An ECC/DCT-Based Robust Video Steganography Algorithm for Secure Data Communication

Ramadhan J. Mstafa* and Khaled M. Elleithy

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

Received 8 October 2016; Accepted 17 April 2017;
Publication 9 May 2017

Abstract

Nowadays, the science of information hiding has gained tremendous significance due to advances in information and communication technology. The performance of any steganographic algorithm relies on the embedding efficiency, embedding payload, and robustness against attackers. Low hidden ratio, less security, and low quality of stego videos are the major issues of many existing steganographic methods. In this paper, we propose a novel video steganography method in discrete cosine transform (DCT) domain based on error correcting codes (ECC). To improve the security of the proposed algorithm, a secret message is first encrypted and encoded by using Hamming and BCH codes. Then, it is embedded into the DCT coefficients of video frames. The hidden message is embedded into DCT coefficients of each Y, U, and V planes excluding DC coefficients. The proposed algorithm is tested under two types of videos that contain slow and fast moving objects. The experiential results of the proposed algorithm are compared with three existing methods. The comparison results show that our proposed algorithm outperformed other algorithms. The hidden ratio of the proposed algorithm is approximately 27.53%, which is considered as a high hiding capacity with a minimal tradeoff of the visual quality. The robustness of the proposed algorithm was tested under different attacks.

Journal of Cyber Security, Vol. 5_3, 167–194.
doi: 10.13052/jcsm2245-1439.531

Keywords: Video steganography, ECC, DCT, embedding efficiency, embedding payload, robustness.

1 Introduction

Steganography is a process that involves hiding important information (message) inside other carrier (cover) data to protect the message from unauthorized users. The mixed data (stego objects) will be seen by the human visual system (HVS) as one piece of data because the HVS will not be able to recognize the small change that occurs in the cover data. Message and cover data could be any type of data format such as text, audio, image, and video [1]. The development of steganalysis tools weakens unsecure steganography schemes and rendering them useless. Hence, researchers have to develop secure steganography algorithms that are protected from both attackers and steganalysis detectors. Any successful steganography system should consider three main important factors: embedding capacity, imperceptibility, and robustness against attacks [2].

First, the embedding payload is defined as the amount of secret information that is going to be embedded inside the cover data. The algorithm has a high embedding payload if it has a large capacity for the secret message. The embedding efficiency includes the stego visual quality, security, and robustness against attackers [3].

Second, both a low modification rate and good quality of the cover data lead to a high embedding efficiency [4]. The steganography algorithm that contains a high embedding efficiency will reduce attacker suspicion of finding hidden data and will be quite difficult to detect through steganalysis tools. However, any distortion to the cover data after the embedding process occurs will increase the attention of attackers. The embedding efficiency is directly affected by the security of the steganographic scheme [5]. In traditional steganographic schemes, embedding payload and embedding efficiency are opposite. Increasing the capacity of the secret message will decrease the quality of stego videos that then weakens the embedding efficiency. Both factors should be considered. The deciding factors depend on the steganography algorithm and the user requirements. To improve steganographic schemes, many of the algorithms use ECC principles such as Hamming, BCH, and Reed-Solomon codes [6].

Third, robustness is another factor which measures the steganography algorithm's resistance against signal processing and attacks. Signal processing operations include compression, geometric transformation, filtering, and cropping. The algorithm is robust when the receiver side extracts the secret message correctly, without any errors. High efficient steganography algorithms are robust against both signal processing and adaptive noises [7].

The remainder of this paper is organized as follows: Section 2 presents steganography versus cryptography and watermarking. Section 3 discusses video steganography techniques in both raw and compressed domains. Section 4 explains discrete cosine transform. Section 5 explains some two ECC principles such as Hamming and BCH codes. Section 6 presents the embedding and extracting phases of the proposed steganography methodology. Section 7 illustrates and explains the experimental results. Section 8 contains the conclusions.

2 Steganography versus Cryptography and Watermarking

The common objective of both steganography and cryptography is to provide confidentiality and protection of data. The steganography "protected writing" establishes a covert communication channel between legitimate parties; while the cryptography "secret writing" creates an overt communication channel [8]. In cryptography, the presence of the secret data is recognizable; however, its content becomes unintelligible to illegitimate parties. In order to increase additional levels of security, steganography and cryptography can operate together in one system [9].

Digital watermarking techniques use a preservation mechanism to protect the copyright ownership information from unauthorized users. This process is accomplished by concealing the watermark information into overt carrier data [10]. Like steganography, watermarking can be used in many different applications such as content authentication, digital fingerprints, broadcast monitoring, copyright protection, and intellectual property protection [11]. Different watermarking techniques can be found in the literature. Figure 1 clarifies the hierarchy of the overall information hiding concept.

3 Video Steganography Techniques

Due to the advancement of Internet and multimedia technologies, digital videos have become a popular choice for data hiding. The video data contains a massive amount of data redundancy which can be utilized for embedding secret data. Recently, there are many useful applications of video steganography techniques such as video error correcting [12, 13], military services,

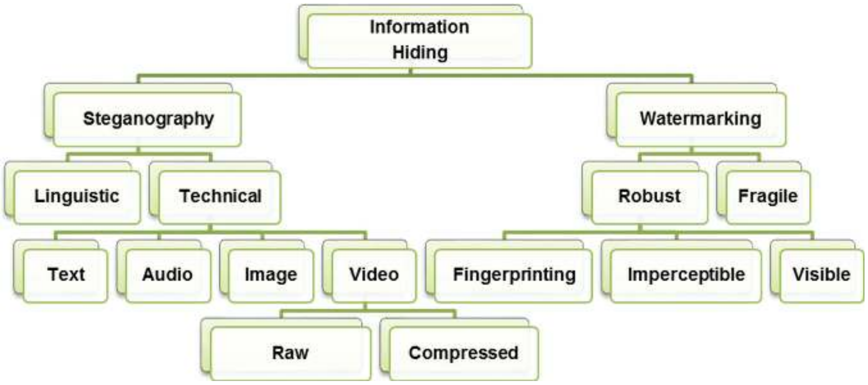


Figure 1 Information hiding concept.

bandwidth saving, video surveillance, and medical video security [14]. Video steganography techniques are classified into compressed and uncompressed domains [15, 16].

3.1 Video Steganography Techniques in Compressed Domain

The H.264 standard has increased the efficiency of video compression when compared to the previous versions. Some new features of H.264 video codec include flexible macroblock ordering, quarter-pixel interpolation, intra prediction in intra frame, deblocking filtering post-processing, and multiple frames reference capability [17]. Usually, H.264 codec comprises a number of group of pictures (GOP). Every GOP includes three types of frames: intra (I) frame, predicted (P) frame, and bidirectional (B) frame. During the video compression process, the motion estimation and compensation processes minimize the temporal redundancy. Since the video stream is a number of correlated still images, a frame can be predicted by using one or more referenced frames based on the motion estimation and compensation techniques. First, frames are divided into 16 × 16 macroblocks (MB) wherein each MB contains blocks that may include the smallest size of 4 × 4. When applying a few searching algorithms, block C in the present frame is compared, individually, to one of the selected block $\tilde{R}$ in the referenced frame $\tilde{F}$ in order to find a corresponding block C. The prediction error between two blocks (C and $\tilde{R}$) of size b can be measured using Sum of Absolute Differences (SAD).

$$e = SAD\left(C, \tilde{R}\right) = \sum_{1 \leq i,j \leq b} |c_{i,j} - \tilde{r}_{i,j}| \tag{1}$$

where $c_{i,j}$ and $\tilde{r}_{i,j}$ refer to block values. The best matched block will have a minimum SAD using C's prediction denoted by $\tilde{P}$. The motion vector and differential error $D = C - \tilde{P}$ are required for the video coding process. Video steganography techniques in compressed domain are categorized according to the video coding stages as venues for data hiding such as intra frame prediction, inter frame prediction, motion vectors, transformed and quantized coefficients, and entropy coding.

3.1.1 Intra frame prediction

During the video compression process, the macroblocks are encoded using a number of intra prediction modes. In H.264 codec, the numbers of intra prediction modes are nine of 4 × 4 blocks and four of 16 × 16 blocks. Figure 2 illustrates intra prediction modes for 16 × 16 blocks. Also, the high efficiency video coding (*HEVC*) codec can support up to 35 intra prediction modes for each 64 × 64, 32 × 32, 16 × 16, 8 × 8, and 4 × 4 block sizes. For data concealing purposes, these modes can be mapped to one or more of secret message bits.

3.1.2 Inter frame prediction

In many video steganographic methods, the seven block sizes that include 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8 and 4 × 4 of H.264 inter frame prediction are commonly utilized as a venue to embed the secret message by mapping each block type to a number of secret bits. Kapotas et al. [18] proposed a data concealing algorithm for scene change detection in H.264 coding. This method uses four different block sizes. Each one is mapped onto one pair of a secret message. In this algorithm, the secret message consists of scene change frames information that will be embedded into the encoded videos. This embedded information will help the scene change detection algorithm, in H.264 video stream, functioning in real time. However, the data hiding methods of the intra frame prediction have a very limited embedding capacity. For example, let "NY" is the secret information that

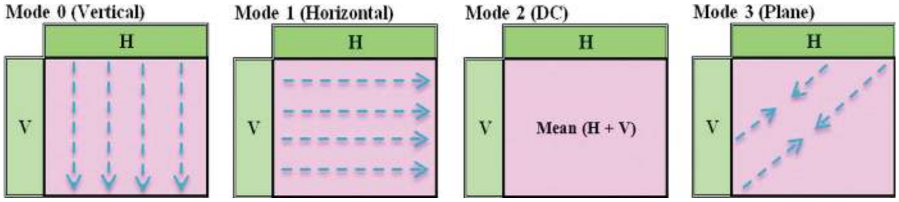
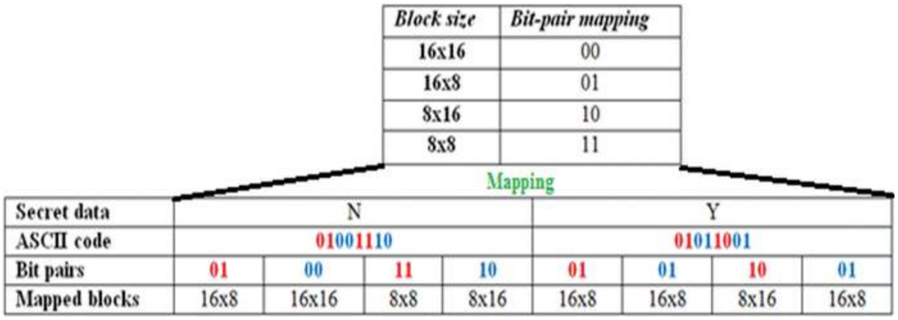


Figure 2 H.264 intra prediction modes for 16 × 16 blocks.



Block size	Bit-pair mapping
16x16	00
16x8	01
8x16	10
8x8	11

Mapping

Secret data	N				Y			
ASCII code	01001110				01011001			
Bit pairs	01	00	11	10	01	01	10	01
Mapped blocks	16x8	16x16	8x8	8x16	16x8	16x8	8x16	16x8

Figure 3 Mapping rules of prediction block type to embed “NY” characters.

must be embedded into the inter frame prediction blocks in H.264 codec. By using mapping rules of different block sizes the embedding goal can be achieved. Figure 3 illustrates the embedding process using mapping rules.

3.1.3 Motion vectors

Motion vector characteristics such as horizontal and vertical components, amplitude, and phase angles are utilized in embedding secret information. Xu et al. [19] proposed a compressed video stream steganography. In this scheme, the embedding process relies on I, P, and B frames. First, the hidden data is concealed into the motion vectors of, both, P and B frames. Only the motion vectors that have high magnitudes are chosen. Here, each macroblock has a motion vector; however, the selected macroblocks are moving rapidly. Secondly, the control information is embedded into I frames. This control information includes the capacity payload and segment range of each GOP. Each GOP contains one I frame which carries the control information necessary for the data extraction phase. In addition, each GOP has a number of P and B frames which contain secret messages in their high magnitude motion vectors. Xu et al.’s method has a low embedding payload because it only used the motion vectors with a high magnitude.

3.1.4 Transform coefficients

DCT, quantized DCT (QDCT), and discrete wavelet transform (DWT) coefficients of the luminance component are also good candidates to conceal the secret message due to their low, middle, and high frequency coefficients for data embedding. Huang et al. [20] presented reliable information bit hiding using the DCT and communication theory. In order to enhance the robustness of this method, the BCH codes and soft-decision decoding have been used. Moreover, the robustness is also achieved by testing both the common signal processing operations and a StirMark attack. The secret data is hidden into the DCT coefficients, especially, in DC with the highest energy coefficient and low-frequency AC coefficients. Barni et al. [21] presented a watermarking technique of MPEG-4 video coding based on the video object planes. This scheme hides the watermark information into the selected inter and intra macroblocks of each video object. Depending on the computed frequency mask, DCT coefficients that exceeds to the predefined threshold were chosen for the embedding process. Barni's is flexible and easy to use for many applications. Moreover, it is robust against some common signal processing.

3.1.5 Entropy coding CAVLC and CABAC

During the H.264 compression, context adaptive variable length coding (CAVLC) and context adaptive binary arithmetic coding (CABAC) entropy coding can be used as host data to carry secret messages within many video steganographic techniques. Ke et al. [22] presented a video steganographic method relies on replacing the bits in H.264 stream. In this algorithm, CAVLC entropy coding has been applied in the data concealing process. The embedding phase can be completed based on the trailing ones sign flag and the level of the codeword parity flag. The sign flag of the trailing ones changes if the embedding bit equals "0" and the parity of the codeword is even. Also, the sign flag changes if the embedding bit equals "1" and the parity of the codeword is odd. Otherwise, the sign flag of the trailing ones does not change. The trailing ones (*TOnes*) are modified as follows:

$$TOnes = \begin{cases} even\ \ codeword;\ if\ secret\, bit = 0 \\ odd\ \ codeword;\ \ if\ secret\, bit = 1 \end{cases} \tag{2}$$

The modification of high frequency coefficients does not have an impact on the video quality. However, the embedding capacity is low because Ke et al.'s method is established on the non-zero coefficients of the high frequencies that consist of a large majority of zeros.

3.2 Video Steganography Techniques in Raw Domain

Unlike the compressed video, raw video steganographic techniques deal with the video as a sequence of frames with the same format. First, digital video is converted into frames as still images, and then each frame is individually used as carrier data to conceal the hidden information. After the embedding process, all frames are merged together to produce the stego video. Raw video steganographic techniques consist of spatial and transform domain techniques [23].

3.2.1 Spatial domain methods

There are many steganographic techniques that rely on the spatial domain such as LSB substitution, bit plane complexity segmentation (BPCS), spread spectrum, region of interest (ROI), histogram manipulation, matrix encoding, and mapping rule. Basically, these techniques utilize the pixel intensities to conceal the secret message. Zhang et al. [24] presented an efficient embedder utilizing BCH encoding for data hiding. This embedder hides the covert information into a block of carrier object. The concealing phase is achieved by modifying different coefficients in the input block to set the syndrome values null. This method enhances embedding payload and execution duration compared to others. The error correcting code and steganographic model of this method is shown in the Figure 4. Zhang et al.'s method modifies the complexity of the algorithm from exponential to linear. On the other hand, Diop et al. [25] presented an adaptive steganography method utilizing the low-density parity-check codes. The method discusses how to reduce the influence of hidden information insertion by this codes. This algorithm demonstrated that the low-density parity-check codes are better for encoding algorithms than other codes. The process of embedding and extraction can be accomplished by Equation (3) and Equation (4).

$$S = Embedding\ (I, m) \tag{3}$$

$$m = Extraction(m) = HS \tag{4}$$

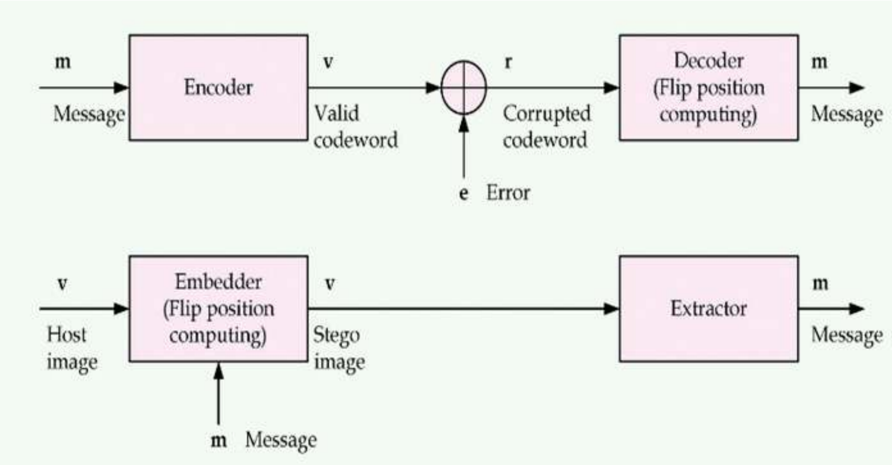


Figure 4 Error correcting code and steganographic model.

where I and S are the cover data and steganogram, respectively, and m is a secret message ($m \in F_2^m$).

3.2.2 Transform domain methods

In the transform domain steganographic methods, each video frame is individually transformed into frequency domain using DCT, DWT, and discrete Fourier transform (DFT) and the secret message is embedded utilizing the low, middle, or high frequencies of the transformed coefficients. Patel et al. [26] presented a new data hiding method using the lazy wavelet transform (LWT) technique, where each video frame is divided into four sub-bands, separating the odd and even coefficients. The secret information is then embedded into the RGB LWT coefficients. For accurate extraction of embedded data, the length of hidden data is concealed into the audio coefficients. The amount of hidden information is high, but this type of wavelet is not a real mathematical wavelet operation. Consequently, Patel et al.'s method will not protect the hidden information from attackers.

Table 1 provides a summary of the related video steganography techniques that operate in both compressed and raw domains, highlighting each of venues for data hiding, robustness against attacks, video preprocessing, secret message preprocessing, performance measures of embedding capacity and video quality.

Table 1 Venues, embedding capacity, video quality, robustness, video and message preprocessing of the existing video steganographic methods

Method	Domain/Venue for Data Hiding	Embedding Capacity	Video Quality	Robust-ness	Preprocessing	
					Video	Message
Pan *et al.* [29]	Compressed domain/Motion vectors	Low embedding capacity (at most 4 bits in 6 bits of high amplitude motion vectors and the modification of 2 bits)	Average PSNR is 37.45 dB	✓	×	×
Jue *et al.* [30]	Compressed domain/Motion vectors	Low embedding capacity (at most 55 bits per P-frame or B-frames macroblocks. Largest amplitude of motion vectors is used)	Average PSNR is 36.27 dB	✓	×	×
Barni *et al.* [21]	Compressed domain/DCT coefficients	Low embedding capacity (at most 30 bits per video object of 500 Kb/s)	Almost the same as compressed video	✓	×	×

(Continued)

Table 1 Continued

Li *et al.* [31]	Compressed domain/DWT coefficients	An average of 38 Kbits per frame of resolution 352×288 when the first level of DWT is used	Average PSNR is 35.50 dB when the first level of DWT is used	✓	✓	×
Li *et al.* [32]	Compressed domain/QDCT coefficients	Low embedding capacity (at most 1 bit per 4×4 luma block)	Average PSNR is 36 dB of Intra frame	✓	×	×
Mobasseri *et al.* [33]	Compressed domain/CAVLC	Low embedding capacity (an average of 1 bit per 8×8 Intra block)	Almost the same as compressed video	✓	×	×
Wang *et al.* [34]	Compressed domain/CABAC	Low embedding capacity (1156 bits are embedded in 50 frames of resolution $176{\times}144$)	Average PSNR is around 37 dB)	✓	×	×
Zhang *et al.* [24]	Raw/Spatial domain	Embedding capacity is $\mathrm{m} \times \mathrm{t}$ bits per $\mathrm{n} = 2^{\mathrm{m}} - 1$ bits block, where $\mathrm{m} > 2$ and $\mathrm{t} = 2$ or 3	N/A	×	×	✓
Cheddad *et al.* [35]	Raw/Spatial domain	Average of embedding capacity ratio is 1.03%	Average PSNR is 59.63 dB	×	✓	×
Alavianmehr *et al.* [36]	Raw/Spatial domain	Average of embedding capacity ratio is 1.34% (4096 bits per video)	Average PSNR is 36.97 dB	✓	×	×
Hu *et al.* [37]	Raw/Spatial domain	Average of embedding capacity 1.5 bpp	Average PSNR is 29.03 dB	×	×	✓
Sun [38]	Raw/Spatial domain	At most the embedding capacity ratio is 45%	Average PSNR is 44.28 dB	×	✓	×
Patel *et al.* [26]	Raw/Transform domain	Average of embedding capacity ratio is 12.5%	Average PSNR is 31.23 dB	×	×	✓
Spaulding *et al.* [39]	Raw/Transform domain	Average of embedding capacity ratio is 25%	Average PSNR is 33 dB	✓	✓	×

4 Discrete Cosine Transform

DCT is a well-known method which is utilized in many applications such as image and video compression. The DCT separates the signal into low, middle, and high frequency regions. The DCT is closely related to the DFT. It is a separable linear transformation; that is, the 2D-DCT is equivalent to a 1D-DCT performed along a single dimension followed by a 1D-DCT in the other dimension [27]. For an input video frame, A, of resolution M × N the DCT frequency coefficients for the transformed frame, B, and the inverse DCT coefficients of the reconstructed frame are calculated according to the following equations, respectively:

$$B_pq = \propto_p \propto_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi\,(2m+1)\,p}{2M} \cos\frac{\pi\,(2n+1)\,q}{2N} \tag{5}$$

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \propto_p \propto_q B_{pq} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \tag{6}$$

Where

$$\propto_p = \begin{cases} \frac{1}{\sqrt{M}}, \; p = 0 \\ \sqrt{\frac{2}{M}}, \; 1 \leq p \leq M-1 \end{cases}$$

And

$$\propto_q = \begin{cases} \frac{1}{\sqrt{N}}, \; q = 0 \\ \sqrt{\frac{2}{N}}, \; 1 \leq q \leq N-1 \end{cases}$$

A (m, n) is the pixel value in row m and column y of the frame A, and B (p, q) is the coefficient in row p and column q of the 2D-DCT matrix. Each of low, middle, and high frequency coefficients were used as cover data to embed the encoded secret message [28].

5 Hamming and BCH ECC

In this paper, Hamming (7, 4) codes are used ($n = 7$, $k = 4$, and $p = 3$), which can correct the identification of a single bit error. A message of size $M\ (m_1,\ m_2, \ldots,\ m_k)$ is encoded by adding $p\ (p_1,\ p_2,\ p_3)$ extra bits as parity to become a codeword of 7-bit length. The codeword is prepared to transmit through a communication channel to the receiver end. The common combination of both message and parity data using these type of codes is to place the parity bits at the position of 2^{i} (i = 0, 1, . . ., n–k) such as $\mathrm{p_1, p_2, m_1, p_3, m_2, m_3, m_4}$ combination. Venn diagram of the hamming codes (7, 4) is illustrated in the Figure 5.

In addition of hamming codes, BCH (7, 4, 1) codes is also used over the GF $\left(2^3\right)$, where $m = 3$, $k = 4$, and $n = 2^3{-}1 = 7$. 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 code is different from the Hamming code because BCH can correct more than one bit. 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 $\propto,\ \propto^2,\ \propto^3, \ldots,\ \propto^{2t}$ as roots

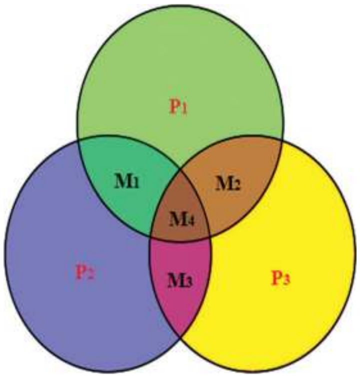


Figure 5 Venn diagram of the hamming codes (7, 4).

on the condition that $\propto$ is a primitive of GF (2^{m}). When $\mathrm{M_i}\,(\mathrm{x})$ is a minimal polynomial of $\propto^{\mathrm{i}}$where $(1 \leq i \leq 2t)$, then the least common multiple (LCM) of *2t* minimal polynomials will be the generator polynomial g(x). The g(x) function and the parity-check matrix *H* of the BCH codes [7, 40] are described as follows:

$$H = \begin{bmatrix} 1 & \propto & \propto^2 & \propto^3 & \ldots & \propto^{n-1} \\ 1 & \left(\propto^3\right) & \left(\propto^3\right)^2 & \left(\propto^3\right)^3 & \ldots & \left(\propto^3\right)^{n-1} \\ 1 & \left(\propto^5\right) & \left(\propto^5\right)^2 & \left(\propto^5\right)^3 & \ldots & \left(\propto^5\right)^{n-1} \\ . & . & . & . & & . \\ . & . & . & . & & . \\ . & . & . & . & & . \\ 1 & \left(\propto^{2t-1}\right) & \left(\propto^{2t-1}\right)^2 & \left(\propto^{2t-1}\right)^3 & \ldots & \left(\propto^{2t-1}\right)^{n-1} \end{bmatrix} \tag{7}$$

$$g\,(x) = lcm\{M_1\,(x)\,,\; M_2\,(x)\,,\; M_3\,(x)\,,\; \ldots,\; M_{2t}\,(x)\} \tag{8}$$

$$g(x) = M_1\,(x)\;\, M_3\,(x)\;\, M_5\,(x)\;\, \ldots\; M_{2t-1}\,(x) \tag{9}$$

A binary BCH (*n*, *k*, *t*) can correct errors of a maximum *t* bits for a codeword $W = \{w_0, w_1, w_2, .\;.\;., w_{n-1}\}$ of length *n* and a message $A\; = \;\{a_0, a_1, a_2, .\;.\;., a_{k-1}\}$ of length *k* [41]. An embedded codeword $C = \{c_0, c_1, c_2, .\,.\,., c_{n-1}\}$ is calculated as follows:

$$C = W * H^T \tag{10}$$

At the receiver side, the codeword $R = \{r_0, r_1, r_2, .\,.\,., r_{n-1}\}$ is obtained. The transmitted and received codewords can both be interpreted as polynomials, where $C\left(X\right)=c_0+c_1x^1+\cdots+c_{n-1}x^{n-1}$, and $R\left(X\right)=r_0+r_1x^1+\cdots+r_{n-1}x^{n-1}$. The error *E* is the difference between *C* and *R*, which indicates the number and location of flipped elements in *C*. The *E* and syndrome *Y* are calculated as follows:

$$E=R-C \tag{11}$$

$$Y=\left(R-C\right)H^T=EH^T \tag{12}$$

6 The Proposed Steganography Methodology

In this section, we proposed a novel video steganography algorithm in DCT domain based on Hamming and BCH (7, 4, 1) codes. At the beginning, the video sequence is separated into frames; each frame is converted to YCbCr color space. The reason for converting to YCbCr color space is to remove the correlation between the red, green, and blue colors. The proposal methodology consists of data embedding stage and data extracting stage.

6.1 Data Embedding Stage

For a security purpose, the hidden message is encrypted using a secret key, and then Hamming and BCH (7, 4, 1) codes will be applied on it producing an encoded message. The whole encoded message is converted from binary to base-8 digits. On the other hand, each video sequence is converted into a number of frames. Each frame separates into the YUV color space. Then, 2D-DCT is applied individually on each plane. Subsequently, the process of embedding is achieved by concealing each base-8 digit of the encoded message into the DCT frequency coefficients except the DC coefficients of each of the Y, U, and V planes. Thereafter, the inverse of 2D-DCT is applied on the three stego components of each frame producing a stego frame. Finally, the stego video is constructed from these stego frames. The secret message is concealed into each of $\mathrm{Y_{ij}},\ \mathrm{U_{ij}},\ \text{and}\ \mathrm{V_{ij}}$ DCT coefficients as follows:

$$\widehat{Y}_{ij}=\begin{cases}Embedding\left(Y_{ij},D_k\right) & ;Y_{ij}\geq 0\\ Embedding\left(abs(Y_{ij}),D_k\right) & ;Y_{ij}<0\end{cases} \tag{13}$$

$$\widehat{U}_{ij}=\begin{cases}Embedding\left(U_{ij},D_k\right) & ;U_{ij}\geq 0\\ Embedding\left(abs(U_{ij}),D_k\right) & ;U_{ij}<0\end{cases} \tag{14}$$

$$\widehat{V}_{ij}=\begin{cases}Embedding\left(V_{ij},D_k\right) & ;V_{ij}\geq 0\\ Embedding\left(abs(V_{ij}),D_k\right) & ;V_{ij}<0\end{cases} \tag{15}$$

where $\widehat{\mathrm{Y}}_{\mathrm{ij}}$, $\widehat{\mathrm{U}}_{\mathrm{ij}}$, and $\widehat{\mathrm{V}}_{\mathrm{ij}}$ are DCT coefficients of stego Y, U, and V planes respectively, and D_k is the encoded digits, $D_k = \{000, \ldots, 111\}$. The data embedding stage of the proposed steganography method is illustrated in Algorithm 1.

Algorithm 1 Data Embedding Stage

```
Input: V //Video ,
        M //Secret message in characters,
        $Key_1$, $Key_2$; // Stego keys
Output: SV; //Stego video
Initialize km, pm;
B←M; //Convert the alphabetic secret message to the binary array
// Stego keys
$Key_1$←Length(B)/4; //Length of the secret message
$Key_2$←rand (2^7,$Key_1$,1)';//Randomization of the seed $Key_1$
EB←Encrypt(B,[$Key_1$]); //Encrypt the binary array by $Key_1$

//Encode each 4 bits of encrypted message by Hamming and BCH (7, 4, 1) codes
$for_1$ i = 1: ($Key_1$*7) do
        g(1:4) ← get(EB(km:km+4));
        E_EB ← encode(g,7,4);
        temp(1:7) ← get($Key_2$(i));
        Ecdmsg(pm:pm+7) ← xor(E_EB, temp);
        pm+7; km+4;
$end_1$
D ← Ecdmsg; //Encoded message is segmented into 3-bit groups
{$Vf_1$, $Vf_2$,..., $Vf_n$} ← V; //Video V is divided into n frames
{Y, U, V} ← Vf; //Each frame Vf is converted into Y, U, and V components
DCT (Y, U, V); //Applying 2D-DCT on each frame components

//Embed the encoded message into YUV coefficients
$for_2$ i = 1:Vfx do
    $for_3$ j = 1:Vfy do
        $\widehat{Y}_{ij} = \begin{cases} Embedding\ (Y_{ij}, D_k) & ; Y_{ij} \geq 0 \\ Embedding\ (abs(Y_{ij}), D_k) & ; Y_{ij} < 0 \end{cases}$
        $\widehat{U}_{ij} = \begin{cases} Embedding\ (U_{ij}, D_k) & ; U_{ij} \geq 0 \\ Embedding\ (abs(U_{ij}), D_k) & ; U_{ij} < 0 \end{cases}$
        $\widehat{V}_{ij} = \begin{cases} Embedding\ (V_{ij}, D_k) & ; V_{ij} \geq 0 \\ Embedding\ (abs(V_{ij}), D_k) & ; V_{ij} < 0 \end{cases}$
    $end_3$
$end_2$
IDCT ($\widehat{Y}$,$\widehat{U}$, $\widehat{V}$); //Applying 2D-IDCT on each frame component
get SVf //Obtain the stego frames
get SV //Obtain the stego video
```

6.2 Data Extracting Stage

Data extracting is the process of retrieving the encoded message from the stego videos. This process is achieved by isolating the stego videos into frames. Each frame is divided into Y, U, and V planes. Then, 2D-DCT is applied separately on each plane. The process of extracting the encoded message is accomplished by taking D_k digits from each of Y, U, and V DCT coefficients, respectively, except DC coefficients. The outcomes data are decoded by Hamming and BCH (7, 4, 1) decoder followed by the deciphering process to extract the valid embedded message. The purpose of using ciphering and encoding methods prior the embedding process is to improve the security and robustness of the proposed algorithm. Moreover, the secret key is only shared between sender and receiver, and used in both the data embedding and extracting processes. The hidden message can be obtained as follows:

$$\widehat{D}_k = \begin{cases} Extracting\left(\widehat{Y}_{ij}\right) & ;(\widehat{Y}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{Y}_{ij}\right)\right) & ;(\widehat{Y}_{ij} < 0) \end{cases} \tag{16}$$

$$\widehat{D}_k = \begin{cases} Extracting\left(\widehat{U}_{ij}\right) & ;\ (\widehat{U}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{U}_{ij}\right)\right) & ;(\widehat{U}_{ij} < 0) \end{cases} \tag{17}$$

$$\widehat{D}_k = \begin{cases} Extracting\left(\widehat{V}_{ij}\right) & ;(\widehat{V}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{V}_{ij}\right)\right) & ;(\widehat{V}_{ij} < 0) \end{cases} \tag{18}$$

where $\widehat{\mathrm{Y}}_{\mathrm{ij}}, \widehat{\mathrm{U}}_{\mathrm{ij}}, \text{ and } \widehat{\mathrm{V}}_{\mathrm{ij}}$ are DCT coefficients of stego YUV planes, and $\widehat{D}_k$ is the retrieved secret message. The data extracting stage of the proposed steganography method is illustrated in Algorithm 2.

7 Experimental Results and Discussion

The experimental environment uses several variables: the cover data comprise a dataset consisting of six video sequences *Akiyo*, *Bus, Coastguard*, *Container, Foreman, and Soccer* of CIF type; also, the format of YUV is 4:2:0. In addition, the resolution of each video is (352×288), and all videos are equal in length with 150 frames. A large text file is used as a secret message. The work is implemented using MATLAB to test the proposed algorithm efficiency.

Algorithm 2: Data Extracting Stage

```
Input: SV; //Stego video
          Key_1, Key_2; //Stego keys
Output: M; //Secret message in characters
Initialize km, pm;
$\{Sf_1, Sf_2, \ldots, Sf_n\} \leftarrow SV$; //Stego Video SV is divided into n frames
$\{\widehat{Y}, \widehat{U}, \widehat{V}\} \leftarrow Sf$; //Each frame Sf is converted into Y, U, and V components
DCT($\widehat{Y}, \widehat{U}, \widehat{V}$); //Applying 2D-DCT on each stego frame component
//Extract the encoded message from stego YUV coefficients
for_1 i = 1:Sfx do
    for_2 j= 1:Sfy do
        $\widehat{D}_k = \begin{cases} Extracting\left(\widehat{Y}_{ij}\right) & ;(\widehat{Y}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{Y}_{ij}\right)\right) & ;(\widehat{Y}_{ij} < 0) \end{cases}$
        $\widehat{D}_k = \begin{cases} Extracting\left(\widehat{U}_{ij}\right) & ;(\widehat{U}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{U}_{ij}\right)\right) & ;(\widehat{U}_{ij} < 0) \end{cases}$
        $\widehat{D}_k = \begin{cases} Extracting\left(\widehat{V}_{ij}\right) & ;(\widehat{V}_{ij} \geq 0) \\ Extracting\left(abs\left(\widehat{V}_{ij}\right)\right) & ;(\widehat{V}_{ij} < 0) \end{cases}$
    end_2
end_1
Ecdmsg ← D; //Collect all 3-bit messages into a single array
//Decode each 7 bits of extracted data by Hamming and BCH (7, 4, 1) decoders
for_3 i = 1: (Key_1*7) do
    Sg(1:7) ← get(Ecdmsg (km:km+7));
    temp(1:7) ← get(Key_2(i));
    E_EB ← xor(Sg, temp)
    EB ← decode(E_EB,7,4);
    pm+4; km+7;
end_3
B ← Decrypt(EB, [Key_1]); //Decrypt the binary array by Key_1
M ← B; //Convert the binary array to the alphabetic characters
get M //Obtain the secret message
```

7.1 Visual Quality

Peak signal to noise ratio (PSNR) is an objective quality measurement used to calculate the difference between the original and the stego video frames. It can be obtained by following equations [23]:

$$PSNR = 10 * Log_{10}\left(\frac{{MAX_O}^2}{MSE}\right) \tag{19}$$

$$MSE = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[O(i,j) - S\left(i,j\right)\right]^2}{m * n} \tag{20}$$

where O and S denote the original and stego YUV frame components, respectively, and m and n are the video resolutions. MAX_O is the maximum possible pixel value of the host YUV frame components. When the pixels are represented using 8 bits per channel, the grayscale image will have 255 maximum value.

Figure 6 shows a sample frame from each *Akiyo*, *Container*, *Foreman*, and *Soccer* video and their corresponding stego frames. In Figure 7, the PSNR of

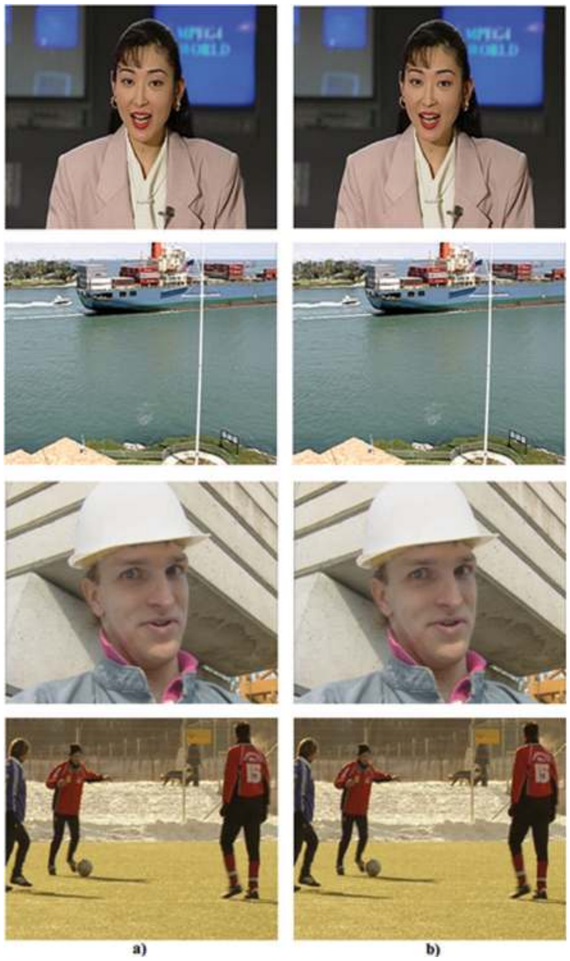


Figure 6 Sample video frames from dataset: a) *Akiyo*, *Container*, *Foreman*, and *Soccer* cover frames, and b) *Akiyo*, *Container*, *Foreman*, and *Soccer* corresponding stego frames.

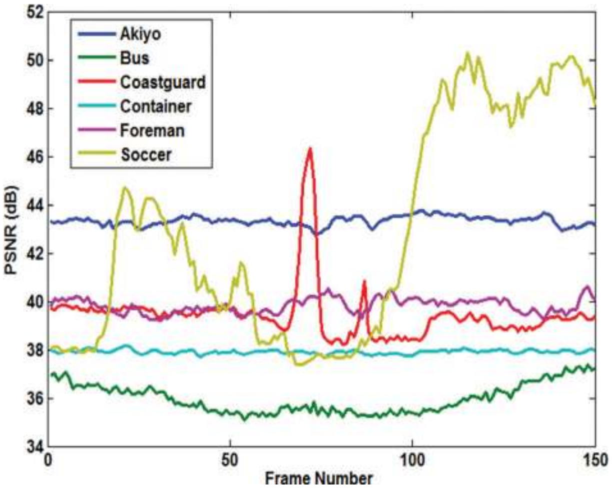


Figure 7 PSNR comparison for Y-components of six experments.

the Y-components are calculated for all six videos. Overall, the *Akiyo* video has the best luminance quality. Figure 8 shows the PSNR of the U-component for all six videos. The PSNR-U of the *Coastguard* video has the highest dBs among the group. Figure 9 shows the PSNR of the V-component for all experiments. The PSNR-V for the *Coastguard* video has a better quality

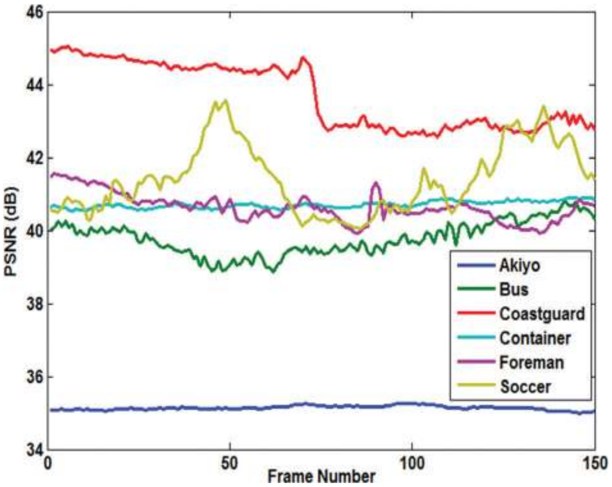


Figure 8 PSNR comparison for U-component of six videos.

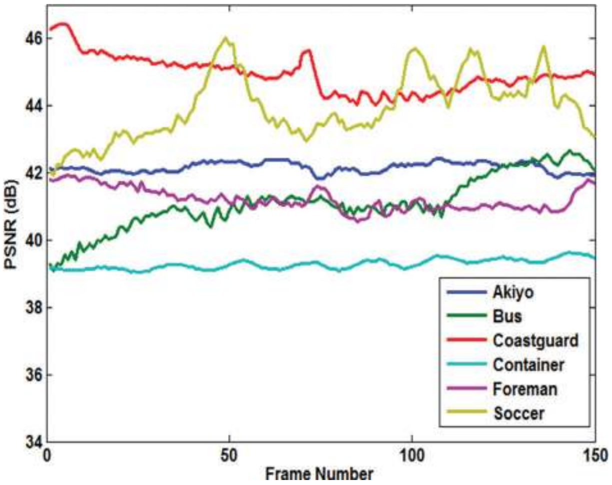


Figure 9 PSNR comparison for V-component of six experments.

among all videos. In Figure 10, the PSNR comparison for 150 frames of each video is shown. The comparison shows that the result of the objective quality for each of the *Akiyo*, *Bus*, *Coastguard*, *Container*, *Foreman*, *and Soccer* videos ranged between **38.95–42.73** dBs. Overall, the results of the PSNR for the *Akiyo*, *Bus*, *Container*, and *Foreman* videos are more stable, while

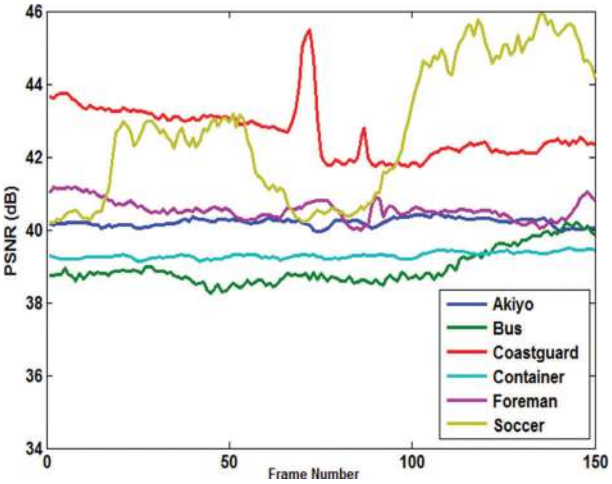


Figure 10 PSNR comparison for 150 frames of six experments.

Table 2 Average PSNR each of Y, U, and V component for six experiments

Sequences	PSNRY	PSNRU	PSNRV	PSNR
Akiyo	43.33	35.14	42.16	40.21
Bus	35.96	39.78	41.10	38.95
Coastguard	39.45	43.69	44.93	42.69
Container	37.91	40.71	39.28	39.30
Foreman	39.82	40.63	41.21	40.55
Soccer	42.85	41.40	43.94	42.73
Average	39.88	40.22	42.10	40.73

in the *Coastguard* and *Soccer* videos the quality is frequently changing. The changes occur because these videos contain faster motion objects that lead to unstable visual quality.

Table 2 shows the averages of the PSNR for each Y, U, and V component for all video sequences. Moreover, the visual quality of each video is measured separately by averaging each of the 150 frames per video. The quality averages are various and depend on type of videos.

7.2 Embedding Payload

The average of the obtained hidden ratio of the proposed algorithm is **27.53%**. A reasonable tradeoff is noticed between the amount of the embedded message in each video (5.99 Mbytes) and the average quality of six experiments (**40.73** dB). The hidden ratio (*HR*) can be calculated as in Equation (21).

A number of experiments were conducted to compare the performance of the proposed with three existing methods. Table 3 illustrates the comparison of our proposed method with the three existing methods in the literature, according to the PSNR and the amount of secret data. Consequently, our proposed algorithm outperformed three existing methods. Table 4 shows the amount of secret message of proposed algorithm in each Y, U, and V planes.

$$HR = \frac{Size\ of\ embedded\ message}{Video\ size} \times 100\% \tag{21}$$

Table 3 Performance comparison of the proposed algorithm with other according to both PSNR and hidden ratio

Criteria	Patel et al. [26]	Alavianmehr et al. [36]	Hu et al. [37]	Proposed Algorithm
PSNR (dB)	31.23	36.97	29.03	40.73
Hidden Ratio	12.5%	1.34%	18.75%	27.53%

Table 4 Embedding capacity of the proposed algorithm

Video Resolution	YUV	Proposed Algorithm (Bits/Frame)
352 × 288	Y	223344
	U	55836
	V	55836

7.3 Robustness

To measure the robustness of the proposed algorithm, the similarity (*Sim*) metric has been utilized. This metric is used to test whether the extracted secret message has been corrupted during communication [42]. The *Sim* $(0 \leq Sim \leq 1)$ can be calculated as in the following equation [43]:

$$Sim = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j)\times\widehat{M}(i,j)]}{\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b}M(i,j)^2} \times \sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b}\widehat{M}(i,j)^2}} \tag{22}$$

where M and $\widehat{M}$ are the embedded and extracted secret messages, respectively, and *a* and *b* are the dimensions of the secret message array. The algorithm is tested under different types of attacks (*Gaussian noise* with the zero mean and variance = **0.01** and **0.001**, *Salt & pepper* noise with the density = **0.01** and **0.001**, and *median filtering*). To achieve the robustness of the algorithm, the higher *Sim* must be obtained. Table 5 illustrates the robustness of the proposed algorithm under attacks while it retrieves the hidden data with a high *Sim*.

Table 5 *Sim* values of the proposed algorithm under attacks

		(Salt & Pepper) Density =		(Gaussian White) Variance =		
Sequences	No Attacks	0.01	0.001	0.01	0.001	Median Filtering
Akiyo	1	0.950	0.957	0.918	0.903	0.981
Bus	1	0.960	0.967	0.928	0.913	0.982
Coastguard	1	0.940	0.947	0.908	0.893	0.981
Container	1	0.970	0.977	0.938	0.923	0.993
Foreman	1	0.960	0.953	0.914	0.892	0.970
Soccer	1	0.918	0.926	0.897	0.868	0.954

8 Conclusion

A novel video steganography method in DCT domain based on Hamming and BCH (7, 4, 1) ECC has been proposed in this paper. The steganography algorithm converts the video into frames; then, it divides each frame into Y, U, and V components. Prior to the embedding process, the secret message is encrypted and encoded using hamming and BCH codes. The 2D-DCT has been applied to each YUV components. DCT coefficients, excluding DC coefficients, are selected for embedding the secret data.

The proposed algorithm has a high embedding payload. The amount of the secret data in each video is approximately **5.99** Mbytes and the *HR* is **27.53**%. The visual quality of the stego videos is also high: the PSNR ranged between **38.95**–**42.73** dBs with an *Sim*=**1**. Moreover, the experimental results showed that the proposed algorithm is robust against several attacks. In addition, the security of the our method is improved by ciphering and encoding processes prior to the embedding process. The result of comparison shows that the proposed algorithm outperformed three existing algorithms. For future work, we would like to improve the embedding payload of the proposed algorithm with the respect of the video quality by using other techniques that operate in frequency domain. Also, we would like to conduct efficient linear block codes to enhance the security of the algorithm.

References

[1] Mstafa, R. J. and Elleithy, K. M. (2016). A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes. *Multimed. Tools Appl.* 75, 10311–10333.

[2] Mstafa, R. J. and Elleithy, K. M. (2014). "A highly secure video steganography using Hamming code (7, 4)," in *Proceedings of the 2014 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, (New York, NY: IEEE), 1–6.

[3] Muhammad, K., Sajjad, M., and Baik, S. W. (2016). Dual-level security based cyclic18 steganographic method and its application for secure transmission of keyframes during wireless capsule endoscopy. *J. Med. Syst.* 40, 1–16.

[4] Muhammad, K., Sajjad, M., Mehmood, I., Rho, S., and Baik, S. W. (2016). Image steganography using uncorrelated color space and its application for security of visual contents in online social networks. *Fut. Gen. Comput. Syst*.

[5] Jyun-Jie, W., Houshou, C., Chi-Yuan, L., and Ting-Ya, Y. (2012). "An embedding strategy for large payload using convolutional embedding codes," in *Proceedings of the 12th International Conference on ITS Telecommunications (ITST),* (New York, NY: IEEE), 365–369.

[6] Zhang, R., Sachnev, V. and Kim, H. (2009). "Fast BCH syndrome coding for steganography," in *Information Hiding*, eds S. Katzenbeisser and A.-R. Sadeghi, Vol. 5806 (Berlin: Springer), 2009, 48–58.

[7] Mstafa, R. J. and Elleithy, K. M. (2015). "A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11)," in *Proceedings of the 2015 International IEEE Wireless Telecommunications Symposium*, (New York, NY: IEEE), 1–8.

[8] Abu-Marie, W., Gutub, A. and Abu-Mansour, H. (2010). Image based steganography using truth table based and determinate array on RGB indicator. *Int. J. Signal Image Process.* 1, 196–204.

[9] Das, R. and Tuithung, T. (2012). "A novel steganography method for image based on Huffman Encoding," in *Proceedings of the 2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, (Piscataway, NJ: IEEE), 14–18.

[10] Khan, A. and Malik, S. A. (2014). A high capacity reversible watermarking approach for authenticating images: exploiting down-sampling, histogram processing, and block selection. *Inform. Sci.* 256, 162–183.

[11] Horng, S.-J., Rosiyadi, D., Fan, P., Wang, X., and Khan, M. K. (2014). An adaptive watermarking scheme for e-government document images. *Multimed. Tools Appl.* 72, 3085–3103.

[12] Mstafa, R. J., and Elleithy, K. M. (2015). A new video steganography algorithm based on the multiple object tracking and hamming codes," in *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, (Piscataway, NJ: IEEE), 335–340.

[13] Mstafa, R. J. and Elleithy, K. M. (2016). "A novel video steganography algorithm in DCT domain based on hamming and BCH codes," in *Proceedings of the 2016 IEEE 37th Sarnoff Symposium*, (Newark, NJ: IEEE), 208–213.

[14] Muhammad, K., Sajjad, M., Lee, M. Y., and Baik, S. W. (2017). Efficient visual attention driven framework for key frames extraction from hysteroscopy videos. *Biomed. Signal Process. Control* 33, 161–168.

[15] Sajjad, M., Muhammad, K., Baik, S. W., Rho, S., Jan, Z., Yeo, S.-S., et al. (2016). Mobile-cloud assisted framework for selective encryption of medical images with steganography for resource-constrained devices. *Multimed. Tools Appl.* 1–18.

[16] Mstafa, R. J. and Elleithy, K. M. (2016). Compressed and raw video steganography techniques: a comprehensive survey and analysis. *Multimed. Tools Appl.* 75, 10311–10333.

[17] Shanableh, T. (2012). Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering. *Inform. For. Secur. IEEE Trans*. 7, 455–464.

[18] Kapotas, S. K. and Skodras, A. N. (2008). "A new data hiding scheme for scene change detection in H. 264 encoded video sequences," in *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo*, Hannover.

[19] Xu, C., Ping, X., and Zhang, T. (2006). "Steganography in compressed video stream," in *Proceedings of the 1st International Conference on Innovative Computing, Information and Control*, (Beijing: IEEE), 269–272.

[20] Huang, J., and Shi, Y. Q. (2002). Reliable information bit hiding. *Circuits Syst. Video Technol. IEEE Trans.* 12, 916–920.

[21] Barni, M., Bartolini, F., and Checcacci, N. (2005). Watermarking of MPEG-4 video objects. *Multimed. IEEE Trans.* 7, 23–32.

[22] Ke, N. and Weidong, Z. (2013). A video steganography scheme based on H. 264 bitstreams replaced, in *Proceedings of the 4th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, 447–450.

[23] Muhammad, K., Sajjad, M., Mehmood, I., Rho, S., and Baik, S. (2015). A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image. *Multimed. Tools Appl.* 1–27.

[24] Zhang, R., Sachnev, V., Botnan, M. B., Kim, H. J., and Heo, J. (2012). "An efficient embedder for BCH coding for Steganography," *Inf. Theory IEEE Trans.* 58, 7272–7279.

[25] Diop, I., Farss, S. M., Tall, K., Fall, P. A., Diouf, M. L., and Diop, A. K. (2014). Adaptive steganography scheme based on LDPC codes," in *Proceedings of the 2014 16th International Conference on Advanced Communication Technology (ICACT)*, Pyeong Chang, 162–166.

[26] Patel, K. Rora, K. K., Singh, K., and Verma, S. (2013). Lazy Wavelet Transform Based Steganography in Video," in *Proceedings of the Communication Systems and Network Technologies (CSNT), International Conference,* Delhi, 497–500.

[27] Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, Inc.

[28] Pennebaker, W. B. and Mitchell, J. L. (1992). *JPEG: Still Image Data Compression Standard*: Berlin: Springer Science & Business Media.

[29] Pan, F., Xiang, L., Yang, X.-Y., and Guo, Y. (2010). "Video steganography using motion vector and linear block codes," in *Proceedings of the Software Engineering and Service Sciences (ICSESS), IEEE International Conference,* Beijing, 592–595.

[30] Jue, W., Min-Qing, Z., and Juan-Li, S. (2011). "Video steganography using motion vector components," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference*, Xi'an, 500–503.

[31] Li, G., Ito, Y., Yu, X., Nitta, N., and Babaguchi, N. (2009). Recoverable privacy protection for video content distribution. *EURASIP J. Inf. Secur.* 2009, 4.

[32] Li, Y., Chen, H.-X., and Zhao, Y. (2010). "A new method of data hiding based on H. 264 encoded video sequences," in *Proceedings of the Signal Processing (ICSP), 2010 IEEE 10th International Conference*, Beijing, 1833–1836.

[33] Mobasseri, B. G. and Marcinak, M. P. (2005). "Watermarking of MPEG-2 video in compressed domain using VLC mapping," in *Proceedings of the 7th Workshop on Multimedia and Security*, New York, NY, 91–94.

[34] Wang, R., Hu, L., and Xu, D. (2011). A watermarking algorithm based on the CABAC entropy coding for H.264/AVC. *J. Comput. Inform. Syst.,* 7, 2132–2141.

[35] Cheddad, A., Condell, J., Curran, K., and Mc Kevitt, P. (2009). A skin tone detection algorithm for an adaptive approach to steganography. *Signal Process.* 89, 2465–2478.

[36] Alavianmehr, M. A., Rezaei, M., Helfroush, M. S., and Tashk, A. (2012). A lossless data hiding scheme on video raw data robust against H.264/AVC compression," in *Proceedings of the 2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, Hingham, MA, 194–198.

[37] Hu, S. and KinTak, U. (2011). "A Novel Video Steganography based on Non-uniform Rectangular Partition," in *Proceedings of the Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference*, Washington, DC, 57–61.

[38] Sun, S. (2015). A new information hiding method based on improved BPCS steganography. *Adv. Multimed.* 2015, 7.

[39] Spaulding, J., Noda, H., Shirazi, M. N., and Kawaguchi, E. (2002). BPCS steganography using EZW lossy compressed images. *Pattern Recognit. Lett.* 23, 1579–1587.

[40] Hoyoung, Y., Jaehwan, J., Jihyuck, J., and In-Cheol, P. (2013). Area-Efficient Multimode Encoding Architecture for Long BCH Codes. *IEEE Trans. Circ. Syst.* 60, 872–876.

[41] Mstafa, R. J. and Elleithy, K. M. (2015). "An efficient video steganography algorithm based on BCH codes," in *American Society for Engineering Education (ASEE Zone 1) Conference*, Boston, MA, 1–10.

[42] Mstafa, R. J. and Elleithy, K. M. (2015). "A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes," in *Proceedings of the 2015 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, New York, NY, 1–7.

[43] He, Y., Yang, G., and Zhu, N. (2012). "A real-time dual watermarking algorithm of H.264/AVC video stream for Video-on-Demand service," *AEU Int. J. Elect. Commun.* 66, 305–312.

Biographies

Ramadhan J. Mstafa is originally from Duhok, Kurdistan Region, Iraq. He is pursuing his Ph.D. 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 an IEEE and ACM Student Member. His research areas of interest include image processing, mobile communication, security, watermarking, and steganography.

Khaled M. 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 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 fifty 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.