# An Efficient Video Steganography Algorithm Based on BCH Codes

**Ramadhan J. Mstafa and Khaled M. Elleithy**

*Department of Computer Science and Engineering*
*University of Bridgeport*
*Bridgeport, CT 06604, USA*
rmstafa@my.bridgeport.edu elleithy@bridgeport.edu

## Abstract

In this paper, in order to improve the security and efficiency of the steganography algorithm, we propose an efficient video steganography algorithm based on the binary BCH codes. First the pixels' positions of the video frames' components are randomly permuted by using a private key. Moreover, the bits' positions of the secret message are also permuted using the same private key. Then, the secret message is encoded by applying BCH codes (*n, k, t*), and XORed with random numbers before the embedding process in order to protect the message from being read. The selected embedding area in each Y, U, and V frame components is randomly chosen, and will differ from frame to frame. The embedding process is achieved by hiding each of the encoded blocks into the 3-2-2 least significant bit (LSB) of the selected YUV pixels. Experimental results have demonstrated that the proposed algorithm have a high embedding efficiency, high embedding payload, and resistant against hackers.

## Keywords

Video Steganography, BCH Codes, Linear Block Code, Embedding Efficiency, Embedding Payload.

## Introduction

Due to technological advances and the speed of the Internet, people are concerned that their personal information will be stolen by hackers. In today's society, many data hiding algorithms and steganographic algorithms have been introduced in order to protect valuable information. Steganography is one of the methods that protects and hides valuable data from unauthorized people without hackers having any suspicion of the data's existence. The Human Visual System (HVS) cannot recognize a slight change that occurs in the cover data such as audio, image and video[1,2]. Unfortunately, many strong steganography analyzing tools have been provided to unauthorized users in order for them to retrieve valuable secret data previously embedded in cover objects. The weakness of some steganography algorithms occur through steganalytical detectors because of the lack of security and embedding efficiency in these algorithms.

The embedding efficiency and the embedding payload are two important factors that every successful steganography system should take into consideration[3]. First, the steganography scheme that has a high embedding efficiency translates to a good visual quality of stego data and a less amount of host (carrier) data are going to be changed[4]. Any obvious distortion to the viewers will increase the probability of the attacker's suspicion, and the secret information can be easily detected by some of the steganalysis tools[5]. These kinds of schemes are difficult to detect

by the steganalytical detectors. The security of the steganography scheme depends directly on the embedding efficiency[6]. Second, a high embedding payload means that the capacity of secret information to be hidden inside host data is large. These two factors, embedding efficiency and embedding payload, contradict one another. Once, the data embedding efficiency is increased, the data embedding payload is decreased. These two factors will change depending on the users' requirements and the type of steganography scheme[4,7]. The remainder of the paper is organized as follows: Section 2 presents some related work. Section 3 introduces an overview of the Linear Block Code and BCH codes, and then presents the proposed steganography algorithm. Section 4 presents experimental results and discussion. Section 5 provides the conclusion.

## Related Work

Feng et al. proposed a novel of a video steganography scheme based on motion vectors as carrier data in order to embed the secret message in H.264 video compression processing. The algorithm also uses the principle of linear block codes to reduce motion vectors' modification rate. The algorithm has a good visual quality of stego data, which is proved by the low modification rate of motion vectors. The Peak Signal to Noise Ratio that was obtained in both *Flower* and *Foreman* videos are more than 37 dB[8]. Hao et al. proposed a novel video steganography method based on a motion vector by using matrix encoding. A motion vector component that has high amplitude among both horizontal and vertical components is chosen to embed the secret message. The HVS can identify the change that occurs when the object is moving slowly. However, if the same object moves quickly, the HVS will not be able to recognize the change that occurs. Motion vectors with large sizes are selected for embedding the secret messages. The macro blocks that are moving quickly will generate motion vectors with large amplitudes. The direction of macro blocks depends on the motion vectors' components. For example, if the vertical component is equal to zero, then the macro block direction is moving vertically. The visual quality of the tested videos that were obtained is more than 36 dB[9]. Rongyue et al. proposed an efficient BCH coding for steganography which is embedding the secret information inside a block of cover data by changing some coefficients. Authors have improved the computational time of the system and the complexity becomes low because of the system's linearity[10]. Liu et al. proposed a robust data hiding scheme in an H.264 compressed video stream, where they have prevented a drift of intra-frame distortion. To give the system more robustness, the authors have encoded the message using BCH codes before the embedding process. The host data is the DCT coefficients of the luminance Intra frame component. The obtained results have a high visual quality and robustness against hackers[11].

## The Proposed Steganography Algorithm

The proposed algorithm uses an uncompressed video stream which is based on the frames as still images. The video sequences are divided into frames, and each frame's color space is converted to *YCbCr*. The reason for using *YCbCr* color space is to remove the correlation between Red, Green, and Blue colors. The luminance (*Y*) component represents the brightness data, which the human eye is more sensitive to than the other color components. As a result, the chrominance (*CbCr*) components can be subsampled in the video stream and some information might be discarded.

*A. Linear block codes*

Any specific block code is defined as a linear block code if the sum of any two codewords equals a new codeword. Furthermore, a binary linear block code includes a linear block code that contains a block of binary bits. A binary linear block code (*n*, *k*) consists of $2^{\mathrm{k}}$ columns and $2^{\mathrm{n-k}}$ rows in a linear code array. This code is an *n*-dimensional vector space and a *k*-dimensional subspace $V_n = \{(C_0, C_1, C_2, \dots, C_{n-1}) | C_j \epsilon\, GF(2)\}$. *N* refers to the length of the codeword and *K* refers to the number of symbols in each codeword. In the standard array, it is not possible for two equal vectors to exist in the same row. For example, if *C* is a (*n*, *k*) code on the Galois Field GF (2), then:

- Each coset has $2^{\mathrm{k}}$ vectors.
- All *F* vectors of length *n* belong to coset of *C*.
- If *C*+*Z* is a coset of *C* and *F* as belonging to (*C*+*Z*), then *C*+*Z*=*C*+*F*.

*B. BCH codes (7, 4, 1)*

Bose, Chaudhuri, and Hocquenghem invented the BCH encoder. It is one of the most powerful random cyclic code methods, which can be used for detecting and correcting errors in a block of data. The BCH codes are different from the Hamming codes because BCH can correct more than one bit. A binary BCH (*n*, *k*, *t*) can correct errors of a maximum *t* bits for codewords of the length *n* $(\mathrm{c}_0, \mathrm{c}_1, \mathrm{c}_2, \dots, \mathrm{c}_{\mathrm{n-1}})$ and message length *k* $(\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_{\mathrm{k-1}})$. Encoded codewords and messages can both be interpreted as polynomials, where $\mathrm{a(x)} = \mathrm{a}_0 + \mathrm{a}_1\mathrm{x}^1 + \cdots + \mathrm{a}_{\mathrm{k-1}}\mathrm{x}^{\mathrm{k-1}}$, and $\mathrm{c(x)} = \mathrm{c}_0 + \mathrm{c}_1\mathrm{x}^1 + \cdots + \mathrm{c}_{\mathrm{n-1}}\mathrm{x}^{\mathrm{n-1}}$. When *m* and *t* are any positive integers where $(\mathrm{m} \geq 3)$and $(\mathrm{t} < 2^{\mathrm{m-1}})$, there will be a binary BCH codes with the following properties:

- Block codeword length $n = 2^m - 1$
- Message length $k$
- Maximum correctable error bits $t$
- Minimum distance $d_{min} \geq 2t + 1$
- Parity check bits $n - k \leq mt$

The BCH codes inventors decided that the generator polynomial g(x) will be the polynomial of the lowest degree in the Galois field GF (2), with $\propto, \propto^2, \propto^3, \dots, \propto^{2\mathrm{t}}$ as roots on the condition that $\propto$ is a primitive of $\mathrm{GF}(2^{\mathrm{m}})$. When $\mathrm{M_i(x)}$ is a minimal polynomial of $\propto^{\mathrm{i}}$ where $(1 \leq \mathrm{i} \leq 2\mathrm{t})$, then the least common multiple (*LCM*) of 2*t* minimal polynomials will be the generator polynomial g(x). In this paper, the BCH codes (7, 4, 1) is used. The parity-check matrix *H* of the BCH codes[12,13] is described as follows:

$$H = \begin{bmatrix} 1 & \propto & \propto^2 & \propto^3 & \dots & \propto^{n-1} \\ 1 & (\propto^3) & (\propto^3)^2 & (\propto^3)^3 & \dots & (\propto^3)^{n-1} \\ 1 & (\propto^5) & (\propto^5)^2 & (\propto^5)^3 & \dots & (\propto^5)^{n-1} \\ . & . & . & . & \dots & . \\ . & . & . & . & \dots & . \\ . & . & . & . & \dots & . \\ 1 & (\propto^{2t-1}) & (\propto^{2t-1})^2 & (\propto^{2t-1})^3 & \dots & (\propto^{2t-1})^{n-1} \end{bmatrix} \quad (1)$$

*C. Data embedding stage*

Data embedding is the process of hiding a secret message inside cover videos. This process converts the video stream into frames. Each frame separates into the Y, U and V components of color space. For security purposes, the pixels' positions of Y, U, and V components are permuted by using a special key ($Key_1$). Also, characters of the secret message are converted into an array of binary bits. In order to change the bits' positions of the secret message, the entire bits' positions within the array are permuted using $Key_1$. After permutation, the array is divided into 4-bit blocks. Then, each block is encoded by the BCH (7, 4, 1) encoder. The outcome of the 7-bit encoded block (consists of 4-bit message and 3-bit parity) is XORed with the 7-bit number. These numbers are randomly generated by using $Key_2$. In order to select the locations for hiding the secret message into the frame components, $Key_2$ is utilized. In other words, $Key_2$ chooses random rows and columns for data embedding in each disordered Y, U, and V component. The embedding process is achieved by hiding each of encoded blocks into the 3-2-2 LSB of the selected YUV pixels. The pixels of the YUV components will be repositioned in order to the original frame pixel positions to produce the stego frames. Finally, the stego video is constructed from these stego frames. The block diagrams of the data embedding stage and the data extracting stage are illustrated in Fig. 1 and Fig. 2, respectively.

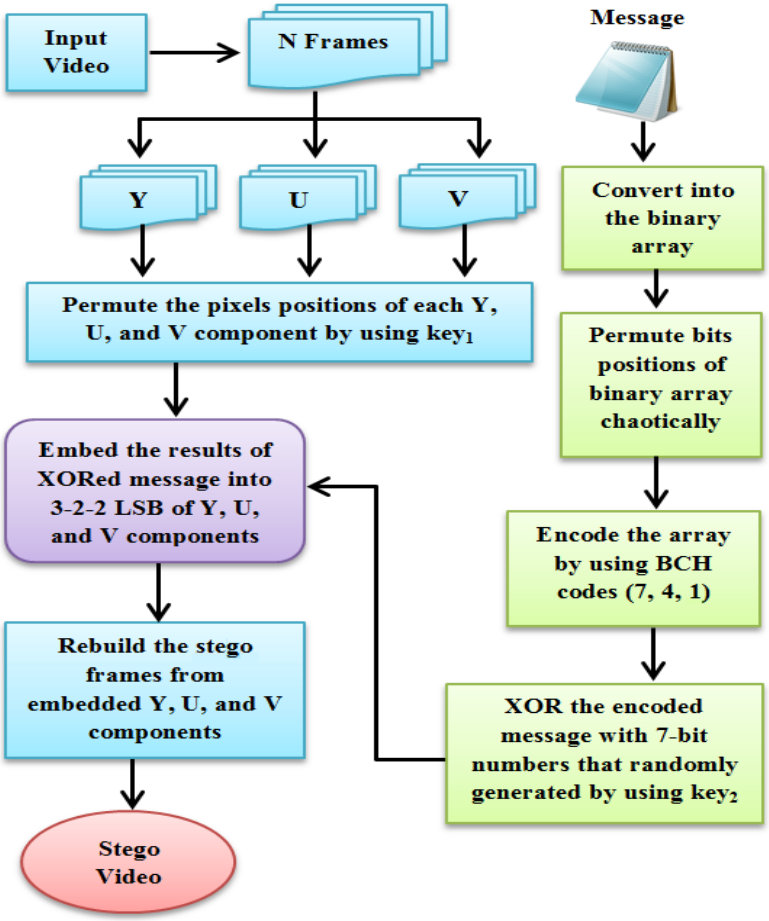


Fig. 1: Block diagram for data embedding stage.

*D. Data extracting stage*

Data extracting is the process of retrieving the secret message from the stego videos. This process is achieved by converting the distorted videos into frames. Then, each frame is partitioned into Y, U and V components. In every Y, U, and V component, the pixels' positions are permuted by using $Key_1$. The process of extracting the secret message from YUV components is accomplished by taking out 3-2-2 LSB in each selected pixel. The obtained message block will be XORed with the 7-bit number that is generated by using $Key_2$. The outcomes of 7 bits are decoded by the BCH (7, 4, 1) decoder in order to produce 4-bit blocks. These blocks are stored into a binary array. Since the message of entire bits is permutated prior to the data embedding process, the permutation process of the entire binary array to the original bits order will be performed again. Then, the binary array of bits will be converted into the characters of the secret message. The purpose of using two keys and the XOR operation is to improve the security and robustness of the proposed algorithm. These keys are only shared between sender and receiver, and used in both the data embedding and extracting processes.

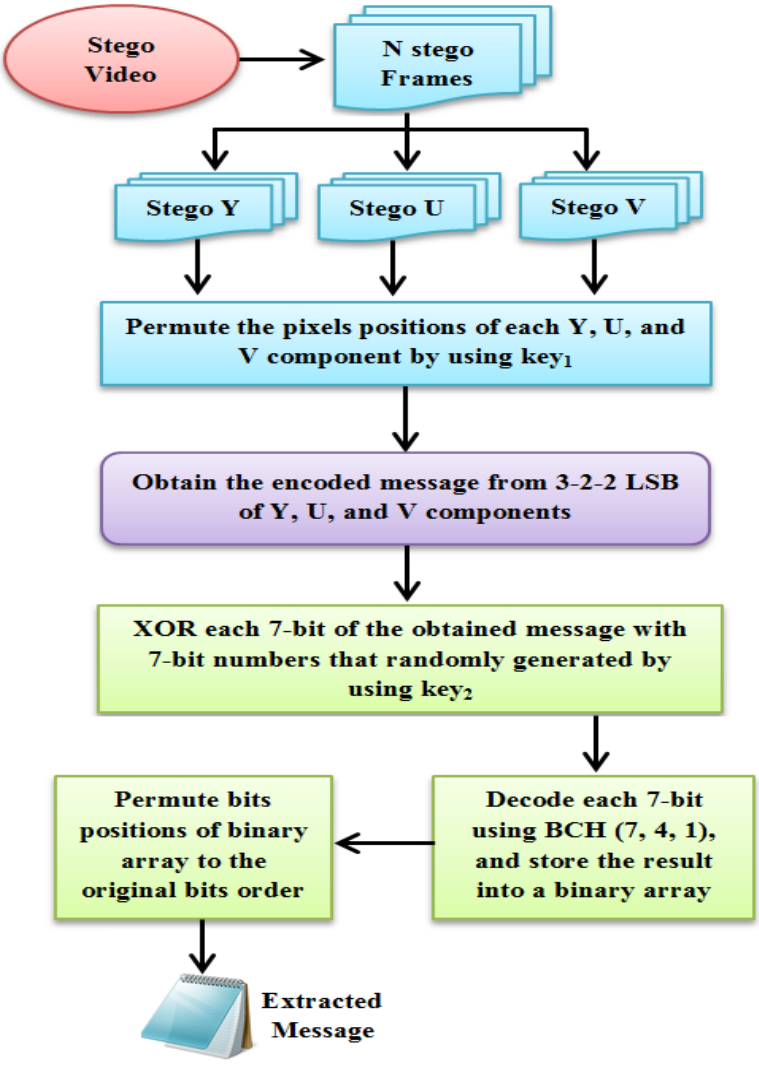


Fig. 2: Block diagram for data extracting stage.

## Experimental Results and Discussion

A database of eight standard of Common Interchange Format (CIF) video sequences is used, with the video resolutions equaled to (288 x 352), and the format represented by 4:2:0 YUV. Video sequences are equal in length to 300 frames. A text file consisting of alphabet characters is used as a secret message. This work is implemented using MATLAB program to test the proposed algorithm's performance. The Peak Signal to Noise Ratio (*PSNR*) is a visual quality measurement which is used to compute the difference between the original and the stego video frames. *PSNR* is calculated by the following equation:

$$PSNR = 10 * Log_{10}\left(\frac{{MAX_O}^2}{MSE}\right) \quad (2)$$

And Mean Square Error (MSE) is calculated as follows:

$$MSE = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[O(i,j) - S(i,j)]^2}{m * n} \quad (3)$$

Where *O* and *S* denote the original and stego YUV frame components, respectively, and *m* and *n* are the video resolutions.

Fig. 3 illustrates the *PSNR*s of 300 stego frames for the *Hall* video. In Fig. 4, the *PSNR*s of 300 *Stefan* stego video frames are shown. Fig. 5 illustrates the *PSNR*s of 300 stego frames for the *Foreman* video. By using our proposed algorithm, the obtained visual quality is similar to the original videos' visual quality. In general, *PSNR*s are greater than 55 dB, and the V component has a better visual quality among the three components. The reason of V component has a better visual quality among other components is because V component has a longest wavelength.

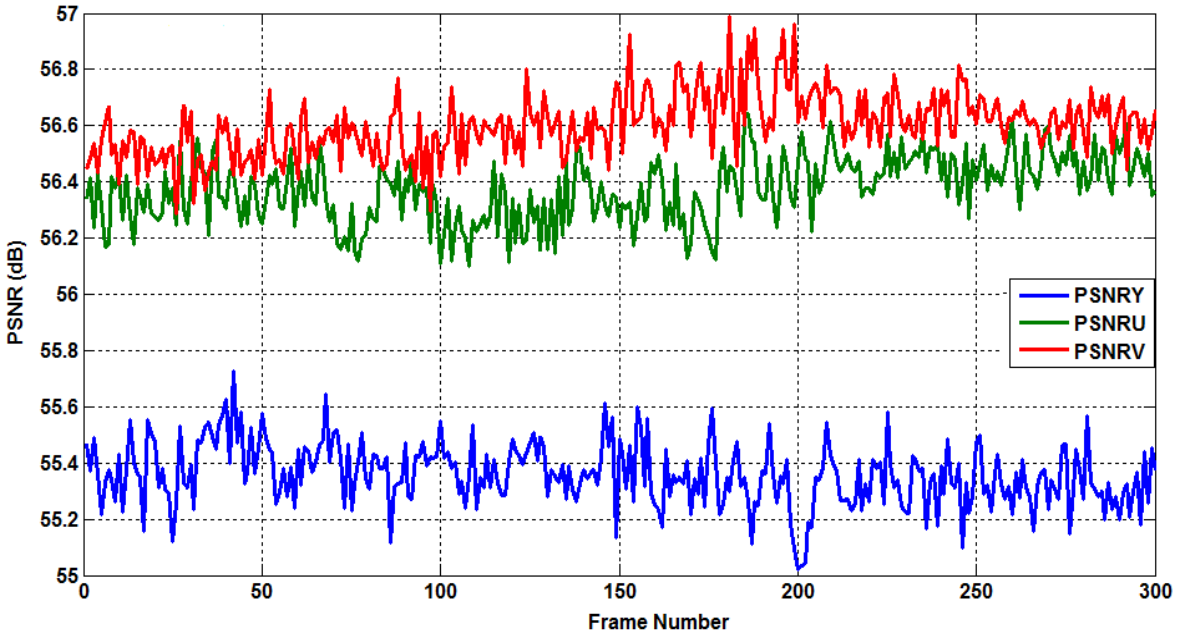


Fig. 3: *PSNR* of 300 stego frames for the *Hall* video.

In Table 1, the *PSNR* for eight video sequences is shown for each Y, U, and V component, separately. The visual quality of the stego videos is the same as the original videos' visual quality because all *PSNR* values are greater than 55 dB.

**TABLE 1 THE AVERAGES OF *PSNRY*, *PSNRU*, AND *PSNRV* FOR ALL VIDEOS**

| Sequences | Frame No. | *PSNRY* | *PSNRU* | *PSNRV* |
|---|---|---|---|---|
| ***Hall*** | *1-100* | **55.401** | **56.34** | **56.52** |
| | *101-200* | **55.357** | **56.324** | **56.649** |
| | *201-300* | **55.317** | **56.459** | **56.638** |
| ***Stefan*** | *1-100* | **55.381** | **56.388** | **57.031** |
| | *101-200* | **55.359** | **56.378** | **57.058** |
| | *201-300* | **55.359** | **56.343** | **57.028** |
| ***Coastguard*** | *1-100* | **55.335** | **56.064** | **56.454** |
| | *101-200* | **55.324** | **56.082** | **56.406** |
| | *201-300* | **55.323** | **56.055** | **56.395** |
| ***Mobile*** | *1-100* | **55.321** | **56.546** | **56.667** |
| | *101-200* | **55.306** | **56.517** | **56.608** |
| | *201-300* | **55.285** | **56.456** | **56.57** |
| ***Foreman*** | *1-100* | **55.287** | **56.484** | **56.621** |
| | *101-200* | **55.297** | **56.479** | **56.616** |
| | *201-300* | **55.285** | **56.463** | **56.605** |
| ***Container*** | *1-100* | **55.362** | **56.527** | **56.665** |
| | *101-200* | **55.329** | **56.476** | **56.674** |
| | *201-300* | **55.334** | **56.464** | **56.672** |
| ***News*** | *1-100* | **55.527** | **56.567** | **56.381** |
| | *101-200* | **55.512** | **56.539** | **56.369** |
| | *201-300* | **55.498** | **56.535** | **56.364** |
| ***Akiyo*** | *1-100* | **55.385** | **56.482** | **56.561** |
| | *101-200* | **55.31** | **56.511** | **56.564** |
| | *201-300* | **55.348** | **56.481** | **56.504** |

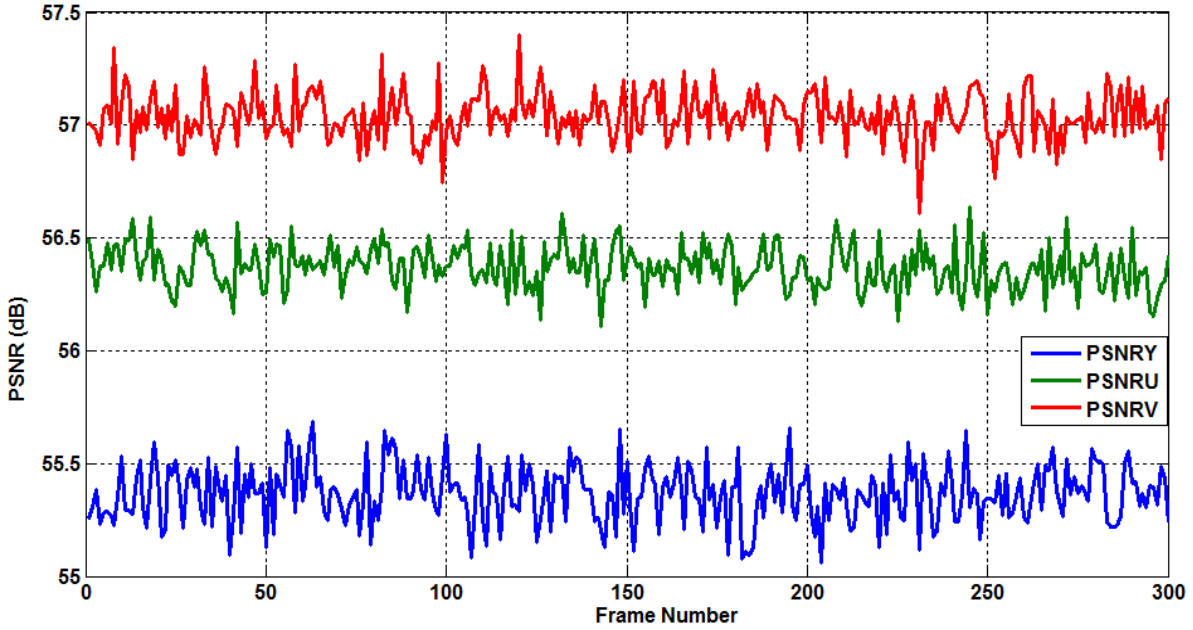


Fig. 4: *PSNR* of 300 stego frames for the *Stefan* video.

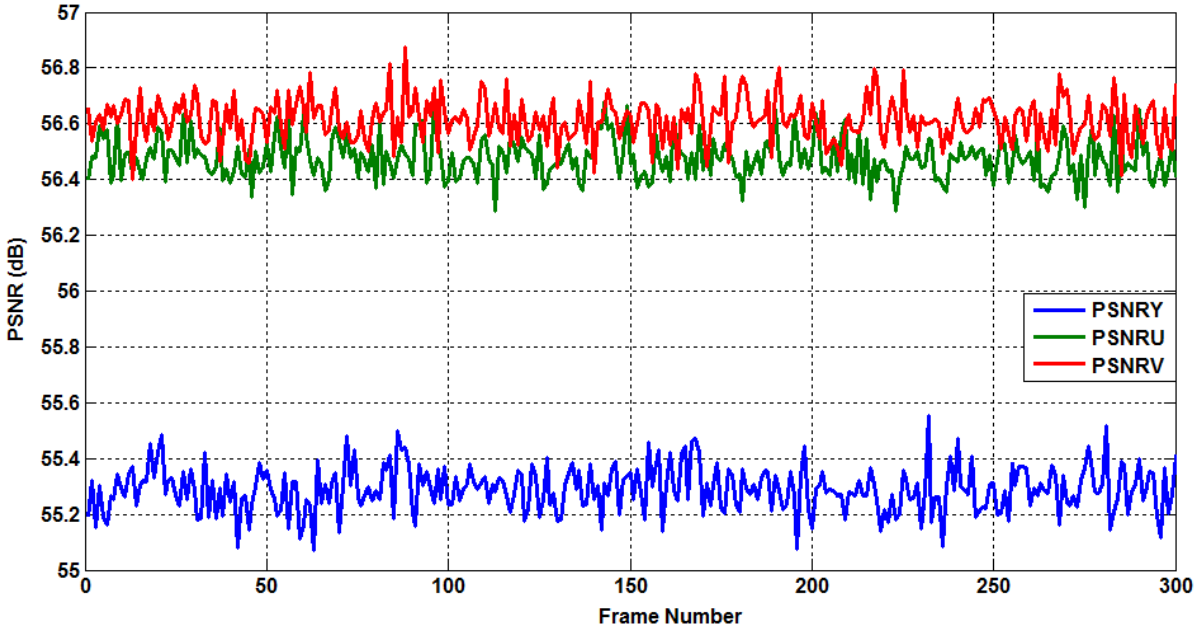


Fig. 5: *PSNR* of 300 stego frames for the *Foreman* video.

Fig. 6 shows the comparison of the visual quality between eight stego videos. The *PSNR* of each component, Y, U, and V is separately calculated, in which the average equals 300 frames per video. The values of *PSNR*s are between 55 and 57 dBs, which are considered excellent visual quality results.

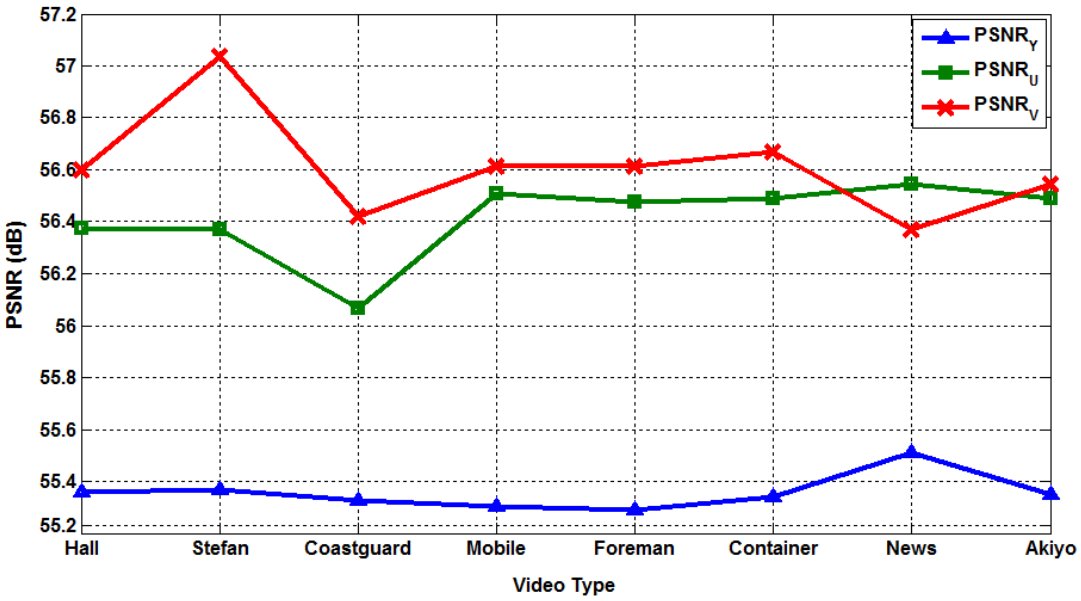


Fig. 6: Comparison between the averages of the *PSNRY*, *PSNRU*, and *PSNRV* components for eight video sequences

## Conclusion

In this paper, an efficient video steganography based on the BCH codes concepts has been proposed. The proposed steganography algorithm utilized frames as still images. It divides the video stream into frames, and then converts the frames to the YUV format. This algorithm is considered a high embedding efficiency algorithm due to the low modification on the cover data that translates into perfect visual quality in the stego videos. The visual quality is measured by the *PSNR* metric, and all the obtained experimental results have a *PSNR* above 55 dB. By achieving a good visual quality for stego videos, hackers will have difficulty retrieving secret messages. The security of our proposed algorithm has been satisfied by having more than one key to embed and extract the secret message. In addition to the secret keys that we have used, we also encoded and decoded the message with the BCH codes (7, 4, 1).

Experimental results prove both a high embedding efficiency and a high embedding payload of the proposed algorithm exist. The visual qualities of the stego videos are the same as the original video visual qualities. The *PSNR* of stego videos is above 55 dB. In each video frame, the embedding capacity is 246 Kbits, and can increase up to 405 Kbits without any noticeable degradation in the visual quality.

## References

1. Yuh-Ming, H. and J. Pei-Wun. *Two improved data hiding schemes*. in *Image and Signal Processing (CISP), 2011 4th International Congress on*. 2011.
2. Mstafa, R.J. and C. Bach. *Information Hiding in Images Using Steganography Techniques*. in *American Society for Engineering Education (ASEE Zone 1), 2013 Zone 1 Conference*. 2013.
3. Mstafa, R.J. and K.M. Elleithy. *A highly secure video steganography using Hamming code (7, 4)*. in *Systems, Applications and Technology Conference (LISAT), 2014 IEEE Long Island*. 2014.

4. Chin-Chen, C., T.D. Kieu, and C. Yung-Chen. *A High Payload Steganographic Scheme Based on (7, 4) Hamming Code for Digital Images*. in *Electronic Commerce and Security, 2008 International Symposium on*. 2008.
5. Guangjie, L., et al. *An Adaptive Matrix Embedding for Image Steganography*. in *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*. 2011.
6. Jyun-Jie, W., et al. *An embedding strategy for large payload using convolutional embedding codes*. in *ITS Telecommunications (ITST), 2012 12th International Conference on*. 2012.
7. Mstafa, R.J. and K.M. Elleithy. *A Novel Video Steganography Algorithm in the Wavelet Domain Based on the KLT Tracking Algorithm and BCH Codes*. in *Systems, Applications and Technology Conference (LISAT), 2015 IEEE Long Island*. 2015.
8. Feng, P., et al. *Video steganography using motion vector and linear block codes*. in *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*. 2010.
9. Hao, B., L.-Y. Zhao, and W.-D. Zhong. *A novel steganography algorithm based on motion vector and matrix encoding*. in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. 2011.
10. Rongyue, Z., et al., *An Efficient Embedder for BCH Coding for Steganography.* Information Theory, IEEE Transactions on, 2012. **58**(12): p. 7272-7279.
11. Liu, Y., et al., *A Robust Data Hiding Algorithm for H. 264/AVC Video Streams.* Journal of Systems and Software, 2013.
12. Hoyoung, Y., et al., *Area-Efficient Multimode Encoding Architecture for Long BCH Codes.* Circuits and Systems II: Express Briefs, IEEE Transactions on, 2013. **60**(12): p. 872-876.
13. Mstafa, R.J. and K.M. Elleithy. *A High Payload Video Steganography Algorithm in DWT Domain Based on BCH Codes (15, 11)*. in *Wireless Telecommunications Symposium (WTS), 2015*. 2015.

## Ramadhan J. Mstafa

Ramadhan J. Mstafa is originally from Duhok, Kurdistan Region, Iraq. He is pursuing his PhD degree in Computer Science and Engineering at the University of Bridgeport, Bridgeport, Connecticut, USA. He received his Bachelor's degree in Computer Science from the University of Salahaddin, Erbil, Iraq. Mr. Mstafa received his Master's degree in Computer Science from University of Duhok, Duhok, Iraq. He is IEEE Student Member. His research areas of interest include image processing, mobile communication, security, and steganography.

## Prof. Khaled M. Elleithy

Dr. Elleithy is the Associate Vice President of Graduate Studies and Research at the University of Bridgeport. He is a professor of Computer Science and Engineering. He has research interests are in the areas of wireless sensor networks, mobile communications, network security, quantum computing, and formal approaches for design and verification. He has published more than three hundred research papers in international journals and conferences in his areas of expertise. Dr. Elleithy has more than 25 years of teaching experience. His teaching evaluations are distinguished in all the universities he joined. He supervised hundreds of senior projects, MS theses and Ph.D. dissertations. He supervised several Ph.D. students. He developed and introduced many new undergraduate/graduate courses. He also developed new teaching / research laboratories in his area of expertise. Dr. Elleithy is the editor or co-editor for 12 books by Springer. He is a member of technical program committees of many international conferences as recognition of his research qualifications. He served as a guest editor for several International Journals. He was the chairman for the International Conference on Industrial Electronics, Technology & Automation, IETA 2001, 19-21 December 2001, Cairo – Egypt. Also, he is the General Chair of the 2005-2013 International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering virtual conferences.