c, d shows the 1st Video Frame, 1st Encoded Video Frame, 1st modified Video Frame and the 1st Recovered Video Frame. Non repeating pixel values of the original Video Frame is encoded and transmitted. The encoded Video Frame is obtained by performing modular multiplication of the selected pixel values of the Video Frame with the generator matrix over GF (2^8). Error has been introduced randomly. At the receiver's end, a modified Video Frame due to the introduced error has been obtained. Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original Video Frame.

 ISSN: 2321-8169
133 – 143

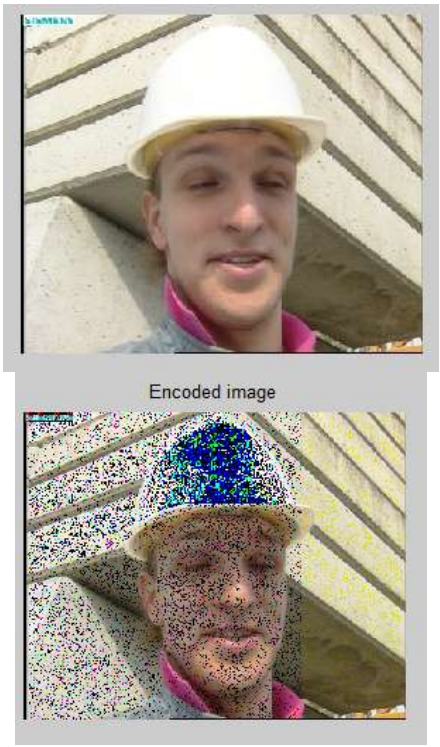

Figure 9a: 1st video frame; Figure 9b: Selectively encoded 1st Video frame

Figure 9c: Received 1st video frame; Figure 9d: Corrected 1st Video frame

Figure 9b shows the encoded Video Frame that needs to be transmitted. The dots in the encoded Video Frame indicate the repeating pixel values that have been replaced by zeros. Figure 9c, shows the modified Video Frame (the darkened portion of the Video Frame) has non zero syndrome values from S108 to S168 and S228 to S240 and has been eliminated using the decoding algorithm. 48,132, missing pixel values are corrected using the proposed algorithm.

Figure 10a, b, c, d shows the 8th Video Frame, 8th Encoded Video Frame, 8th modified Video Frame and the 8th Recovered Video Frame.

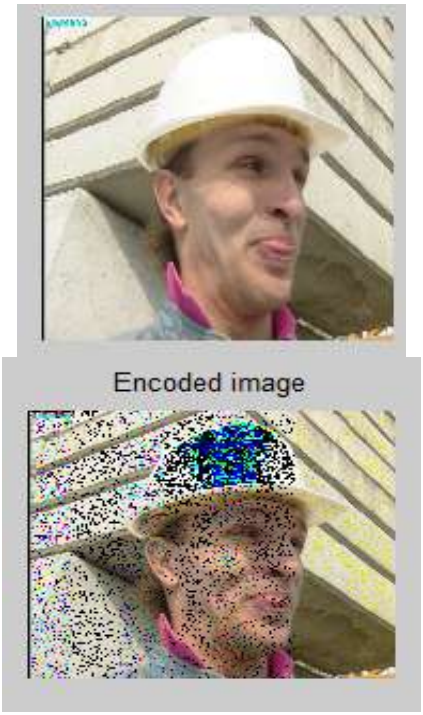

Figure 10a: 8th video frame; Figure 10b Encoded 8th Video Frame

Figure 10c: Received 1st Video Frame; Figure 10d: Corrected 1st Video Frame

Case2: With Selective Encoding

Here for the purpose of experimentation, White Gaussian Noise with value of mean=0.3 and variance=0.1 has been introduced. The BER is a parameter used to derive conclusion about the performance of the algorithm and is taken as 1000/2048 in this case.

The proposed algorithm is tested for various values of mean μ and variance σ^2.

141st Video frame has been randomly chosen experimental purpose.

Figure 11a, b, c, d shows the 141st Video Frame, 141st Encoded Video Frame, 141st modified Video Frame and the 141st Recovered Video Frame. The original Video Frame is selectively encoded and transmitted. The encoded Video Frame is obtained by performing modular multiplication of the non repeating pixel values of the Video Frame with the generator matrix over GF (2^8). Error has been introduced by white Gaussian noise. At the receiver's end, a modified Video Frame due to the introduced error has been obtained. Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original Video Frame.

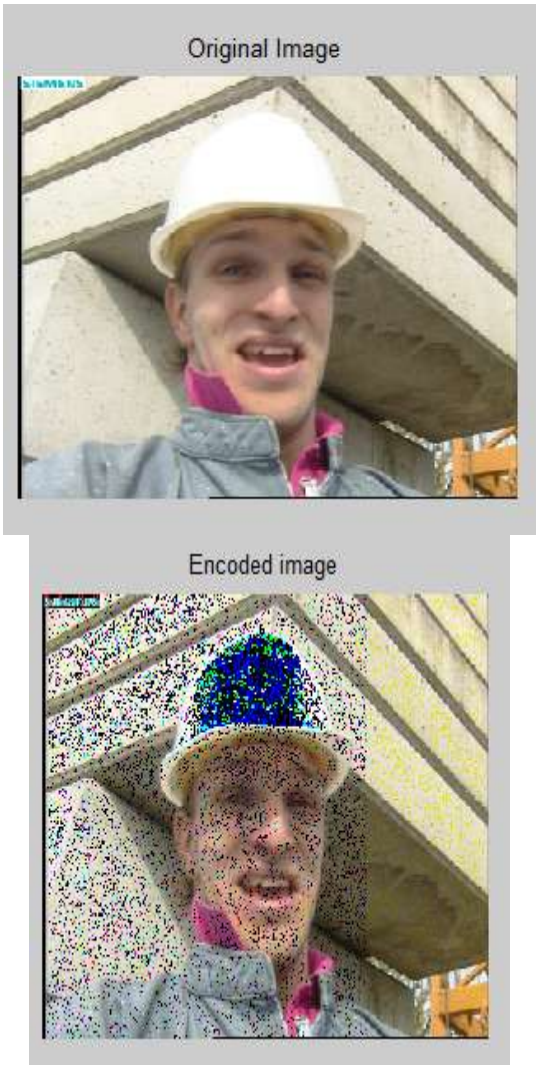

Figure 11a: 141st Video Frame; Figure 11b: 141st selectively encoded Video frame

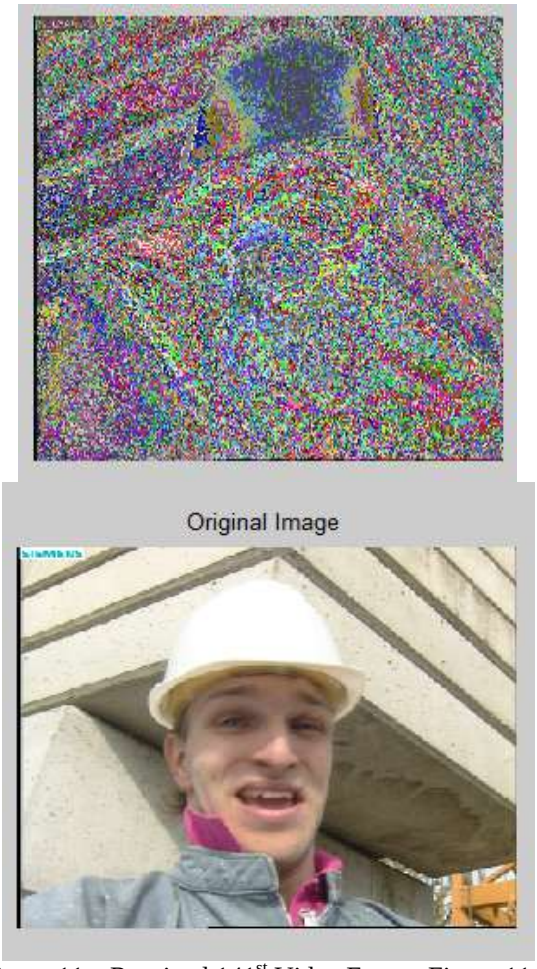

Figure 11c: Received 141st Video Frame; Figure 11d: Reconstructed 141st Video frame

Figure 11a, shows the 141st video frame. Figure 11b shows the selectively encoded Video Frame. Figure 11c, shows the 141st video frame that is received and has 49,500 missing pixel values and has been corrected using the decoding algorithm. This reconstructed 141st Video Frame is shown in Figure 11d.

A mean square error of zero is obtained between the original video frame and the encoded video frame that implies the corrected video frame to be same as the original video frame. Figure 12 shows the mean square error between the original 141st video frame and the encoded 141st video frame, as well as the mean square error between the original video frame and the corrected video frame.

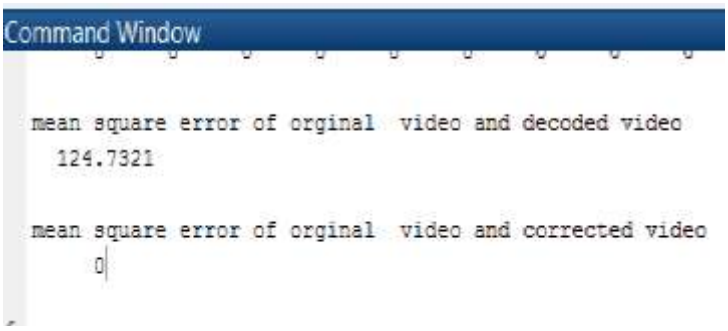

Figure 12: Mean Square Error of Original Video Frame and Corrected Video frame.

From figure 12, the Mean Square Error between the original video frame and the encoded video frame is close to 125, which implies many of the repeating consecutive pixel values are replaced by zeros. The Mean Square Error between the original video frame and the Corrected video frame is 0, indicating both the video frames to be the same, after performing selective encoding and selective decoding.

In Selective encoding, the repeating consecutive values are replaced by zeros.

IV. RESULTS

Figure 13 shows a plot of the execution times for different values of BER with using selective encoding and without using selective encoding. When Selective encoding is used, the repeating consecutive pixel values are replaced by zeros. The number of modular arithmetic operations gets significantly reduced. Thus the decoding time for selective encoding is lesser compared to the method which does not use selective encoding.

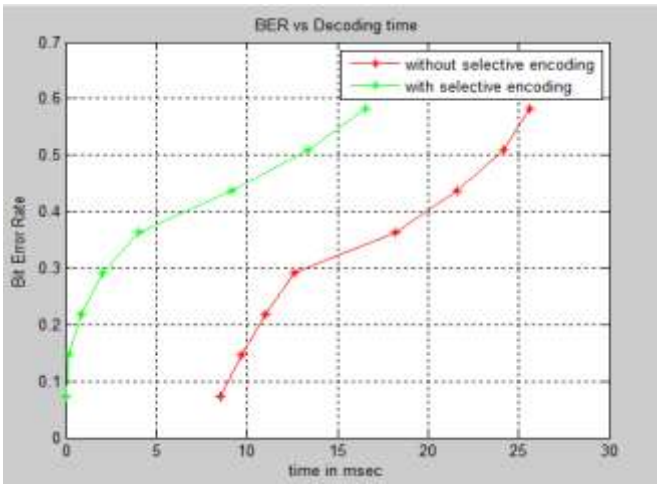

Figure 13: Comparison between Selective Encoding and without selective encoding.

Table 5 shows the significance of using parallelization to improve the time complexity of the decoder.

Data	Selective Encoding Without parallelization	Selective Encoding With Parallelization using Workers in MATLAB
Video1 with 60 frames	220.8333mSec	3.680mSec
Video2 with 100 frames	368.0555mSec	5.52mSec
Video3 with 152 frames	568.992mSec	7.48mSec

Table 5: Comparative execution times (msec) for selective encoding of video frames with Parallelization and without Parallelization

From Table 5, it can be inferred that the proposed parallelization approach has helped to improve the time complexity while reconstructing the video frames.

Figure 15 shows the graph of the execution time with and without parallelization.

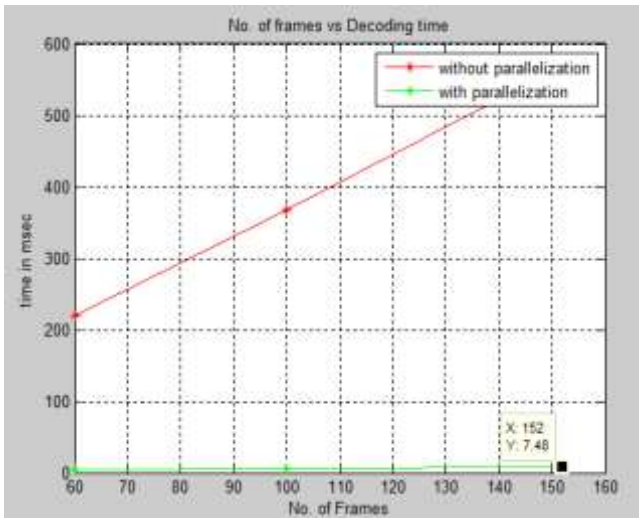

Figure 15: Comparative execution times, with and without parallelization

It can be inferred that using Parallelization, selectively encoding the Video Frames reduces the execution time significantly.

V. CONCLUSION

This paper, establishes the working of Selective Encoding using LDPC with cubic Bezier curve over Galois field GF (2^8). Bezier curves are used for the construction of the generator matrix G and parity check matrix H. The generator matrix is used for encoding while the Parity check matrix is used for decoding.

The proposed decoder is able to detect and correct errors in Video frames. These video frames of size 352X288 can be reconstructed from modified Video frames, which have 52,800 missing pixel values.

The proposed algorithm uses Selective Encoding, where in the repeating consecutive pixel values of the video frame are replaced by zeros. Using this approach, it is possible to encode a few non zero pixel values. The Encoding involves modular arithmetic operations. At the receiver, decoding a few pixel values still preserves the concept of Error detection and correction.

Selective encoding speeds up the encoding and decoding process as the repeating consecutive pixel values are replaced by zeros, thus enhancing the speed of communication.

It is found that this method is more convenient as the encoding and the error correction involves modular addition over Galois field and also reduces the hardware complexity.

Parallelization reduces the time complexity due to the usage of parfor loop construct in MATLAB.

REFERENCES

[1] Costello Daniel J, J., Imai H, Wicker S.B," Applications of Error-Control Coding", IEEE Transactions on Information Theory, Vol. 44, No. 6, October 1998 .

[2] Ph.D thesis onby Azmi, M.H, UNSW "Design of low-density parity-check codes in relay channels" Electrical Engineering & Telecommunications, 2012, Faculty of Engineering .

[3] Sheryl L. Howard , C.S.a.K.I, "Error control coding in wireless sensor networks" EURASIP Journal onWireless Communications and Networking, DOI 10.1155/WCN/2006/74812, Volume 2006, Article ID 74812, Pages 1-14.

[4] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," SI AM Journal of Applied Mathematics, 1960, Volume 8, (pp. 300-304).

[5] Sanjeev kumar, R.G., " Performance Comparison of Different Forward Error Correction Coding Techniques for Wireless Communication Systems " (I S S N : 2 2 2 9 - 4 3 3 3 (P r i n t) | I S S N : 0 9 7 6 - 8 4 9 1 (On l i n e)) .

[6] V.S.Ganepola et.al "Performance study of non-binary LDPC codes over Galois field" CSNDSP08, IEEE, 2008

[7] Padmini U Wasule, Shubhagini Ugale," Review paper on decoding of LDPC codes using Advanced Gallagers algorithm", IJAICT Volume 1, Issue 7, November 2014.

[8] AlinSindhu A " Galois field based very fast and compact error correcting technique" Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 4, Issue 1(Version 4), January 2014, pp.94-97.

[9] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low density parity check nodes based on belief propagation," IEEE Trans. on Communication., vol. 47, no. 5, pp. 673- 680, May 1999.

[10] J P chen and M P C Fossorier" Density evolution for two improved BP-based decoding algorithm for LDPC codes" , IEEE Communication letters, vol 6, no.5, pp 208-210, May 2002

[11] Meng Xu, Jianhui Wu, Meng Zhang, "A modified offset Min-sum decoding algorithm for LDPC codes", 3rd IEEE International Conference on computer science and information technology, (ICCSIT), vol 3, 2010.

[12] Jaehong Kim, Aditya Ramamoorthy, "The Design of Efficiently Encodable Rate- Compatible LDPC Codes"IEEE transactions on communications, vol.57, no. 2, February 2009.

[13] L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF (2^q)," The Proc. 2003 Inform. Theory Workshop, pp. 70–73, 2003

[14] Robert Gallager, "Low Density Parity Check Codes ",[On-line] http://www.rle.mit.edu/rgallager/documents/ldpc.pdf

[15] Shu Lin, D.L.C., "Error Control coding fundamentals and application", 2nd Edition, Editor, Prentice Hall series in computer application in Electrical Engineering.

[16] Weisstein, Eric W, "Bézier Curve", From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/BezierCurve.html

[17] The Encyclopedia of design theory: Galois fields by Peter J.Cameron, May 30, 2003.

[18] World wide web: http://in.mathworks.com/help/distcomp/parfor.html