EFFICIENT AND ROBUST VIDEO STEGANOGRAPHY ALGORITHMS FOR SECURE DATA COMMUNICATION

Ramadhan J. Mstafa

Under the Supervision of: Dr. Khaled M. Elleithy

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

June, 2017

EFFICIENT AND ROBUST VIDEO STEGANOGRAPHY

ALGORITHMS FOR SECURE DATA COMMUNICATION

Ramadhan J. Mstafa

Under the Supervision of Dr. Khaled M. Elleithy

Approvals

Committee Members

Name	Signature	Date
Dr. Khaled M. Elleithy		5/12/17
Dr. Navarun Gupta		5/10/17
Dr. Xingguo Xiong	Xingguo Xiong	05/09/2017
Dr. Miad Faezipour	Miad Faezipour	05,11,2017
Dr. Eman Abdelfattah	Eman Abdelfattah	5/23/2017

Ph.D. Program Coordinator

Dr. Khaled M. Elleithy		5/23/17

Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood	Ausif Mahmood	5-12-2017

Dean, School of Engineering

Dr. Tarek M. Sobh	T. Sobh	5-27-2017

EFFICIENT AND ROBUST VIDEO STEGANOGRAPHY ALGORITHMS FOR SECURE DATA COMMUNICATION

EFFICIENT AND ROBUST VIDEO STEGANOGRAPHY ALGORITHMS FOR SECURE DATA COMMUNICATION

ABSTRACT

Over the last two decades, the science of secretly embedding and communicating data has gained tremendous significance due to the technological advancement in communication and digital content. Steganography is the art of concealing secret data in a particular interactive media transporter such as text, audio, image, and video data in order to build a covert communication between authorized parties. Nowadays, video steganography techniques are important in many video-sharing and social networking applications such as Livestreaming, YouTube, Twitter, and Facebook because of noteworthy developments in advanced video over the Internet.

The performance of any steganography method, ultimately, relies on the imperceptibility, hiding capacity, and robustness against attacks. Although many video steganography methods exist, several of them lack the preprocessing stages. In addition, less security, low embedding capacity, less imperceptibility, and less robustness against attacks are other issues that affect these algorithms.

This dissertation investigates and analyzes cutting edge video steganography techniques in both compressed and raw domains. Moreover, it provides solutions for the aforementioned problems by proposing new and effective methods for digital video steganography.

The key objectives of this research are to develop: 1) a highly secure video steganography algorithm based on error correcting codes (ECC); 2) an increased payload video steganography algorithm in the discrete wavelet domain based on ECC; 3) a novel video steganography algorithm based on Kanade-Lucas-Tomasi (KLT) tracking and ECC; 4) a robust video steganography algorithm in the wavelet domain based on KLT tracking and ECC; 5) a new video steganography algorithm based on the multiple object tracking (MOT) and ECC; and 6) a robust and secure video steganography algorithm in the discrete wavelet and discrete cosine transformations based on MOT and ECC.

The experimental results from our research demonstrate that our proposed algorithms achieve higher embedding capacity as well as better imperceptibility of stego videos. Furthermore, the preprocessing stages increase the security and robustness of the proposed algorithms against attacks when compared to state-of-the-art steganographic methods.

ACKNOWLEDGEMENTS

My thanks are wholly devoted to God who has helped me all the way to complete this work successfully.

I would like to express my heartfelt gratitude to my supervisor, **Prof. *Khaled M. Elleithy***, for his support, encouragement, and valuable guidance.

Also, I would like to acknowledge the committee members **Dr. *Navarun Gupta***, **Dr. *Xingguo Xiong***, **Dr. *Miad Faezipour***, and **Dr. *Eman Abdelfattah*** for their valuable feedback and recommendation regarding this work.

My deepest gratitude goes to my family for their unflagging love and support throughout my life, and provides the best possible environment for me to grow up; this work would simply be impossible without them. I am indebted to my father, ***Jahfar Mstafa***, for his care and love, and thanks to my mother for all the moral support and continuing prayers she has given me over the years. Also, I would like thank my cousin, ***Mohammad***, for his support since he has constantly been one of my best supporters. I would like to extend my sincerest thanks and appreciation to my sisters and brothers who waited me, patiently, for years to accomplish this dissertation.

I owe a debt of gratitude to my beloved and dearest wife, ***Hozan***, and my three precious children, ***Rebeen***, ***Rugash***, and ***Kani***, for their time waiting, understanding, and encouragement.

I am also grateful to Ms. ***Camy Deck*** of English department, University of Bridgeport, Bridgeport, USA for proofreading and editing of my dissertation.

I would like to take this opportunity to thank all colleagues, friends, and relatives who have helped and inspired me during my doctoral study.

Finally, I would like to thank all the staff of the School of Engineering for their support that made my study in the University of Bridgeport a wonderful and exciting experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS AND NOMENCLATURE

AVI	Audio Video Interleave
B	Bi-directionally Predicted Frame
BCH	Bose, Chaudhuri, and Hocquenghem
BER	Bit Error Rate
BPCS	Bit-Plane Complexity Segmentation
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
Cb	Chrominance Blue
CGC	Canonical Gray Coding
Cr	Chrominance Red
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DST	Discrete Sine Transform
DWT	Discrete Wavelet Transform
ECC	Error Correcting Codes
EZW	Embedded Zerotree Wavelet
GF	Galois Field

GMM	Gaussian Mixture Model
GOP	Group of Pictures
HCV	Histogram Constant Value
HDC	Histogram Distribution Constrained
HEVC	High Efficiency Video Coding
HH	Diagonal Sub-band
HL	Vertical Sub-band
HR	Hidden Ratio
HVS	Human Visual System
I	Intra Predicted Frame
KLT	Kanade-Lucas-Tomasi
LCM	Least Common Multiple
LH	Horizontal Sub-band
LL	Approximation Sub-band
LSB	Least Significant Bit
LWT	Lazy Wavelet Transform
MB	Macroblock
MOT	Multiple Object Tracking
MSB	Most Significant Bit
MSE	Mean Square Error
MV	Motion Vector
P	Predicted Frame

PSNR	Peak Signal to Noise Ratio
QDCT	Quantized Discrete Cosine Transform
QP	Quantization Parameter
RGB	Red, Green, and Blue
ROI	Region of Interest
SAD	Sum of Absolute Differences
Sim	Similarity
SPIHT	Set Partitioning in Hierarchical Trees
Y	Luminance

CHAPTER 1: INTRODUCTION

In spite of the fact that the Internet is utilized as a medium to access desired information, it has also opened a new door for attackers to obtain precious information of other users with little effort [1]. Steganography has functioned in a complementary capacity to offer a protection mechanism that hides communication between an authorized transmitter and its recipient. Steganography is defined as the art of concealing secret information in specific carrier data, establishing covert communication channels between official parties [2]. Subsequently, a stego object (steganogram) should appear the same as an original data that has a slight change of the statistical features. Carrier data is also referred as cover or host data [3, 4]. Carriers can be acknowledged in various forms such as text, audio, image, and video. A hidden message can also appear in any form of data such as text, audio, image, and video [5]. The primary objective of steganography is to eliminate any suspicion to the transmission of hidden messages and provide security and anonymity for legitimate parties. The simplest way to observe the steganogram's visual quality is to determine its accuracy, which is achieved through the human visual system (HVS). The HVS cannot identify slight distortions in steganogram, thus avoiding suspiciousness [6]. However, if the size of the hidden message in proportion with the size of the carrier object is large, then the steganogram's degradation will be visible to the human eye resulting in a failed steganographic method [7].

Embedding efficiency, hiding capacity, and robustness are the three major requirements incorporated in any successful steganographic methods [8]. First, embedding efficiency can be determined by answering the following questions [9, 10]: 1) how safe is the steganographic method to conceal the hidden information inside the carrier object? 2) how precise are the steganograms' qualities after the hiding procedure occurs? and 3) is the secret message undetectable from the steganogram? In other words, the steganography method is highly efficient if it includes encryption, imperceptibility, and undetectability characteristics. The high efficient algorithm conceals the covert information into the carrier data by utilizing some of the encoding and encryption techniques prior to the embedding stage for improving the security of the underlying algorithm [11]. Figure 1.1 represents the general model of steganographic method.

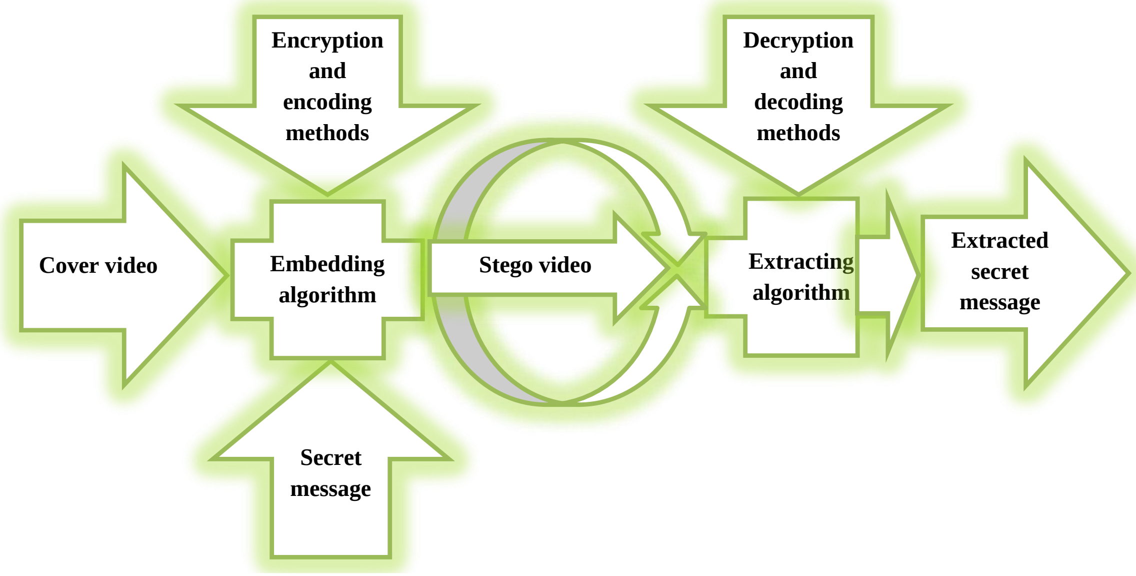


Figure 1.1 General diagram of the steganography method.

Steganograms with low alteration rate and high quality do not draw the hacker's attention, and thus will avoid any suspicion for the covert information. If the steganography method is more effective, then the steganalytical detectors will find it more challenging to detect the hidden message [12].

The hiding capacity is the second fundamental requirement which permits any steganography method to increase the size of hidden message taking into account the visual quality of the steganograms. The hiding capacity is the quantity of the covert messages needed to be inserted inside the carrier object. In ordinary steganographic methods, both hiding capacity and embedding efficiency are contradictory [13, 14]. Conversely, if the hiding capacity is increased, then the quality of steganograms will be diminished, decreasing the efficiency of underlying method. The embedding efficiency is affected by embedding capacity. To increase the hiding capacity with the minimum alteration rate of the carrier object, many steganographic methods have been presented using different strategies. These methods utilize linear block codes and matrix encoding principles which include Bose, Chaudhuri, and Hocquenghem (BCH) codes, Hamming codes, Cyclic codes, Reed-Solomon codes, and Reed-Muller codes [15, 16].

Robustness is the third requirement which measures the steganographic method's strength against attacks and signal processing operations [17]. These operations contain geometrical transformation, compression, cropping, and filtering. A steganographic method is robust whenever the recipient obtains the secret message accurately, without bit errors. An efficient steganography methods withstand against both adaptive noises and signal processing operations [18, 19].

1.1 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 [20]. 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 [21].

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 [22]. Like steganography, watermarking can be used in many different applications such as content authentication, digital fingerprints, broadcast monitoring, copyright protection, and intellectual property protection [22-26]. Different watermarking techniques can be found in the literature [27-35]. Table 1.1 shows the general similarities and differences between steganography, cryptography, and watermarking techniques.

1.2 Motivations and Research Problem

Video steganography is getting the attention of researchers in the area of video processing due to substantial growth in video data. The recent literature reports a significant amount of video steganography algorithms. Unfortunately, many of these algorithms lack the preprocessing stages. Particularly, there is no video steganography algorithm that includes preprocessing stages for both secret messages and cover videos. Moreover, existing steganography techniques suffer major weakness in several aspects including security, embedding capacity, imperceptibility, and robustness against attacks.

Table 1.1 Comparison of steganography, cryptography, and watermarking techniques.

Description	**Steganography**	**Cryptography**	**Watermarking**
Goal achieved:	Communication channels are covert	Data content of communication channels are covert	Copyright protection exists
Goal failed:	Communication is detected	Plain-text is retrieved	Watermark is erased or exchanged
Common carrier file:	Text, audio, image, or video	Plain-text or image	Image or video
Secret information:	Any type of data	plain-text	watermark
Secret keys:	May exist	Must exist	May exist
Extraction phase:	Carrier data is unnecessary	Carrier data is unnecessary during deciphering process	Carrier data availability depends on the application
Output file:	Steganogram	Cryptogram	Watermarked object
Security level:	Depends on the embedding algorithms	Depends on the secret keys	Depends on the watermarking algorithms
Information transparency:	Invisible	Visible	Transparency depends on the application
Robustness level:	Against detection	Against deciphering	Robust watermarking, fragile watermarking, and semi-fragile watermarking
Common attacks:	Steganalysis	Cryptanalysis	Signal processing operations
Requirements:	Embedding efficiency, embedding payload, undetectability, and robustness	Robustness	Robust watermarking requires robustness while fragile and semi-fragile watermarking do not need robustness

This research is motivated by the limitations of the existing video steganography algorithms, and is based on the following reasons to improve the performance of these algorithms:

1) By utilizing the preprocessing stages to include the procedure of manipulating both secret messages and cover videos prior to the embedding stage in order to enhance the security and robustness of the steganographic method.
2) Using a portion of each frame throughout all video as regions of interest for the embedding process, the imperceptibility of stego videos will improve. Accordingly, we track the facial regions and moving objects in video. Since it is very challenging for attackers to determine the location of secret message in video frames because the secret message is only embedded into facial regions and moving objects which changes from frame to frame, it is necessary to preserve the security and robustness of embedded data.
3) Applying encryption methods and ECC such as Hamming codes and BCH codes to encode the secret message prior to the embedding process will produce a secure and robust steganographic algorithm.
4) Transforming video frames into frequency domain such as DWT and DCT transformations will improve the robustness of the steganographic method against attacks, hence preserving imperceptibility of stego videos.

1.3 Main Contributions of the Proposed Research

Our research investigates some innovative approaches to improve video steganography methods. The main objective of this thesis is to develop and validate a new method to outperform the existing video steganography techniques from the literature. In this dissertation, the key contributions are as follows:

- A highly secure video steganography algorithm based on ECC is proposed. In order to enhance the security and robustness of this algorithm against attacks, the secret message is embedded into specific areas of each video frame, randomly. The algorithm achieves better imperceptibility of stego videos as well as higher embedding capacity of secret data [36, 37].
- An increased payload video steganography algorithm in the discrete wavelet domain based on ECC is proposed. This method not only improved the capacity of the encoded secret message, but also increased the robustness against attacks, providing a reasonable tradeoff with the imperceptibility [19, 38-40].
- A novel video steganography algorithm based on KLT tracking and ECC is proposed, which controls the limitations of some state-of-the-art steganographic algorithms in terms of security and imperceptibility. This algorithm utilizes facial regions as carrier data to conceal the secret message, which operates in the spatial domain [41].
- A robust video steganography algorithm in the wavelet domain based on KLT tracking and ECC is proposed. This method uses wavelet coefficients of facial regions as cover data to embed the secret information, hence enhancing the security and robustness of the hidden data [9].
- A new video steganography algorithm based on the MOT and ECC is proposed. This algorithm utilizes multiple motion objects in the video frames as regions of interest to hide the secret information, which operates in the spatial domain. This method improves each of imperceptibility and embedding capacity when compared to the related video steganography methods [42].

- A robust and secure video steganography algorithm in DWT-DCT domains based on MOT and ECC is proposed. The proposed algorithm has utilized MOT and ECC as the preprocessing stages which in turn provides a better confidentiality to the secret message prior to embedding phase. In order to enhance the security and robustness of the proposed method, both wavelet and cosine frequency coefficients of moving objects through the video frames are utilized as host data to embed the secret message [43].
- A comprehensive survey and analysis of state-of-the-art video steganography techniques in both compressed and raw domains is provided. This survey will guide the reader to comprehend the existing methods of video steganography and its main issues [44, 45].

CHAPTER 2: LITERATURE SURVEY

2.1 Introduction

Due to the advancement of the Internet and multimedia technologies, digital videos have become a popular choice for data hiding. The digital video contains a massive amount of data redundancy which can be utilized for embedding secret messages. Recently, there are many useful applications of video steganography techniques such as video error correcting [42, 46-48], military services [49], bandwidth saving [50, 51], video surveillance [41, 52, 53], and medical video security [54, 55].

In the past, many video steganography techniques have been proposed in the literature. Unfortunately, this literature lacks video steganography survey articles. So, we conducted an extensive study that included all video steganography techniques from past decades. This thesis provides a comprehensive survey and analysis of state-of-the-art video steganography methods. Figure 2.1 clarifies the hierarchy of the overall system protection which includes video steganography. Video steganography methods are classified into compressed and raw domains.

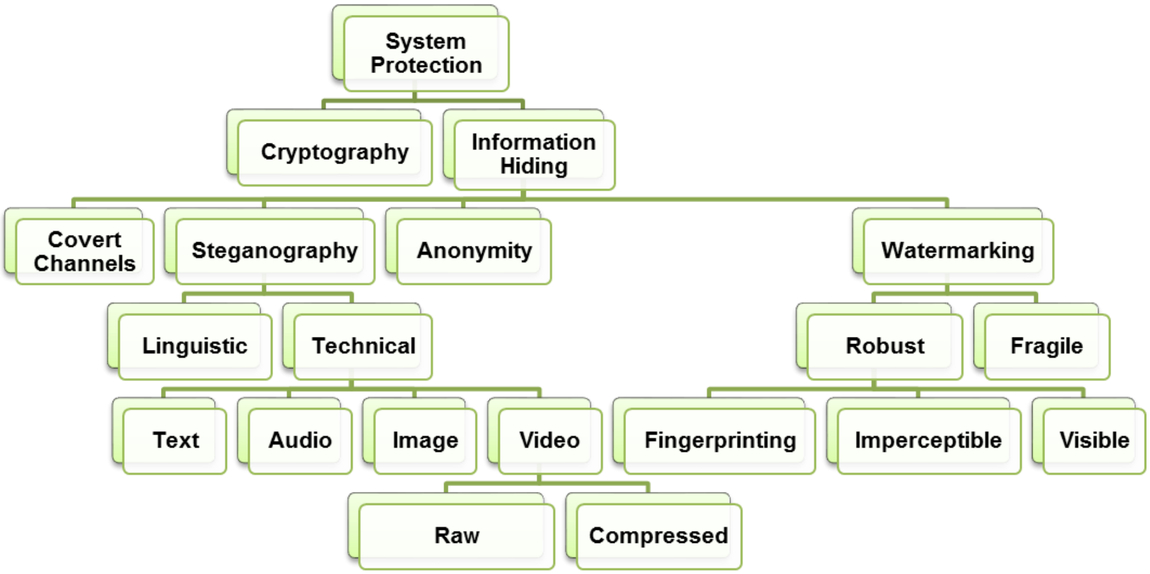


Figure 2.1 Disciplines of overall system protection.

2.2 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 [56-59]. 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 16x16 macroblocks (MB) wherein each MB contains blocks that may include the smallest size of 4x4. 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).

$$e = SAD\left(C, \tilde{R}\right) = \sum_{1 \leq i,j \leq b} \left| c_{i,j} - \tilde{r}_{i,j} \right| \tag{2.1}$$

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 (MV) and differential error $D = C - \tilde{P}$ are required for the coding process. In compressed domain, video steganography techniques are categorized according to the video coding stages. These stages are used as venues for data hiding such as intra frame prediction, inter frame prediction, motion vectors, transformed and quantized coefficients, and entropy coding. Figure 2.2 illustrates the H.264 video codec standard that indicates some venues for data hiding.

2.2.1 Video steganography techniques in 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 for 4x4 blocks and four for 16x16 blocks which are illustrated in Figure 2.3 and Figure 2.4, respectively. Also, the high efficiency video coding (HEVC) codec can support up to 35 intra prediction modes for each 64x64, 32x32, 16x16, 8x8, and 4x4 block sizes as shown in Figure 2.5. For data concealing purposes, these modes can be mapped to one or more of secret message bits.

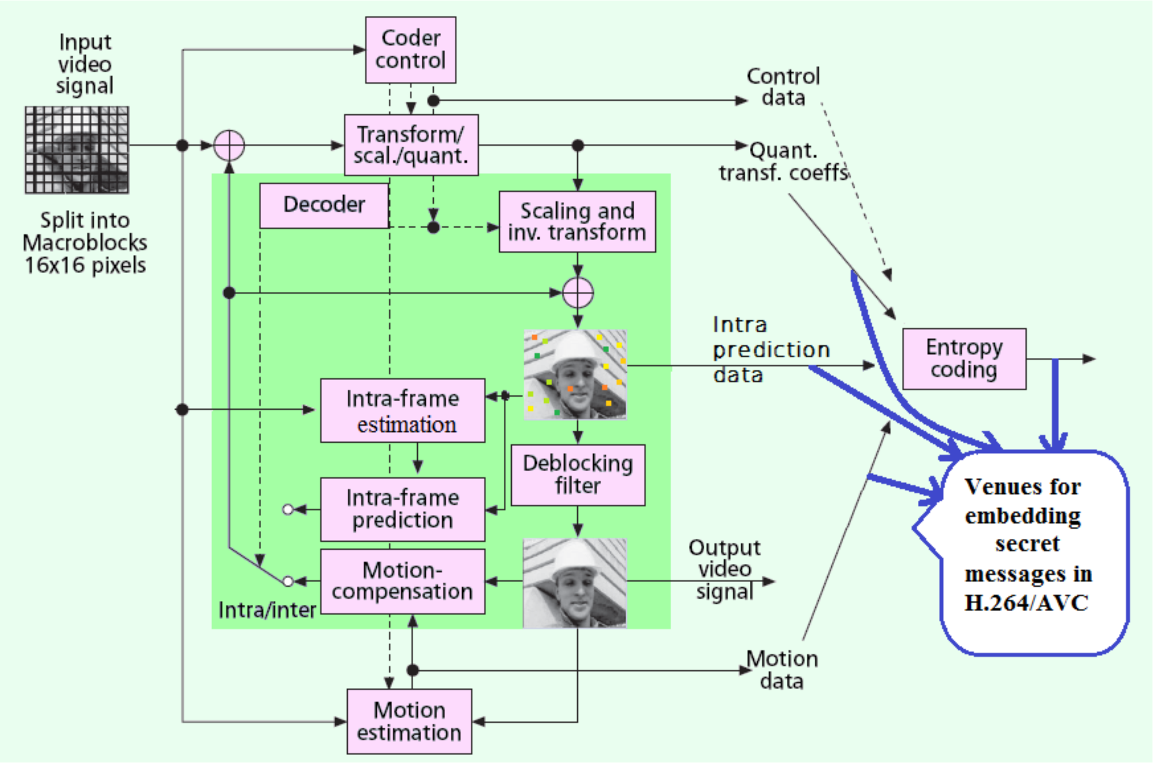


Figure 2.2 H.264 hybrid video codec standard shows venues for data hiding.

Liu *et al.* [60] presented a new secure data hiding technique which performs entirely in a compressed domain. The framework of this algorithm consists of four stages. First, in the video sequences parser stage, the video sequences are coded, and discrete cosine transform (DCT) coefficients are obtained. In addition, the motion vectors, and the intra coded macroblocks are acquired. In the second stage, scene detection is performed on the consecutive intra frames to identify the fluctuation scenes. The fluctuation scene is identified using a histogram variation of DC coefficients within intra frame DCT coefficients. In the third stage, the embedding process is achieved using only intra frames of fluctuation scenes. The last stage is called video steganalysis. Here, the security level of the stego video is statistically measured to determine whether it is high or low. If the stego video is detected by the steganalysis, then it will adjust the scale factor to make it stronger. The algorithm introduced by Liu et al. has limited capacity for hidden data because the fluctuation scenes of intra frames are only used for data embedding.

Chang *et al.* [61] presented a data concealing algorithm using HEVC utilizing both DCT and discrete sine transform (DST) methods. In this scheme, HEVC intra frames are used to conceal the hidden message without propagating the error of the distortion drift to the adjacent blocks. Blocks of quantized DCT and DST coefficients are selected for embedding the secret data by using a specific intra prediction mode. The combination modes of adjacent blocks will produce three directional patterns of error propagation for data hiding, consisting of vertical, horizontal, and diagonal. Each of the error propagation patterns has a range of intra prediction modes that protect a group of pixels in any particular direction. The range of the modes begins at 0 and ends at 34. Chang et al.'s algorithm has a low embedding payload because the selection of blocks for the embedding process must meet predefined conditions.

Hu *et al.* [62] and Zhu *et al.* [63] presented data hiding methods using intra prediction modes for H.264/AVC. During the intra frame coding process, the secret message is embedded into the 4x4 luminance block. These algorithms utilize the 4x4 intra prediction modes in order to hide one bit of secret information per block. The 4x4 intra prediction modes are divided into two subsets based on the predefined mapping rules between the secret message and intra prediction modes in order to embed 0 or 1 of the secret message bits. Table 2.1 illustrates the mapping rule of 4x4 intra prediction modes of the Hu et al.'s method, which shows that each most probable mode and its candidate modes mapped to 0 or 1. Both Hu and Zhu methods achieve a negligible degradation of video quality as well as a small increase on the bit rate. In general, the steganographic techniques that use the intra frame prediction as venues for data hiding have low embedding capacities to conceal secret messages.

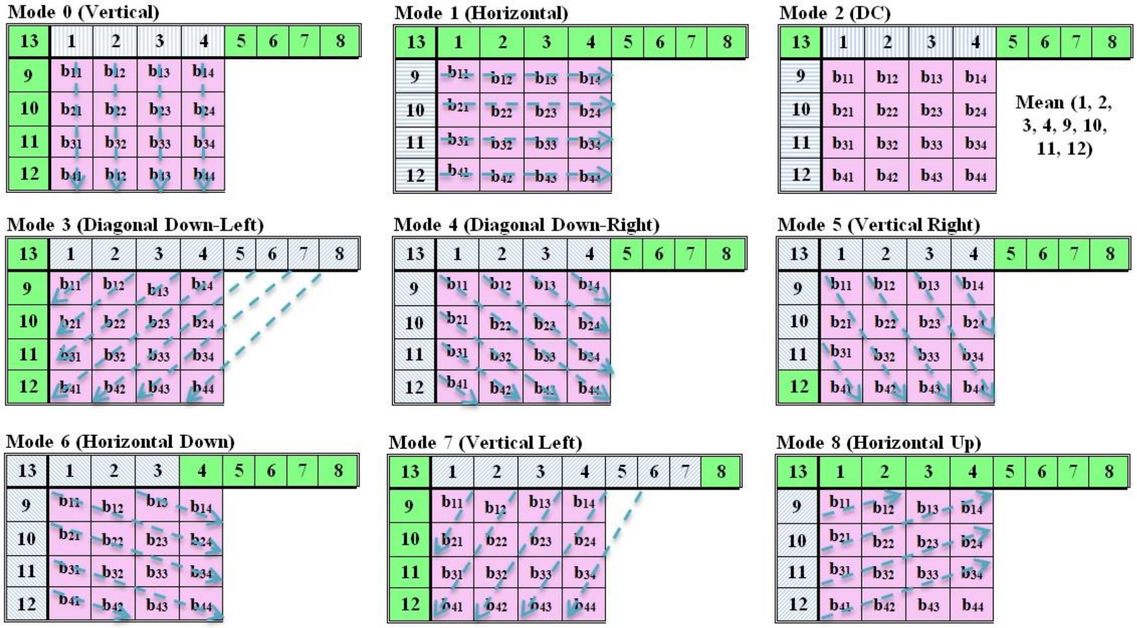


Figure 2.3 H.264 intra prediction modes for 4x4 blocks.

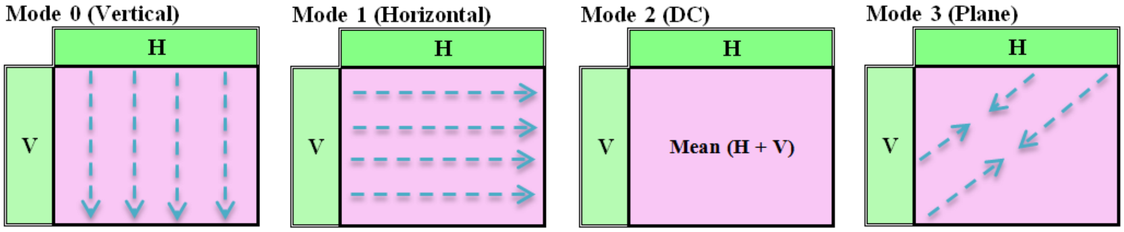


Figure 2.4 H.264 intra prediction modes for 16x16 blocks.

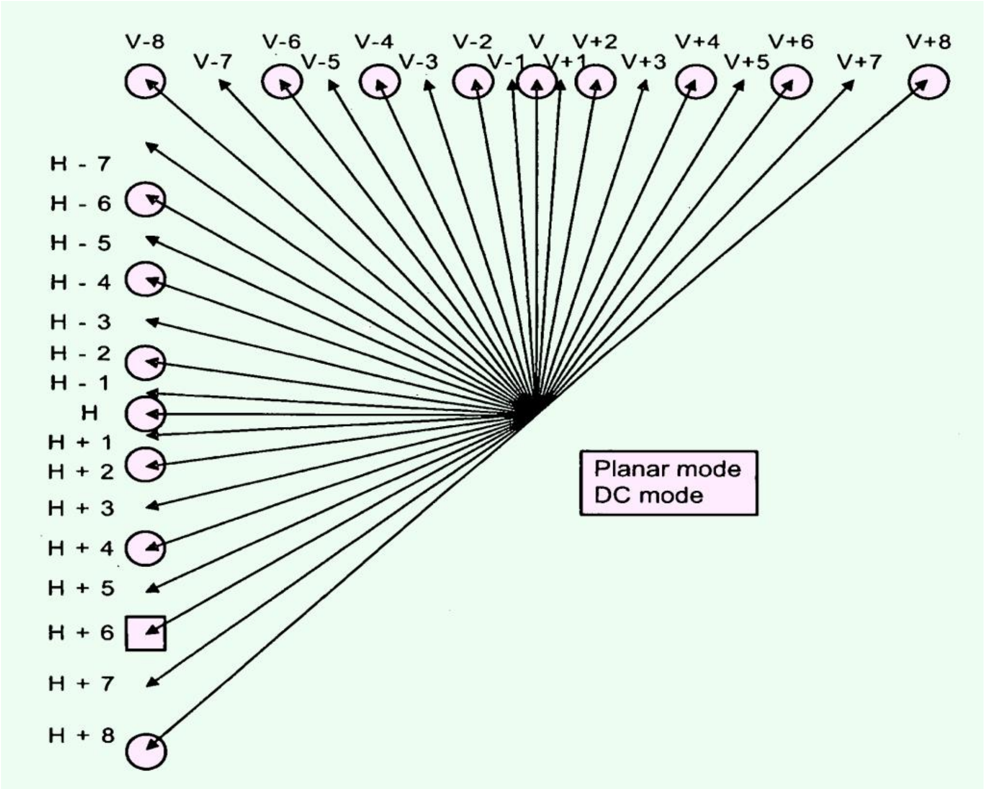


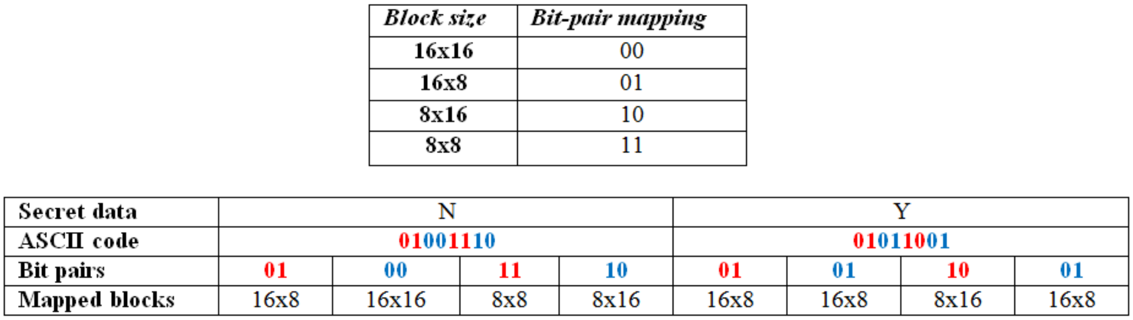
Figure 2.5 The 35 HEVC intra prediction modes [64].

Table 2.1 Mapping rules for 4x4 intra prediction modes [62].

Most Probable Mode	**Candidate Modes Mapping to 0**	**Candidate Modes Mapping to 1**
Mode 0	1, 2, 3, 4	5, 6, 7, 8
Mode 1	0, 3, 4, 8	2, 5, 6, 7
Mode 2	0, 3, 4, 8	1, 5, 6, 7
Mode 3	0, 5, 6, 8	1, 2, 4, 7
Mode 4	0, 3, 6, 8	1, 2, 5, 7
Mode 5	0, 3, 6, 8	1, 2, 4, 7
Mode 6	0, 3, 4, 8	1, 2, 5, 7
Mode 7	0, 5, 6, 8	1, 2, 3, 4
Mode 8	0, 1, 3, 4	2, 5, 6, 7

2.2.2 Video steganography techniques in inter frame prediction

In many video steganography methods, the seven block sizes that include 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 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.* [65] 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. An example of steganographic method that uses mapping rules, let "NY" is the secret information that must be embedded into the inter frame prediction blocks in H.264 codec. The embedding goal can be achieved by using mapping rules of different block sizes. Figure 2.6 illustrates the embedding process using mapping rules.



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

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 2.6 Using mapping rules for prediction block type to conceal "NY" characters.

2.2.3 Video steganography techniques in motion vectors

Motion vector characteristics such as horizontal, vertical, amplitude, and phase angles components are utilized in embedding secret information. Xu *et al.* [66] 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 P and B frames. Only the motion vectors that have a high magnitude 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.

Pan *et al.* [67] presented a new steganography method in the H.264 video standard based on the motion vectors and linear block codes. The embedding process is achieved by using motion vectors of inter frames macroblocks, and, then discarding the surrounding macroblocks. By using a predefined threshold, a group of motion vectors are selected in each video inter frame. The values (0 or 1) of selected motion vectors (MV_r) are obtained by calculating the phase angles (φ) illustrated in Figure 2.7. By definition, phase angles are the arctangents of both vertical ($\mathrm{MV_v}$) and horizontal ($\mathrm{MV_h}$) motion vectors' components as given in Eq. 2.2.

$$\varphi = arctan\left(\frac{MV_v}{MV_h}\right) \qquad (0^{\circ} \leq \ \varphi_i < 360^{\circ}) \tag{2.2}$$

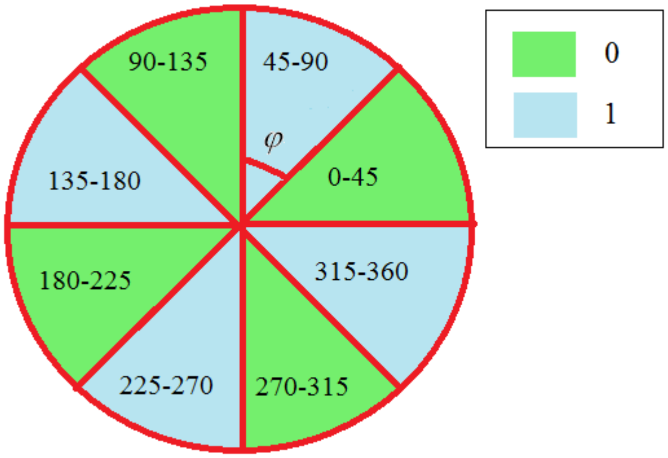


Figure 2.7 Motion vector representation in [67].

Once the MV_r values are obtained, the hidden information is concealed into the motion vector array utilizing the linear block code principle. The reason for using the linear block codes is to minimize the motion vectors' alteration rate and increase embedding capacity. The results of this algorithm have demonstrated that in every 6 bits of motion vector array, 4 bits of the secret data can be hidden. The peak signal to noise ratio (PSNR) of the obtained stego videos is 37.45 dB, which is proven by reducing alteration rate of motion vectors. However, this method has a limited hiding capacity due to it is based on the number of motion vectors. The data embedding and extracting phases of the Pan et al.'s method is illustrated in Eq. 2.3-2.6 as follows:

$$SY = \ MV_r H^T \tag{2.3}$$

$$b = SY \oplus S \tag{2.4}$$

$$MV_r^w = MV_r \oplus E_b \tag{2.5}$$

$$S' = MV_r H^T \oplus E_b H^T \tag{2.6}$$

Where S and S$'$ are embedded and extracted messages. MV_r and MV_r^w are original and stego selected motion vectors. SY, E_b, and H^T are syndrome, coset leader of b, and transpose of parity check matrix, respectively [67].

Bin *et al.* [68] presented a new data concealing algorithm using the motion vector and matrix encoding processes. The naked eye can realize the difference that happens when the object moves tardily, while if the object transfers rapidly, then the change will be unnoticeable. The motion vectors that have large amplitudes are produced from the macroblocks that move quickly. The sizable motion vectors will be utilized for concealing the hidden message. The selected motion vectors for data embedding include two properties: 1) the motion vector's amplitude must be greater than the predefined threshold T; and 2) both the vertical and the horizontal motion vector components must not be equal. Moreover, the best component ($\mathrm{MV_w}$) of both the vertical (MV_v) and the horizontal ($\mathrm{MV_h}$) motion vectors are chosen based on their phase angles (θ). Then, the process of hiding the secret message is performed using matrix encoding, reducing the modification rate of selected motion vectors. The least significant bit (LSB) of the selected motion vectors (MV_{w_LSB}) is utilized for embedding secret bits. The average PSNR of the stego videos is 38.18 dB [68]. However, this algorithm has a low embedding capacity because the selected motion vectors have restricted conditions. The embedding stage of the algorithm introduced by Bin et al. can be carried out as follows:

$$MV_w = \begin{cases} MV_h & 0 \leq \theta < \pi/4 \\ MV_v & \pi/4 \leq \theta < \pi/2 \end{cases} \tag{2.7}$$

$$\theta = \arctan|MV_v/MV_h| \tag{2.8}$$

$$MV_{w\_LSB} = \begin{cases} unchanged & ;if \quad MV_w = 0 \\ 1 & ;if \quad MV_{w\_LSB} = 0 \; and \; MV_w \neq 0 \\ 0 & ;if \quad MV_{w\_LSB} = 1 \; and \; MV_w \neq 0 \end{cases} \tag{2.9}$$

In a different work, Jue *et al.* [69] designed a new algorithm for H.264/AVC video steganography using motion vectors as cover data. In this scheme, the luminance macroblocks for inter frames (P and B) video coding is used. Using a predefined threshold, the motion vectors with a large magnitude will be selected, while the motion vectors of slow objects will be discarded. Then, the hidden data bits will be concealed into the difference of both horizontal and vertical components for the selected motion vectors. This algorithm has improved the utilization ratio and the embedding efficiency. The modified motion vector's feature ($\widehat{\mathrm{P}}_{\mathrm{i}}$) including the secret message can be calculated as follows:

$$\hat{P}_i = \begin{cases} mod\left[|V_{dx}| - |V_{dy}|, 2\right] & ;if \quad P_i = S_i \\ mod\left[|V_{dx} + 0.25| - |V_{dy}|, 2\right] & ;if \quad P_i \neq S_i \quad and \\ & \quad |V_{dx}| - |V_{dy}| \geq 0 \\ mod\left[|V_{dx}| - |V_{dy} + 0.25|, 2\right] & ;if \quad P_i \neq S_i \quad and \\ & \quad |V_{dx}| - |V_{dy}| < 0 \end{cases} \tag{2.10}$$

P_i and S_i are motion vector features and secret message bits. V_{dx} and V_{dy} are horizontal and vertical motion vector components, respectively. However, Jue et al.'s scheme is limited to the embedding payload due to the high value of the predefined threshold.

The steganographic techniques that commonly utilize motion vectors as carrier objects to hide the secret messages have low embedding capacities. Moreover, a high modification rate on the motion vectors will negatively influence the quality of the stego videos.

2.2.4 Video steganography techniques in transform coefficients (DCT, QDCT, and DWT)

The DCT, quantized discrete cosine transform (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. Shahid *et al.* [70] proposed a reconstruction loop for information embedding of intra and inter frames for H.264/AVC video codec. This method embeds the secret message into the LSB of QDCT coefficients. Only non-zero QDCT coefficients are chosen for data hiding process, utilizing the predefined threshold which directly depends on the size of secret information. Edges, texture, and motion regions of intra and inter frames are utilized in the concealing process. Shahid et al.'s algorithm extracts the hidden message easily and maintains the efficiency of compression domain.

On the other hand, Thiesse *et al.* [71-73] presented a steganography of motion data in the chrominance and luminance of video frame components. In order to control the modification of bitrate in the H.264 codec, the motion vector indices are embedded into the selected DCT coefficients of both luminance and chrominance components. In addition, the hidden indices minimize the distortion drift propagation of the prediction process to the next frames utilizing the rate-distortion optimization. The summation of the selected QDCT coefficients (S_i^w) is modified as follows:

$$S_i^w = \begin{cases} S_i & ; if\ |S_i|\ mod\ 2 = I_i \\ S_i + m_i & ; if\ |S_i|\ mod\ 2 \neq I_i \end{cases} \tag{2.11}$$

$$S_i = \sum_{n=1}^{N} a_n \tag{2.12}$$

Where a_n represents quantized coefficients, and S_i represents the summation of quantized coefficients of the i^{th} block. I_i is the prediction index and m_i represents shifted coefficients. In this method, the distortion drift propagation is low. However, the embedding capacity of the secret message is limited.

Meuel *et al.* [74] proposed information concealing in H.264 codec for lossless reconstruction of the region of interest (ROI). This method protects the facial features of video stream by embedding facial regions into the DCT coefficients. Two LSBs of non-zero QDCT coefficients are utilized to embed the facial information. Only the skip mode is used during inter coded prediction of the ROI. Both DC and AC DCT coefficients of ROI macroblocks are set to 1 and 0, respectively, in order to guarantee predicting the original ROI macroblocks during the decoding process. The facial pixels are determined as skin pixels if the Euclidean distance is lower than the predefined threshold value d using the following formula:

$$\sqrt{(P_u - Ref_u)^2 + (P_v - Ref_v)^2} < d \tag{2.13}$$

Where P_u and Ref_u are the *Cb* and its reference components, respectively, P_v and Ref_v are the *Cr* and its reference components, respectively. The suggested method of

Meuel *et al.* achieved a high quality of the region of interest based on the lossless reconstruction.

In a different work, Li *et al.* [75] proposed a new algorithm for H.264 video steganography. During the video coding process, the quantized coefficients in each 4x4 luminance of inter frame macroblocks are used for embedding the secret message. The majority zero values of quantized coefficients are located on the bottom-right corner because it is a high frequency region. Conversely, the majority of non-zero values of quantized coefficients belonging to low frequency band are located on the top-left corner. An array of inverse zigzag scan mode equaled to every 16 quantized coefficients will be produced in order to obtain the last non-zeros more efficiently. Using a predefined threshold T (0-15), based on the scan point, the last non-zero coefficient is selected in every macroblock. Depending on the parity of odd and even, the secret message of 1-bit per block is concealed. If the hidden bit is 1, then the selected DCT coefficients (V) modifies as follows:

$$\hat{V} = \begin{cases} V & if\ V\ mod\ 2 = 1 \\ V - 1 & if\ V\ mod\ 2 = 0\ and\ V > 0 \\ V + 1 & if\ V\ mod\ 2 = 0\ and\ V < 0 \end{cases} \tag{2.14}$$

Otherwise, the selected DCT coefficients (V) are modified as follows:

$$\hat{V} = \begin{cases} V & if\ V\ mod\ 2 = 0 \\ V + 1 & if\ V\ mod\ 2 = 1\ and\ V > 0 \\ V - 1 & if\ V\ mod\ 2 = 1\ and\ V < 0 \end{cases} \tag{2.15}$$

Li’s method has limited data embedding payload because the selected blocks embed only one bit per 4x4 block. Correspondingly, both Ma *et al.* [76] and Liu *et al.* [77] presented a video data hiding for H.264 coding without having an error accumulation in intra video frames. In the intra frame coding, the current block predicts its data from the encoded adjacent blocks, specifically from the boundary pixels of upper and left blocks. Thus, any embedding process that occurs in these blocks will propagate the distortion, negatively, to the current block. In addition, the distortion drift will be increased toward the lower right intra frame blocks. To prevent this distortion drift, authors have developed three conditions to determine the directions of intra frame prediction modes. The 4x4 blocks have nine prediction modes (0-8) and 16x16 blocks have four prediction modes (vertical, horizontal, DC, and plane). In the 4x4 block, the first condition is the right mode $\{0,3,7\}$; the second condition is both the under-left mode $\{0,1,2,4,5,6,8\}$ and the under mode $\{1,8\}$; and the third condition is the under right-mode $\{0,1,2,3,7,8\}$. To select 4x4 QDCT coefficients of the luminance component for data embedding, the three conditions must be presented together. However, the two methods have a low embedding payload because only the luminance of the intra frame blocks that meet the three conditions are selected for hiding data.

Later, Liu *et al.* [78, 79] presented a robust data hiding using H.264/AVC codec without a deformation accumulation in the intra frame based on BCH codes. By using the directions of the intra frame prediction, the deformation accumulation of the intra frame can be prevented. Some blocks will be chosen as carrier object for concealing the covert message. This procedure will rely on the prediction of the intra frame modes of adjacent blocks to prevent the deformation that proliferates from the neighboring blocks. The authors used BCH encoding to the hidden message before the embedding phase to enhance the method performance. Then, the encoded information is concealed into the 4x4

QDCT coefficients using only a luminance plane of the intra frame. Liu *et al.* defined *N* as a positive integer and $\widetilde{\mathrm{Y}}_{\mathrm{ij}}$ as selected DCT coefficients (i, j=0,1,2,3). The embedding process of this method is carried out by the following steps:

1. If $\left|\widetilde{\mathrm{Y}}_{\mathrm{ij}}\right| = \mathrm{N} + 1$ or $\left|\widetilde{\mathrm{Y}}_{\mathrm{ij}}\right| \neq \mathrm{N}$, then the $\widetilde{\mathrm{Y}}_{\mathrm{ij}}$ will be modified as follows:

$$\tilde{Y}_{ij} = \begin{cases} \tilde{Y}_{ij} + 1 & \text{if } \tilde{Y}_{ij} \geq 0 \text{ and } \left|\tilde{Y}_{ij}\right| = N + 1 \\ \tilde{Y}_{ij} - 1 & \text{if } \tilde{Y}_{ij} < 0 \text{ and } \left|\tilde{Y}_{ij}\right| = N + 1 \\ \tilde{Y}_{ij} & \text{if } \left|\tilde{Y}_{ij}\right| \neq N + 1 \; or \; \left|\tilde{Y}_{ij}\right| \neq N \end{cases} \tag{2.16}$$

2. If the secret bit is 1 and $\left|\tilde{Y}_{ij}\right| = N$, then the $\tilde{Y}_{ij}$ will be changed as follows:

$$\tilde{Y}_{ij} = \begin{cases} \tilde{Y}_{ij} + 1 & \text{if } \tilde{Y}_{ij} \geq 0 \text{ and } \tilde{Y}_{ij} = N \\ \tilde{Y}_{ij} - 1 & \text{if } \tilde{Y}_{ij} < 0 \text{ and } \tilde{Y}_{ij} = N \end{cases} \tag{2.17}$$

3. If the secret bit is 0 and $\left|\widetilde{\mathrm{Y}}_{\mathrm{ij}}\right| = \mathrm{N}$, then the $\widetilde{\mathrm{Y}}_{\mathrm{ij}}$ will not be modified.

Overall, the previously mentioned methods that use DCT, QDCT, and DWT coefficients as venues to hide secret messages are restricted to a limited number of coefficients in the embedding phase. Moreover, these algorithms do not include the secret message and cover data preprocessing stages, which are necessary to improve security and robustness of any of the steganographic methods.

2.2.5 Video steganography techniques in 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 steganography techniques. Ke *et al.* [80] presented a video steganography method relies on replacing the bits in H.264 stream.

In this algorithm, CAVLC entropy coding has been applied in the data concealing process. During the video coding and after the quantization stage, authors used non-zero coefficients of high frequency regions for the luminance component of the embedding process. Here, non-zero coefficients in high frequency bands are almost "+1" or "-1". 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 are modified as follows:

$$Trailing\ Ones = \begin{cases} even\ codeword & ; if\ secret\ bit = 0 \\ odd\ \ codeword & ; if\ secret\ bit = 1 \end{cases} \tag{2.18}$$

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

Similarly, Liao *et al.* [81] proposed real-time data concealing in H.264/AVC codec. During the process of CAVLC in 4x4 blocks, the trailing ones are utilized for embedding the secret data. The performance of this method was achieved through low computational complexity, negligible degradation of the video quality, and an unchangeable bit-steam size. This method employed random sequences as secret data. It is embedded into the selected blocks of CAVLC trailing ones as follows:

$$\hat{T}_{Ones} = \begin{cases} 2 & ; if\ w = 0\ and\ Trailing\ Ones \geq 3 \\ 1 & ; if\ w = 1\ and\ Trailing\ Ones = 2, \\ & \quad or \\ & \quad w = 1\ and\ \ Trailing\ Ones = 0 \\ 0 & ; if\ w = 0\ and\ Trailing\ Ones = 1 \\ unchanged & ; otherwise \end{cases} \tag{2.19}$$

Where w represents secret data that is hidden into the trailing ones codeword within range of 0 to 3. $\hat{T}_{Ones}$ represents modified trailing ones.

Additionally, Lu *et al.* [82] proposed real-time frame dependent video watermarking in CAVLC coding. In order to achieve the real-time detection, the CAVLC encoder is applied during this algorithm. During the process of video coding, the secret data is embedded into the run-level pairs of each frame's macroblocks. Table 2.2 illustrates run-level pairs (r, l) and codewords of the CAVLC encoder. This algorithm keeps the bit-rate almost unchangeable. However, it has limited embedding capacity for secret messages. The block diagram of the data hiding process that was introduced by Lu et al. is illustrated in Figure 2.8.

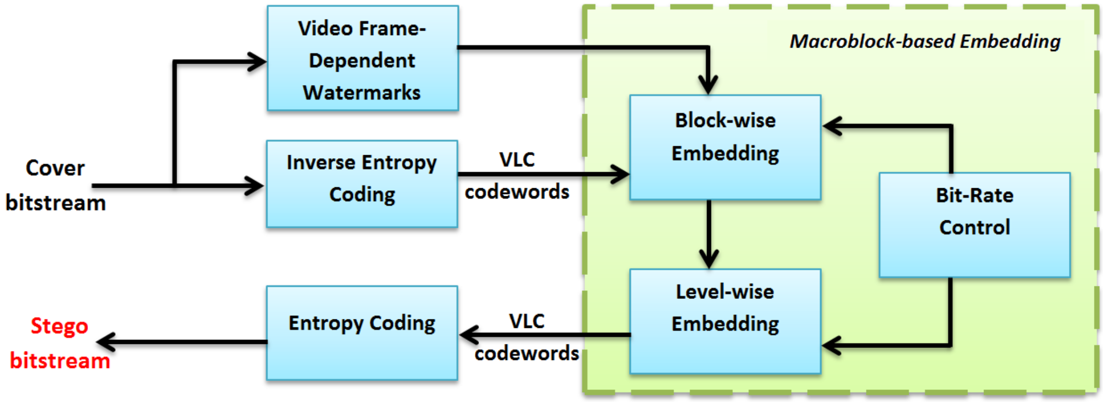


Figure 2.8 Block diagram of the embedding process in method [82].

Table 2.2 VLC table (s denotes the sign bit).

(run, level)	**Variable length code**	**Bit length**
(0,1)	11s	3
(0,2)	0100s	5
(0,3)	0010 1s	6
(0,4)	0000 110s	8
(0,5)	0010 0110s	9

Wang *et al.* [83] presented a real-time watermarking method in the H.264/AVC codec based on the CABAC features. The CABAC encoder uses a unary binarization, which is a process of concatenating all binary values of syntax elements. A certain number of motion vectors for both P and B frames are utilized for the data hiding process using the CABAC properties. The secret watermark is embedded by displacing the binary sequence of the selected syntax elements orderly. This method achieves a low degradation of the video quality because of the difference between the original code and the replacement code is very small (at most 1 bit is altered out 8-bits of the selected motion vector). This small difference is also the reason of achieving a little bit-rate increase. The percentage of the increased bit-rate μ is calculated as follows:

$$\mu = \frac{m - u}{u} \times 100\% \tag{2.20}$$

where u and m indicate the bit-rate of the original and the watermarked videos respectively. The flowchart of Wang’s method is illustrated in Figure 2.9. Also, the diagram of the CABAC encoder is shown in Figure 2.10.

Generally, the previous methods that utilize CAVLC and CABAC entropy coding as venues to conceal secret messages are limited in capacity due to the restricted number of selected blocks in the embedding stage. Moreover, when using the entropy coding, the quality of the steganogram is severely distorted.

Table 2.3 clarifies the advantages and limitations of each venue for concealing secret messages in the compressed domain. These venues include: intra frame prediction, inter frame prediction, motion vectors, DCT coefficients, QDCT coefficients, DWT coefficients, CAVLC entropy coding, and CABAC entropy coding.

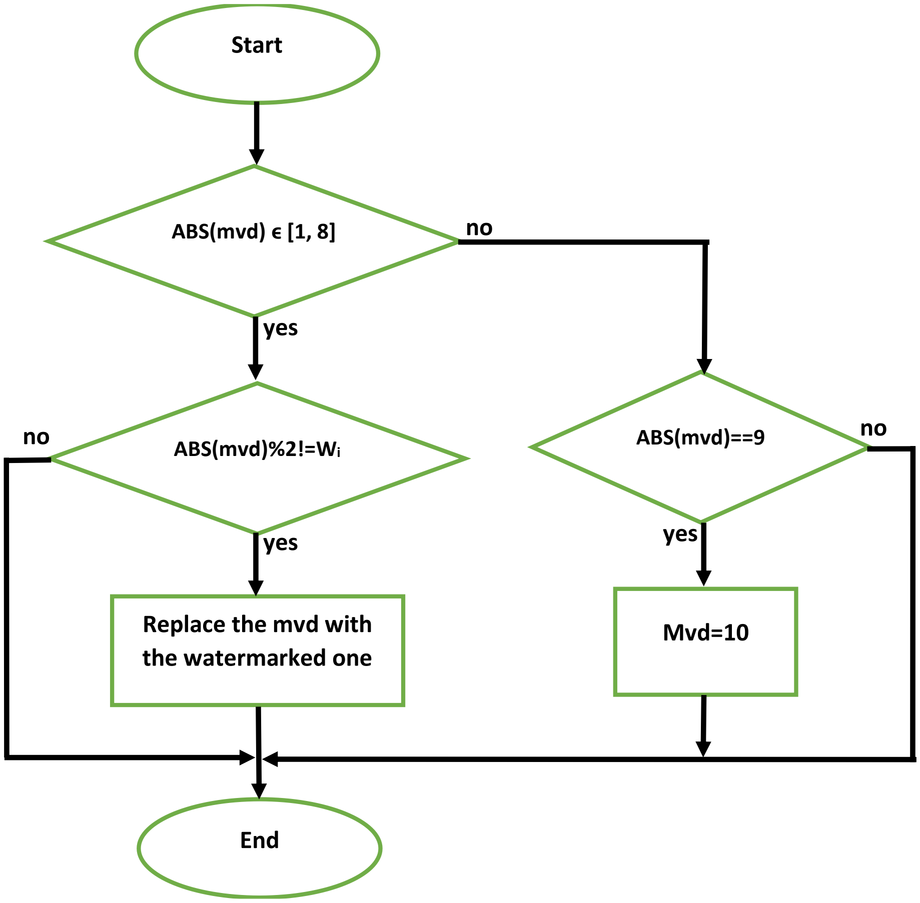


Figure 2.9 The data embedding framework in [83].

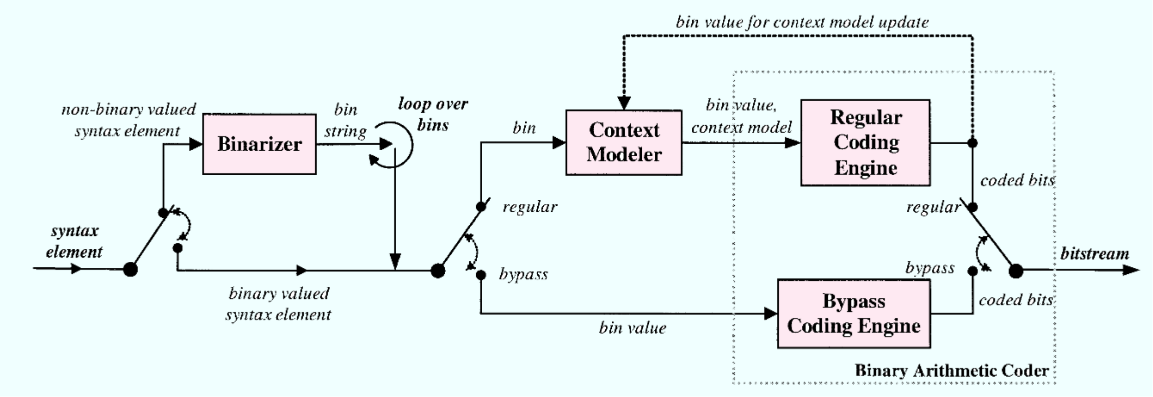


Figure 2.10 General block diagram of the CABAC encoder.

Table 2.3 Advantages and disadvantages of each venue for data concealing in compressed domain.

Venues for data hiding	**Characteristics (According to compressed video steganography techniques)**	**Limitations**
Intra frame prediction	The computational complexity is moderate	The embedding capacity is low and the impact on the video quality is high
Inter frame prediction	The influence on the video quality and the computational complexity are low	The embedding capacity needs to be improved
Motion vectors	Both embedding payload and computational complexity are moderate	The impact on the video quality is high
DCT/QDCT/DWT coefficients	Achieve a high embedding payload as well as a low computational complexity	The influence on the video quality is high
CAVLC/CABAC entropy coding	Achieve a high embedding payload as well as a low computational complexity	The quality of the steganogram is severely distorted

2.3 Video steganography Techniques in Raw Domain

Unlike the compressed video, raw video steganography 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 steganography techniques operate in both spatial and transform domains [84].

2.3.1 Video steganography techniques in spatial domain

There are many steganography techniques that rely on the spatial domain such as LSB substitution, bit-plane complexity segmentation (BPCS), spread spectrum, ROI, histogram manipulation, matrix encoding, and mapping rule. Basically, these techniques utilize the pixel intensities to conceal the secret message. Zhang *et al.* [85] 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. Zhang et al.'s method modifies the complexity of the algorithm from exponential to linear.

Cheddad *et al.* [86] presented a skin tone data concealing method which depends on the *YCbCr* color space. *YCbCr* is utilized in different methods such as object detection and compression techniques. In *YCbCr,* the correlation between RGB colors is isolated by separating the luminance (*Y*) from the chrominance blue (*Cb*) and the chrominance red (*Cr*). In this method, the human skin areas are recognized, and *Cr* of these areas are used for embedding the hidden information. Overall, the method has a limited embedding capacity because the secret message is embedded only in the *Cr* plane of the skin region.

Similarly, Sadek *et al.* [87] proposed a robust video steganography method based on the skin region of interest. The secret message is concealed into the wavelet coefficients of skin regions for each blue and red components. This method is robust against MPEG compression. However, the results of comparison demonstrated that Cheddad et al.'s method outperformed Sadek et al.'s algorithm in both imperceptibility and embedding capacity.

Alavianmehr *et al.* [88] presented a robust uncompressed video steganography by utilizing the histogram distribution constrained (HDC). In this method, the Y component of every frame is segmented into non-overlapping blocks (C) of size $m \times n$. Then, the secret message is concealed into these blocks based on the shifting process. The selected blocks are changed only when the secret message bits are "1". The modified frame S of the k^{th} block is calculated as follows:

$$\hat{S}^k(i,j) = \begin{cases} S^k(i,j) + \gamma & ; if\ \alpha \in [0,T]\ and\ mod(i,2) = mod(j,2) \\ S^k(i,j) - \gamma & ; if\ \alpha \in [-T,0]\ and\ mod(i,2)\ \neq mod(j,2) \\ S^k(i,j) & ; otherwise \end{cases} \tag{2.21}$$

$$\gamma = \frac{(G+T) \times 2}{m \times n} \tag{2.22}$$

$$\alpha^k = \sum_{i=1}^{m} \sum_{j=1}^{n} C^k(i,j) \times N(i,j) \tag{2.23}$$

$$N(i,j) = \begin{cases} 1 & ; if\ \ mod(i,2) = mod(j,2) \\ -1 & ; if\ \ mod(i,2)\ \neq mod(j,2) \end{cases} \tag{2.24}$$

Where $\gamma, \alpha,$ and N are the shift quantity, the arithmetic difference, and the computed matrix for each block, respectively. Also, T and G are two predefined thresholds used in this method. Alavianmehr et al.'s method withstands against compression attack. However, it utilizes only *Y* plane for data embedding process.

Eltahir *et al.* [89] proposed a high rate data concealing algorithm. In each frame, a 3-3-2 approach is used based upon the LSB of three color channels (RGB). A 3-3-2 method refers to 3-bits of Red, 3-bits of Green, and 2-bits of Blue in each pixel that are used to hide the secret data as shown in the Figure 2.11.

Later, Dasgupta *et al.* [90] optimized the [89] method based on the genetic algorithm in order to enhance both the security of the covert information and the visual quality of the steganogram. The reason for this improvement is to develop an objective function that is based on the weights of different parameters such as MSE and HVS. However, [89] and [90] algorithms are not robust against signal processing, noises, and video compression due to the fact that they operate in the spatial domain.

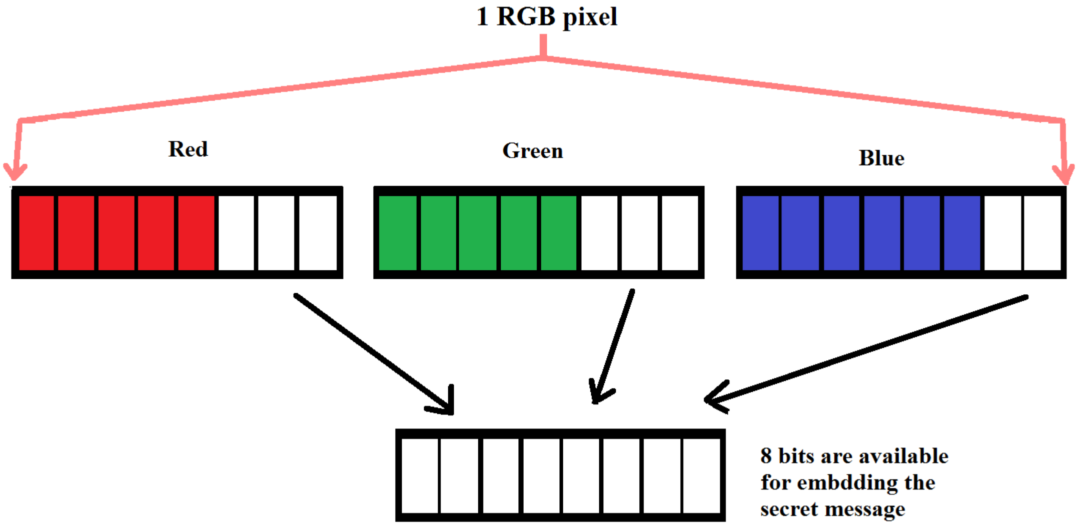


Figure 2.11 The hiding capacity in each RGB pixel [89].

Kawaguchi *et al.* [91] proposed principles and applications of BPCS steganography. In this method, the video frame is first converted into 8 bit-planes, and then each bit-plane is divided into informative (simple) and noise-like (complex) regions. The BPCS technique differs from the LSB technique in the number of bit-planes that are utilized for embedding secret message. The BPCS technique uses all bit-planes (0 – 7) for data hiding while the LSB technique only uses a bit-plane 0 for the embedding process.

Figure 2.12 clarifies how one of the video frames converts to 8 bit-planes by applying the BPCS technique. The secret information is embedded into the complex regions to achieve a high embedding payload. Moreover, modifying the noise-like areas in each bit-plane for data hiding purposes has a minimal influence to the human visual system. The complexity (α) level is measured in each region whether informative or complex, and α can be defined as follows:

$$\alpha = \frac{k}{2m(m-1)} \, , (0 \leq \alpha \leq 1) \qquad (2.25)$$

Where k equals the total length of the black-and-white border in the selected region, and $2m(m-1)$ is the highest possibility of the border length gained from the selected region. $m \times\ m$ represents the size of the selected region. Figure 2.13 illustrates the complexity degree of the BPCS regions according to Kawaguchi’s method.

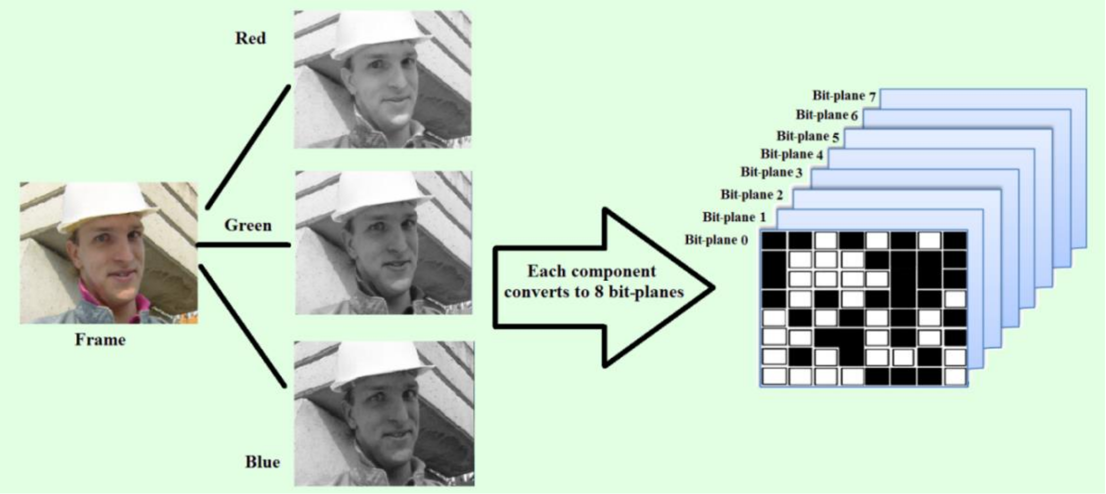


Figure 2.12 The process of converting one of the Foreman video frames into 8 bit-planes using the BPCS technique.

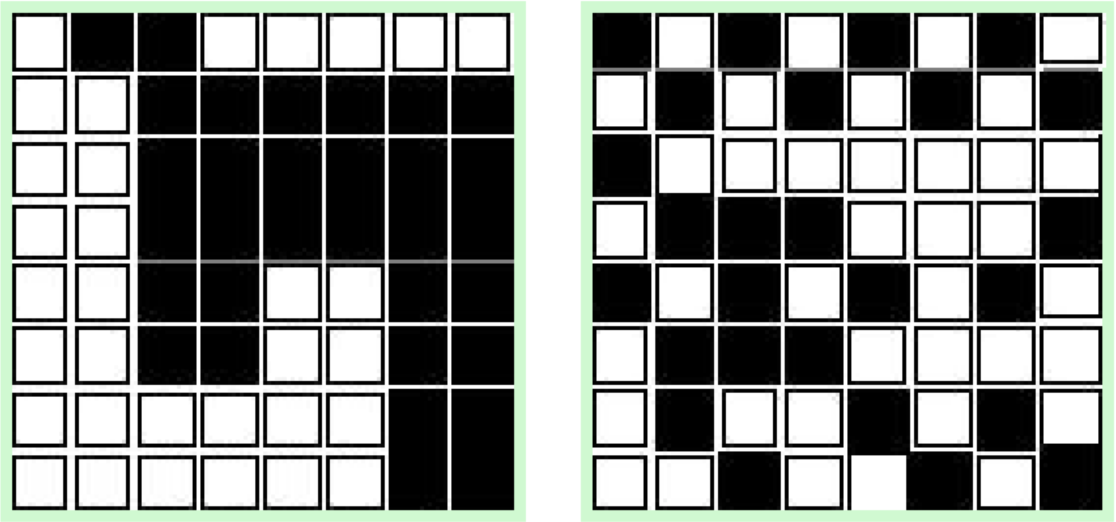


Figure 2.13 BPCS complexity degree of different regions: left informative region and right noise-like region.

Sun [92] proposed a new information hiding method based on the improved BPCS steganography. The regular BPCS method computes the complexity of the selected region based on the total length of the black-and-white border. This technique introduces a new method that identifies the noise-like regions which is useful, especially, in periodical patterns. Each canonical gray coding (CGC), run-length irregularity, and border noisiness are utilized to measure the complexity level of the selected regions. Based on the complexity degree, the secret data is embedded into the noise-like areas. In order to expand the capacity of the secret information, the informative regions are converted into the complex regions using the conjugation operation. If n is the length of pixels and $h[i]$ is the repetition of run-lengths in each black-or-white of i pixels, then run-length irregularity of the binary pixels (H_s) in Sun's algorithm can be calculated as follows:

$$H_s = -\sum_{i=1}^{n} h[i] \, log_2 \, P_i \tag{2.26}$$

$$P_i = \frac{h[i]}{\sum_{j=1}^{n} h[j]} \tag{2.27}$$

In conclusion, the steganographic methods that operate in the spatial domain are simple and obtained a high payload of secret messages. However, these techniques are not robust against signal processing, noises, and compression. Moreover, most of the above-mentioned methods do not take advantage of applying the preprocessing stages on both cover videos and secret messages. Once applied, the robustness and security of the steganographic algorithms will be enhanced.

2.3.2 Video steganography techniques in transform domain

In video steganography methods that operate in transform domain, 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.* [93] 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 the hidden data is embedded into the audio coefficients. The amount of the secret message in Patel's method is high. However, this type of wavelet is not a real mathematical wavelet operation which is not robust enough against attacks.

On the other hand, Spaulding *et al.* [94] presented the BPCS steganography method using an embedded zerotree wavelet (EZW) lossy compression. In this method, the DWT's coefficients are representing the original frame's pixels. Therefore, the BPCS steganography can be applied to DWT coefficient sub-bands which contain different features. The features of DWT sub-bands include correspondence, complexity, and resiliency against attacks. Each DWT sub-band is divided into pit-planes, and then the quantized coefficients are used for hiding the covert data. This method achieves a high embedding capacity around a quarter of the size of the compressed frame. Figure 2.14 illustrates the data embedding process of Spaulding's method.

Noda *et al.* [95] presented a video steganography technique utilizing the BPCS and wavelet compressed video. The 3D set partitioning in hierarchical trees (SPIHT) and motion-JPEG2000 are the two coding techniques that use the DWT domain. First, each bit-plane of the video frame and the secret message is segmented into 8x8 blocks. Then, the noise-like, bit-plane blocks are selected using the threshold of the noise-like complexity measurement. The two wavelet compression techniques are applied on the selected blocks by using the BPCS method, hiding the secret data into the quantized DWT coefficients. The experimental results of Noda et al.'s algorithm demonstrated that the 3D SPIHT coding method has a higher embedding payload than the Motion-JPEG2000 coding method when using BPCS steganography. However, the suggested algorithm of Noda *et al.* is not guaranteed that all type of cover videos contain enough noise-like bit-plane regions. Moreover, this method is only applied to the wavelet-based compression domain.

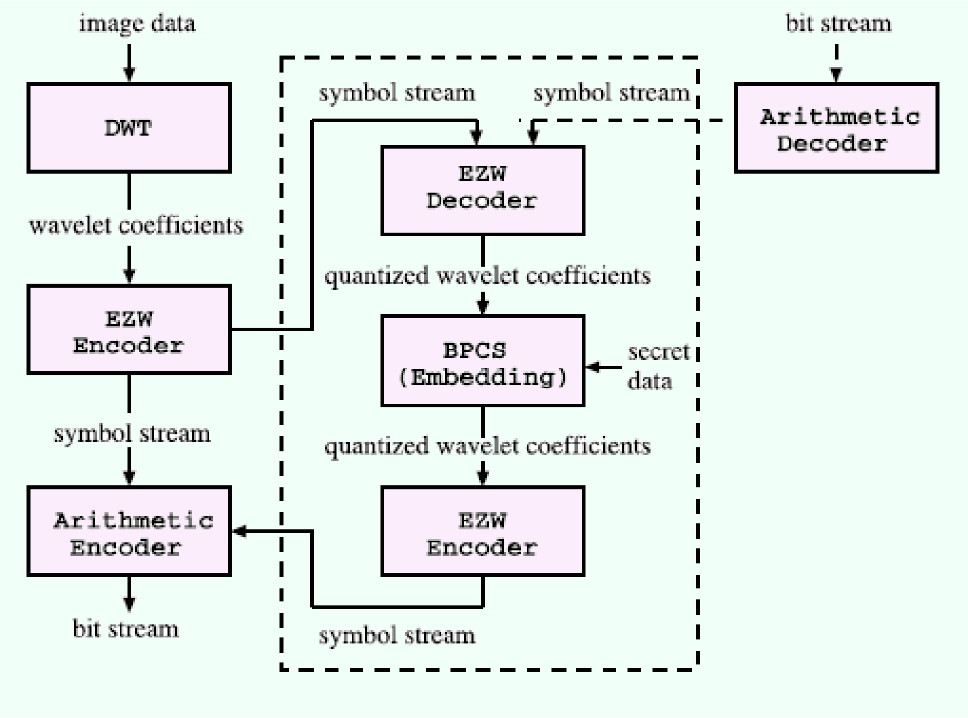


Figure 2.14 A block diagram representing the data concealing phase of the method [94].

Ordinarily, the steganographic techniques based on the transform domains improve the robustness against signal processing, noises, and compression. However, these techniques are more complex than the spatial domain methods.

2.4 Hamming Codes

In this section, the Hamming codes technique will be explained and discussed through a specific Hamming (15, 11) example. Hamming codes are defined as one of the most powerful binary linear codes. These types of codes can detect and correct errors that occur in the binary block of data during the communication between parties [96]. The codeword includes both original and extra data with a minimum amount of data redundancy, and is the result of the encoded message that uses the Hamming codes technique. In general, if p is parity bits of a positive integer number $p \geq 2$; then, the length of the codeword is $n = 2^p - 1$. The size of the message that needs to be encoded is defined as $k = 2^p - p - 1$. The number of parity bits that must be added to the message is $p = n - k$ with the rate of $r = k/n$ [51, 97].

In this dissertation, Hamming codes (15, 11) and (7, 4) are utilized to encode the secret message prior to the embedding process. For instance, the Hamming code (15, 11), which *n*=15, *k*=11, and *p*=4, can correct the identification of a single bit error. A message of size M $(m_1,\ m_2, \dots,\ m_{11})$ is encoded by adding p $(p_1,\ p_2,\ p_3,\ p_4)$ extra bits as parity to become a codeword of 15-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*) as follows:

$$\mathrm{p_1, p_2, m_1, p_3, m_2, m_3, m_4, p_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}} \tag{2.28}$$

During the encoding and decoding processes, the generator matrix *G* and parity-check matrix *H* are being used by the Hamming code (15, 11). At the transmitter channel, a message *M*, which includes of 11-bit, will be multiplied by the generator matrix *G*, and then, manipulated by having modulo of 2. The codeword *X* of 15-bit is obtained and ready to be sent.

$$X_{(1\times n)} = M_{(1\times k)} \times G_{(k\times n)} \qquad (2.29)$$

At the receiver channel, the encoded data (message + parity) which is a codeword *R* of 15-bit will be received and checked for errors. Once the received codeword *R* is multiplied by the parity-check matrix *H*, modulo of 2 will then be applied. A syndrome vector *Z* (z_1, z_2, z_3, z_4) of 4-bit is obtained. If the received message is correct, then *Z* must have all zero bits (0000); otherwise, during the transmission, one or more bits of the received message might be flipped. In that case, the error correction process must occur.

$$Z_{(1\times p)} = R_{(1\times n)} \times H^T \qquad (2.30)$$

$$\text{Where} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (2.31)$$

The reason of using parity bits in the Hamming codes is to protect the message during communication. In the Hamming code (15, 11), 7 bits of the message are used to calculate each of parity bit (total 8-bit), which is illustrated in the Figure 2.15. A group of 3 bits of the message are used to consider each of parity bit in the Hamming code (7, 4), which is shown in the Figure 2.16.

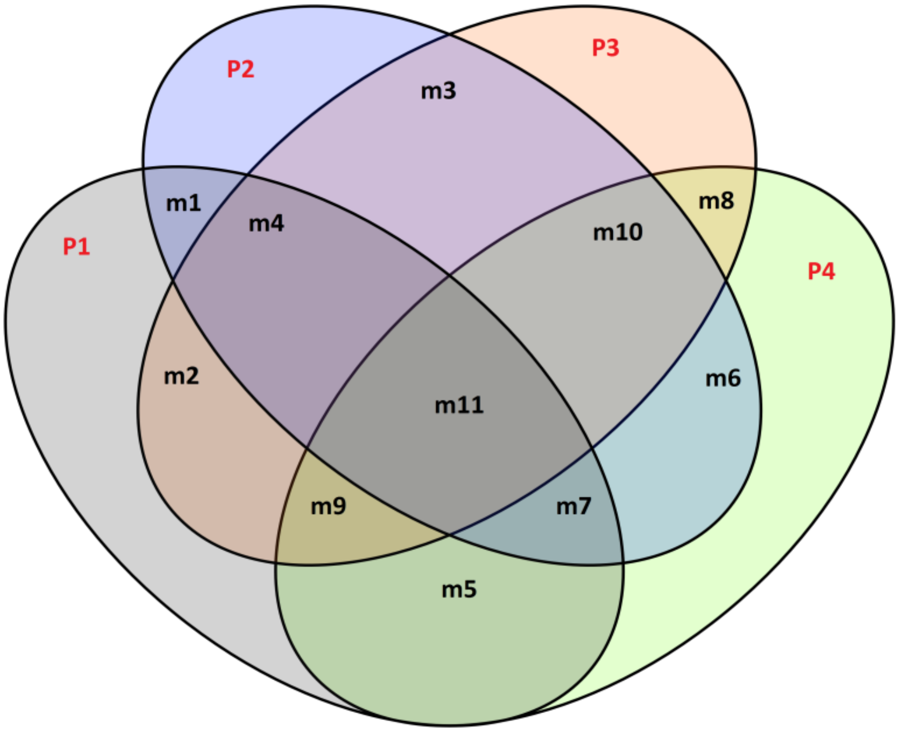


Figure 2.15 Venn diagram of the Hamming codes (15, 11).

Hamming code (15, 11) is explained through the example stated below. A message *M₁* consists of 11-bit (11111111111), *X₁* is a transmitted codeword, *R₁* is a received codeword, and *Z₁* a syndrome, then the process of finding the Hamming code (15, 11) is conducted as follows:

1) Calculate: $X_1 = M_1 \times G$, then X_1 vector equals to (777711111111111). By applying modulo of 2 to the X_1 vector, 15-bit codeword (111111111111111) is obtained. Then, this 15-bit codeword is sent to the destination side.
2) To obtain the correct message on the receiver side, the syndrome Z_1 vector must have all zero bits after taking Z_1's modulo of 2.

3) For example, if R_1=111111111111111 is received error-free, then Z_1 vector will be (0000).

4) However, assume we have a noisy channel and one of the bits has flipped during the transmission. Then, the received codeword R_1=111111111011111 contains one bit error. Thus, the syndrome Z_1 will become (0101).

5) In an error example, checking Z_1 vector with the parity-check matrix H, it showed that Z_1 (0101) is equal to the row number 10 of the parity-check matrix H, which appears that the 10th bit of R_1 has flipped.

6) Upon changing the 10th bit of R_1 from 0 to 1, R_1 will be correct (111111111111111).

7) Hence, the original message M_1 (11111111111) of 11-bit can be obtained by taking R_1 and ignoring the first 4-bit.

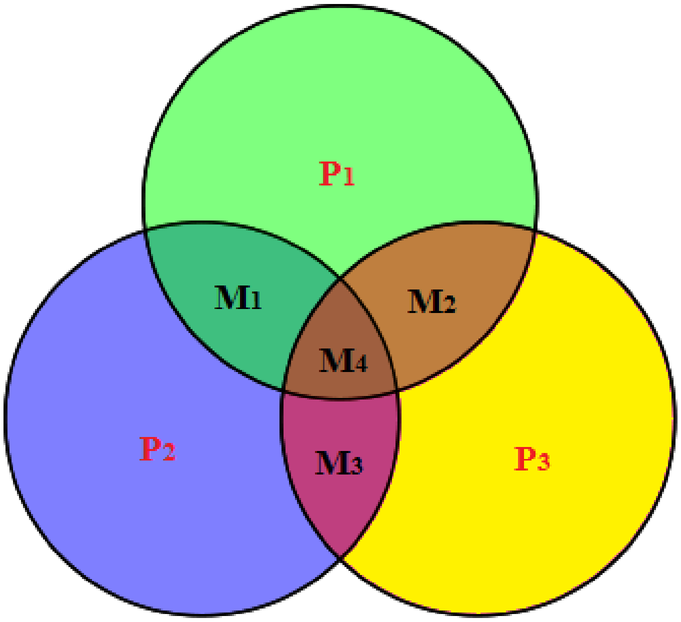


Figure 2.16 Venn diagram of the Hamming codes (7, 4).

2.5 BCH Codes

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 code 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 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 $(1 \leq i \leq 2t)$, 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 [19, 98] are described as follows:

$$H = \begin{bmatrix} 1 & \propto & \propto^2 & \propto^3 & \ldots & \propto^{n-1} \\ 1 & (\propto^3) & (\propto^3)^2 & (\propto^3)^3 & \ldots & (\propto^3)^{n-1} \\ 1 & (\propto^5) & (\propto^5)^2 & (\propto^5)^3 & \ldots & (\propto^5)^{n-1} \\ . & . & . & . & & . \\ . & . & . & . & & . \\ . & . & . & . & & . \\ 1 & (\propto^{2t-1}) & (\propto^{2t-1})^2 & (\propto^{2t-1})^3 & \ldots & (\propto^{2t-1})^{n-1} \end{bmatrix} \tag{2.32}$$

$$g(x) = lcm\{M_1(x), M_2(x), M_3(x), \ldots, M_{2t}(x)\} \tag{2.33}$$

$$g(x) = M_1(x)\, M_3(x)\, M_5(x)\ \ldots\ M_{2t-1}(x) \tag{2.34}$$

A binary BCH code (*n*, *k*, *t*) can correct errors of a maximum *t* bits for a codeword $W = \{w_0, w_1, w_2, \ldots, w_{n-1}\}$ of length *n* and a message $A = \{a_0, a_1, a_2, \ldots, a_{k-1}\}$ of length *k* [37]. An embedded codeword $C = \{c_0, c_1, c_2, \ldots, c_{n-1}\}$ is calculated as follows:

$$C = W * H^T \tag{3.35}$$

At the receiver side, the codeword $R = \{r_0, r_1, r_2, \dots, r_{n-1}\}$ is obtained. The transmitted and received codewords can both be interpreted as polynomials, where $C(X) = c_0 + c_1x^1 + \cdots + c_{n-1}x^{n-1}$, and $R(X) = r_0 + r_1x^1 + \cdots + r_{n-1}x^{n-1}$. 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:

$$E = R - C \tag{3.36}$$

$$Y = (R - C)H^T = EH^T \tag{3.37}$$

In this dissertation, we use BCH codes (15, 11, 1) and (7, 4, 1) over the $\mathrm{GF}(2^{\,4})$ and $\mathrm{GF}(2^{\,3})$, respectively, to encode the secret message prior to the embedding process. The parameters for the BCH code (15, 11, 1) are *m*=4, *k*=11, and $n = 2^{\,4} - 1 = 15$, while for the BCH code (7, 4, 1) are *m*=3, *k*=4, and $n = 2^{\,3} - 1 = 7$.

2.6 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 discrete Fourier transform (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 [99]. For an input video frame, *A*, of resolution *M x 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:

$$B_{pq} = \propto_p \propto_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \, cos \frac{\pi(2m+1)p}{2M} cos \frac{\pi(2n+1)q}{2N} \tag{2.38}$$

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \propto_p \propto_q B_{pq} \, cos \frac{\pi(2m+1)p}{2M} cos \frac{\pi(2n+1)q}{2N} \tag{2.39}$$

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 *n* 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 [100].

2.7 Discrete Wavelet Transform

DWT is a well-known method that transfers the signal from the time domain to the frequency domain [101]. The DWT separates high, middle, and low frequencies and its boundaries from one another, while other methods, such as DCT, group the various frequencies into estimated regions. The first level of the 2D-DWT image decomposition is applied to the cover video frame. It splits the frame into four sub-bands, LL (approximation), LH (horizontal), HL (vertical), and HH (diagonal), using both a low pass filter $Lo\_D(z)$ and a high pass filter $Hi\_D(z)$ for the decomposition process. LL is a low frequency sub-band, which is an approximation of the original frame reduced to a quarter of its size. The LH, HL, and HH sub-bands are middle and high frequencies that contain detailed information about any image. In the second level of frame decomposition, the 2D-DWT is applied to the LL sub-band, producing four new sub-bands [102, 103]. In this dissertation, some results are demonstrated based on the first level of decomposition. Figure 2.17 illustrates the first level of a two-dimensional DWT decomposition showing each of LL, LH, HL, and HH sub-bands. The LH, HL, and HH coefficients were used as cover data to embed the encoded secret message. In addition, Figure 2.18 shows the third level of the decomposition process.

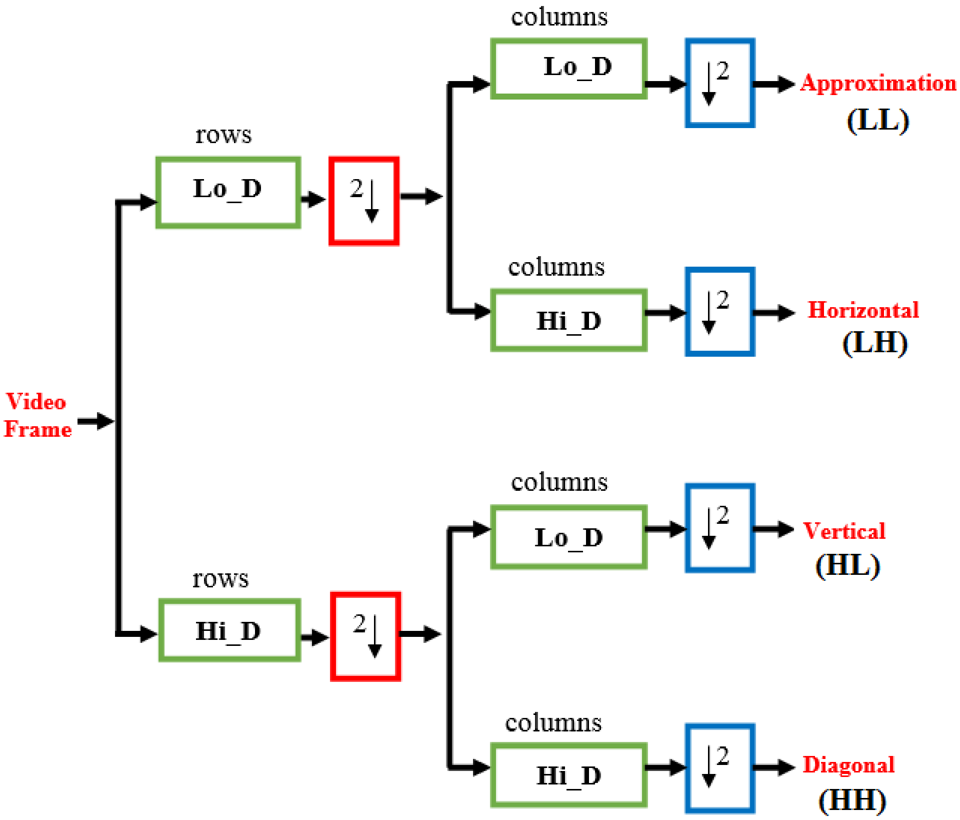


Figure 2.17 First level of a two-dimensional DWT decomposition [104].

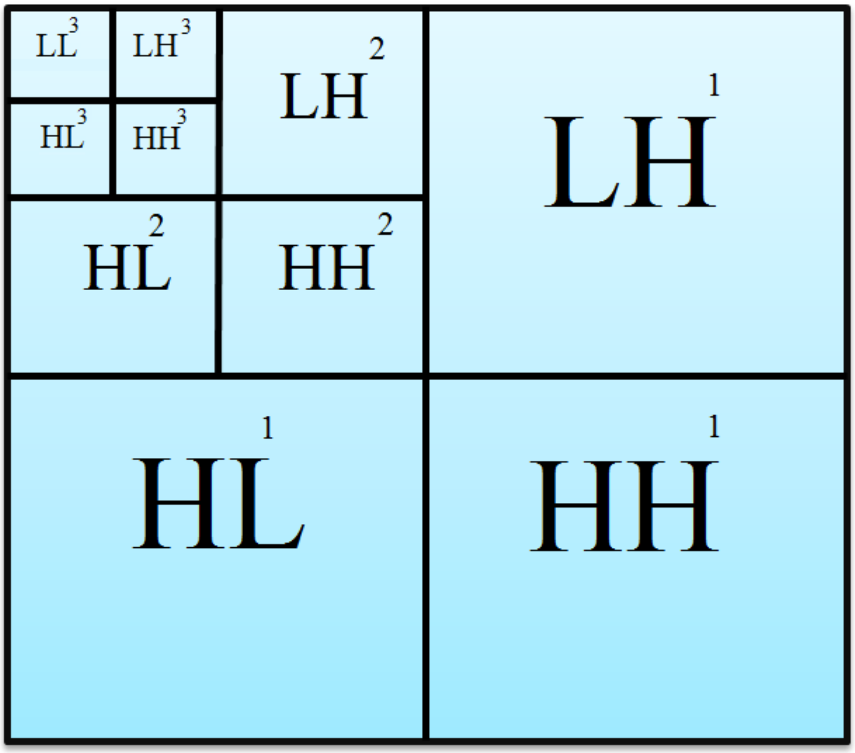


Figure 2.18 Third level of the 2D-DWT decomposition.

To achieve a complete reconstruction process, the following wavelet equations must be satisfied:

$$\{Lo\_D(z)Hi\_D(z) + \ \ Lo\_R(z)Hi\_R(z)\} = 2 \tag{2.40}$$

$$Lo\_R(z) = z^{-k}Hi\_D(-z), Hi\_R(z) = z^{k}Lo\_D(-z) \tag{2.41}$$

In the above equations, $Lo\_D(z)$ and $Hi\_D(z)$ represent the wavelet filter bank of the decomposition process. Furthermore, $Lo\_R(z)$ and $Hi\_R(z)$ signify the wavelet filter bank of the reconstruction process. The following equations are the transfer functions of the Haar wavelet transform filters:

$$Lo\_D(z) = \frac{1}{2}(1 + z^{-1}) \tag{2.42}$$

$$Hi\_D(z) = (z + 1) \tag{2.43}$$

$$Hi\_R(z) = \frac{1}{2}(z - 1) \tag{2.44}$$

$$Lo\_R(z) = (z^{-1} - 1) \tag{2.45}$$

2.8 Face Detection

To detect the facial area in the first video frame, one of the most powerful and fast algorithm in object detection has been used. It is called the Viola-Jones object detection algorithm. The reason of using the Viola-Jones detector in this dissertation is that this detector consists of three major contributions. The first contribution is the integral image, which introduces a new image representation. The integral image representation can compute the selected features (Haar-like features) much faster than other detectors [105]. The second contribution of the Viola-Jones detector is building a specific feature based classifier using an AdaBoost algorithm. The third contribution of the Viola-Jones algorithm is identifying a cascade structure, which consists of combining many complex classifiers [106]. The cascade object detector eliminates unimportant areas such as an image’s background, and focuses on the important areas of the image that contain a given object such as a facial region [107, 108]. Figure 2.19 shows the process of detecting the facial region in the video frame using the Viola-Jones face detection algorithm.

2.9 KLT Face Tracking

In this section, we will introduce the KLT tracking algorithm which is used for feature selection and tracking objects. The process of facial detection in all video frames is costly, because this process requires a high computation time [109, 110]. In addition, when a person moves fast or tilts his head the result will cause the detector to fail based on the training stage of the classifier. Therefore, it is important to have an alternative algorithm which tracks the face throughout the video frames. Once the Viola-Jones detector algorithm is applied to the first frame for purposes of detecting the facial region, the KLT tracking algorithm will be applied throughout the remaining video frames.

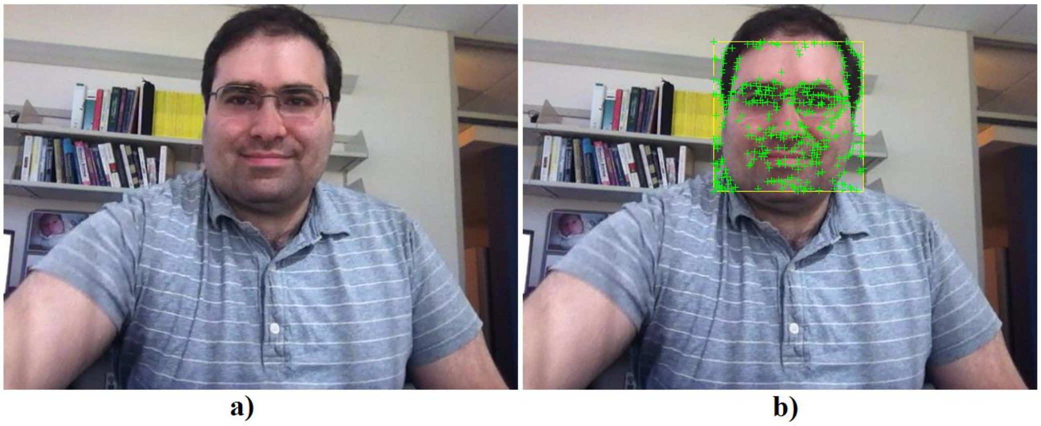


Figure 2.19 Detecting the ROI in the first video frame. a) original frame, b) detected facial region frame after applying the Viola-Jones face detection [107].

The KLT algorithm operates by finding good feature points (Harris corners) in the facial area from the first frame. These feature points are tracked throughout all the video frames [111, 112]. Each feature point will have a corresponding point between two consecutive frames. The displacement of the corresponding point pairs can be computed as motion vectors. The process of tracking the facial region depends on the movement of the centers of the features in two successive video frames. The following equations show the process of face tracking across the video frames [113, 114]:

$$\mathrm{R_t} = \mathrm{R_{t-1}} + (\mathrm{C_t} - \mathrm{C_{t-1}}) \qquad (2.46)$$

$$\mathrm{C_t} = \frac{1}{|\mathrm{f_t}|}\sum_{\mathrm{i}} \mathrm{f_t(i)} \tag{2.47}$$

$$\mathrm{C_{t-1}} = \frac{1}{|\mathrm{f_{t-1}}|}\sum_{\mathrm{i}} \mathrm{f_{t-1}(i)} \tag{2.48}$$

Where $\mathrm{R_t}$ and $\mathrm{R_{t-1}}$ represent the face areas in two adjacent video frames, respectively. $\mathrm{C_t}$ and $\mathrm{C_{t-1}}$ are the position centers of features in two consecutive frames, respectively. Also, $\mathrm{f_t}$ and $\mathrm{f_{t-1}}$ are the feature points in current and previous frames, respectively [115]. Figure 2.20 displays the process of tracking the facial regions throughout the video frames using KLT algorithm.

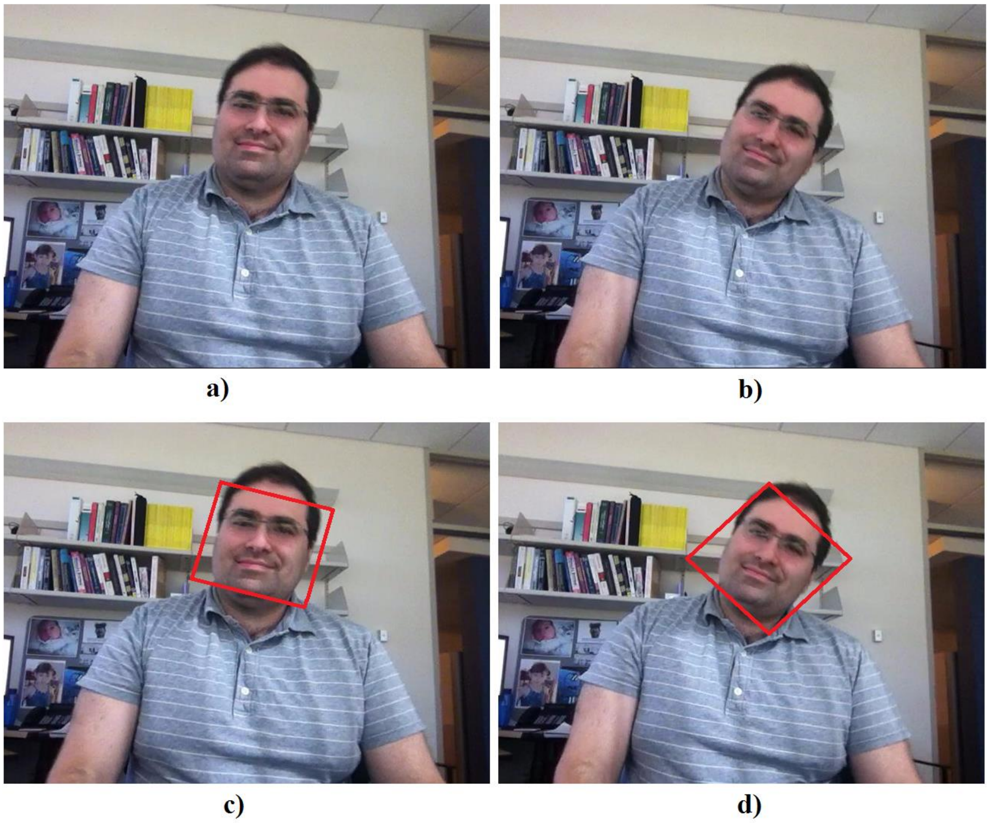


Figure 2.20 Face tracking in video frames. a) and b) two original different frames in tested video, c) and d) show facial regions that are tracked in the two frames using KLT tracking algorithm [111, 112].

2.10 Motion-based Multiple Object Tracking

Due to its various applications, computer vision is one of the fastest emerging fields in computer science. The detection and tracking of moving objects within the computer vision field has recently gained significant attention [116]. The tracking of moving objects is commonly divided into two major phases: 1) detection of moving objects in an individual video frame, and 2) association of these detected objects throughout all video frames in order to construct complete tracks [42, 117].

In the first phase, the background subtraction technique is utilized to detect the regions of interest such as moving objects. This technique is based on the Gaussian mixture model (*GMM*), which is the probability of density function that equals to a weighted sum of component Gaussian densities. The background subtraction method computes the differences between consecutive frames that generate the foreground mask. Then, the noises will be eliminated from the foreground mask by using morphological operations. As a result, the corresponding moving objects are detected from groups of connected pixels.

The second phase is called data association. It is based on the motion of the detected object. A Kalman filter is employed to speculate the motion of each trajectory. In each video frame, the location of each trajectory is predicted by the Kalman filter. Moreover, the Kalman filter is utilized to determine the probability of a specific detection that belongs to each trajectory [117]. Figure 2.21 shows four video frames that contain multiple objects and their foreground masks.

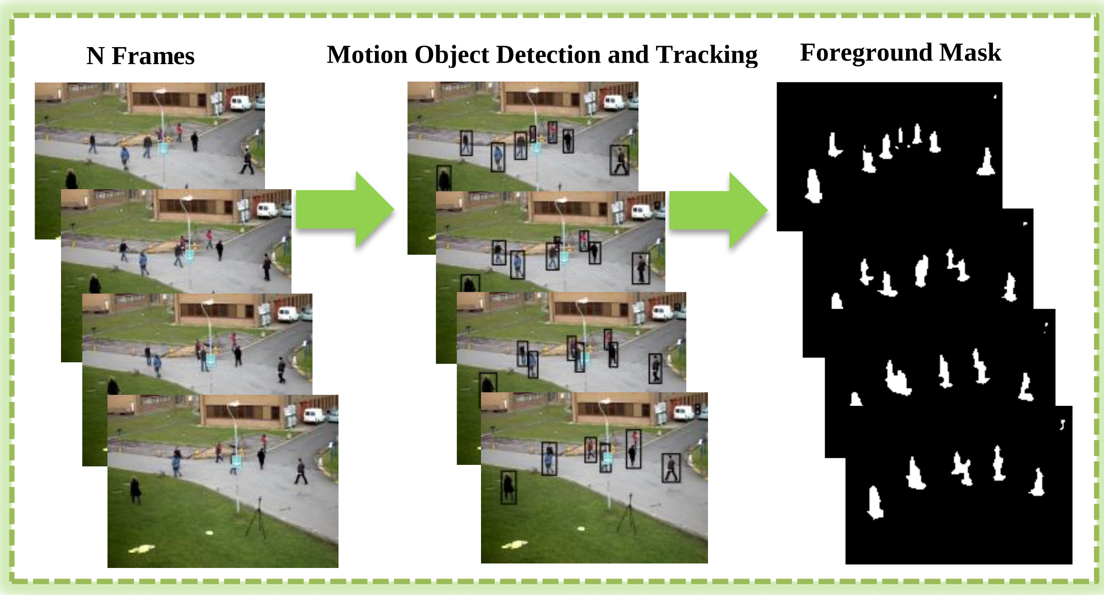


Figure 2.21 Left column: four video frames from S2L1 PETS2009 dataset [118], middle column: detecting multiple motion objects in the corresponding frames, and right column: foreground masks for the corresponding frames.

2.11 Performance Assessment Metrics

The main purpose of steganography techniques is to conceal the secret information inside the cover video data, thus the quality of the cover data will be changed ranging from a slight modification to a severe distortion. In order to evaluate whether the distortion level is acceptable or not, statistically, different metrics have been utilized [119]. PSNR is a common metric utilized to calculate the difference between the carrier and stego data. The *PSNR* can be calculated as follows [120]:

$$MSE = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{h}[C(i,j,k) - S(i,j,k)]^2}{m \times n \times h} \tag{2.49}$$

$$PSNR = 10 * Log_{10}\left(\frac{{MAX_C}^2}{MSE}\right) \quad (dB) \tag{2.50}$$

C and *S* represent the carrier and stego frames. Both *m* and *n* indicate the frame resolutions, and *h* represents the RGB colors (k=1, 2, and 3). MAX_C is the highest pixel value of the frame *C*. PSNR-HVS (*PSNRH*) and PSNR-HVS-M (*PSNRM*) objective measurements are utilized to enhance the quality of the steganograms. The *PSNRM* is an upgraded form of the *PSNRH*. Each of *PSNRH* and *PSNRM* relies on the DCT coefficients of the transform domain [121]. *PSNRH* and *PSNRM* can be calculated using Eq. 2.51 and Eq. 2.52 [122]:

$$PSNRH = 10 * Log_{10}\left(\frac{{MAX_C}^2}{MSE^{hvs}}\right) \quad (dB) \tag{2.51}$$

$$PSNRM = 10 * Log_{10}\left(\frac{{MAX_C}^2}{MSE^{hvs_m}}\right) \quad (dB) \tag{2.52}$$

$\mathrm{MSE}^{\mathrm{hvs}}$ and $\mathrm{MSE}^{\mathrm{hvs\_m}}$ utilize the factor matrix and the 8x8 DCT coefficients of the carrier and stego frame blocks [121]. On the other hand, the performance of steganographic method in terms of embedding capacity is a major factor that any method tried to increase it with the respect of the visual quality. According to [123], any steganographic method has a high embedding capacity if the hidden ratio exceeds 0.5%. The embedding ratio is calculated in the following formula:

$$Hidden\ Ratio\ = \frac{Size\ of\ embedded\ message}{Cover\ video\ size} \times 100\% \tag{2.53}$$

To further evaluate the performance of any steganographic algorithm in terms of robustness, two objective metrics including bit error rate (*BER*) and similarity (*Sim*) are used. These metrics are applied to determine whether the secret messages are retrieved from the stego videos successfully by comparing the embedded and extracted secret data. The *BER* and *Sim* are computed in the following formulas [124]:

$$BER = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j)\oplus \widehat{M}(i,j)]}{a \times b} \times 100\% \qquad (2.54)$$

$$Sim = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j) \times \widehat{M}(i,j)]}{\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} M(i,j)^2} \times \sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} \widehat{M}(i,j)^2}} \qquad (2.55)$$

Where M and $\widehat{M}$ are the embedded and extracted secret data, and a and b are the size of the secret data.

2.12 Summary

In this chapter, many past and present video steganography methods are introduced. The advantages and drawbacks of each approach in different video compression stages are highlighted. These stages include: intra frame prediction, inter frame prediction, motion vectors, transformed coefficients, and entropy coding. In addition, advantages and limitations of raw video steganography techniques that operate in both spatial and transform domains are given. Furthermore, Hamming codes, BCH codes, discrete cosine transform, discrete wavelet transform, face detection, KLT face tracking, motion-based multiple object tracking, and performance assessment metrics are deeply reviewed. Table 2.4 provides a summary of related works of all video steganography techniques that operate in both compressed and raw domains. This summary highlights each of venues for data hiding, robustness against attacks, video preprocessing, secret message preprocessing, performance measures of embedding capacity and video quality.

Table 2.4 Venues, embedding capacity, video quality, robustness, video and message preprocessing of the discussed video steganography methods

Technique	**Domain/venue for data hiding**	**Embedding capacity**	**Video quality**	**Robustness**	**Video preprocessing**	**Message preprocessing**
Liu *et al.* [60]	Compressed domain/ Intra frame prediction	Low embedding capacity (only luma DCT coefficients of scene change Intra frames are used)	PSNR ranges 36 – 42 dB	Robust against video compression	Not used	Encryption
Chang *et al.* [61]	Compressed domain/ Intra frame prediction	Average of embedding capacity ratio is 1.04%	37 dB when the bitrate is 20000 Kb/s	Robust against HEVC compression	Not used	Not used
Hu *et al.* [62] Zhu *et al.* [43]	Compressed domain/ Intra frame prediction modes	At most 1 bit per qualified Intra 4×4 luma block	Almost the same as compressed video	Robust against H.264/AVC compression	Not used	Not used
Kapotas *et al.* [65]	Compressed domain/ Inter frame prediction modes	Low embedding capacity (at most 3960 bits' capacity with the bitrate variation 85% for 20 scene change frames of luma component of resolution 176×144)	Almost the same as compressed video	Robust against H.264 compression	Scene change detector	Not used
Xu *et al.* [66]	Compressed domain/ Motion vectors	Low embedding capacity (at most 537 bits in 990 P-frame macroblocks, 4519 bits in 2640 B-frame macroblocks	I-frame 35.22 dB, P-frame 34.61 dB, and B-frame 33.31 dB	Robust against MPEG compression	Not used	Not used
Pan *et al.* [67]	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	Robust against H.264 compression	Not used	Not used
Bin *et al.* [68]	Compressed domain/ Motion vectors	Low embedding capacity (motion vector amplitude must exceed the threshold value and both components must not be equal)	Average PSNR is 38.18 dB	Robust against H.264 compression	Not used	Not used
Jue *et al.* [69]	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	Robust against H.264/AVC compression	Not used	Not used
Huang *et al.* [125]	Compressed domain/ DCT coefficients	Low embedding capacity (32 character per frame/image of resolution 352×288)	Average PSNR is 44 dB	Robust against StirMark 3.1 attack (common signal processing operations)	Not used	BCH codes
Barni *et al.* [126]	Compressed domain/ DCT coefficients	Low embedding capacity (at most 30 bits per video object of 500 Kb/s)	Almost the same as compressed video	Robust against MPEG-4 compression at lower bit rates and frame dropping	Not used	Not used
Shahid *et al.* [70]	Compressed domain/ QDCT coefficients	Average of embedding capacity ratio is 0.98% (at most 195 kbps or 20 bits per macroblock)	Average PSNR is 43.39 dB	Robust against H.264/AVC codec	Not used	Not used

Thiesse *et al.* [71-73]	Compressed domain/ QDCT coefficients	The motion vector indices are embedded into QDCT coefficients of luma and chroma	Almost the same as compressed video	Robust against H.264/AVC compression	Not used	Not used
Meuel *et al.* [74]	Compressed domain/ QDCT coefficients	An average of 25 Kbits per frame when the bitrate is 3828 Kbits/s	Average PSNR is 47.71 dB when the bitrate is 3828 Kbits/s	Robust against H.264 compression	RIO (skin detection)	Not used
Yilamz *et al.* [47]	Compressed domain/ QDCT coefficients	Intra frame: 8-13 bits of bit-length for resynchronization and 4 bits of edge-direction for damage. Inter frame: MV row hides in a corresponding row of the next frame	PSNR of Y: 36 dB, U: 39.96 dB, and V: 41.24 dB when the bitrate is 500 Kbits/s	Robust against H.263+ codec	Not used	Not used
Li *et al.* [127]	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	Robust against JPEG/JPEG2000 compression	RIO (object detection by GMM)	Not used
Stanescu *et al.* [50]	Compressed domain/ DCT coefficients	Low embedding capacity (an average of 1 bit per 8 × 8 block)	N/A	Robust against MPEG-2 codec	Not used	Not used
Li *et al.* [75]	Compressed domain/ QDCT coefficients	Low embedding capacity (at most 1 bit per 4 × 4 luma block)	Average PSNR is 36 dB of Intra frame	Robust against H.264 codec	Not used	Not used
Ma *et al.* [76]	Compressed domain/ QDCT coefficients	Average of embedding capacity ratio is 0.10% (at average 798 bits per Intra frame of resolution 176 × 144)	Average PSNR is 40.74 dB of all Intra frames	Robust against H.264/AVC codec	Not used	Not used
Liu *et al.* [59]	Compressed domain/ QDCT coefficients	Low embedding capacity (at average 758 bits per Intra frame of size 176 × 144 or 15155 bits in 20 Intra frames)	Average PSNR is 40.73 dB of all Intra frames	Robust against H.264/AVC codec	Not used	Not used
Liu *et al.* [78, 79]	Compressed domain/ QDCT coefficients	Average of embedding capacity ratio is 0.09% (at most 3541 bits are embedded in 20 Intra frames of resolution 176 × 144)	Average PSNR is 46.35 dB of all Intra	Robust against H.264/AVC codec	Not used	BCH codes
Ke *et al.* [80]	Compressed domain/ CAVLC	Embedding rate 2.44% when QP=28 and video resolution is 352 × 288 or 1 bit per 4 × 4 residual block	Average PSNR is 34.54 dB when QP=28 and video resolution is 352 × 288	Robust against H.264 compression	Not used	Not used
Liao *et al.* [81]	Compressed domain/ CAVLC	Low embedding capacity (at most 100 bits are embedded in 20th Intra frame of resolution 352 × 288)	Average PSNR is 34.37 dB when video resolution is 352 × 288	Robust against H.264/AVC codec	Not used	Not used
Lu *et al.* [82]	Compressed domain/ CAVLC	N/A	Average PSNR is 37 dB	Robust against MPEG-2 codec and geometric attacks	Not used	Not used
Mobasseri *et al.* [128]	Compressed domain/ CAVLC	Low embedding capacity (an average of 1 bit per 8 × 8 Intra block)	Almost the same as compressed video	Robust against MPEG-2 encoder	Not used	Not used
Wang *et al.* [83]	Compressed domain/ CABAC	Average of embedding capacity ratio is 0.57% (1156 bits are embedded in 50 frames of resolution 176 × 144)	average PSNR is around 37.05 dB	Robust against H.264/AVC codec	Not used	Not used
Zhang *et al.* [85]	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	Not robust against signal processing operations	Not used	BCH
Cheddad *et al.* [86]	Raw / Spatial domain	Average of embedding capacity ratio is 0.08%	Average PSNR is 61.22 dB	Not robust against signal processing	Skin region detection	Not used
Sadek *et al.* [87]	Raw / Spatial domain	Average of embedding capacity ratio is 0.23%	Average PSNR is 54.64 dB	Robust against MPEG-4 codec	Skin region detection	Not used

Khupse *et al.* [129]	Raw / Spatial domain	Low embedding capacity only frame is used (2120 bits per video)	Almost the same as original video	Not robust against signal processing	Skin region detection	Not used
Alavianmehr *et al.* [88]	Raw / Spatial domain	Average of embedding capacity ratio is 1.34% (4096 bits per video)	Average PSNR is 36.97 dB	Robust against H.264/AVC codec	Not used	Not used
Moon *et al.* [130]	Raw / Spatial domain	Average of embedding capacity ratio is 12.5%	N/A	Not robust against signal processing	Not used	Encryption
Kelash *et al.* [131]	Raw / Spatial domain	Average of embedding capacity ratio is 1.1%	Average PSNR is 48.84 dB	Not robust against signal processing	Not used	Not used
Paul *et al.* [132]	Raw / Spatial domain	Average of embedding capacity 8 bpp only in sudden scene change frames	Almost the same as original video (frames that are sudden scenes)	Not robust against signal processing	Scene change detector	Not used
Bhole *et al.* [133]	Raw / Spatial domain	Average of embedding capacity ratio is 0.2%	N/A	Not robust against signal processing	Not used	Not used
Hanafy *et al.* [134]	Raw / Spatial domain	Average of embedding capacity 0.65 bpp	Average PSNR is 51.35 dB	Not robust against signal processing	Randomization	Randomization
Lou *et al.* [135]	Raw / Spatial domain	Average of embedding capacity ratio is 12%	Average PSNR is 50.51 dB	Robust against x^2-detection and regular-singular	Not used	Not used
Tadiparthi *et al.* [136]	Raw / Spatial domain	Average of embedding capacity ratio is 2%	N/A	Not robust against signal processing	Not used	Encryption
Eltahir *et al.* [89]	Raw / Spatial domain	Average of embedding capacity 8 bpp	N/A	Not robust against signal processing	Not used	Not used
Dasgupta *et al.* [90]	Raw / Spatial domain	Average of embedding capacity 8 bpp	Average PSNR is 38.45 dB	Not robust against signal processing	Not used	Not used
Hu *et al.* [137]	Raw / Spatial domain	Average of embedding capacity 1.5 bpp	Average PSNR is 29.03 dB	Not robust against signal processing	Not used	Non-uniform Rectangular Partition
Kawaguchi et al. [91]	Raw / Spatial domain	At most the embedding capacity ratio is 41% when the threshold is 25	N/A	Not robust against signal processing	BPCS	Not used
Sun [92]	Raw / Spatial domain	At most the embedding capacity ratio is 45%	Average PSNR is 44.28 dB	Not robust against signal processing	BPCS	Not used
Patel *et al.* [93]	Raw / Transform domain	Average of embedding capacity ratio is 12.5%	Average PSNR is 31.23 dB	Not robust against signal processing	Not used	Rijndael 256 encryption
Spaulding *et al.* [94]	Raw / Transform domain	Average of embedding capacity ratio is 25%	Average PSNR is 33 dB	Robust lossy compression	BPCS	Not used
Noda *et al.* [95]	Raw / Transform domain	Average of embedding capacity ratios are 18% for 1 bit-plane and 28% for 2 bit-planes	Average PSNR of 2 bit-planes are 42.55 dB	Robust against 3D-SPIHT and Motion-JPEG2000 compression	BPCS	Not used

CHAPTER 3: PROPOSED VIDEO STEGANOGRAPHY ALGORITHMS

In this chapter, we propose six video steganography algorithms: 1) a highly secure video steganography algorithm based on ECC; 2) an increased payload video steganography algorithm in DWT domain based on ECC; 3) a novel video steganography algorithm based on KLT tracking and ECC; 4) a robust video steganography algorithm in the wavelet domain based on KLT tracking and ECC; 5) a new video steganography algorithm based on the multiple object tracking and ECC; and 6) a robust and secure video steganography algorithm in DWT-DCT domains based on multiple object tracking and ECC. The proposed algorithms provide solutions for the issues of less imperceptibility, low embedding capacity, less security, and less robustness against attacks, which exist in many steganographic algorithms, by adding some preprocessing stages and security levels to these algorithms.

3.1 A Highly Secure Video Steganography Algorithm Based on ECC

This algorithm uses an uncompressed video stream which is based on the frames as still images. First the video stream is separated into frames and each frame's color space is converted to *YCbCr*. The reason for using *YCbCr* color space is that it removes correlation between Red, Green, and Blue colors. This algorithm is divided into two phases.

3.1.1 Data embedding phase

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 shuffled by using a secret key (*Key_1*). Figure 3.1 shows the process of shuffling YUV pixels. 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 also shuffled using *Key_1*. After shuffling, the array is divided into 4-bit blocks. Then, each block is encoded by the Hamming (7, 4, 1) and BCH (7, 4, 1) encoders. 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 another secret key (*Key_2*). 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 to the original frame pixel positions to produce the stego frames. Finally, the stego video is constructed from these stego frames. The block diagram of the data embedding stage is illustrated in Figure 3.2.

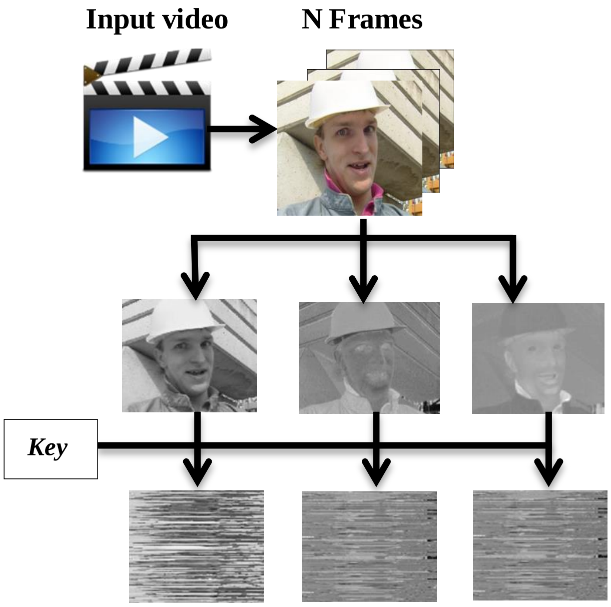


Figure 3.1 The process of shuffling pixels in each Y, U, and V frame component by secret key.

3.1.2 Data extracting phase

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 shuffled to the original positions by using secret *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 secret *Key_2*. The outcomes of 7-bit groups are decoded by the Hamming (7, 4, 1) and BCH (7, 4, 1) decoder in order to produce 4-bit blocks. These blocks are stored into a binary array and the inverse of the permutation process will be applied on these blocks to produce original bits' order. Then, the binary array of bits will be converted into the characters of the secret message. The purpose of using two secret keys and the XOR operation is to improve the security and robustness of the proposed algorithm. Secret keys are only shared between sender and receiver, and used in both data embedding and extracting processes. The block diagram of the data extracting stage is illustrated in Figure 3.3.

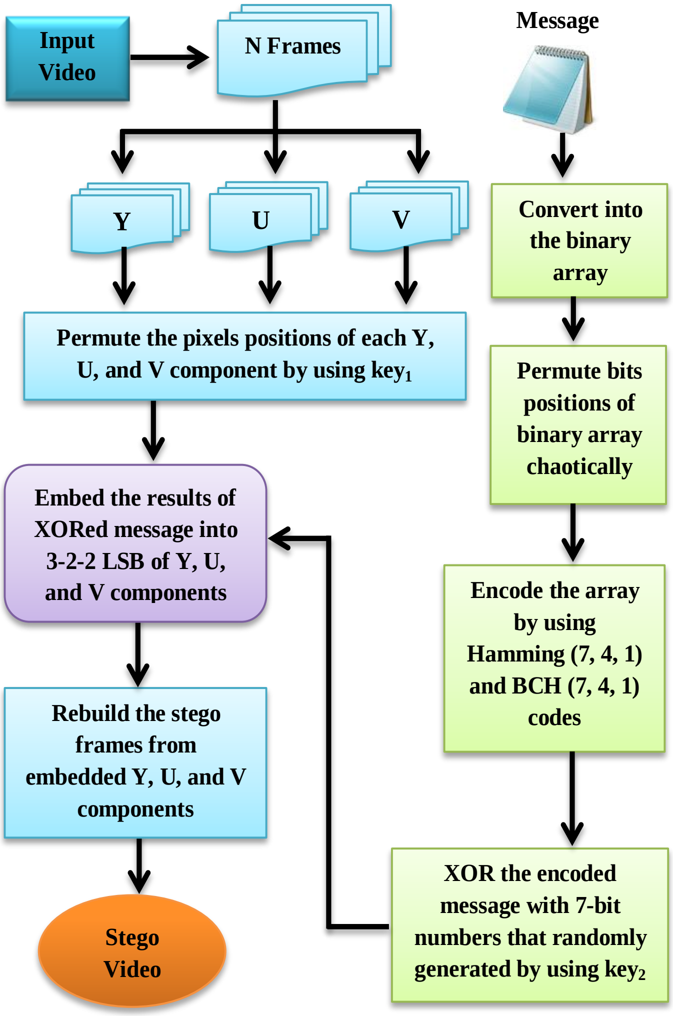

Input Video
N Frames
Message
Y
U
V
Convert into the binary array
Permute the pixels positions of each Y, U, and V component by using key$_1$
Permute bits positions of binary array chaotically
Embed the results of XORed message into 3-2-2 LSB of Y, U, and V components
Encode the array by using Hamming (7, 4, 1) and BCH (7, 4, 1) codes
Rebuild the stego frames from embedded Y, U, and V components
XOR the encoded message with 7-bit numbers that randomly generated by using key$_2$
Stego Video

Figure 3.2 Block diagram of the data embedding phase.

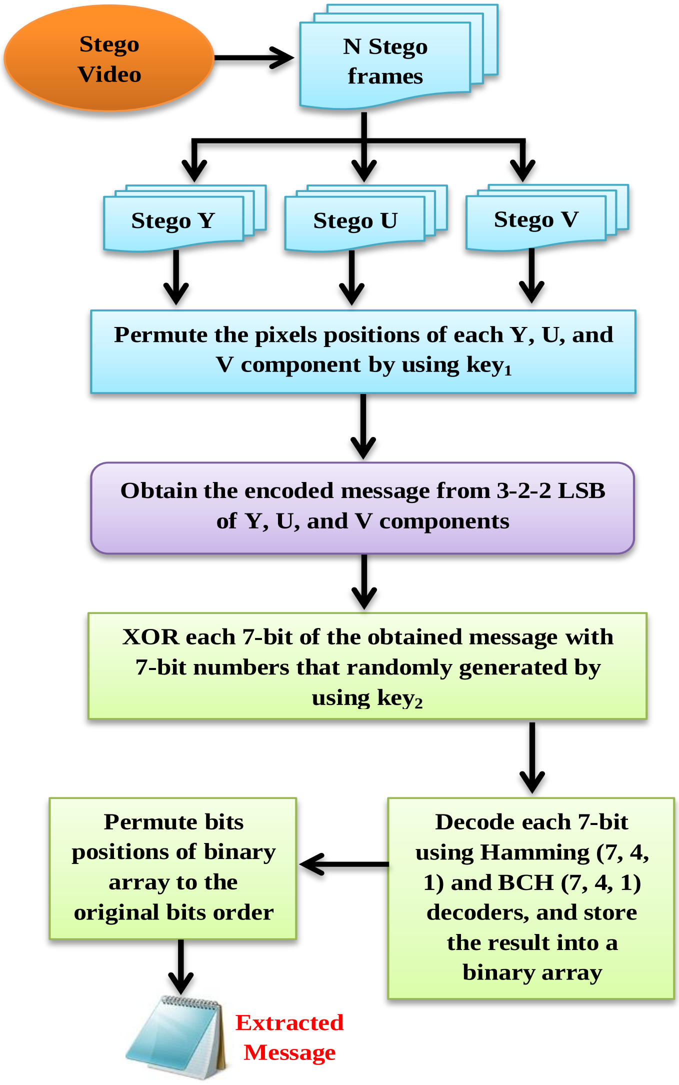

Stego Video
N Stego frames
Stego Y
Stego U
Stego V
Permute the pixels positions of each Y, U, and V component by using key₁
Obtain the encoded message from 3-2-2 LSB of Y, U, and V components
XOR each 7-bit of the obtained message with 7-bit numbers that randomly generated by using key₂
Decode each 7-bit using Hamming (7, 4, 1) and BCH (7, 4, 1) decoders, and store the result into a binary array
Permute bits positions of binary array to the original bits order
Extracted Message

Figure 3.3 Block diagram of the data extracting phase.

3.2 An Increased Payload Video Steganography Algorithm in DWT Domain Based on ECC

In this algorithm, we use uncompressed video sequences based on the frames as still images. This method is illustrated by using two phases: 1) data embedding phase and

2) data extracting phase. Block diagrams of data embedding and extracting stages are illustrated in Figure 3.4. and 3.5, respectively.

3.2.1 Data Embedding Phase

The process of embedding the secret message consists of two stages: first encoding the message using the BCH code (Step 1 to 5) and then embedding the encoded message into the cover videos (Step 6 to 13). This process can be completed by the following steps:

Step1: Input the secret message (text file).

Step2: Change bits' positions of the whole secret message by secret *Key₁*.

Step3: Convert the whole secret message to a one dimensional array (1-D).

Step4: Encode the message by using the BCH (15, 11) encoder.

Step5: XOR the encoded data, which consists of 15 bits (11 bits of message + 4 bits of parity), with the 15 bits of random value using secret *Key₂*.

Step6: Input the cover video stream.

Step7: Convert the video sequence into a number of frames.

Step8: Split each frame into the YUV color space.

Step9: Apply the 2D-DWT individually on each Y, U, and V frame component.

Step10: Embed the message into the middle and high frequency coefficients (LH, HL, and HH) of each of the Y, U, and V components as follows:

$$\hat{Y}_{ij} = \begin{cases} E[floor(Y_{ij\_bit_{1,2,3}}), S] & if\ (Y_{ij} \geq 0) \\ E[floor(|Y_{ij\_bit_{1,2,3}}|), S] & if\ (Y_{ij} < 0) \end{cases} \tag{3.1}$$

$$\hat{U}_{ij} = \begin{cases} E[floor(U_{ij\_bit_{1,2,3}}), S] & if\ (U_{ij} \geq 0) \\ E[floor(|U_{ij\_bit_{1,2,3}}|), S] & if\ (U_{ij} < 0) \end{cases} \tag{3.2}$$

$$\hat{V}_{ij} = \begin{cases} E[floor(V_{ij\_bit_{1,2,3}}), S] & if\ (V_{ij} \geq 0) \\ E[floor(|V_{ij\_bit_{1,2,3}}|), S] & if\ (V_{ij} < 0) \end{cases} \tag{3.3}$$

Where $Y_{ij}, U_{ij}, and\ V_{ij}$ are the Y, U, and V coefficients, and *S* is the encoded secret message, $S = \{000, \dots, 111\}$. *E* is the embedding process.

Step11: Apply the inverse of 2D-DWT on the frame components.

Step12: Rebuild the stego frames from the YUV stego components.

Step13: Output the stego videos, which are reconstructed from all embedded frames.

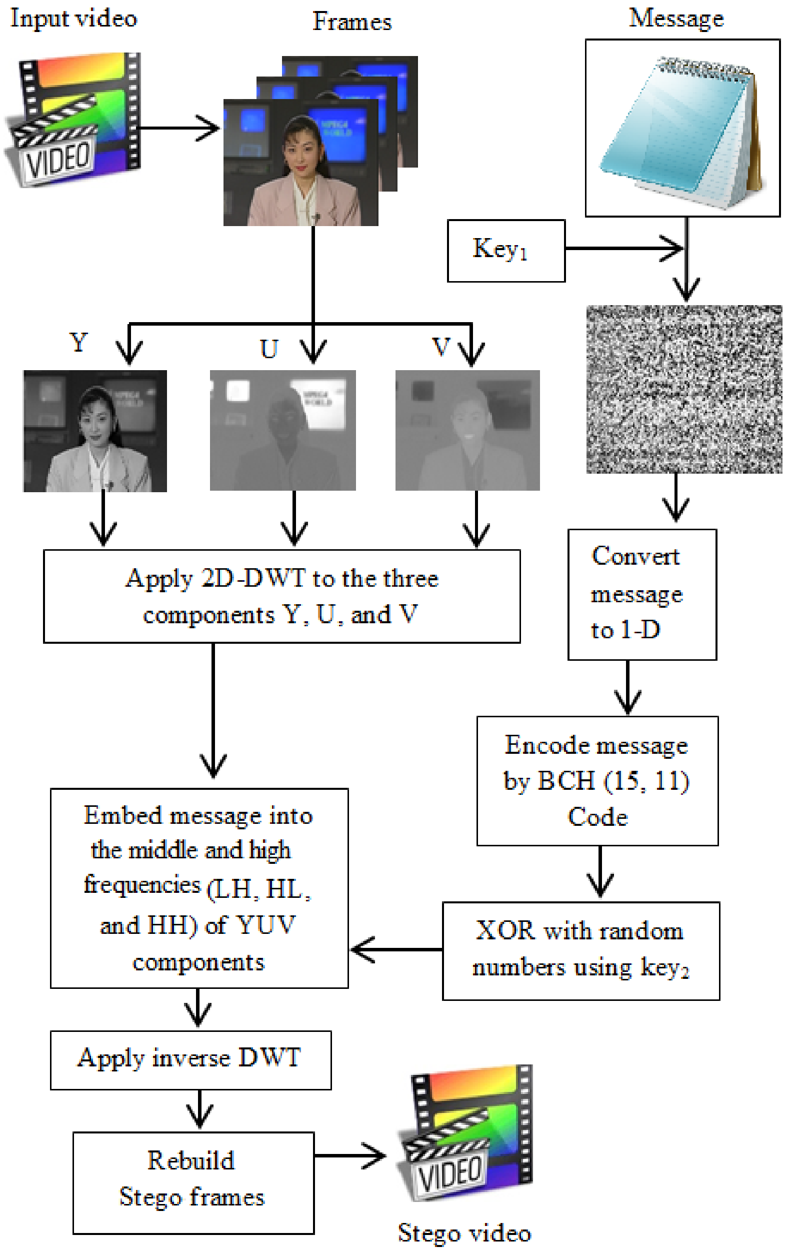

Input video
Frames
Message
VIDEO
Key1
Y
U
V
Apply 2D-DWT to the three components Y, U, and V
Convert message to 1-D
Encode message by BCH (15, 11) Code
Embed message into the middle and high frequencies (LH, HL, and HH) of YUV components
XOR with random numbers using key2
Apply inverse DWT
Rebuild Stego frames
VIDEO
Stego video

Figure 3.4 Block diagram of the data embedding process.

In this steganographic algorithm, two secret keys were used; each secret key was predefined by the sender and receiver in both embedding and the extracting processes. The first secret key (*Key*$_1$) is used to randomly change the position of all bits in the secret message to make the message unreadable and chaotic before encoding by the BCH. The second secret key (*Key2*) is used after the encoding process; the encoded message is divided into 15-bit groups, and each group is XORed with the 15-bit numbers (the 15-bit numbers were randomly generated). One of the strengths of the proposed algorithm is the usage of the two secret keys, which improve the security and robustness of our algorithm.

3.2.2 Data Extracting Phase

This section introduces the process of retrieving the encoded message from the stego videos first, and then decoding the encoded message using the BCH decoder. This process can be completed by the following steps:

Step1: Input the stego videos.

Step2: Convert the stego video sequences into a number of frames.

Step3: Divide each frame into the YUV color space.

Step4: Apply the 2D-DWT separately on each Y, U, and V component.

Step5: Extract the encoded message from the middle and high frequency coefficients (LH, HL, and HH) of each Y, U, and V component.

$$\hat{S}_{1,2,3} = \begin{cases} EX[floor(\hat{Y}_{ij\_bit_{1,2,3}})] & if\ (\hat{Y}_{ij} \geq 0) \\ EX[floor(|\hat{Y}_{ij_{bit_{1,2,3}}}|)] & if\ (\hat{Y}_{ij} < 0) \end{cases} \tag{3.4}$$

$$\hat{S}_{1,2,3} = \begin{cases} EX[floor(\widehat{U}_{ij\_bit_{1,2,3}})] & if\ (\widehat{U}_{ij} \geq 0) \\ EX[floor(|\widehat{U}_{ij_{bit_{1,2,3}}}|)] & if\ (\widehat{U}_{ij} < 0) \end{cases} \tag{3.5}$$

$$\hat{S}_{1,2,3} = \begin{cases} EX[floor(\hat{V}_{ij\_bit_{1,2,3}})] & if\ (\hat{V}_{ij} \geq 0) \\ EX[floor(|\hat{V}_{ij_{bit_{1,2,3}}}|)] & if\ (\hat{V}_{ij} < 0) \end{cases} \tag{3.6}$$

Where $\hat{Y}_{ij}, \widehat{U}_{ij}, and\ \hat{V}_{ij}$ are the stego YUV coefficients, and $\hat{S}$ is the retrieved secret message. *EX* is the extracting process.

Step6: Segment the entire encoded message into 15-bits groups.

Step7: XOR each group with the random 15-bit numbers that were generated by the same secret key (*Key2*) at the sender side.

Step8: Decode the message by the BCH (15, 11) decoder.

Step9: Produce an array from the resulted groups.

Step10: Reposition the message again to the original bit order using secret Key_1

Step11: Output the secret message as a text file.

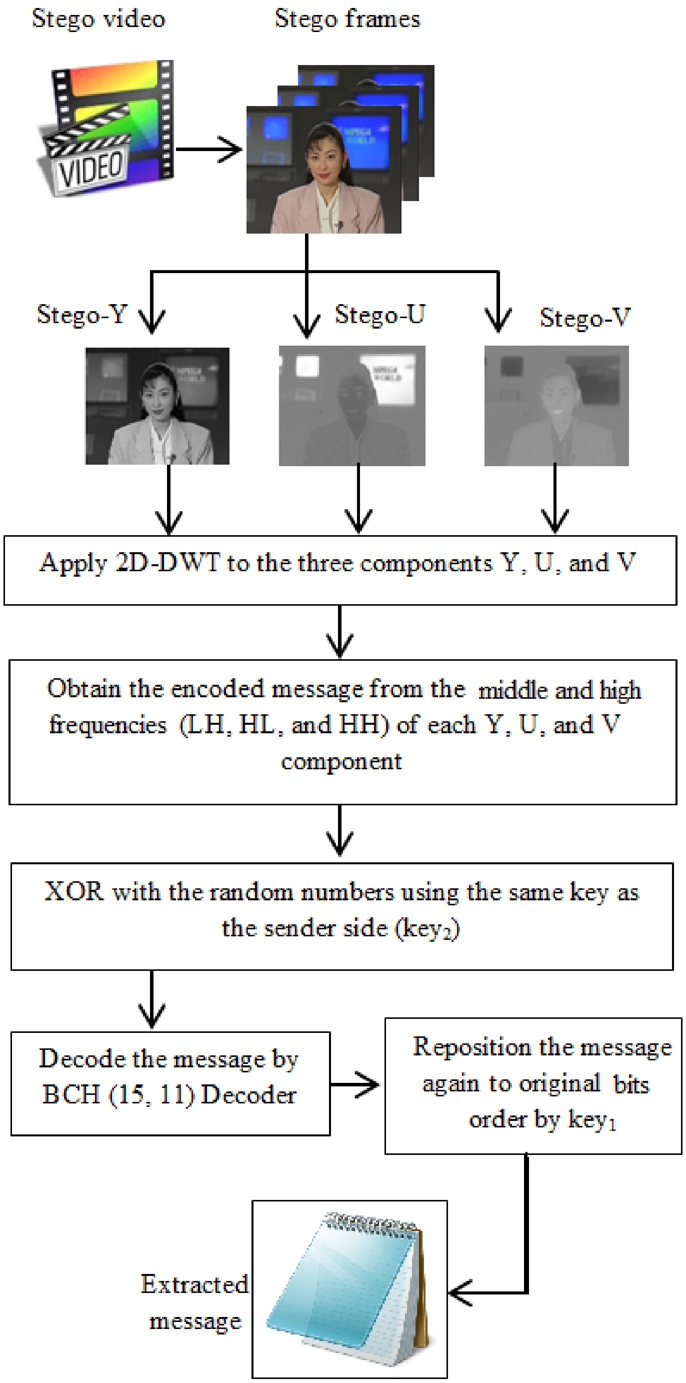

Stego video
Stego frames
VIDEO
Stego-Y
Stego-U
Stego-V
Apply 2D-DWT to the three components Y, U, and V
Obtain the encoded message from the middle and high frequencies (LH, HL, and HH) of each Y, U, and V component
XOR with the random numbers using the same key as the sender side (key₂)
Decode the message by BCH (15, 11) Decoder
Reposition the message again to original bits order by key₁
Extracted message

Figure 3.5 Block diagram of the data extracting process.

3.3 A Novel Video Steganography Algorithm Based on KLT Tracking and ECC

In this section, we propose a novel video steganographic method based on the KLT tracking algorithm using Hamming codes (15, 11). Algorithms 1 and 2 clarify the major steps of our embedding and extracting algorithms, respectively. Our proposed method is divided into four stages:

3.3.1 Secret message preprocessing stage

The secret message is a digital data type which is based on the ASCII of characters. Each character has a unique 8-bit code that is unchangeable. In this work, a sizable text file is used as a secret message, and it is preprocessed before the embedding phase. First, the whole characters in the text file are converted into ASCII codes in order to generate an array of binary bits. Then, for security purposes, the binary array is encrypted by using a secret key (*Key_1*) that represents the size of the secret message. A shuffle encryption has been used to change the index of entire bits of the binary array. This process will encode the message, and protect it from hackers. Since the binary linear block of Hamming code (15, 11) is used, the encrypted array is divided into 11-bit blocks. Then, every block is encoded by the Hamming code (15, 11) that will generate 15-bit blocks. Consequently, this encoder extends the size of the message by adding four parity bits into each block. Another secret key (*Key_2*) is utilized as a seed to generate randomized 15-bit numbers, and each number is XORed with the 15-bit encoded block.

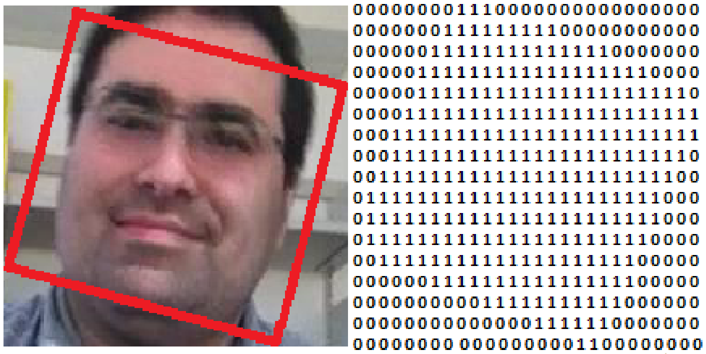
By using two secret keys, the Hamming codes, and XOR operation the security of our algorithm will be enhanced.

3.3.2 Face detection and tracking stage

At the beginning, the process of extracting the facial regions in the video frames must be identified because these regions are used as cover data for embedding the secret message. To detect the facial region in the first video frame, the Viola-Jones detector algorithm has been applied. Then, throughout the remaining video frames the face will be tracked by using the KLT tracking algorithm. Since our algorithm is based on the face detection and KLT tracking algorithms, the secret message will be hidden into only video frames that contain facial features. Otherwise non-facial frames will be discarded without embedding. Hiding secret messages inside facial regions make it more challenging for attackers during data extraction as these regions changes in every frame of the underlying video. The process of face detection and KLT tracking have been previously explained in Sections **2.8** and **2.9**, respectively.

3.3.3 Data embedding stage

After the facial region is detected and tracked, it will be extracted from the video frames. In each frame, the cover data of this algorithm is the facial region of interest. The ROI changes in every frame based on the size of the facial bounding box. The ROI extracting process is not always accurate, and may include pixels outside the facial bounding box. For example, when the face is tilted, a binary mask is applied. This mask sets the pixels that are located inside the polygon bounding box to “1” and sets the pixels outside the bounding polygon to “0”. The binary mask applies to all video frames. The main advantage of the binary mask is to determine the number of pixels and their positions that will involve in the embedding process of every frame. Figure 3.6 illustrates the role of the binary mask that identifies the facial region in each frame.



```
000000001110000000000000000
000000011111111100000000000
000000111111111111110000000
000001111111111111111110000
000001111111111111111111110
000011111111111111111111111
000111111111111111111111111
000111111111111111111111110
001111111111111111111111100
011111111111111111111111000
011111111111111111111111000
011111111111111111111110000
001111111111111111111100000
000000111111111111111100000
000000000011111111111000000
000000000000001111110000000
00000000 0000000001100000000
```

Figure 3.6 Binary mask of one video frame.

Every four edges of the bounding box in each frame are embedded into the specific non-facial area known by both sender and receiver. Every box needs 80 bits per frame (40 bits for X-axis and 40 bits for Y-axis). Moreover, in order to transmit keys to the receiver party, both keys will be embedded into the non-facial region of the first video frame. Then, the next stage in the process of embedding the hidden data begins. This stage hides the secret message blocks by placing them into the LSB of each red, green, and blue color components of the facial region in all video frames. In our algorithm, one LSB, two LSBs, three LSBs, and four LSBs of the three color components from each facial pixel are utilized in order to embed 3, 6, 9, and 12 bits of the hidden message, respectively. Upon completion, the stego frames will be reconstructed into a stego video format that sends via the communication channel to the receiver party. Figure 3.7 illustrates the block diagram of data embedding stage.

Algorithm 1: Data embedding of the proposed algorithm

```
Input: V //Video, M //Secret message in characters, Key1, Key2; //Stego keys
Output: SV; //Stego video
Initialize km, pm,p;
B ← M; //Convert the alphabetic secret message to the binary array
// Stego keys
Key1 ← Length(B)/11; //Length of the secret message
Key2 ← rand (2^15,Key1,1)'; //Randomization of the seed Key1
EB ← E(B, [Key1]); //Encrypt the binary array by Key1

for1 i = 1: (Key1*15) do //Encode each 11 bits of encoded message by Hamming (15,11)
      g(1:11) ← get(EB(km:km+11));
      E_EB ← encode(g,15,11);
      temp(1:15) ← get(Key2(i));
      Ecdmsg(pm:pm+15) ← xor(E_EB,temp);
      pm+15; km+11;
end1
Read (V); //Read input video, {Vf1, Vf2,…, Vfn} are video frames (n frames)
FBox1 ← Face_detector (Vf1); //Calling the Viola-Jones face detector for first frame Vf1
Non_Face(Vf1) ← Key1, Key2; //Embed keys (Key1 and Key$_2$) into the non-facial regions of the first frame Vf1

for2 t = 1:n do //For each video frame, track the face and its corner box points
FBoxt ← Face_KLT(Vft); //Calling KLT face tracking algorithm
Non_Face(Vft )← Edges (FBoxt(xz,yz)); //Embed edge points of each facial box into the non-facial area of its frame (z=1,2,3, and 4)
B_mat = mask(Edges (FBoxt(xz,yz)),Vfx,Vfy); //Identify the binary mask of the facial regions of size (Vfx,Vfy)
  //Embed the encoded message into the 1 LSB, 2 LSBs, 3 LSBs, or 4 LSBs of each frame's facial region (FBox1,2,…n)
    for3 i = 1:Vfx do
        for4 j= 1:Vfy do
           if5 B_mat(i,j) == 1
             LSB_R1,2,3, or 4(FBoxt(i,j)) ← Ecdmsg(p+1,4,7, or 10);
             LSB_G1,2,3, or 4(FBoxt(i,j)) ← Ecdmsg(p+2,5,8, or 11);
             LSB_B1,2,3, or 4(FBoxt(i,j)) ← Ecdmsg(p+3,6,9, or 12);
                         p+3,6,9, or 12;
     end5 end4 end3
end2

get SV //Obtain the stego video
```

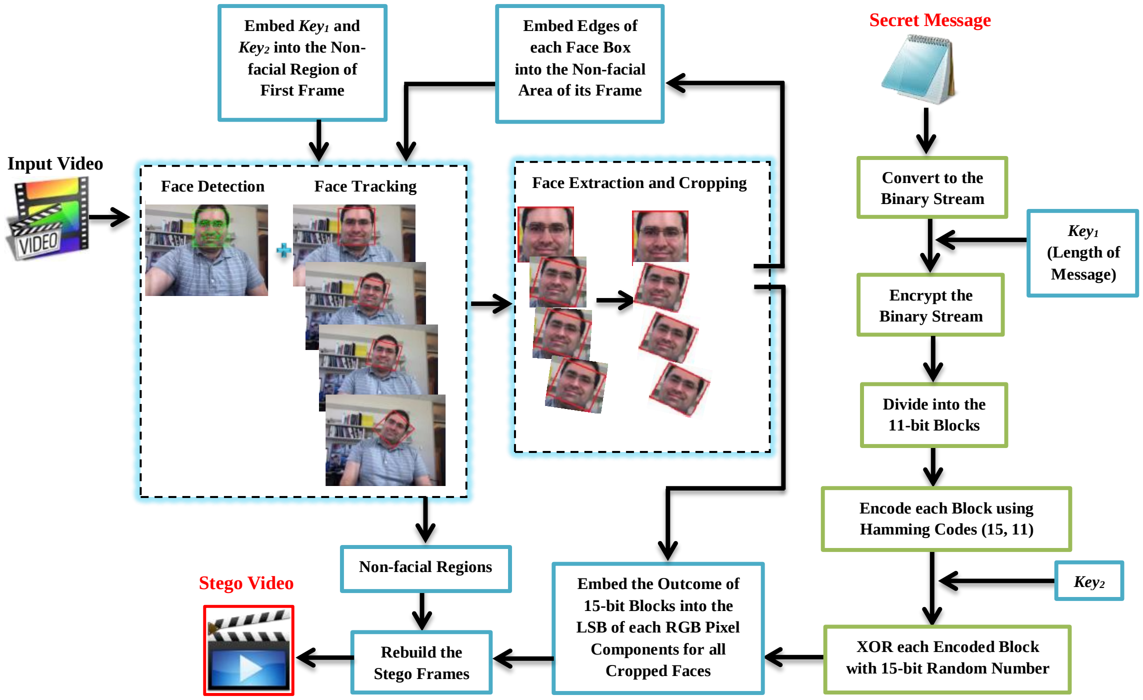

Embed Key_1 and Key_2 into the Non-facial Region of First Frame
Embed Edges of each Face Box into the Non-facial Area of its Frame
Secret Message
Input Video
Face Detection
Face Tracking
Face Extraction and Cropping
Convert to the Binary Stream
Key_1 (Length of Message)
Encrypt the Binary Stream
Divide into the 11-bit Blocks
Encode each Block using Hamming Codes (15, 11)
Key_2
Stego Video
Non-facial Regions
Rebuild the Stego Frames
Embed the Outcome of 15-bit Blocks into the LSB of each RGB Pixel Components for all Cropped Faces
XOR each Encoded Block with 15-bit Random Number

Figure 3.7 Block diagram for the data embedding stage.

3.3.4 Data extraction stage

At the receiver side, the stego video will be divided into frames, and both two secret keys are extracted from the non-facial area of the first frame. Moreover, in each video frame, the four corner points of the facial box will be extracted from the non-facial regions. The binary mask of each frame is generated from these points, and the exact facial region will be identified. Then, the process of extracting the hidden message is conducted by taking out the 3, 6, 9, and 12 bits from first LSB, second LSBs, third LSBs, and fourth LSBs, respectively, in each facial pixel of all video frames. The extracted bits from all video frames are stored in a binary array. The binary array will be segmented into the 15-bit blocks. Each block will be XORed with the 15-bit number that was randomly generated by secret Key_2. The results of 15-bit blocks are decoded using the Hamming (15, 11) decoder to produce 11-bit blocks. Since the sender has encrypted the secret message, the obtained array must be decrypted using secret Key_1. The final array divides into 8-bit codes (ASCII) for generating the right characters of the original message. Figure 3.8 illustrates the block diagram of the data extracting stage.

Algorithm 2: Data extracting of the proposed algorithm

```
Input: SV; //Stego video
Output: M; //Secret message in characters
Initialize km, pm,p;
{Sf1, Sf2,…, Sfn} ← Read (SV);//Convert the stego video into frames (n stego frames)
Extract[Key1, Key2] from (Sf1); //Extract keys (Key1 and Key2) from the non-facial region of the first stego frame Sf1
for1 t = 1:n do
Extract[Edges (FBoxt(xz,yz))] from (Non_Face(Sft )); //Extract edge points of each facial box FBox from non-facial areas of its stego frame (z=1,2,3, and 4)
Extract[FBoxt] from (Sft); //Identify the region of interest (facial region) by edges
B_mat = mask(Edges (FBoxt(xz,yz)),Sfx,Sfy); //Identify the binary mask of facial regions of size (Sfx,Sfy)
//Extract the encoded message from the 1 LSB, 2 LSBs, 3 LSBs, or 4 LSBs of each frame's facial region (FBox1,2,…n)
  for2 i = 1:Sfx do
       for3 j= 1:Sfy do
            if4 B_mat(i,j) == 1
               Ecdmsg(p+1,4,7, or 10) ←  LSB_R1,2,3, or 4(FBoxt(i,j));
               Ecdmsg(p+2,5,8, or 11) ←  LSB_G1,2,3, or 4 (FBoxt(i,j));
               Ecdmsg(p+3,6,9, or 12) ←  LSB_B1,2,3, or 4 (FBoxt(i,j));
                    p+3,6,9, or 12;
   end4  end3  end2  end1
for5 i = 1: (Key1*15) do //Decode each 15 bits of extracted data by Hamming (15,11)
      Sg(1:15) ← get(Ecdmsg (km:km+15));
      temp(1:15) ← get(Key2(i));
      E_EB   ← xor(Sg,temp)
      EB ←  decode(E_EB ,15,11);
      pm+11; km+15;
end5
B ← D(EB, [Key1]); //Decrypt the binary array by Key1
M ← B; //Convert the binary array to the alphabetic characters
get M; //Recover the secret message
```

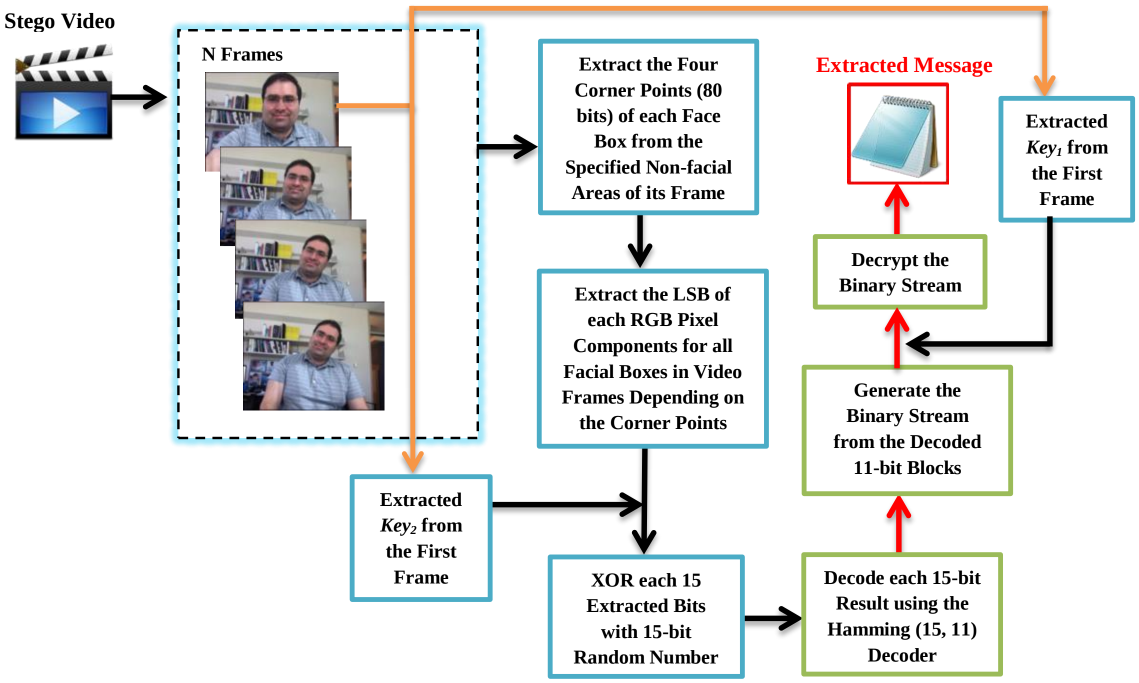

Stego Video
N Frames
Extract the Four Corner Points (80 bits) of each Face Box from the Specified Non-facial Areas of its Frame
Extracted Message
Extracted Key_1 from the First Frame
Decrypt the Binary Stream
Extract the LSB of each RGB Pixel Components for all Facial Boxes in Video Frames Depending on the Corner Points
Generate the Binary Stream from the Decoded 11-bit Blocks
Extracted Key_2 from the First Frame
XOR each 15 Extracted Bits with 15-bit Random Number
Decode each 15-bit Result using the Hamming (15, 11) Decoder

Figure 3.8 Block diagram for the data extracting stage.

3.4 A Robust Video Steganography Algorithm in the Wavelet Domain Based on KLT Tracking and ECC

In this section, we introduce a video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes (15, 11, 1). Our proposed steganography algorithm has four phases:

3.4.1 Secret Message Preprocessing Phase

In this work, a large size text file has been used as a secret message, and it is preprocessed before the embedding phase. Here, the whole characters in the text file are converted into ASCII codes in order to generate an array of binary bits. Then, for security purposes, the binary array is encrypted by using a secret key (Key_1) that represents the size of the secret message. This process will encode the message and protect it from attackers. Since the binary linear block of BCH codes (15, 11, 1) are used, the encrypted array is divided into 11-bit blocks. Then, every block is encoded by the BCH codes (15, 11, 1) producing 15-bit blocks. The size of the message is extended by adding four parity bits into each block. Another secret key (Key_2) is utilized to generate randomized 15-bit numbers, and each number is XORed with the 15-bit encoded block. The security of our algorithm will be improved by using two keys, BCH codes, and XOR operation.

3.4.2 Face Detection and Tracking Phase

First, the process of extracting the facial regions in the video frames must be identified since facial regions are used as cover data. The Viola-Jones detector algorithm is applied in order to detect the facial region in the first video frame. Then, the KLT tracking algorithm is used throughout the remaining video frames in order to track the facial regions. Sections **2.8** and **2.9** have previously discussed face detection and tracking using KLT algorithm.

3.4.3 Data Embedding Phase

In each video frame, the cover data of our algorithm is the facial region of interest. The facial regions will be extracted from the video frames after they are detected and tracked. The region of interest changes in every frame based on the size of the facial bounding box. In each video frame, the 2D-DWT is applied on each R, G, and B color components of the facial region producing LL, LH, HL, and HH sub-bands. Then, the secret message is embedded into the LH, HL, and HH coefficients of the facial region in all video frames. Once this stage is completed, the secret keys and facial box edges are hidden into the non-facial areas. Every four edges of the bounding box in each frame are embedded into the specific non-facial area known by both transmitter and receiver. The length of each facial box is 80 bits per frame (40 bits for each X-axis and Y-axis). Moreover, in order to transmit keys to the receiver party, both secret keys will be embedded into the non-facial region of the first video frame. Upon completion, the stego frames will be reconstructed in order to produce the stego video format that transmits via the communication channel to the receiver party. Figure 3.9 illustrates the block diagram of data embedding phase.

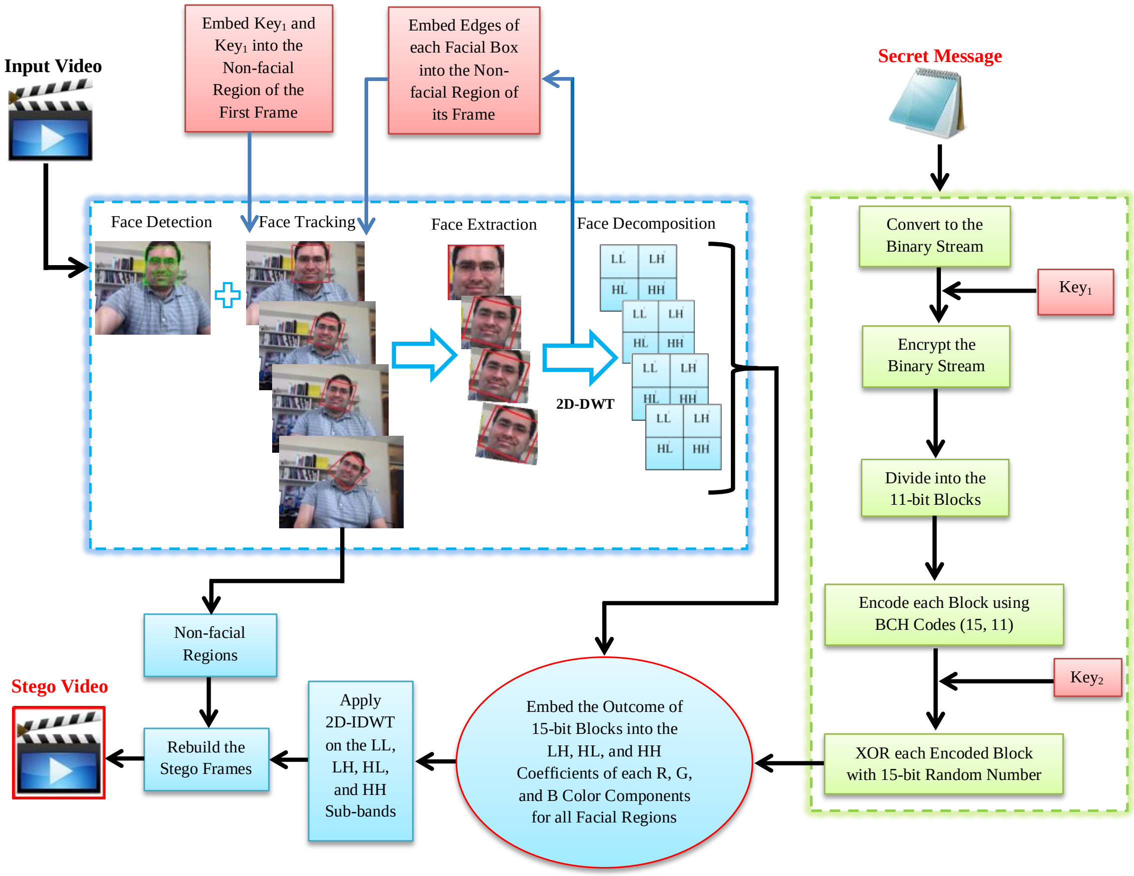

Input Video
Embed Key$_1$ and Key$_1$ into the Non-facial Region of the First Frame
Embed Edges of each Facial Box into the Non-facial Region of its Frame
Secret Message
Face Detection
Face Tracking
Face Extraction
Face Decomposition
LL
LH
HL
HH
2D-DWT
Convert to the Binary Stream
Key$_1$
Encrypt the Binary Stream
Divide into the 11-bit Blocks
Encode each Block using BCH Codes (15, 11)
Key$_2$
XOR each Encoded Block with 15-bit Random Number
Non-facial Regions
Stego Video
Rebuild the Stego Frames
Apply 2D-IDWT on the LL, LH, HL, and HH Sub-bands
Embed the Outcome of 15-bit Blocks into the LH, HL, and HH Coefficients of each R, G, and B Color Components for all Facial Regions

Figure 3.9 Block diagram for the data embedding phase.

3.4.4 Data Extraction Phase

The process of the data extracting phase is illustrated in Figure 3.10. In order to retrieve a secret message correctly, the stego video is divided into frames through the receiver, and two secret keys are extracted from the non-facial area of the first frame. In addition, in each video frame, the four edge points of the facial box are extracted from the non-facial region. Thus, the exact facial region is identified in each video frame. Then, the 2D-DWT is applied on each of the R, G, and B color components of the facial box in order to generate the LL, LH, HL, and HH sub-bands.

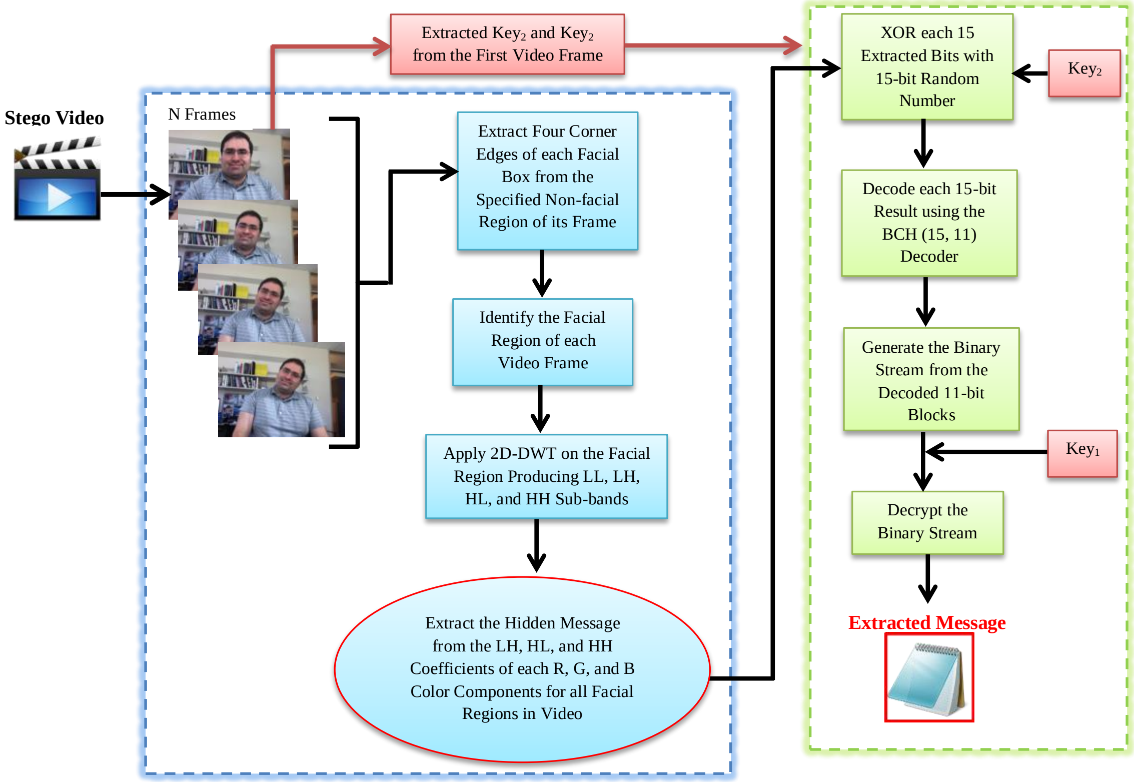

Extracted Key$_2$ and Key$_2$ from the First Video Frame
Stego Video
N Frames
Extract Four Corner Edges of each Facial Box from the Specified Non-facial Region of its Frame
Identify the Facial Region of each Video Frame
Apply 2D-DWT on the Facial Region Producing LL, LH, HL, and HH Sub-bands
Extract the Hidden Message from the LH, HL, and HH Coefficients of each R, G, and B Color Components for all Facial Regions in Video
XOR each 15 Extracted Bits with 15-bit Random Number
Key$_2$
Decode each 15-bit Result using the BCH (15, 11) Decoder
Generate the Binary Stream from the Decoded 11-bit Blocks
Key$_1$
Decrypt the Binary Stream
Extracted Message

Figure 3.10 Block diagram for the data extracting phase.

Then, the process of extracting the hidden message is conducted by taking out the secret message from LH, HL, and HH coefficients of each RGB color channels of the facial region. The extracted bits from all video frames are stored in a binary array. The binary array is divided into the 15-bit blocks. Each block will be XORed with the 15-bit number randomly generated by secret *Key*$_2$. The results of the 15-bit blocks are decoded using the BCH (15, 11, 1) decoder to produce 11-bit blocks. Since the sender has encrypted the secret message, the obtained array is decrypted using secret *Key*$_1$. The final array is divided into an 8-bit code (ASCII) in order to generate the right characters of the original message.

3.5 A New Video Steganography Algorithm Based on the Multiple Object Tracking and ECC

In this section, we present a new video steganography algorithm based on the multiple object tracking algorithm and Hamming codes (15, 11). Our proposed steganography is divided into the following four stages:

3.5.1 Secret Message Preprocessing Stage

In this work, a large size text file is used as a secret message, and it is preprocessed before the embedding stage. Here, the whole characters in the text file are converted into ASCII codes in order to generate an array of binary bits. Then, for security purposes, the binary array is encrypted by using a secret key (*Key*$_1$) that represents the size of the secret message. This process will encode the message and protect it from attackers. Since the binary linear block of Hamming codes (15, 11) are used, the encrypted array is divided into 11-bit blocks. Then, every block is encoded by the Hamming codes (15, 11) that will produce 15-bit blocks. The size of the message is extended by adding four parity bits into each block. Another secret key (*Key*$_2$) is utilized to generate randomized 15-bit numbers, and each number is XORed with the 15-bit encoded block. The security of the proposed algorithm will be improved by using two secret keys, Hamming codes, and XOR operation.

3.5.2 Motion-Based Multiple Object Tracking Stage

The motion-based multiple object tracking algorithm has been previously explained in Section **2.10**. In this stage, the process of identifying the moving objects in the video frames must be performed since motion object regions are used as cover data. This process is achieved by detecting each moving object within an individual frame, and then associating these detections throughout all of the video frames. The background subtraction method is applied to detect the moving objects. Then, the Kalman filter is used to predict estimation trajectory of each moving object.

3.5.3 Data Embedding Stage

In each video frame, the cover data of the proposed algorithm is the motion objects that are considered as regions of interest. The motion regions are identified through the video frames after they are detected and tracked. The region of interest changes in every frame based on the size and the number of the moving objects. The motion-based multiple object tracking algorithm is applied in order to predict trajectories of all moving objects. In each video frame, the background subtraction method is administered to generate a foreground mask which will determine the regions of the moving objects. Then, the R, G, and B components of each motion object's pixels are used for embedding purposes. In our algorithm, one LSB and two LSBs are utilized in order to embed 3 and 6 bits of the secret message in each motion pixel. Moreover, in order to transmit secret keys to the receiver party, both keys are embedded into the non-motion region of the first video frame. Upon completion, the stego frames will be reconstructed in order to produce the stego video format that transmits via the communication medium to the receiver party. Figure 3.11 shows the block diagram of the data embedding stage.

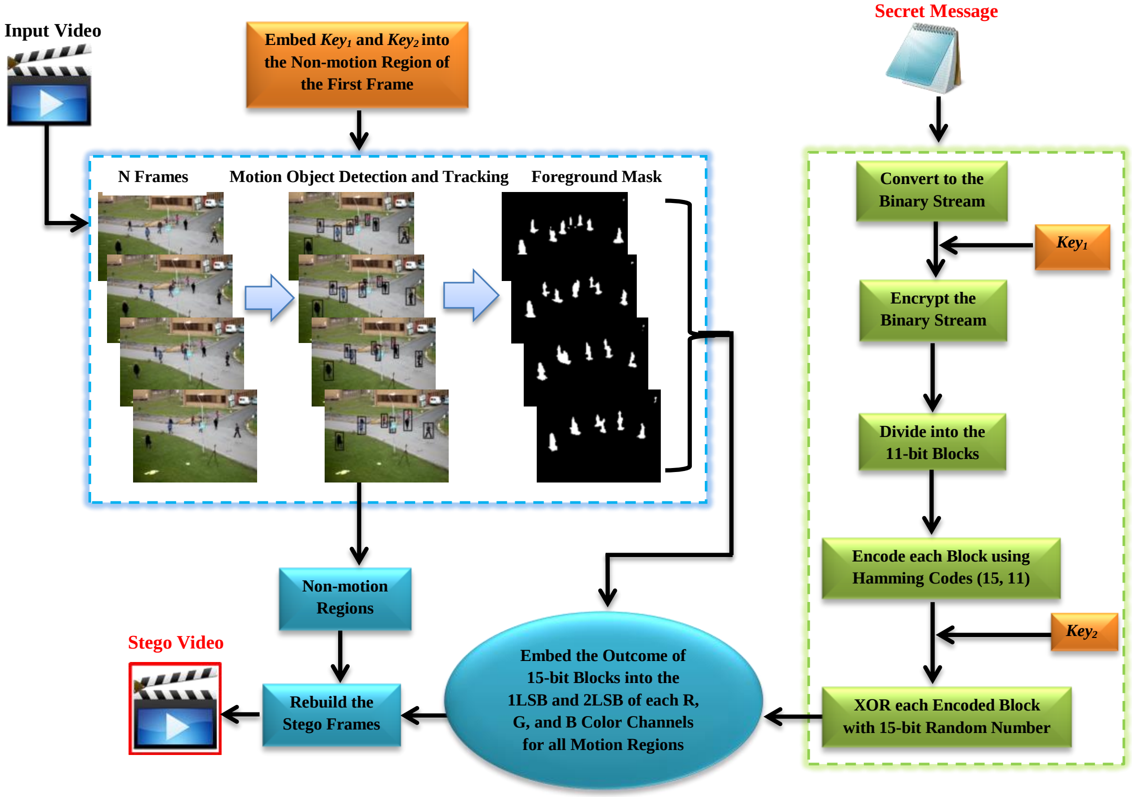

Input Video
Embed Key_1 and Key_2 into the Non-motion Region of the First Frame
Secret Message
N Frames
Motion Object Detection and Tracking
Foreground Mask
Convert to the Binary Stream
Key_1
Encrypt the Binary Stream
Divide into the 11-bit Blocks
Encode each Block using Hamming Codes (15, 11)
Key_2
XOR each Encoded Block with 15-bit Random Number
Non-motion Regions
Stego Video
Rebuild the Stego Frames
Embed the Outcome of 15-bit Blocks into the 1LSB and 2LSB of each R, G, and B Color Channels for all Motion Regions

Figure 3.11 Block diagram of the data embedding stage of the proposed algorithm.

3.5.4 Data Extraction Stage

The process of the data extracting stage is illustrated in Figure 3.12. In order to retrieve a secret message correctly, the stego video is divided into frames through the receiver, and then two secret keys are extracted from the non-motion region of the first frame. To predict trajectories of motion objects, the motion-based multiple object tracking algorithm is applied again by the receiver. Moreover, in each video frame, a foreground mask that is similar to the embedding stage's mask is produced by using the background subtraction method.

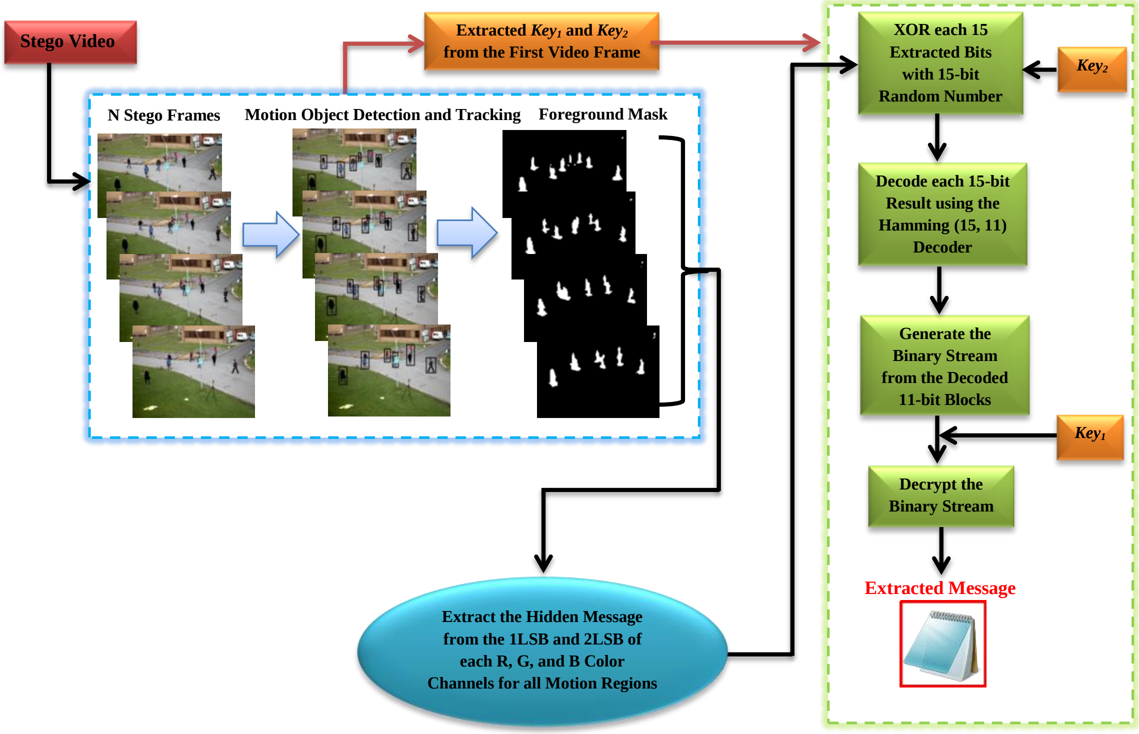

Stego Video
Extracted Key_1 and Key_2 from the First Video Frame
N Stego Frames
Motion Object Detection and Tracking
Foreground Mask
Extract the Hidden Message from the 1LSB and 2LSB of each R, G, and B Color Channels for all Motion Regions
XOR each 15 Extracted Bits with 15-bit Random Number
Key_2
Decode each 15-bit Result using the Hamming (15, 11) Decoder
Generate the Binary Stream from the Decoded 11-bit Blocks
Key_1
Decrypt the Binary Stream
Extracted Message

Figure 3.12 Block diagram of the data extraction stage of the proposed algorithm.

Then, the process of extracting the hidden message is conducted by taking out 3 and 6 bits from first LSB and second LSBs of RGB color components in each motion object's pixels of all the video frames. The extracted bits from all the video frames are stored in a binary array. The binary array is divided into 15-bit blocks. Each block will be XORed with the 15-bit number randomly generated by secret Key_2. The results of the 15-bit blocks are decoded by using the Hamming (15, 11) decoder to produce 11-bit blocks. Since the sender has encrypted the secret message, the obtained array is decrypted by using secret Key_1. The final array is divided into an 8-bit code (ASCII) in order to generate the right characters of the original message.

3.6 A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC

In this section, a robust and secure video steganography algorithm in DWT-DCT domains based on MOT and ECC is presented. The major stages of the proposed video steganography framework are illustrated in Figure 3.13. A sizeable text data is utilized as s secret messages, and it is preprocessed prior the data embedding interval, which is ciphered and coded by Hamming and BCH (7, 4) codes. Figure 3.14 illustrates the process of securing input messages prior the embedding stage. The proposed steganographic algorithm is separated into the three stages:

3.6.1 Motion-Based MOT Stage

The motion-based MOT algorithm has previously discussed in Section **2.10**. The process of identifying the moving objects in the video frames must be carried out when motion object regions are utilized as host data. This process is achieved by detecting each moving object within an individual frame, and then associating these detections throughout all of the video frames. The background subtraction method is applied to detect the moving objects based on the GMM. It also computes the differences between consecutive frames that generate the foreground mask. Then, the Kalman filter is employed to predict estimation trajectory of each moving region.

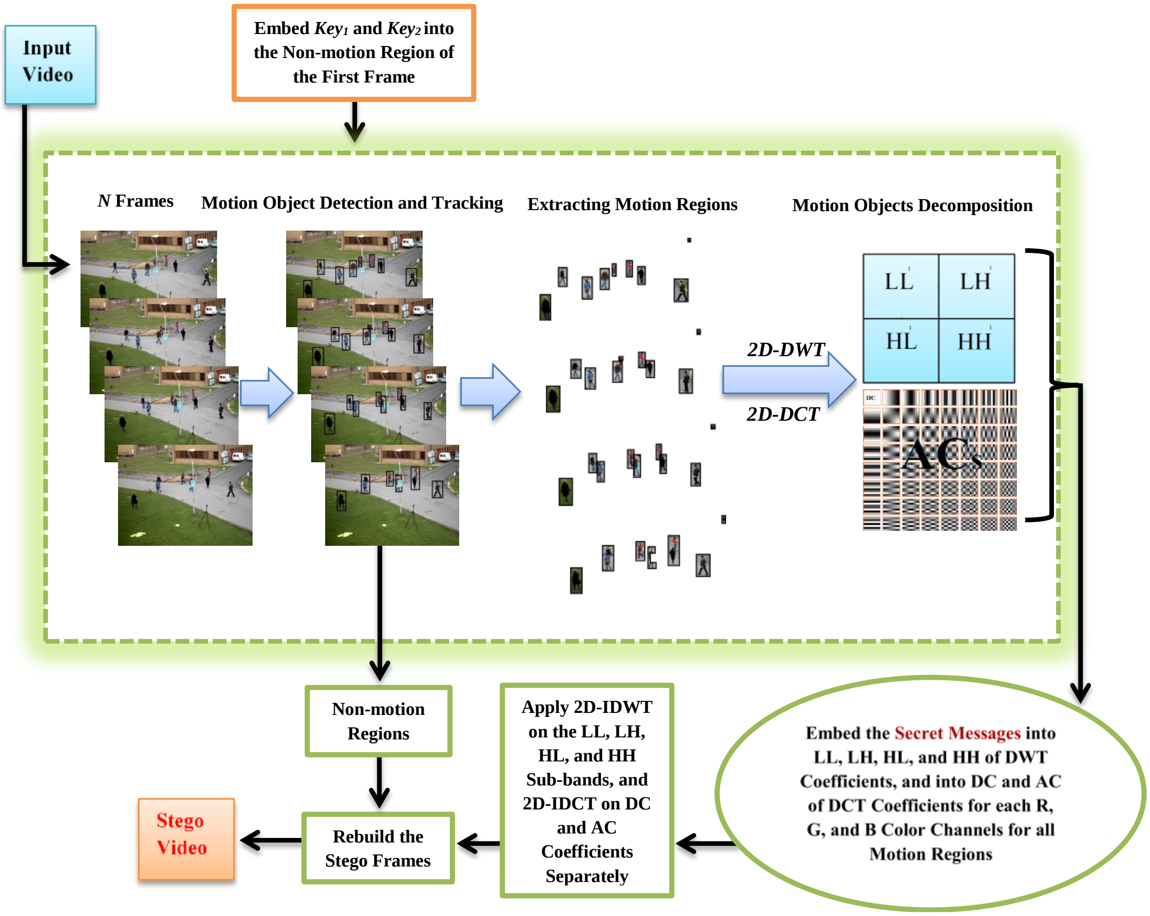

Input Video
Embed *Key₁* and *Key₂* into the Non-motion Region of the First Frame
N Frames
Motion Object Detection and Tracking
Extracting Motion Regions
Motion Objects Decomposition
LL
LH
HL
HH
2D-DWT
2D-DCT
DC
ACs
Non-motion Regions
Apply 2D-IDWT on the LL, LH, HL, and HH Sub-bands, and 2D-IDCT on DC and AC Coefficients Separately
Embed the Secret Messages into LL, LH, HL, and HH of DWT Coefficients, and into DC and AC of DCT Coefficients for each R, G, and B Color Channels for all Motion Regions
Stego Video
Rebuild the Stego Frames

Figure 3.13 The proposed video steganography framework.

3.6.2 Data Embedding Stage

In entire video frames, the host data of our proposed method is the motion objects that are considered as regions of interest. By using the motion-based MOT algorithm, the process of detecting and tracking the motion regions over all video frames are achieved.

The regions of interest altered in each video frame is dependent on the number and the size of the moving objects. In every frame, 2D-DWT is implemented on RGB channels of each motion region resulting LL, LH, HL, and HH subbands. In addition, 2D-DCT is also applied on the same motion regions generating DC and AC coefficients. Thereafter, the secret messages are concealed into LL, LH, HL, and HH of DWT coefficients, and into DC and AC of DCT coefficients of each motion object separately based on its foreground mask. Furthermore, both secret keys are transmitted to the receiver side by embedding them into the non-motion area of the first frame. Upon accomplishment, the stego video frames are rebuild in order to construct the stego video that can be transmitted through the unsecure medium to the receiver. Algorithm 3 clarifies the major steps of our embedding algorithm.

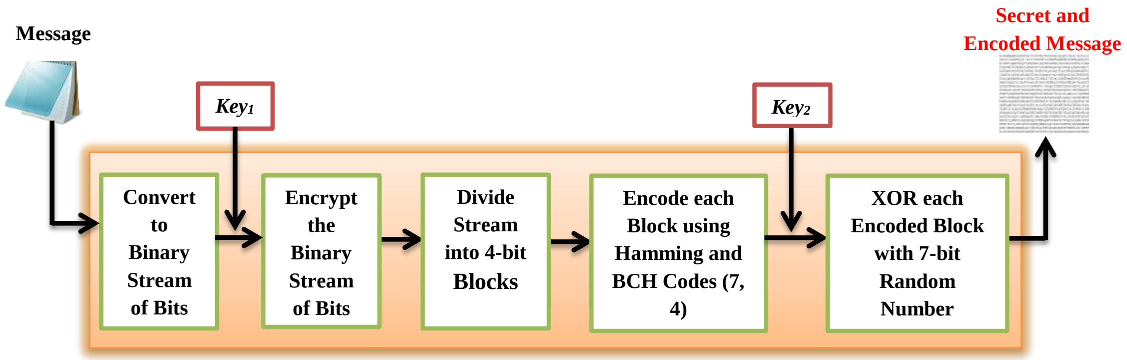

Message
Key1
Key2
Secret and Encoded Message
Convert to Binary Stream of Bits
Encrypt the Binary Stream of Bits
Divide Stream into 4-bit Blocks
Encode each Block using Hamming and BCH Codes (7, 4)
XOR each Encoded Block with 7-bit Random Number

Figure 3.14 Process of encrypting and encoding input messages.

3.6.3 Data Extraction Stage

In order to recover hidden messages accurately, the embedded video is separated into a number of frames through the receiver side, and then two secret keys are obtained from the non-motion region of the first video frame. To predict trajectories of motion objects, the motion-based MOT algorithm is applied again by the receiver. Then, 2D-DWT and 2D-DCT are employed on the RGB channels of each motion object in order to create LL, LH, HL, and HH subbands, and DC and AC coefficients, respectively.

Algorithm 3: Data Embedding Stage

```
Input: V //Video, M //Secret message in characters, $Key_1$, $Key_2$; //Secret keys
Output: SV; //Output of Stego videos
Initialize km1, pm1, p1;
Bin ← Msm; //Change the text message to binary vector
$Key_1$ ← Len(Bin)/4; //Size of the hidden messages
$Key_2$ ← rand (2^7, $Key_1$,1)'; //Randomizing the secret $Key_1$
EnB ← En(Bin, [$Key_1$]);  //Ciphering the binary vector by $Key_1$
$for_1$ i = 1: ($Key_1$*7) do //Encode each 4 bits of secret message by Hamming and BCH (7,4)
    g(1:4) ← get(EnB(km1:km1+4));
    En_EnB ← encode(g,7,4);
    tem(1:7) ← get($Key_2$(i));
    Encdmsg(pm1:pm1+7) ← xor(En_EnB,tem);
    pm1+7; km1+4;
$end_1$
{$Vf_1$, $Vf_2$,…, $Vf_n$}x ← V; //Video V is divided into n frames.
MODTBox ← MODT(Vf); //Calling the Motion Object Detection and Tracking for each video frame Vf.
Non_Motion($Vf_1$) ← $Key_1$, $Key_2$; //Embed keys ($Key_1$ and $Key_2$) into the non-motion areas of the first frame $Vf_1$.
FMask = mask(Vf); //Identify the foreground mask of each motion region in Vf frame of size (Vfx, Vfy).
[CoeffR, CoeffG, CoeffB] ← DWT/DCT (MODTBox); //Applying 2D-DWT and 2D-DCT separately on each motion object for RGB frame components
//Conceal the secret messages into coefficients of R, G, and B for each motion object.
   $for_2$ i = 1:Vfx do
      $for_3$ j= 1:Vfy do
         $if_4$ FMask(i,j) == 1
            $CoeffR_{1,2, or 3}$ ← Encdmsg(p1+1,4, or7);
            $CoeffG_{1,2, or 3}$ ← Encdmsg(p1+2,5, or 8);
            $CoeffB_{1,2, or 3}$ ← Encdmsg(p1+3,6, or 9);
                     p1+3,6, or 9;
    $end_4$ $end_3$ $end_2$
SVf ← IDWT/IDCT (CoeffR, CoeffG, CoeffB); //Applying 2D-IDWT and 2D-IDCT separately on each component coefficients to produce the stego frame.
SV ← {$SVf_1$, $SVf_2$,…, $SVf_n$}x; //Obtain the stego video
```

Next, the extracting process of the embedded data is achieved by obtaining the secret messages from LL, LH, HL, HH, DC, and AC coefficients of each motion region over all video frames based on the same foreground masks used in the embedding stage. The extracted secret message is decoded by Hamming and BCH (7, 4), and then decrypted to obtain the original message. The essential steps of data extracting algorithm are shown in the Algorithm 4.

Algorithm 4: Data Extracting Stage

```
Input: SV; //Input Stego videos
Output: Msm; //Secret text data in characters
Initialize km1, pm1, p1;
{$Sf_1$, $Sf_2$,…, $Sf_n$} ← SV; //Stego Video SV is divided into n frames
Extract[$Key_1$, $Key_2$] from ($Sf_1$); //Extract keys ($Key_1$ and $Key_2$) from the non-motion areas of the first embedded frame $Sf_1$
MODTBox ← MODT(SVf); //Calling the Motion Object Detection and Tracking for each stego frame SVf.
FMask = mask(SVf); //Identify the foreground mask of each motion region in SVf frame of size (SVfx, SVfy).
[CoeffR, CoeffG, CoeffB] ← DWT/DCT (MODTBox); //Applying 2D-DWT and 2D-DCT separately on each motion object in RGB stego frame.
//Recover the secret messages from the coefficients of R, G, and B for each motion object.
   $for_1$ i = 1:Sfx do
        $for_2$ j= 1:Sfy do
            $if_3$ FMask(i,j) == 1
                Encdmsg(p1+1,4, or 7) ←  $CoeffR_{1,2, or 3}$ ;
                Encdmsg(p1+2,5, or 8) ←  $CoeffG_{1,2, or 3}$ ;
                Encdmsg(p1+3,6, or 9) ←  $CoeffB_{1,2, or 3}$ ;
                    P1+3,6, or 9;
        $end_3$ $end_2$ $end_1$
$for_4$ i = 1: ($Key_1$*7) do //Decode each 7 bits of retrieved data by Hamming and BCH (7,4).
      Sg(1:7) ← get(Encdmsg (km1:km1+7));
      tem(1:7) ← get($Key_2$(i));
      En_EnB   ← xor(Sg,tem)
      EnB ← decode(En_EnB ,7,4);
      Pm1+4; km1+7;
$end_4$
Bin ← D(EnB, [$Key_1$]); //Decipher the binary vector by $Key_1$
Msm ← Bin; //Alter the binary vector to the text data.
get Msm   //Obtain secret messages
```

CHAPTER 4: EXPERIMENTAL RESULTS

4.1 A Highly Secure Video Steganography Algorithm Based on ECC

A database of nine standard Common Interchange Format (CIF) video sequences is used, with the size (288 x 352) and the format 4:2:0 YUV. All video sequences are equal in length with 300 frames in each one. A large text file is used as a secret message. The MATLAB software program is used to implement this work and test our experiment results.

In Figure 4.1, an example of one frame (frame number 111) in the Foreman video is chosen. The first part of the figure shows that the three components of the 111th frame are separated. Then it shows some locations that have been selected randomly for the secret message. The embedded locations are different in each component inside one frame and they differ from one frame to next, which mainly depends on the secret key. The second part of the figure shows frame number 111 before and after the embedding process.

In Table 4.1, the average PSNR for all video sequences is shown for each Y, U, and V component and all are greater than 51 dB. The quality of the stego videos are mostly the same as the original videos. Figure 4.2 shows the PSNRs of 300 stego frames in the Mother-daughter video. The qualities of the results that have been obtained from our proposal are very close to the quality of the original videos before embedding. In general, PSNRs are greater than 51 dBs, and the V component has a better quality among the three components.

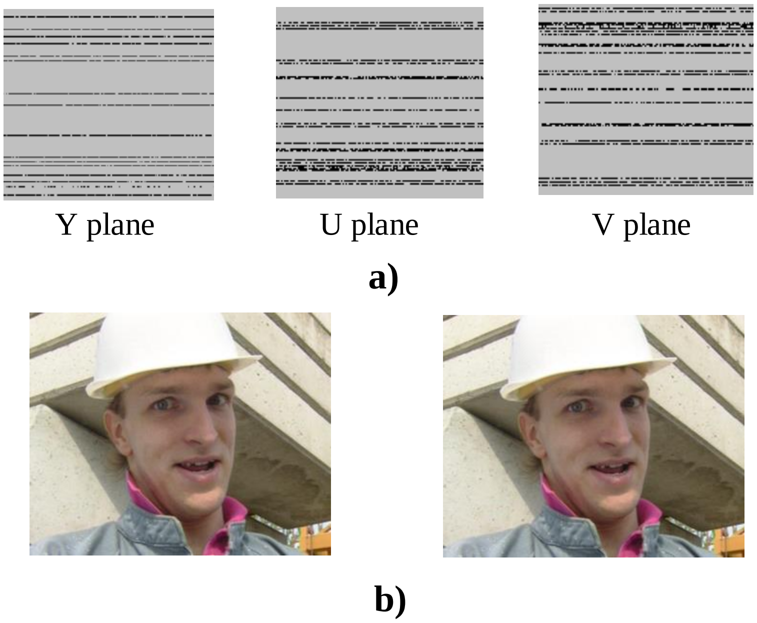


Figure 4.1 A sample result of frame number 111 for the Foreman video. a) shows the selected areas for embedding in YUV components for frame number 111, b) shows the 111th frame both the original and the stego frames.

Figure 4.3 shows the comparison of visual quality between nine stego videos. The PSNR of each component, Y, U, and V, is calculated, of which the average is 300 frames per video. All the results of PSNRs are between 51 and 52.5 dBs, which are considered very good results with regard to the purpose of quality.

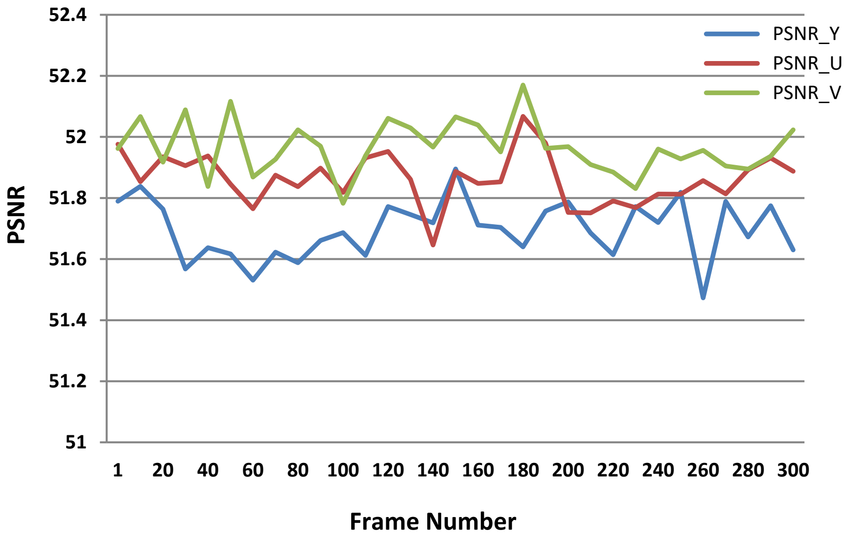


Figure 4.2 PSNR of 300 stego frames for the Mother-daughter video.

Table 4.2 shows the experimental results of the proposed algorithm and other five methods based on embedding capacity, visual quality, cover video preprocessing, and secret message preprocessing. Our algorithm clearly dominates the five existing algorithms by attaining highest values of the PSNR and hidden ratio (HR).

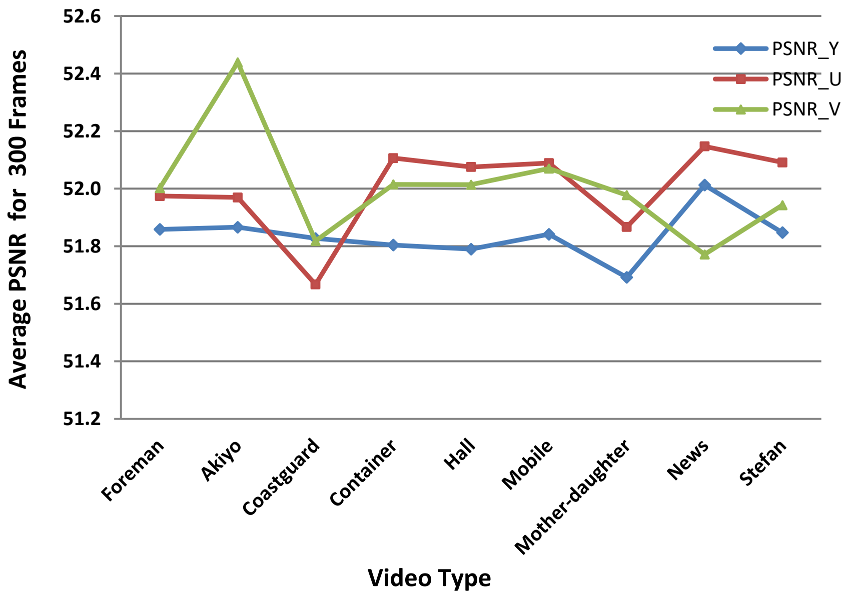


Figure 4.3 Comparison between the averages of the PSNR Y, U, and V components for nine videos.

Table 4.1 The average PSNR of Y, U, and V for all video sequences.

Sequences	*Frame No.*	*PSNRY*	*PSNRU*	*PSNRV*
Foreman	**1-100**	**51.901**	**51.940**	**51.920**
	101-200	**51.857**	**51.924**	**52.049**
	201-300	**51.817**	**52.059**	**52.038**
Akiyo	**1-100**	**51.881**	**51.988**	**52.431**
	101-200	**51.859**	**51.978**	**52.458**
	201-300	**51.859**	**51.943**	**52.428**
Coastguard	**1-100**	**51.835**	**51.664**	**51.854**
	101-200	**51.824**	**51.682**	**51.806**
	201-300	**51.823**	**51.655**	**51.795**
Container	**1-100**	**51.821**	**52.146**	**52.067**
	101-200	**51.806**	**52.117**	**52.008**
	201-300	**51.785**	**52.056**	**51.970**
Hall	**1-100**	**51.787**	**52.084**	**52.021**
	101-200	**51.797**	**52.079**	**52.016**
	201-300	**51.785**	**52.063**	**52.005**
Mobile	**1-100**	**51.862**	**52.127**	**52.065**
	101-200	**51.829**	**52.076**	**52.074**
	201-300	**51.834**	**52.064**	**52.072**
Mother-daughter	**1-100**	**51.686**	**51.857**	**51.995**
	101-200	**51.702**	**51.868**	**51.992**
	201-300	**51.687**	**51.876**	**51.946**
News	**1-100**	**52.027**	**52.167**	**51.781**
	101-200	**52.012**	**52.139**	**51.769**
	201-300	**51.998**	**52.135**	**51.764**
Stefan	**1-100**	**51.885**	**52.082**	**51.961**
	101-200	**51.810**	**52.111**	**51.964**
	201-300	**51.848**	**52.081**	**51.904**

Table 4.2 Performance comparison of the proposed algorithm with existing five methods.

Method	*HR*	*PSNR*	*Video Preprocessing*	*Message Preprocessing*
Chang et al. [53]	1.04%	37.00 dB	✕	✕
Ma et al. [58]	0.10%	40.74 dB	✕	✕
Wang et al. [83]	0.57%	37.05 dB	✕	✕
Ke et al. [80]	2.44%	34.54 dB	✕	✕
Alavianmehr et al. [88]	1.34%	36.97 dB	✕	✕
Proposed algorithm	**7.39%**	**51.75 dB**	✔	✔

4.2 An Increased Payload Video Steganography Algorithm in DWT Domain Based on ECC

In this section, the performance of our second algorithm is evaluated through several experiments. The experimental environment utilizes several variables: the cover data comprise a dataset consisting of seven video sequences 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 300 frames. A large text file is used as a secret message. The results are implemented using both fast and slow motion videos. In Figure 4.4, the PSNR of the Y-components are calculated for all seven videos. The results of the PSNR-Y for the *Akiyo*, *Container*, *Bus*, and *Foreman* videos are more stable, while in the *Soccer* and *Tennis* videos, the quality is frequently changing. The reason for varying the visual quality is because the sporting videos contain faster motion objects than the others. Overall, the *Akiyo* video has the best visual quality.

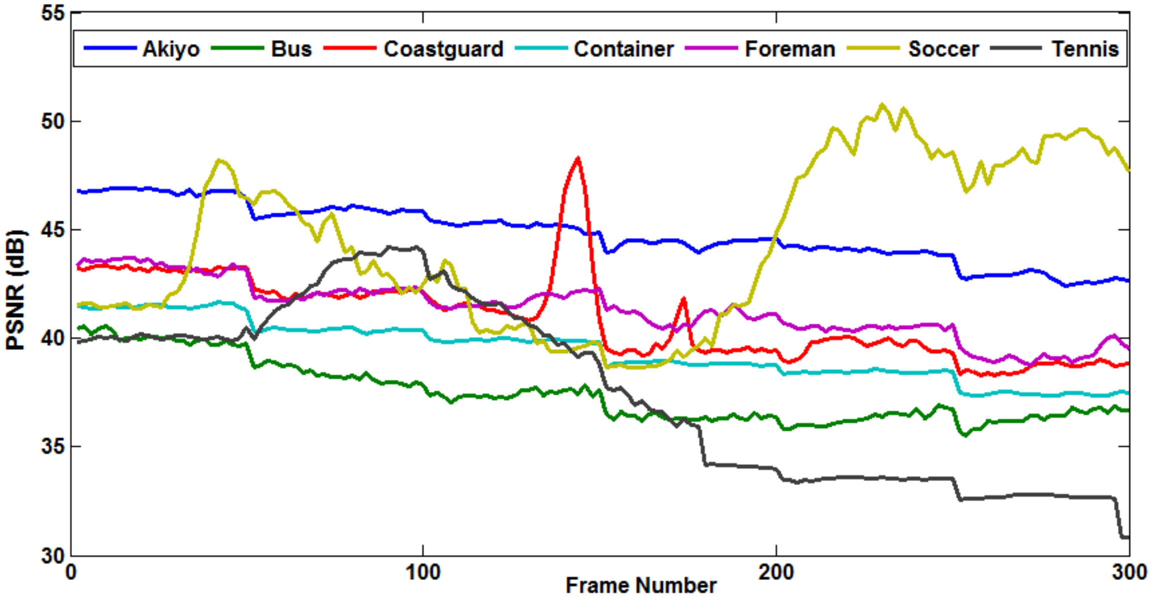


Figure 4.4 PSNR comparisons for the Y-components of all videos.

Figures 4.5 and 4.6 show the PSNR of the U-component and the V-component, respectively, for all seven videos. In the first figure, the demonstrated results of the PSNR-U for the *Akiyo* and *Container* videos are changing slightly from one frame to other, as compared to other five videos. The PSNR-U of the *Coastguard* video has the highest dBs among the group. In the second figure, the PSNR-V for all video streams has been calculated; the *Coastguard* and *Soccer* videos have a better quality. In Figure 4.7, the average PSNR comparison for 300 frames of each video is shown. The comparison shows that the result of the visual quality for each of the *Akiyo*, *Container*, *Bus*, and *Foreman* videos ranged between **40.44** – **42.05** dBs; these videos all contain slower motion objects while the PSNR of the *Coastguard*, *Soccer*, and *Tennis* videos change frequently over time (ranges between **37.64** – **44.23** dB). The changes occur because these videos contain faster motion objects that lead to unstable visual quality.

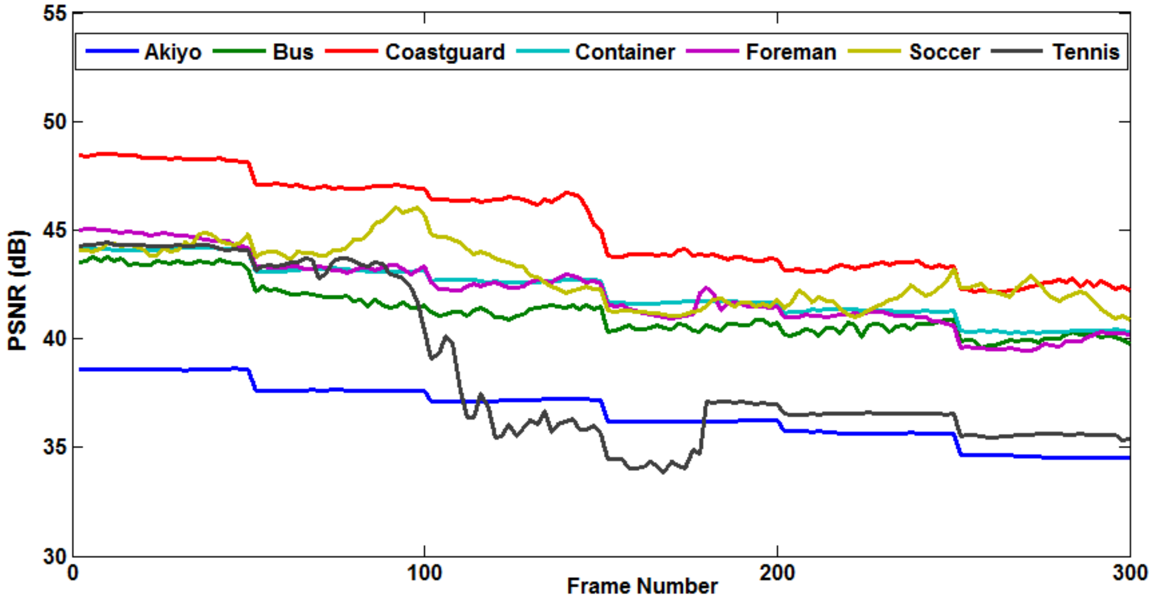


Figure 4.5 PSNR comparisons for the U-component of all videos.

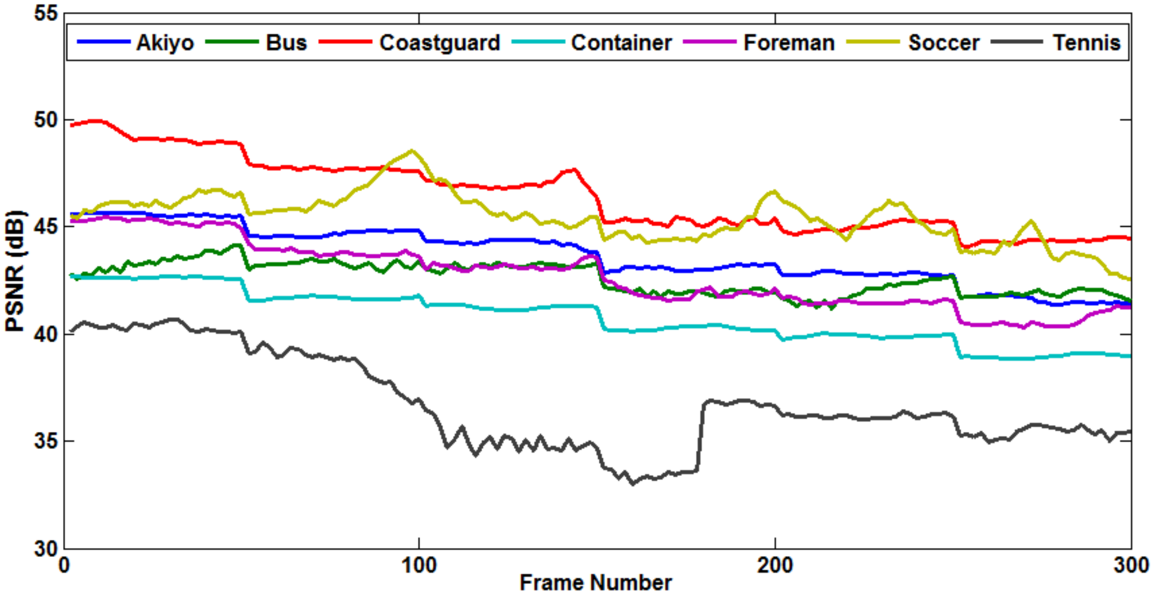


Figure 4.6 PSNR comparisons for the V-component of all videos.

Table 4.3 shows the average of the PSNR for each Y, U, and V component for all video sequences. The visual quality of each part is measured separately by averaging each of the 100 frames per video. The average results' are different and depend on both the type of videos and the speed of the motion object.

In Table 4.4, there are five videos of both fast and slow motion objects (the *Soccer*, *Tennis*, and *Coastguard* have fast motion objects; the *Akiyo* and *Container* videos have slow motion objects). Part (a) of the table shows the stego frame that has the lowest *PSNR* and its original frame in each video. Part (b) of the table indicates the stego frame that has the highest *PSNR* and its original frame in each video. It can be observed that the minimum and maximum *PSNR* of the videos that have slow motion objects are very close to one another, as in the *Akiyo* and *Container* videos. However, the minimum and maximum *PSNR* of the videos that have fast motion objects are different in *dBs* range such as the *Soccer*, *Tennis*, and *Coastguard* videos. Overall, the objective quality of both video types is considerable.

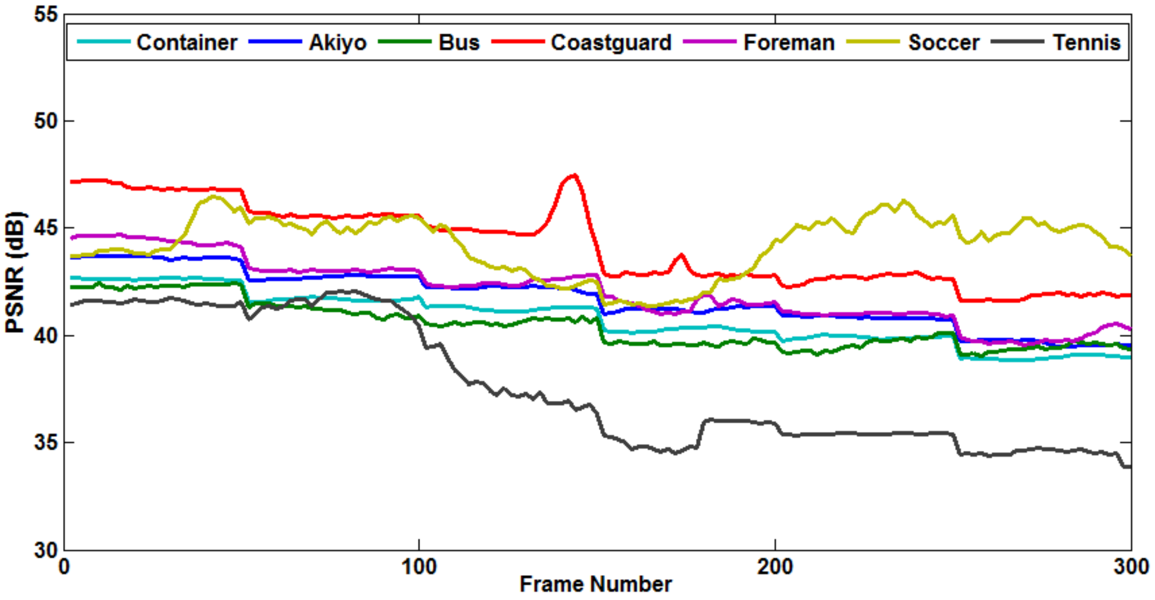


Figure 4.7 PSNR comparisons for 300 frames of all videos.

Table 4.3 The average PSNR for each Y, U, and V component for all seven videos.

Video Sequences	***Frame Number***	***PSNRY***	***PSNRU***	***PSNRV***
Akiyo	**1-100**	**46.299**	**38.104**	**45.122**
	101-200	**44.787**	**36.681**	**43.663**
	201-300	**43.405**	**35.121**	**42.192**
Coastguard	**1-100**	**42.626**	**47.666**	**48.490**
	101-200	**41.064**	**45.054**	**46.126**
	201-300	**39.148**	**42.858**	**44.677**
Container	**1-100**	**40.923**	**43.632**	**42.153**
	101-200	**39.365**	**42.171**	**40.752**
	201-300	**37.942**	**40.812**	**39.434**
Bus	**1-100**	**39.161**	**42.687**	**43.318**
	101-200	**36.902**	**40.916**	**42.563**
	201-300	**36.330**	**40.237**	**41.917**
Soccer	**1-100**	**43.929**	**44.447**	**46.387**
	101-200	**40.471**	**42.357**	**45.349**
	201-300	**48.663**	**41.907**	**44.579**
Foreman	**1-100**	**42.706**	**43.989**	**44.525**
	101-200	**41.374**	**41.982**	**42.532**
	201-300	**39.870**	**40.409**	**41.065**
Tennis	**1-100**	**41.452**	**43.694**	**39.421**
	101-200	**38.278**	**36.114**	**34.966**
	201-300	**33.038**	**36.050**	**35.783**

On the other hand, according to [123], this algorithm has a high embedding payload. The obtained hiding ratio is **28.12**%. A reasonable tradeoff is noticed between the amount of the embedded message in each video (**12.23** Mbytes) and the quality (**37.64** - **44.23** dBs). A number of experiments were conducted to compare the embedding capacity of our algorithm and the embedding capacity of both the LSB algorithm and [88].

Table 4.5 shows the comparison between the three algorithms, according to the amount of secret data in each frame.

Table 4.4 Minimum and maximum PSNR frames for each of five videos.

Minimum PSNR		***Maximum PSNR***	
Original Frames	**Stego frames**	**Original Frames**	**Stego frames**
74th in *Akiyo*	PSNR **41.44** dB	107th of *Akiyo*	PSNR 41.95 dB
26th in *Container*	PSNR **40.63** dB	143rd in *Container*	PSNR **40.99** dB
2nd in *Soccer*	PSNR **41.68** dB	136th in *Soccer*	PSNR **47.43** dB
86th in *Tennis*	PSNR **34.99** dB	39th in *Tennis*	PSNR **41.08** dB
90th in *Coastguard*	PSNR **43.24** dB	72nd in *Coastguard*	PSNR **47.00** dB
a)		**b)**	

Table 4.5 Embedding capacity comparison of our algorithm with both Alavianmehr and LSB Algorithms.

Video Resolution	YUV	Proposed Algorithm (Bits/Frame)	*Alavianmehr et al.* (Bits/Frame)	LSB Algorithm (Bits/Frame)
176 X 144	Y	57024	4096	25344
	U	14256	Not used	6336
	V	14256	Not used	6336
352 X 288	Y	228096	8192	101376
	U	57024	Not used	25344
	V	57024	Not used	25344

This algorithm has improved the embedding capacity of [88] and the LSB algorithm by approximately **41.7** and **2.2** times, respectively, without visual quality degradation. Figure 4.8 shows the embedding capacity improvement of our algorithm. Table 4.6 shows the experimental results of our method and the other five methods based on embedding capacity, visual quality, cover video preprocessing, and secret message preprocessing. The proposed method obviously outperforms the five existing methods by obtaining highest values of the *PSNR* and *HR*.

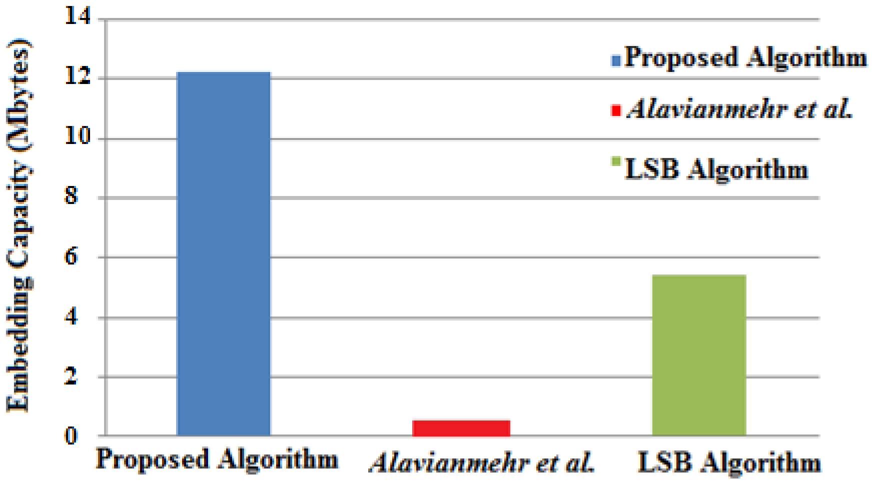


Figure 4.8 Comparison of the proposed algorithm with [88] and LSB.

Table 4.6 Performance comparison of the proposed method with existing five methods.

Method	***HR***	***PSNR***	***Video Preprocessing***	***Message Preprocessing***
Chang et al. [53]	1.04%	37.00 dB	✕	✕
Ma et al. [76]	0.10%	40.74 dB	✕	✕
Wang et al. [83]	0.57%	37.05 dB	✕	✕
Ke et al. [80]	2.44%	34.54 dB	✕	✕
Alavianmehr et al. [88]	1.34%	36.97 dB	✕	✕
Proposed algorithm	**28.12%**	**41.58 dB**	✔	✔

Finally, the robustness of our method 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* and the lower *BER* must be obtained. Table 4.7 illustrates the performance of the proposed algorithm under attacks while it retrieves the hidden data with a high *Sim* and a low *BER*.

Table 4.7 Performance of the proposed algorithm under attacks.

Type of Attack		***Akiyo***		***Bus***		***Coastguard***		***Container***		***Foreman***		***Soccer***		***Tennis***	
		Sim	**BER %**	**Sim**	**BER %**	**Sim**	**BER %**	**Sim**	**BER %**	**Sim**	**BER %**	**Sim**	**BER %**	**Sim**	**BER %**
No attacks		**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
(Salt & Pepper) *Density=*	0.01	**0.955**	**4.5**	**0.965**	**3.5**	**0.945**	**5.5**	**0.975**	**2.5**	**0.965**	**3.5**	**0.923**	**7.7**	**0.921**	**7.9**
	0.001	**0.963**	**3.7**	**0.973**	**2.7**	**0.953**	**4.7**	**0.983**	**1.7**	**0.959**	**4.1**	**0.932**	**6.8**	**0.933**	**6.7**
(Gaussian white) *Variance=*	0.01	**0.923**	**7.7**	**0.933**	**6.7**	**0.913**	**8.7**	**0.943**	**5.7**	**0.919**	**8.1**	**0.902**	**9.8**	**0.901**	**9.9**
	0.001	**0.909**	**9.1**	**0.919**	**8.1**	**0.899**	**10.1**	**0.929**	**7.1**	**0.898**	**10.2**	**0.874**	**12.6**	**0.865**	**13.5**
Median filtering		**0.986**	**1.4**	**0.987**	**1.3**	**0.986**	**1.4**	**0.998**	**0.2**	**0.975**	**2.5**	**0.959**	**4.1**	**0.961**	**3.9**

The proposed algorithm not only achieved a high embedding capacity of the secret message but also improved the secrecy of hidden data. This is mainly because two secret keys and BCH (15, 11) code had been used prior to the embedding process to produce the unreadable message to safeguard it against attackers.

4.3 A Novel Video Steganography Algorithm Based on KLT Tracking and ECC

This section presents experimental results that are implemented by the MATLAB software version R2013a. A dataset of five videos ($Video_1$, $Video_2$, $Video_3$, $Video_4$, and $Video_5$) with the format of audio video interleave (AVI) are used. The implemented videos are videoconferencing sequences taken by the laptop camera (from the computer vision MATLAB toolbox). The cover videos have a 640x480 pixel resolution at 30 frames per second, and a data rate of 8856 kbps. Each video contains a face object through the entire 413 frames. In all video frames, the secret message is a large text file segmented according to the size of the ROI.

In order to evaluate the transparency of our algorithm, *PSNR*, *PSNR-HVS (PSNRH)*, and modified *PSNR-HVS (PSNRM)* metrics are applied. Figure 4.9 explains the visual quality comparison when one LSB is used for embedding the hidden message from each of R, G, and B color channels. Here, the averages of *PSNR*, *PSNRH*, and *PSNRM* for the five experiments are 53.07, 64.01, and 74.51 dBs, respectively. Visual qualities of the stego videos are close to visual qualities of the original videos.

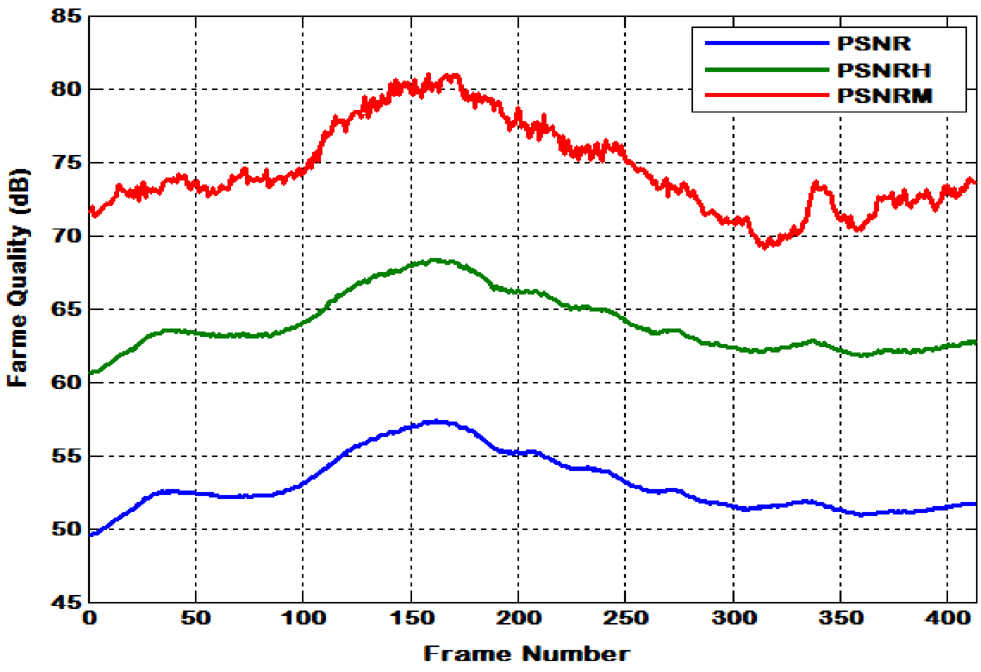


Figure 4.9 Averages of visual qualities for five videos using one LSB.

Figure 4.10 shows the visual quality comparison when two LSBs are utilized to hide the secret message from each of three color components. The averages of *PSNR*, *PSNRH*, and *PSNRM* for the five videos are 43.10, 56.91, and 64.75 dBs, respectively.

Table 4.8 summarizes the results of the visual qualities included in each *PSNR*, *PSNRH*, and *PSNRM* for the five experiments using one LSB, two LSBs, three LSBs, and four LSBs.

In Figure 4.11, the visual quality comparison is shown by using three LSBs for the embedding payload from each of the RGB pixel components. Here, the averages of each *PSNR*, *PSNRH*, and *PSNRM* for the five experiments equal 36.86, 50.29, and 55.56 dBs, respectively.

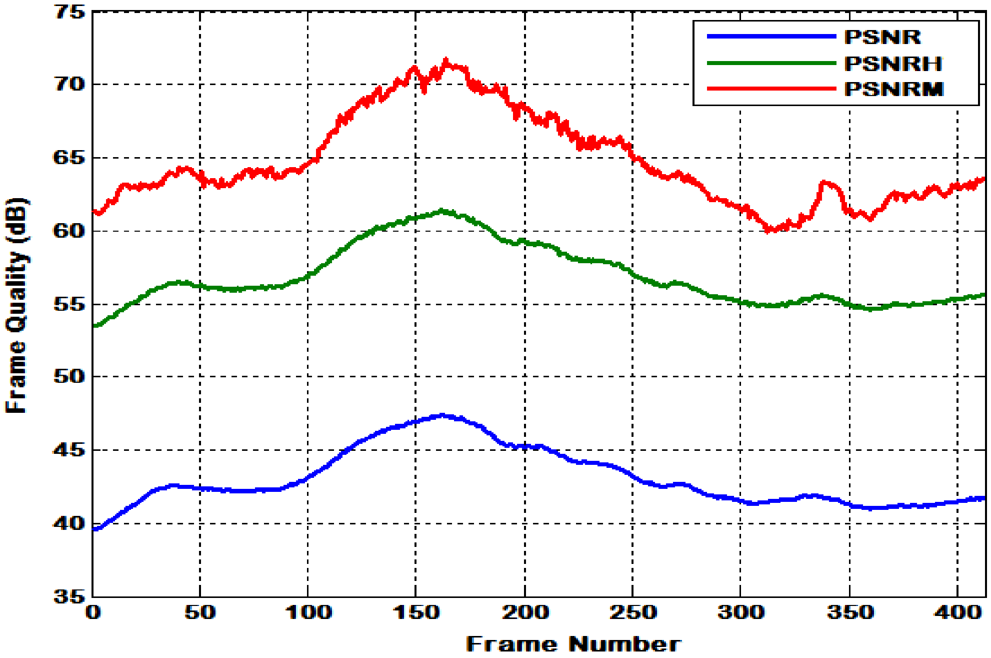


Figure 4.10 Averages of visual qualities for five experiments using two LSBs.

Table 4.8 Visual qualities comparison for five experiments using each of one LSB, two LSBs, three LSBs, and four LSBs into R, G, and B color components.

Visual quality	*No. of LSBs in each R, G, and B*	*Video1*	*Video2*	*Video3*	*Video4*	*Video5*
PSNR	***1 LSB***	53.08	53.53	52.23	53.93	52.63
	2 LSBs	43.11	43.56	42.26	43.96	42.66
	3 LSBs	36.87	37.32	36.02	37.72	36.42
	4 LSBs	33.81	34.26	32.96	34.66	33.36
PSNRH	***1 LSB***	64.02	64.47	63.17	64.87	63.57
	2 LSBs	56.91	57.36	56.06	57.76	56.46
	3 LSBs	50.30	50.75	49.45	51.15	49.85
	4 LSBs	43.54	43.99	42.69	44.39	43.09
PSNRM	***1 LSB***	74.51	74.96	73.66	75.36	74.06
	2 LSBs	64.75	65.20	63.90	65.60	64.30
	3 LSBs	55.56	56.01	54.71	56.41	55.11
	4 LSBs	46.81	47.26	45.96	47.66	46.36

Figure 4.12 illustrates the visual quality comparison when four LSBs are used for embedding the hidden message in each of the color channels. The averages of each *PSNR*, *PSNRH*, and *PSNRM* for five videos are 33.81, 43.54, and 46.81 dBs, respectively. The *PSNRM* metric has enhanced the visual quality. In conclusion, due to the high values of *PSNR*, *PSNRH*, and *PSNRM*, the proposed algorithm has consistent visual qualities for stego videos.

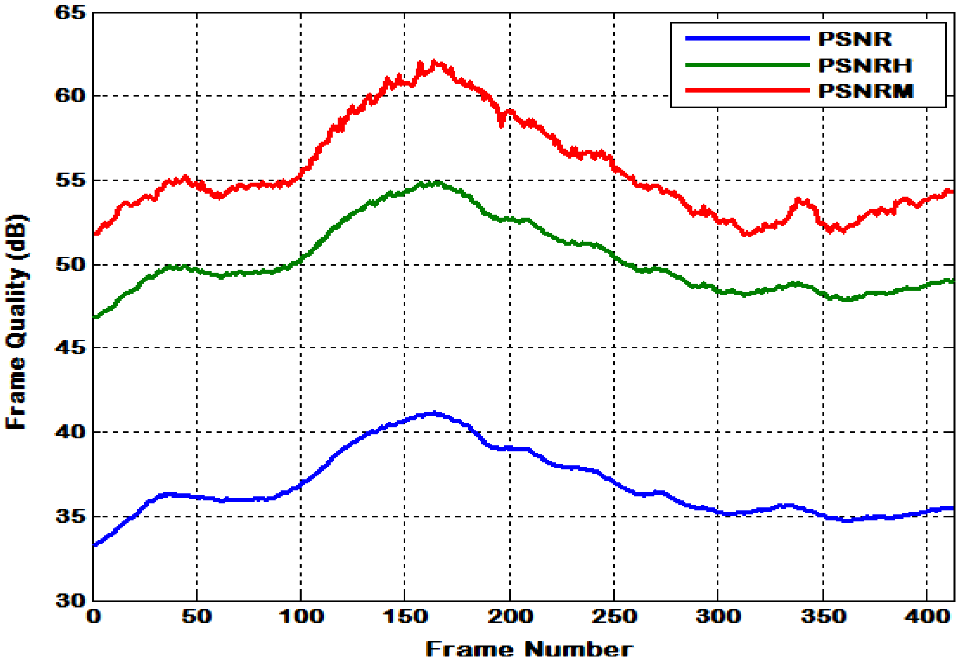


Figure 4.11 Averages of visual qualities for five videos using three LSBs.

On the other hand, according to the [123], our method has a high embedding payload. The obtained embedding capacity ratios of our algorithm, when using one LSB, two LSBs, three LSBs, and four LSBs, are 5.5%, 10.9%, 16.4%, and 21.9%, respectively. In other words, the average of hidden data in five experiments are 20.8, 41.6, 62.4, and 83.2 Megabits when using one LSB, two LSBs, three LSBs, and four LSBs, respectively.

A number of experiments were conducted that compares both embedding payload and visual quality of the proposed algorithm with other related algorithms.

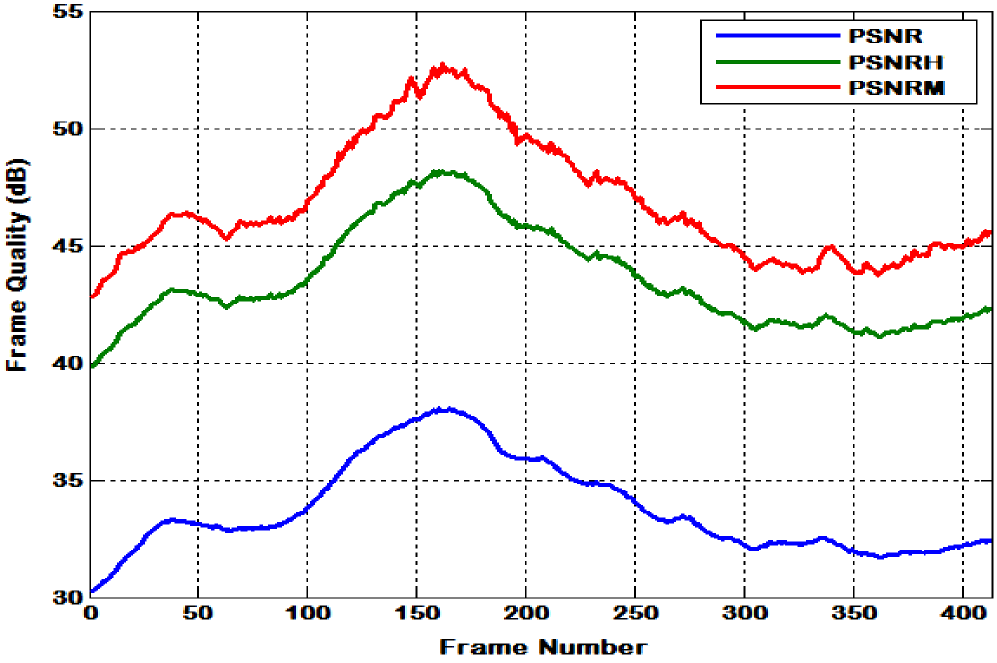


Figure 4.12 Averages of visual qualities for five experiments using four LSBs

Table 4.9 shows the comparison of data embedding ratios of our proposed algorithm with others. Figure 4.13 illustrates the comparison of average visual quality of our algorithm with other related algorithms. The results of comparison demonstrated that our algorithm outperformed the three related algorithms in the literature in both visual quality and embedding capacity. Figure 4.14 summarizes the average of the data embedding payload of five experiments for this algorithm using one LSB, two LSBs, three LSBs, and four LSBs.

Table 4.10 shows the experimental results of our algorithm and other nine algorithms based on embedding capacity, visual quality, host video preprocessing, and secret message preprocessing. The proposed algorithm clearly dominates nine related algorithms by obtaining highest values of *PSNR* and *HR*.

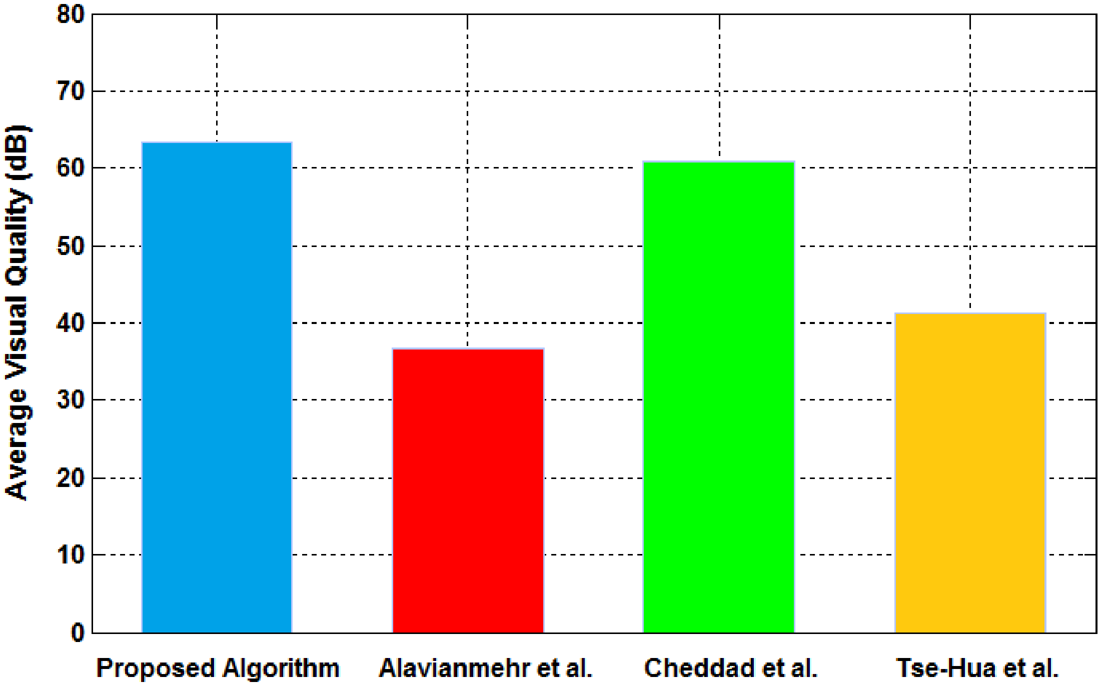


Figure 4.13 Visual quality comparison of our algorithm with Alavianmehr et al. [88], Cheddad et al. [86], and Tse-Hua et al. [123] existing algorithms.

Table 4.9 The comparison of *HR*s for the proposed algorithm with other existing algorithms.

	Proposed Algorithm	***Alavianmehr et al. [88]***	***Cheddad et al. [86]***	***Tse-Hua et al. [123]***
1 LSB	**5.5 %**	1.34 %	0.08 %	0.50 %
2 LSBs	**10.9 %**	2.68 %	0.17 %	1.00 %
3 LSBs	**16.4 %**	4.02 %	0.26 %	1.50 %
4 LSBs	**21.9 %**	5.36 %	0.34 %	2.00 %

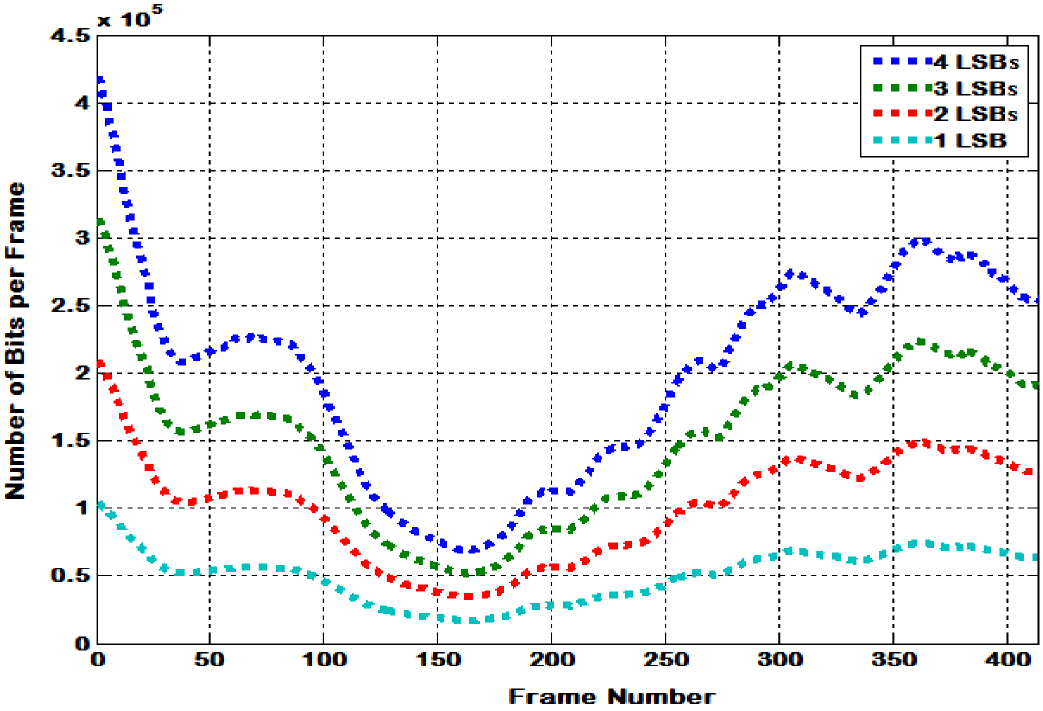


Figure 4.14 Average of the data embedding payload for five videos using each of one LSB, two LSBs, three LSBs, and four LSBs

Table 4.10 Performance comparison of the proposed algorithm with existing methods.

Method	***HR***	***PSNR***	***Video Preprocessing***	***Message Preprocessing***
Chang et al. [53]	1.04%	37.00 dB	✗	✗
Ma et al. [76]	0.10%	40.74 dB	✗	✗
Shahid et al. [70]	0.98%	43.39 dB	✗	✗
Wang et al. [83]	0.57%	37.05 dB	✗	✗
Liu et al. [79]	0.09%	46.35 dB	✗	✓
Ke et al. [80]	2.44%	34.54 dB	✗	✗
Alavianmehr et al. [88]	1.34%	36.97 dB	✗	✗
Cheddad et al. [86]	0.08%	61.22 dB	✓	✗
Sadek et al. [87]	0.23%	54.64 dB	✓	✗
Proposed algorithm	**5.50%**	**63.78 dB**	✓	✓

In order to evaluate the performance of the proposed algorithm for retrieving the secret message successfully, it is tested against various 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*). Table 4.11 illustrates the values of *BER* and *Sim* for the five experiments. Since our algorithm is applied on time domain, the *BER* and *Sim* values are reasonable but not ideal. Table 4.12 summarizes the performance of our algorithm under such attacks. The visual qualities (*PSNR*, *PSNRH*, and *PSNRM*) of the distorted videos are calculated and the visual qualities are acceptable.

Table 4.11 *BER* and *Sim* values for the five distorted videos against three attacks.

Type of Attack		***Video1***		***Video2***		***Video3***		***Video4***		***Video5***	
		Sim	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***
No attacks		**1.00**	**0**	**1.00**	**0**	**1.00**	**0**	**1.00**	**0**	**1.00**	**0**
Salt & Pepper	***D=0.01***	**0.76**	**24.5**	**0.76**	**24.2**	**0.75**	**25.1**	**0.76**	**23.9**	**0.75**	**24.8**
	D=0.001	**0.76**	**23.7**	**0.77**	**23.4**	**0.76**	**24.3**	**0.77**	**23.1**	**0.76**	**24**
Gaussian white	***V=0.01***	**0.69**	**31.4**	**0.69**	**31.1**	**0.68**	**32**	**0.69**	**30.8**	**0.68**	**31.7**
	V=0.001	**0.69**	**30.9**	**0.69**	**30.6**	**0.69**	**31.5**	**0.70**	**30.3**	**0.69**	**31.2**
Median filtering		**0.79**	**21.4**	**0.79**	**21.1**	**0.78**	**22**	**0.79**	**20.8**	**0.78**	**21.7**

Finally, the security of our steganographic algorithm has been improved. The reason for this security improvement is based upon the cover video (facial regions) being changed frame to frame. Therefore, the attackers have an extremely difficult time determining the location of the hidden message. In addition, since two secret keys have been used prior to the embedding process, attackers will be further prevented from reading the secret message. Moreover, applying the Hamming codes (15, 11) on the secret message as a part of the preprocessing stage, hackers will have additional obstacles to overcome in order to read the secret message.

Table 4.12 Visual qualities comparison for the five distorted videos against three attacks.

Visual quality	*Type of Attack*		*Video1*	*Video2*	*Video3*	*Video4*	*Video5*
PSNR	***Impulsive***	***D=0.01***	**23.17**	**23.62**	**22.32**	**24.02**	**22.72**
		D=0.001	**31.20**	**31.65**	**30.35**	**32.05**	**30.75**
	Gaussian white	***V=0.01***	**18.95**	**19.40**	**18.10**	**19.80**	**18.50**
		V=0.001	**28.87**	**29.32**	**28.02**	**29.72**	**28.42**
	Median filtering		**25.26**	**25.71**	**24.41**	**26.11**	**24.81**
PSNRH	***Impulsive***	***D=0.01***	**35.23**	**35.68**	**34.38**	**36.08**	**34.78**
		D=0.001	**45.27**	**45.72**	**44.42**	**46.12**	**44.82**
	Gaussian white	***V=0.01***	**30.01**	**30.46**	**29.16**	**30.86**	**29.56**
		V=0.001	**39.93**	**40.38**	**39.08**	**40.78**	**39.48**
	Median filtering		**34.47**	**34.92**	**33.62**	**35.32**	**34.02**
PSNRM	***Impulsive***	***D=0.01***	**38.06**	**38.51**	**37.21**	**38.91**	**37.61**
		D=0.001	**48.17**	**48.62**	**47.32**	**49.02**	**47.72**
	Gaussian white	***V=0.01***	**33.37**	**33.82**	**32.52**	**34.22**	**32.92**
		V=0.001	**45.08**	**45.53**	**44.23**	**45.93**	**44.63**
	Median filtering		**38.99**	**39.44**	**38.14**	**39.84**	**38.54**

4.4 A Robust Video Steganography Algorithm in the Wavelet Domain Based on KLT Tracking and ECC

This section presents experimental results that are implemented by the MATLAB software version R2013a. The same dataset as in Section **4.3** is used. *PSNR*, *PSNRH*, and *PSNRM* metrics are applied to measure the quality of the stego videos. Figure 4.15 summarizes the visual quality comparison between the *PSNR*, *PSNRH*, and *PSNRM* metrics totaling the average of the five experiments. Here, the averages of each *PSNR*, *PSNRH*, and *PSNRM* for the five experiments equal 56.90, 65.84, and 76.44 dBs, respectively. Overall, due to the high values of *PSNR*, *PSNRH*, and *PSNRM*, the proposed method has visual qualities the same as the original videos' visual qualities. The *PSNRM* metric has improved the visual quality of all five stego videos better than the other two metrics.

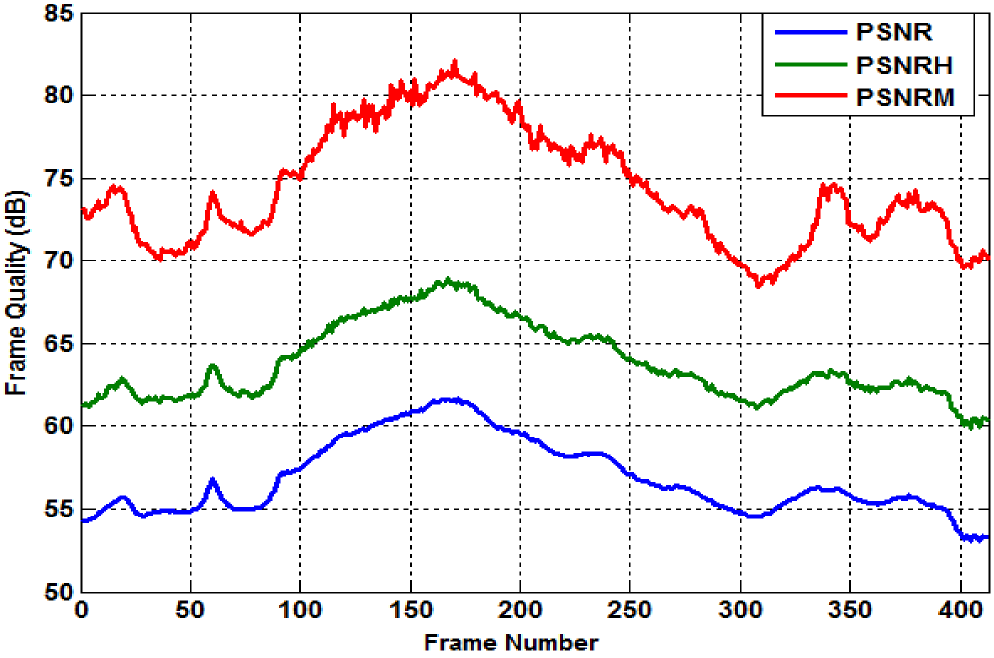


Figure 4.15 The PSNR, PSNRH, and PSNRM comparison for the average of the five experiments.

In addition, the proposed algorithm has a high embedding payload according to the reference [123]. The average of obtained hiding ratios for five videos is 4.1%. Moreover, the average amount of the embedded secret message in experiments is 15.60 Megabits. Figure 4.16 illustrates the average of the data embedding payload of five experiments for our steganographic algorithm.

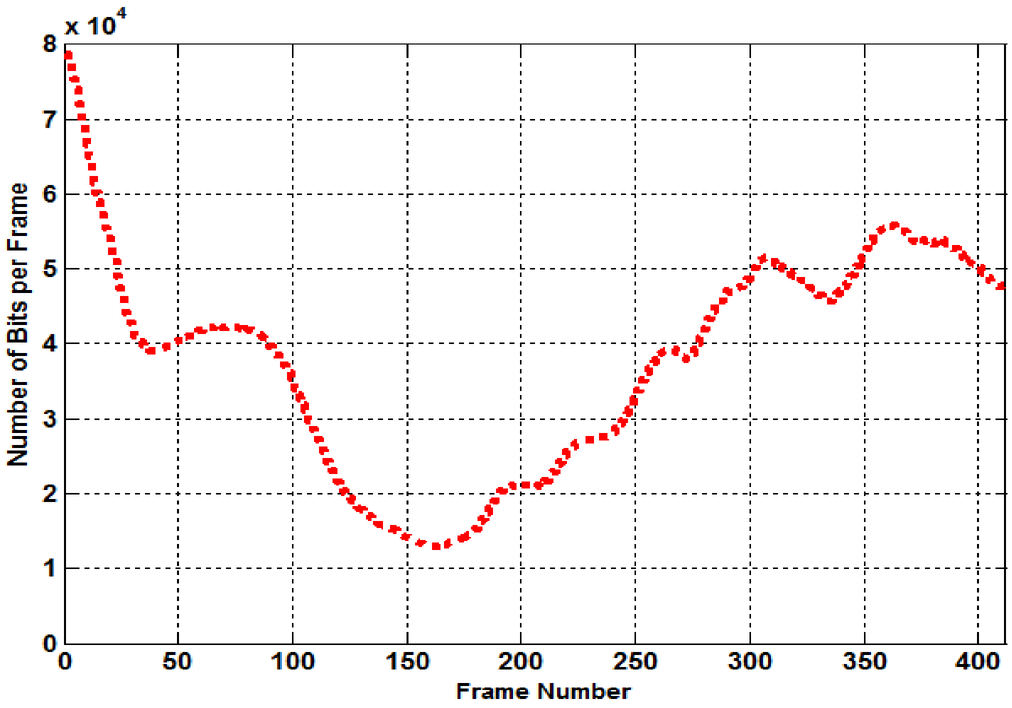


Figure 4.16 Average of the data embedding payload of the five experiments.

In contrast to the previous algorithm, mentioned in Section **4.3**, this algorithm withstands against attacks because it operates in the transform domain. To assess the robustness of this algorithm against attacks, we have conducted the same experiments as mentioned in Section **4.3** (*Gaussian noise* with the zero mean and *variance*=0.01 and 0.001, *Salt & Peppe*r noise with the *density*=0.01 and 0.001, and *Median Filtering*). Table 4.13 illustrates the values of *BER* and *Sim* for the five experiments, which a low *BER* and a high *Sim* are achieved. Table 4.14 summarizes the performance of our algorithm under such attacks. The visual qualities (*PSNR*, *PSNRH*, and *PSNRM*) of the distorted videos are calculated and the visual qualities are reasonable.

Table 4.13 *BER* and *Sim* values for the five distorted videos against three attacks.

Type of Attack		***Video1***		***Video2***		***Video3***		***Video4***		***Video5***	
		Sim	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***	***Sim***	***BER %***
No attacks		**1.00**	**0**	**1.00**	**0**	**1.00**	**0**	**1.00**	**0**	**1.00**	**0**
Salt & Pepper	***D=0.01***	**0.946**	**5.4**	**0.956**	**4.4**	**0.941**	**5.9**	**0.952**	**4.8**	**0.943**	**5.7**
	D=0.001	**0.987**	**1.3**	**0.989**	**1.1**	**0.979**	**2.1**	**0.993**	**0.7**	**0.984**	**1.6**
Gaussian white	***V=0.01***	**0.912**	**8.8**	**0.931**	**6.9**	**0.909**	**9.1**	**0.918**	**8.2**	**0.909**	**9.1**
	V=0.001	**0.968**	**3.2**	**0.979**	**2.1**	**0.959**	**4.1**	**0.974**	**2.6**	**0.965**	**3.5**
Median filtering		**0.965**	**3.5**	**0.974**	**2.6**	**0.961**	**3.9**	**0.971**	**2.9**	**0.962**	**3.8**

Table 4.14 Visual qualities comparison for the five distorted videos against three attacks.

Visual Quality	***Type of Attack***		***Video1***	***Video2***	***Video3***	***Video4***	***Video5***
PSNR	***Impulsive***	***D=0.01***	**32.87**	**33.32**	**32.02**	**33.72**	**32.42**
		D=0.001	**37.70**	**38.15**	**36.85**	**38.55**	**37.25**
	Gaussian white	***V=0.01***	**31.45**	**31.90**	**30.60**	**32.30**	**31.00**
		V=0.001	**35.37**	**35.82**	**34.52**	**36.22**	**34.92**
	Median filtering		**35.56**	**36.01**	**34.71**	**36.41**	**35.11**
PSNRH	***Impulsive***	***D=0.01***	**44.93**	**45.38**	**44.08**	**45.78**	**44.48**
		D=0.001	**51.77**	**52.22**	**50.92**	**52.62**	**51.32**
	Gaussian white	***V=0.01***	**42.51**	**42.96**	**41.66**	**43.36**	**42.06**
		V=0.001	**46.43**	**46.88**	**45.58**	**47.28**	**45.98**
	Median filtering		**44.77**	**45.22**	**43.92**	**45.62**	**44.32**
PSNRM	***Impulsive***	***D=0.01***	**47.76**	**48.21**	**46.91**	**48.61**	**47.31**
		D=0.001	**54.67**	**55.12**	**53.82**	**55.52**	**54.22**
	Gaussian white	***V=0.01***	**45.87**	**46.32**	**45.02**	**46.72**	**45.42**
		V=0.001	**51.58**	**52.03**	**50.73**	**52.43**	**51.13**
	Median filtering		**49.29**	**49.74**	**48.44**	**50.14**	**48.84**

4.5 A New Video Steganography Algorithm Based on the Multiple Object Tracking and ECC

Three *S2L1* video sequences of different views (*View$_1$*, *View$_3$*, and *View$_4$*) were used from the well-known *PETS2009* dataset [118]. The implemented videos contain moving objects which are taken by different stationary cameras. Experimental results are obtained by using the R2013a version of the MATLAB software program. The videos contain a 768 x 576 resolution at 30 frames per second, and a data rate of 12684 kbps. Each cover video sequence contains 795 frames. In all the video frames, the secret message appears as a large text file split in accordance with the size and number of the moving objects.

The visual quality of the proposed algorithm is measured by applying the *PSNR* metric. Figure 4.17 shows the *PSNR* comparison of the first video (*View$_1$*) when using one LSB and two LSBs of each motion object's RGB pixels. Here, the *PSNR* values equal 47.73 dB for one LSB and 40.45 dB for two LSBs. Figure 4.18 illustrates the *PSNR* comparison of the *View$_3$* experiment when using one LSB and two LSBs of each motion pixel in the video frames. The *PSNR* values equal 50.93 and 43.88 dBs for one LSB and two LSBs, respectively. Figure 4.19 shows the *PSNR* comparison of the *View$_4$* video when using one LSB and two LSBs of each motion object's RGB pixels. Here, the *PSNR* values equal 51.35 dB for one LSB and 44.16 dB for two LSBs. The third experiment (*View$_4$*) has better visual quality among other experiments because it has fewer regions of moving objects than others. This means that *View$_4$* video can embed less size of the secret data than the other two experiments. Overall, the stego videos' visual qualities are close to the original videos' visual qualities due to the high values of *PNSR*s for our proposed algorithm.

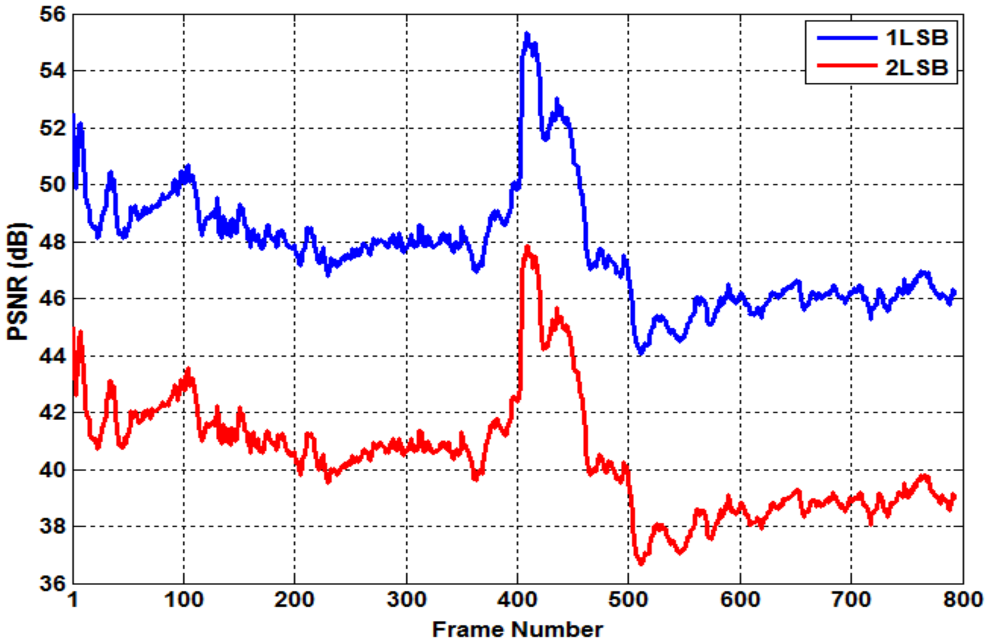


Figure 4.17 The PSNR comparison of the *View1* experiment.

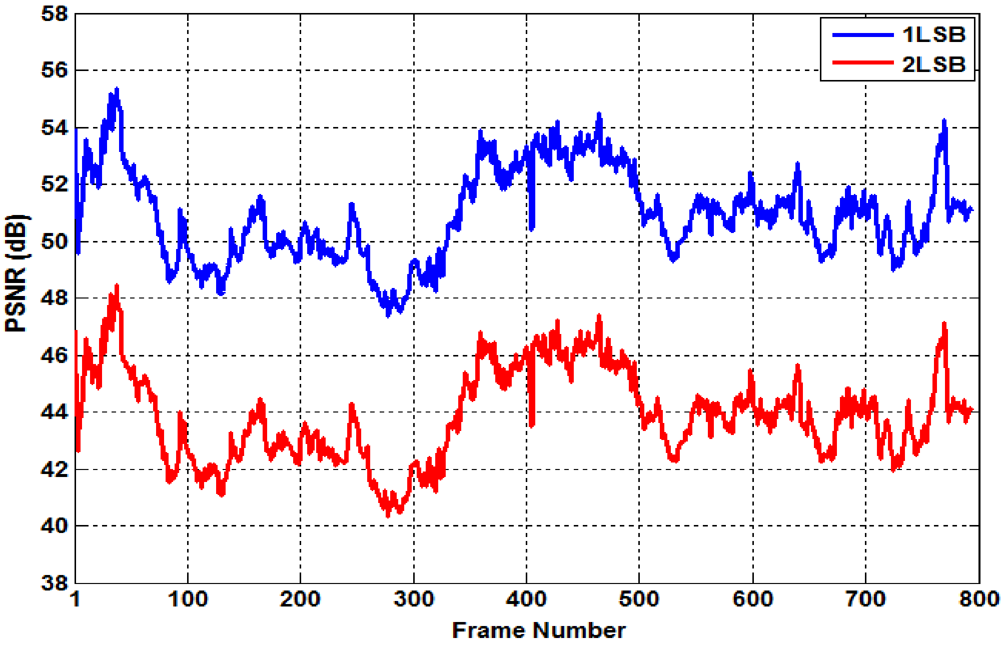


Figure 4.18 The PSNR comparison of the *View3* video.

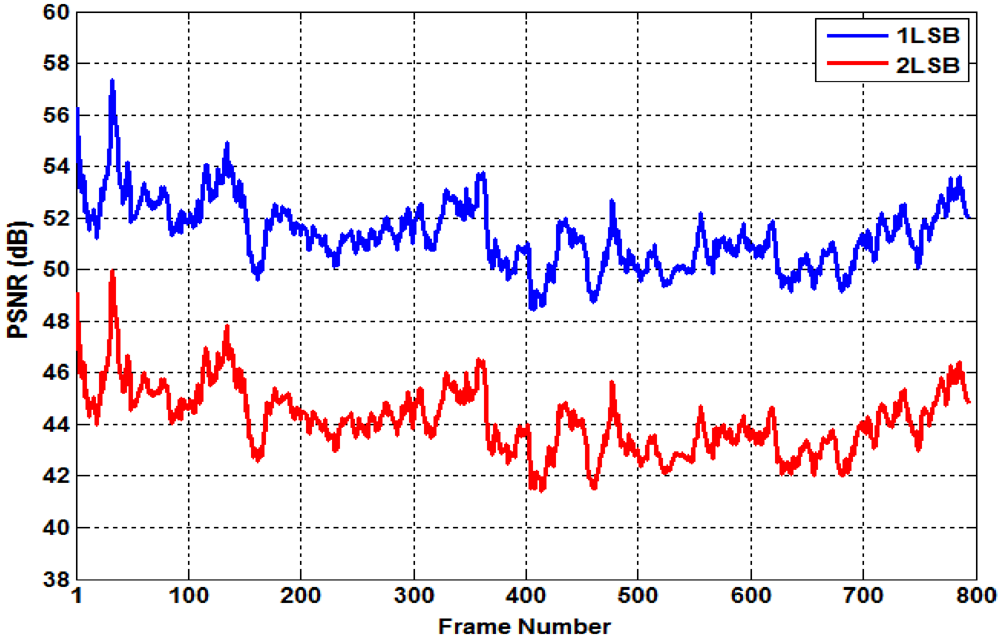


Figure 4.19 The PSNR comparison of the *View4* experiment.

On the other hand, according to the reference [123], our proposed algorithm has a high embedding payload. Here, the average of obtained hiding ratios for three experiments is 3.37%. The size of the hidden secret message in each *View$_1$*, *View$_3$*, and *View$_4$* videos using one LSB is 31.38, 14.62, and 12.95 Megabits, respectively. Moreover, when using two LSBs, the amount of the secret message in each *View$_1$*, *View$_3$*, and *View$_4$* experiments will be 62.77, 29.25, and 25.92 Megabits, respectively. Figure 4.20, 4.21, and 4.22 illustrate the data embedding payload of the proposed steganography algorithm for each *View$_1$*, *View$_3$*, and *View$_4$* experiments, respectively. These three figures have shown the comparison of the embedding capacity of each video when one LSB and two LSBs of the moving objects' pixels are utilized. The two LSBs were implemented in order to double the amount of the secret message in each experiment. Table 4.15 illustrates the experimental results of our fifth algorithm with other algorithms based on embedding capacity, visual quality, cover video preprocessing, and secret message preprocessing.

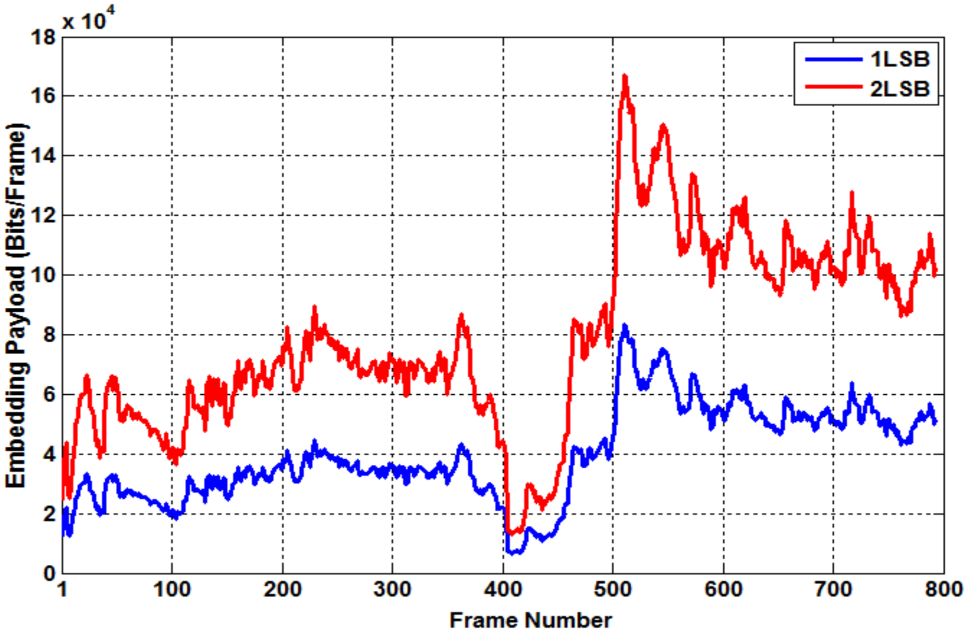


Figure 4.20 The embedding payload comparison of the *View1* experiment.

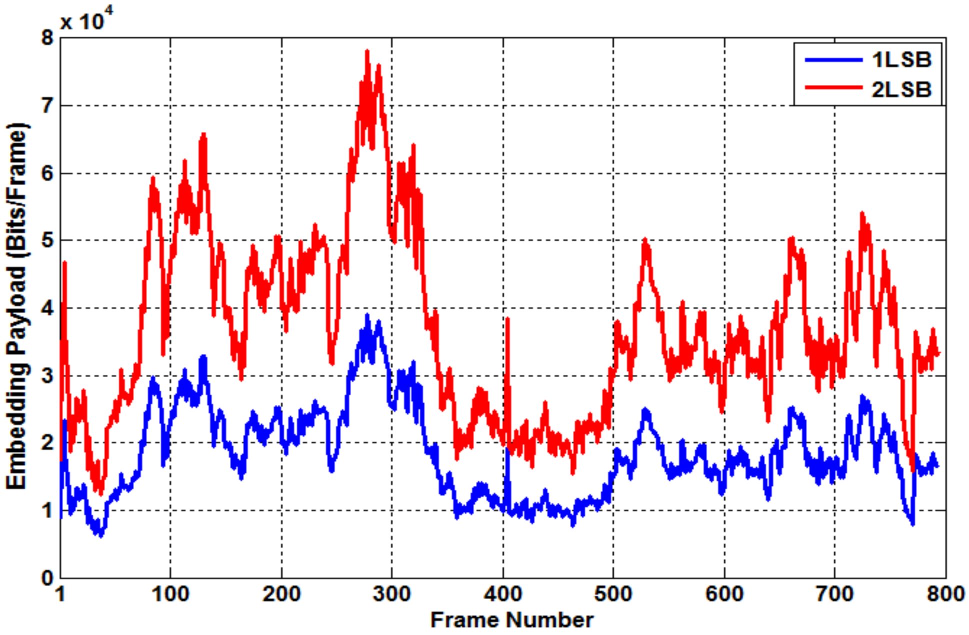


Figure 4.21 The embedding payload comparison of the *View3* video.

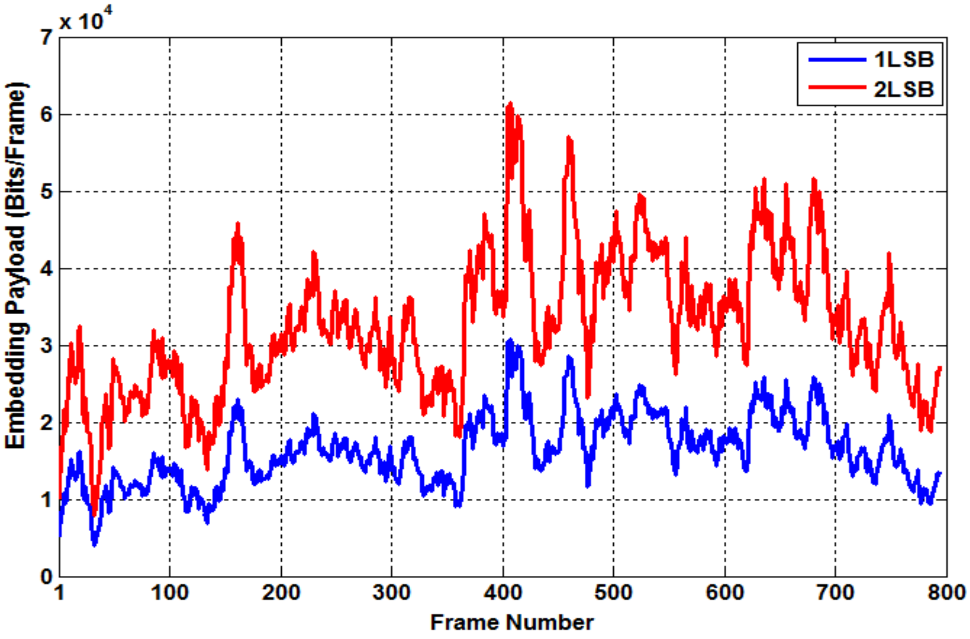


Figure 4.22 The embedding payload comparison of the *View4* experiment.

Table 4.15 Performance comparison of the proposed method with other existing methods.

Method	***HR***	***PSNR***	***Video Preprocessing***	***Message Preprocessing***
Chang et al. [53]	1.04%	37.00 dB	✗	✗
Ma et al. [76]	0.10%	40.74 dB	✗	✗
Shahid et al. [70]	0.98%	43.39 dB	✗	✗
Wang et al. [83]	0.57%	37.05 dB	✗	✗
Liu et al. [79]	0.09%	46.35 dB	✗	✓
Ke et al. [80]	2.44%	34.54 dB	✗	✗
Alavianmehr et al. [88]	1.34%	36.97 dB	✗	✗
Proposed algorithm	**3.37%**	**50.01 dB**	✓	✓

4.6 A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC

A *S2L1* video sequence was used from the well-known PETS2009 dataset [118]. The proposed algorithm results are achieved using MATLAB implementation of the algorithm. The cover video consists of a 768x576 video dimension at 30 frames/sec, and a 12684 kbps data rate. The video sequence also includes 795 frames; each frame has multiple moving objects. In the entire video frames, the text messages appear as a sizeable file divided based on the number and size of the moving objects.

The imperceptibility of our proposed algorithm is measured by utilizing *PSNR*. Figure 4.23 illustrates the *PSNR* comparison of the experiment video when using one LSB, two LSBs, and three LSBs of each motion object's DWT coefficients, including each of LL, LH, HL, and HH subband. The *PSNR* values equal **49.01**, **42.70**, and **36.41** dBs when using one LSB, two LSBs, and three LSBs of each coefficient, respectively. Figure 4.24 illustrates the *PSNR* comparison of the tested video when using one LSB, two LSBs, and three LSBs of each motion object's DCT coefficients, including both DCs and ACs. Here, the *PSNR* values equal **48.67**, **41.45**, and **35.95** dBs for each one LSB, two LSBs, and three LSBs, respectively.

Figure 4.25 illustrates the original and stego 574th frame of the tested video along with histograms of their RGB components. The histograms show no obvious alteration in the video quality. Table 4.16 clarifies the average of visual qualities based on DWT and DCT domains. Overall, the embedded videos' qualities are near to the host videos' qualities because of the high values of *PNSR*s for our proposed algorithm.

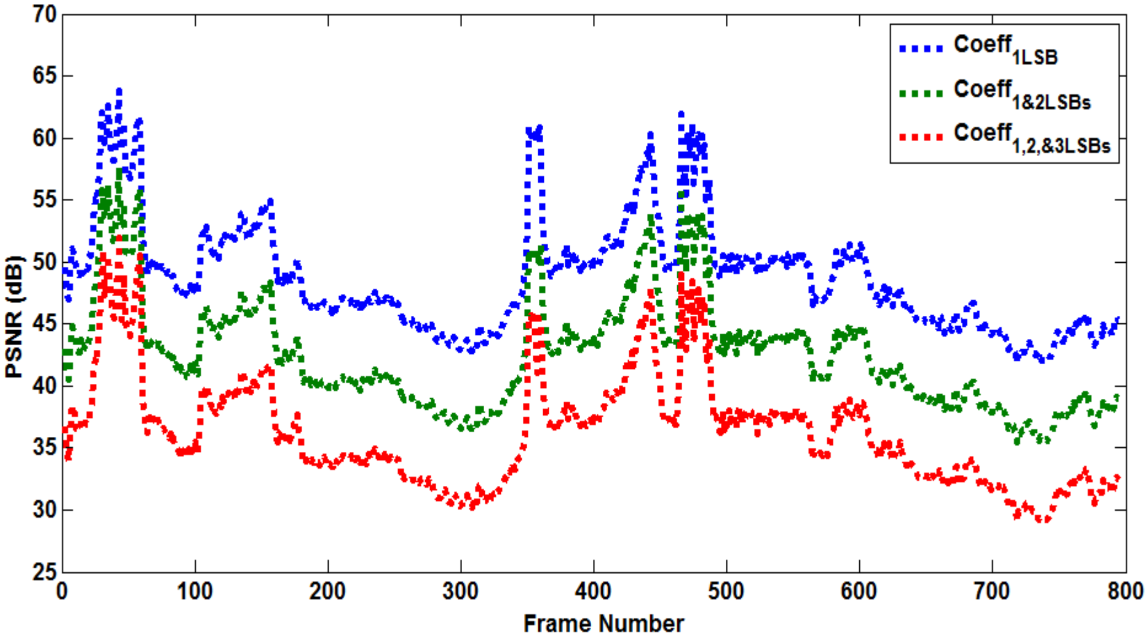


Figure 4.23 The PSNR comparison of the experiment video in DWT domain.

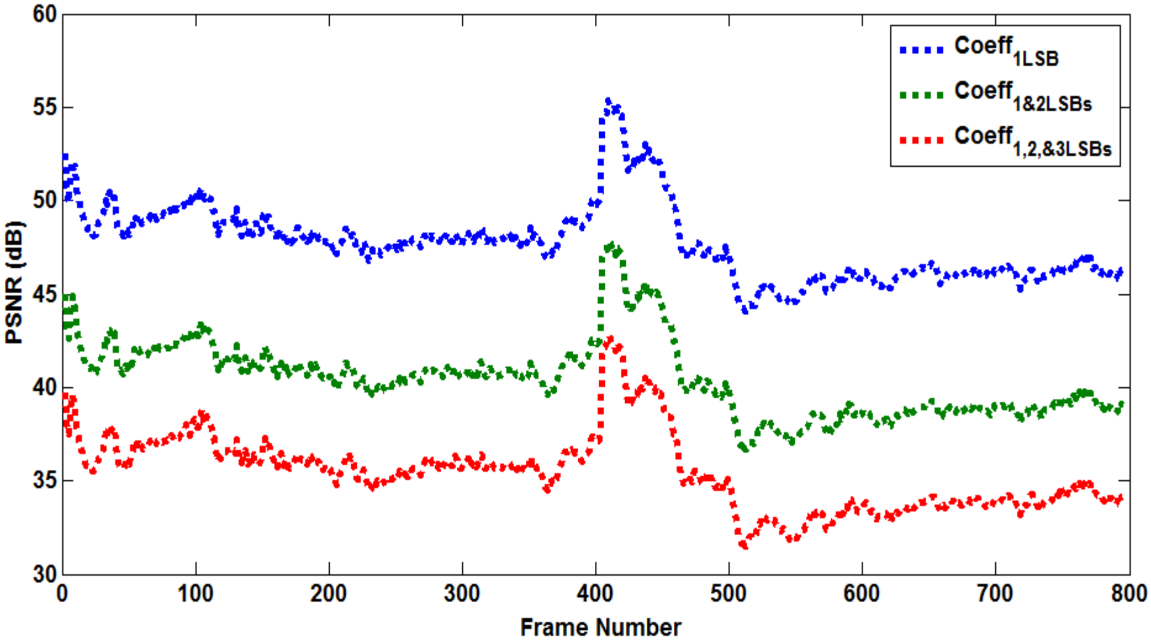


Figure 4.24 The PSNR comparison of the tested video in DCT domain.

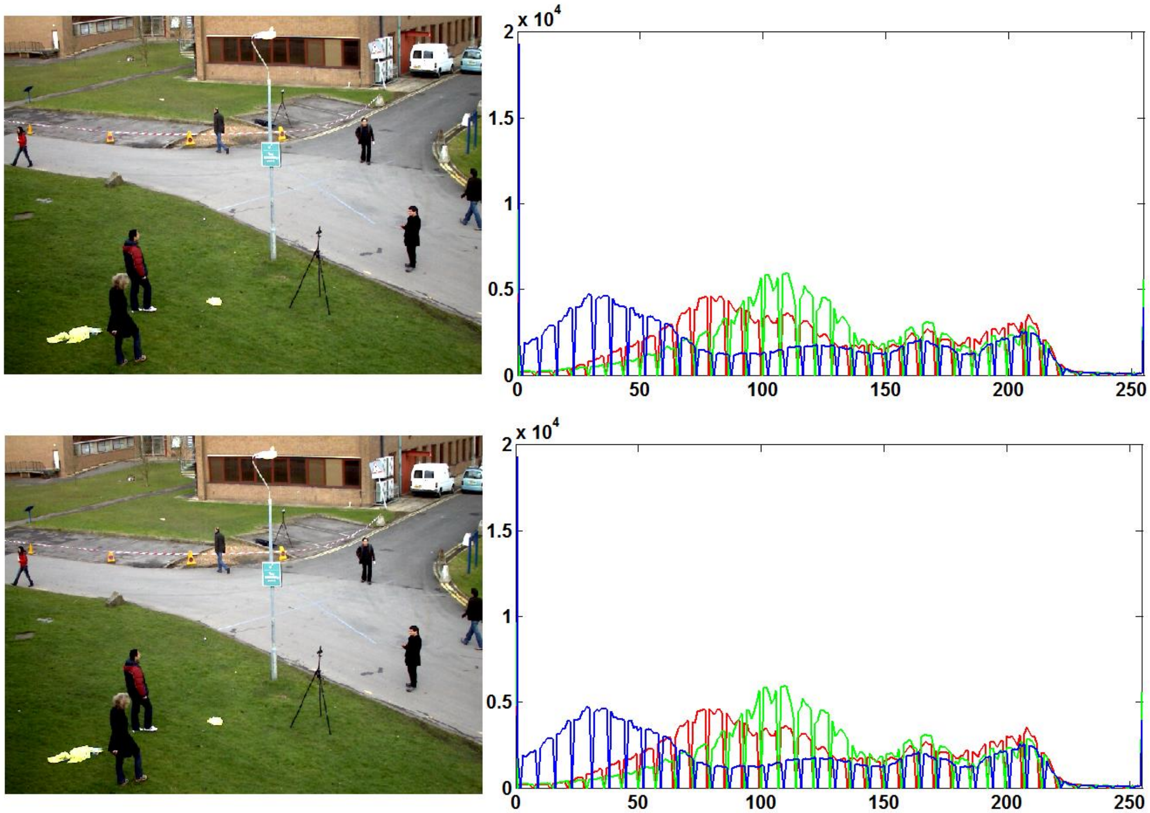


Figure 4.25 Visual quality assessment: The first line illustrates the original 574th frame of the tested video along with histograms of its RGB channels. The second line shows the stego 574th frame of the tested video and histograms of RGB channels after embedding stage.

Table 4.16 Average PSNR each of R, G, and B component of the experiment video after applying DWT and DCT transform domains.

Transform Domains	***No. of coefficients***	***PSNR_R***	***PSNR_G***	***PSNR_B***	***PSNR***
DWT (LL, LH, HL, and HH)	$Coff_{1LSB}$	49.39	49.16	48.49	**49.01**
	$Coff_{1\&2LSBs}$	43.08	42.85	42.18	**42.70**
	$Coff_{1,\ 2,\ \&3LSBs}$	36.79	36.56	35.89	**36.41**
DCT (DC and AC)	$Coff_{1LSB}$	48.84	48.74	48.44	**48.67**
	$Coff_{1\&2LSBs}$	41.62	41.52	41.22	**41.45**
	$Coff_{1,\ 2,\ \&3LSBs}$	36.12	36.02	35.72	**35.95**

According to [123], our proposed algorithm has a high embedding capacity. Here, the average of the gained hiding ratio is **3.40**% when our algorithm operates in DWT domain. This average has increased to **3.46**% when the proposed algorithm operates in DCT domain. The average sizes of secret messages in both domains are **31.38**, **62.77** and **94.15** Megabits when using one LSB, two LSBs, and three LSBs of DWT and DCT coefficients, respectively. Figure 4.26 illustrates the data embedding capacity of the proposed steganography algorithm when using each of DWT and DCT domains. The figure has shown the comparison of the embedding capacity of the tested video when one LSB, two LSBs, and three LSBs of the moving objects' DWT and DCT coefficients are utilized separately. Table 4.17 shows that our proposed algorithm outperforms other existing methods.

Table 4.17 Performance comparison of the proposed method with other existing methods.

Method		***HR***	***PSNR***	***Video Preprocessing***	***Message Preprocessing***
Chang et al. [53]		1.04%	37.00 dB	✗	✗
Ma et al. [76]		0.10%	40.74 dB	✗	✗
Shahid et al. [70]		0.98%	43.39 dB	✗	✗
Wang et al. [83]		0.57%	37.05 dB	✗	✗
Liu et al. [79]		0.09%	46.35 dB	✗	✓
Ke et al. [80]		2.44%	34.54 dB	✗	✗
Alavianmehr et al. [88]		1.34%	36.97 dB	✗	✗
Proposed algorithm	***DWT***	**3.40%**	**49.01 dB**	✓	✓
	DCT	**3.46%**	**48.67 dB**	✓	✓

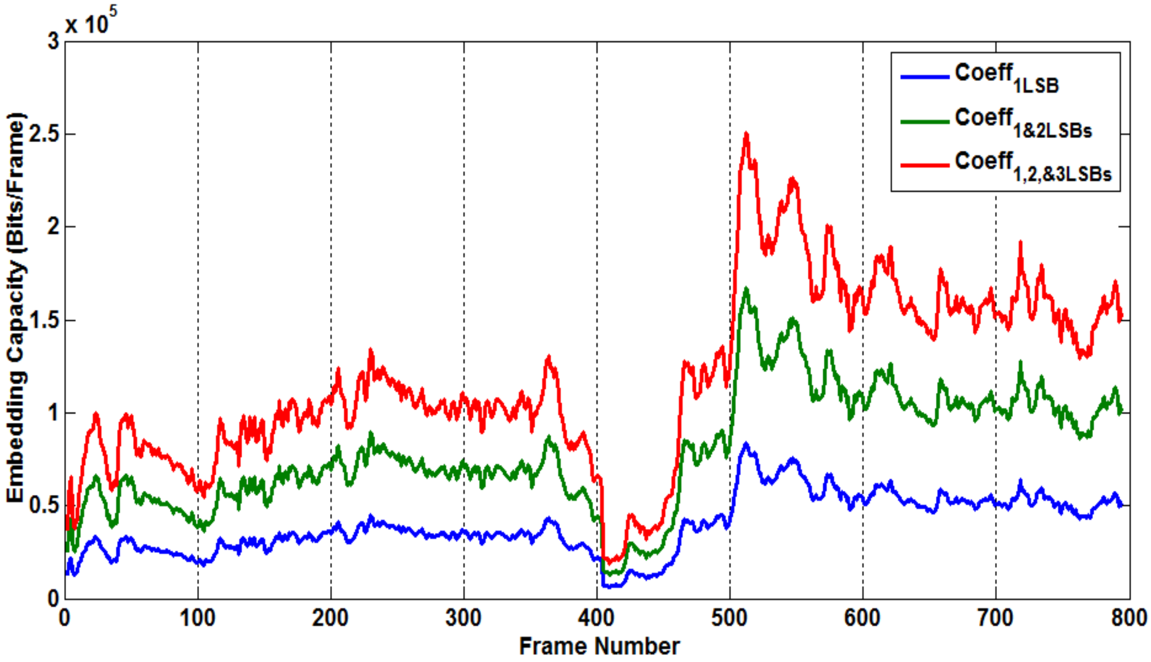


Figure 4.26 The embedding capacity comparison of the experiment in DWT-DCT domains.

To measure the robustness of the proposed algorithm, the *Sim* and *BER* metrics have been utilized. 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* and lower *BER* must be obtained. Table 4.18 illustrates the robustness of the proposed method under various attacks.

4.7 Additional Experiments

Table 4.19 shows some additional experiments that have been conducted in order to validate the efficiency of our six video steganography algorithms. Also, it illustrates the reasonable tradeoff between *PSNR* and *HR* of our proposed algorithms when compared to the existing algorithms.

Table 4.18 *Sim* and *BER* values of our method under various attacks.

Type of Attack		*DWT domain*			*DCT domain*		
		PSNR (dB)	*BER %*	*Sim*	*PSNR (dB)*	*BER %*	*Sim*
No attacks		49.01	0	1	48.67	0	1
Salt & Pepper	***D=0.01***	34.37	6.5	0.935	33.77	8.3	0.917
	D=0.001	39.2	2.4	0.976	38.6	4.2	0.958
Gaussian white	***V=0.01***	32.95	9.9	0.901	32.35	11.7	0.883
	V=0.001	36.87	4.3	0.957	36.27	6.1	0.939
Median filtering		37.06	4.6	0.954	36.46	6.4	0.936

Table 4.19 Additional experiments to validate the efficiency of our algorithms.

Algorithm Number		***Our Proposed Results***	***Additional Experiments***			***Average Time/Sec***
Algorithm1	***HR***	**7.39%**	8.50%	9.61%	**10.72%**	**201.69**
	PSNR	**51.75 dB**	43.99 dB	36.23 dB	**28.46 dB**	
Algorithm2	***HR***	**28.12%**	32.34%	**36.56%**	40.77%	**212.63**
	PSNR	**41.58 dB**	35.34 dB	**29.11 dB**	22.87 dB	
Algorithm3	***HR***	**5.50%**	7.15%	7.98%	**8.80%**	**311.196**
	PSNR	**63.78 dB**	44.65 dB	35.08 dB	**25.51 dB**	
Algorithm4	***HR***	**4.10%**	5.33%	5.95%	**6.56%**	**401.225**
	PSNR	**66.39 dB**	46.47 dB	36.51 dB	**26.56 dB**	
Algorithm5	***HR***	**3.37%**	4.38%	4.89%	**5.39%**	**544.286**
	PSNR	**50.01 dB**	42.51 dB	35.01 dB	**27.51 dB**	
Algorithm6	***HR***	**3.43%**	4.46%	4.97%	**5.49%**	**602.779**
	PSNR	**48.84 dB**	41.51 dB	34.19 dB	**26.86 dB**	

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

The main objective of this dissertation is to enhance video steganography methods. Hence, six new and efficient algorithms are proposed to maintain a reasonable trade-off between imperceptibility, hiding capacity, and robustness against various attacks.

First, we proposed a highly secure video steganography algorithm based on ECC. In this algorithm, we combined steganography concepts with other system protection methods such as cryptography and ECC. Thus, encrypting and encoding the secret message prior to the embedding process provided an additional security level to the secret message and made it more robust against attacks during the transmission process.

Then, an increased payload video steganography algorithm in the discrete wavelet domain based on ECC was proposed. This algorithm enhanced the hiding capacity of the secret message as compared to other algorithms reported in the literature.

After that, we proposed a novel video steganography algorithm based on KLT tracking and ECC. This algorithm focused only on the facial regions of the video as a host data for embedding the secret message instead of using the entire video. These methods enhanced imperceptibility. Furthermore, it will be challenging for unauthorized users and intruders to define the position of hidden data in each video frame since the hidden data is embedded into the ROI which changes from frame to frame, thus maintaining the security of hidden message.

Next, we proposed a robust video steganography algorithm in the wavelet domain based on the KLT tracking and ECC. The robustness against attacks of this algorithm was enhanced due to the use of wavelet coefficients of facial regions as cover data to embed the secret message.

Later, a new video steganography algorithm based on the MOT and ECC was proposed. This method used multiple motion objects throughout the video frames as regions of interest to conceal the secret message, thus improving each of imperceptibility and embedding capacity.

Finally, we proposed a robust and secure video steganography algorithm in DWT-DCT domains based on MOT and ECC. Each of DWT and DCT frequency coefficients of moving objects are used as cover data to embed the secret message. The security and robustness of this method enhanced as compared to other methods reported in the literature.

Our experimental results demonstrate that the proposed algorithms achieve higher embedding capacity as well as better visual quality of stego videos. Furthermore, the preprocessing steps increase the security and robustness of the proposed algorithms when compared to state-of-the-art methods.

5.2 Future Directions

In continuation of this research, it is planned to propose a real-time video steganography algorithm based on multiple object tracking and ECC. Such algorithm will use multiple moving objects in security cameras or video surveillance systems as regions of interest to embed the secret information.

It is also planned to apply our proposed algorithms in some other frequency domains such as curvelet transform for further improving the efficiency, visual quality, and security.

In addition, it is planned to consider different types of error correcting codes in our proposed video steganography methods for further enhancing the robustness against signal processing operations and various attacks.

REFERENCES

[1] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," *Signal Processing*, vol. 89, pp. 2324-2332, 2009.

[2] X.-y. Wang, C.-p. Wang, H.-y. Yang, and P.-p. Niu, "A robust blind color image watermarking in quaternion Fourier transform domain," *Journal of Systems and Software*, vol. 86, pp. 255-277, 2013.

[3] M. Masoumi and S. Amiri, "A blind scene-based watermarking for video copyright protection," *AEU - International Journal of Electronics and Communications*, vol. 67, pp. 528-535, 2013.

[4] Z. Qian, G. Feng, X. Zhang, and S. Wang, "Image self-embedding with high-quality restoration capability," *Digital Signal Processing*, vol. 21, pp. 278-286, 2011.

[5] F. Lusson, K. Bailey, M. Leeney, and K. Curran, "A novel approach to digital watermarking, exploiting colour spaces," *Signal Processing*, vol. 93, pp. 1268-1294, 2013.

[6] M. S. Subhedar and V. H. Mankar, "Current status and key issues in image steganography: A survey," *Computer Science Review*, vol. 13–14, pp. 95-113, 2014.

[7] S. Islam, M. R. Modi, and P. Gupta, "Edge-based image steganography," *EURASIP Journal on Information Security*, vol. 2014, pp. 1-14, 2014.

[8] M. Hasnaoui and M. Mitrea, "Multi-symbol QIM video watermarking," *Signal Processing: Image Communication*, vol. 29, pp. 107-127, 2014.

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

[10] K. Qazanfari and R. Safabakhsh, "A new steganography method which preserves histogram: Generalization of LSB++," *Information Sciences,* vol. 277, pp. 90-101, 2014.

[11] L. Guangjie, L. Weiwei, D. Yuewei, and L. Shiguo, "Adaptive Steganography Based on Syndrome-Trellis Codes and Local Complexity," in *Fourth International Conference on Multimedia Information Networking and Security (MINES)*, pp. 323-327, 2012.

[12] C. Rupa, "A Digital Image Steganography using Sierpinski Gasket Fractal and PLSB," *Journal of The Institution of Engineers (India): Series B,* vol. 94, pp. 147-151, 2013.

[13] M. M. Sadek, A. S. Khalifa, and M. G. M. Mostafa, "Video steganography: a comprehensive review," Multimedia Tools and Applications, vol. 74, pp. 7063-7094, 2015.

[14] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing,* vol. 90, pp. 727-752, 2010.

[15] R. Zhang, V. Sachnev, and H. Kim, "Fast BCH Syndrome Coding for Steganography," in *Information Hiding*. S. Katzenbeisser and A.-R. Sadeghi, Eds., ed: Springer Berlin Heidelberg, vol. 5806, pp. 48-58, 2009.

[16] C. Fontaine and F. Galand, "How Can Reed-Solomon Codes Improve Steganographic Schemes?," in *Information Hiding*. T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds., ed: Springer Berlin Heidelberg, vol. 4567, pp. 130-144, 2007.

[17] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences,* vol. 279, pp. 251-272, 2014.

[18] T. Yiqi and W. KokSheik, "An Overview of Information Hiding in H.264/AVC Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 24, pp. 305-319, 2014.

[19] R. J. Mstafa and K. M. Elleithy, "A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11)," in *Wireless Telecommunications Symposium (WTS)*, pp. 1-8, 2015.

[20] W. Abu-Marie, A. Gutub, and H. Abu-Mansour, "Image Based Steganography Using Truth Table Based and Determinate Array on RGB Indicator," *International Journal of Signal and Image Processing,* vol. 1, pp. 196-204, 2010.

[21] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman Encoding," in *Emerging Trends and Applications in Computer Science (NCETACS), 3rd National Conference on*, pp. 14-18, 2012.

[22] A. Khan and S. A. Malik, "A high capacity reversible watermarking approach for authenticating images: Exploiting down-sampling, histogram processing, and block selection," *Information Sciences,* vol. 256, pp. 162-183, 2014.

[23] D. Rosiyadi, S.-J. Horng, P. Fan, X. Wang, M. K. Khan, and Y. Pan, "Copyright protection for e-government document images," *MultiMedia, IEEE,* vol. 19, pp. 62-73, 2012.

[24] W.-H. Lin, S.-J. Horng, T.-W. Kao, R.-J. Chen, Y.-H. Chen, C.-L. Lee, and T. Terano, "Image copyright protection with forward error correction," *Expert systems with applications,* vol. 36, pp. 11888-11894, 2009.

[25] S.-J. Horng, D. Rosiyadi, T. Li, T. Takao, M. Guo, and M. K. Khan, "A blind image copyright protection scheme for e-government," *Journal of Visual Communication and Image Representation,* vol. 24, pp. 1099-1105, 2013.

[26] S.-J. Horng, D. Rosiyadi, P. Fan, X. Wang, and M. K. Khan, "An adaptive watermarking scheme for e-government document images," *Multimedia Tools and Applications,* vol. 72, pp. 3085-3103, 2014.

[27] D. Rosiyadi, S.-J. Horng, N. Suryana, and N. Masthurah, "A comparison between the hybrid using genetic algorithm and the pure hybrid watermarking scheme," *Int J Comput Theory and Eng (IJCTE),* vol. 4, pp. 329-331, 2012.

[28] P. C. Ritchey and V. J. Rego, "A context sensitive tiling system for information hiding," *J Inf Hiding and Multimed Sig Process,* vol. 3, pp. 212-226, 2012.

[29] W.-H. Lin, Y.-R. Wang, S.-J. Horng, T.-W. Kao, and Y. Pan, "A blind watermarking method using maximum wavelet coefficient quantization," *Expert systems with applications,* vol. 36, pp. 11509-11516, 2009.

[30] W.-H. Lin, Y.-R. Wang, and S.-J. Horng, "A wavelet-tree-based watermarking method using distance vector of binary cluster," *Expert systems with applications,* vol. 36, pp. 9869-9878, 2009.

[31] W.-H. Lin, S.-J. Horng, T.-W. Kao, P. Fan, C.-L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Transactions on Multimedia*, vol. 10, pp. 746-757, 2008.

[32] H.-C. Huang, S.-C. Chu, J.-S. Pan, C.-Y. Huang, and B.-Y. Liao, "Tabu search based multi-watermarks embedding algorithm with multiple description coding," *Information Sciences,* vol. 181, pp. 3379-3396, 2011.

[33] F.-C. Chang, H.-C. Huang, and H.-M. Hang, "Layered access control schemes on watermarked scalable media," *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology,* vol. 49, pp. 443-455, 2007.

[34] A. Khan, S. A. Malik, A. Ali, R. Chamlawi, M. Hussain, M. T. Mahmood, and I. Usman, "Intelligent reversible watermarking and authentication: Hiding depth map information for 3D cameras," *Information Sciences,* vol. 216, pp. 155-175, 2012.

[35] M. Arsalan, S. A. Malik, and A. Khan, "Intelligent reversible watermarking in integer wavelet domain for medical images," *Journal of Systems and Software,* vol. 85, pp. 883-894, 2012.

[36] R. J. Mstafa and K. M. Elleithy, "A highly secure video steganography using Hamming code (7, 4)," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT),* pp. 1-6, 2014.

[37] R. J. Mstafa and K. M. Elleithy, "An Efficient Video Steganography Algorithm Based on BCH Codes," in *American Society for Engineering Education (ASEE) Conference*, pp. 1-10, 2015.

[38] R. J. Mstafa and K. M. Elleithy, "A novel video steganography algorithm in DCT domain based on hamming and BCH codes," in *IEEE 37th Sarnoff Symposium*, pp. 208-213, 2016.

[39] R. J. Mstafa and K. M. Elleithy, "An ECC/DCT-Based Robust Video Steganography Algorithm for Secure Data Communication," *Journal of Cyber Security and Mobility, vol. 5, pp. 167-194, 2016*.

[40] R. J. Mstafa and K. M. Elleithy, "A DCT-based robust video steganographic method using BCH error correcting codes," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1-6, 2016.

[41] R. J. Mstafa and K. M. Elleithy, "A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes," *Multimedia Tools and Applications*, vol. 75, pp. 10311-10333, 2016.

[42] R. J. Mstafa and K. M. Elleithy, "A New Video Steganography Algorithm Based on the Multiple Object Tracking and Hamming Codes," in *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 335-340, 2015.

[43] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC," *IEEE Access*, vol. 5, pp. 5354-5365, 2017.

[44] R. J. Mstafa and K. M. Elleithy, "Compressed and raw video steganography techniques: a comprehensive survey and analysis," *Multimedia Tools and Applications*, pp. 1-38, 2016.

[45] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "Video Steganography Techniques: Taxonomy, Challenges, and Future Directions," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1-6, 2017.

[46] W.-N. Lie and C.-W. Lin, "Enhancing video error resilience by using data-embedding techniques," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 16, pp. 300-308, 2006.

[47] A. Yilmaz and A. A. Alatan, "Error concealment of video sequences by data hiding," in *International Conference on Image Processing (ICIP),* pp. 679-82, 2003.

[48] D. L. Robie and R. M. Mersereau, "Video error correction using steganography," *EURASIP Journal on Applied Signal Processing,* vol. 2002, pp. 164-173, 2002.

[49] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, pp. 1062-1078, 1999.

[50] D. Stanescu, M. Stratulat, B. Ciubotaru, D. Chiciudean, R. Cioarga, and M. Micea, "Embedding Data in Video Stream using Steganography," in *4th International Symposium on Applied Computational Intelligence and Informatics SACI '07*, pp. 241-244, 2007.

[51] R. J. Mstafa and K. M. Elleithy, "A highly secure video steganography using Hamming code (7, 4)," in *IEEE Long Island, Systems, Applications and Technology Conference (LISAT)*, pp. 1-6, 2014.

[52] Z. Wei, S. S. Cheung, and C. Minghua, "Hiding privacy information in video surveillance system," in IEEE International Conference on Image Processing, pp. II-868-871, 2005.

[53] I. Mehmood, M. Sajjad, S. Rho, and S. W. Baik, "Divide-and-conquer based summarization framework for extracting affective video content," *Neurocomputing,* vol. 174, pp. 393-403, 2016.

[54] K. Muhammad, A. Jamil, F. Haleem, J. Zahoor, S. Muhammad, and B. Sung Wook, "A Secure Method for Color Image Steganography using Gray-Level Modification and Multi-level Encryption," *KSII Transactions on Internet and Information Systems (TIIS),* vol. 9, pp. 1938-1962, 2015.

[55] K. Muhammad, M. Sajjad, and S. W. Baik, "Dual-Level Security based Cyclic18 Steganographic Method and its Application for Secure Transmission of Keyframes during Wireless Capsule Endoscopy," *Journal of Medical Systems,* vol. 40, pp. 1-16, 2016.

[56] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE transactions on circuits and systems for video technology,* vol. 13, pp. 614-619, 2003.

[57] T. Shanableh, "Data Hiding in MPEG Video Files Using Multivariate Regression and Flexible Macroblock Ordering," *IEEE Transactions on Information Forensics and Security,* vol. 7, pp. 455-464, 2012.

[58] T. Wedi, "Adaptive interpolation filter for motion compensated prediction," in *Proceedings International Conference on Image Processing*, vol. 2, pp. II-509-II-512, 2002.

[59] G. Yang, J. Li, Y. He, and Z. Kang, "An information hiding algorithm based on intra-prediction modes and matrix coding for H. 264/AVC video stream," *AEU-*

International Journal of Electronics and Communications, vol. 65, pp. 331-337, 2011.

[60] B. Liu, F. Liu, C. Yang, and Y. Sun, "Secure steganography in compressed video bitstreams," in *Third International Conference on Availability, Reliability and Security, ARES 08.*, pp. 1382-1387*, 2008.*

[61] P.-C. Chang, K.-L. Chung, J.-J. Chen, C.-H. Lin, and T.-J. Lin, "A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames," *Journal of Visual Communication and Image Representation,* vol. 25, pp. 239-253, 2014.

[62] Y. Hu, C. Zhang, and Y. Su, "Information hiding based on intra prediction modes for H. 264/AVC," in *IEEE International Conference on Multimedia and Expo*, pp. 1231-1234, 2007.

[63] H. Zhu, R. Wang, D. Xu, and X. Zhou, "Information Hiding Algorithm for H. 264 Based on the predition difference of Intra_4× 4," in *3rd International Congress on Image and Signal Processing (CISP)*, pp. 487-490, 2010.

[64] X. Zhang and S. Liu, "Method and Apparatus for Intra Mode Coding in HEVC," ed: Google Patents, 2012.

[65] S. K. Kapotas and A. N. Skodras, "A new data hiding scheme for scene change detection in H. 264 encoded video sequences," in IEEE International Conference on Multimedia and Expo, pp. 277-280, 2008.

[66] C. Xu, X. Ping, and T. Zhang, "Steganography in compressed video stream," in *First International Conference on Innovative Computing, Information and Control, ICICIC'06,* pp. 269-272, 2006.

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

[68] H. Bin, Z. Li-Yi, and Z. Wei-Dong, "A novel steganography algorithm based on motion vector and matrix encoding," in *IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 406-409, 2011.

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

[70] Z. Shahid, M. Chaumont, and W. Puech, "Considering the reconstruction loop for data hiding of intra-and inter-frames of H. 264/AVC," *Signal, Image and Video Processing*, vol. 7, pp. 75-93, 2013.

[71] J. M. Thiesse, J. Jung, and M. Antonini, "Rate Distortion Data Hiding of Motion Vector Competition Information in Chroma and Luma Samples for Video Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 729-741, 2011.

[72] J. M. Thiesse, J. Jung, and M. Antonini, "Data hiding of motion information in chroma and luma samples for video compression," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 217-221, 2010.

[73] J. M. Thiesse, J. Jung, and M. Antonini, "Data hiding of intra prediction information in chroma samples for video compression," in *17th IEEE International Conference on Image Processing (ICIP)*, pp. 2861-2864, 2010.

[74] P. Meuel, M. Chaumont, and W. Puech, "Data hiding in H. 264 video for lossless reconstruction of region of interest," in *EUSIPCO 07: 15th European Signal Processing Conference*, pp. 2301-2305, 2007.

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

[76] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H. 264/AVC video streams without intra-frame distortion drift," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 20, pp. 1320-1330, 2010.

[77] Y. Liu, Z. Li, X. Ma, and J. Liu, "A novel data hiding scheme for H. 264/AVC video streams without intra-frame distortion drift," in *IEEE 14th International Conference on Communication Technology (ICCT),* pp. 824-828, 2012.

[78] Y. Liu, Z. Li, X. Ma, and J. Liu, "A Robust Data Hiding Algorithm for H. 264/AVC Video Streams without Intra-frame Distortion Drift," in *Proceedings of the Second International Conference on Electric Information and Control Engineering-Volume 01*, pp. 182-186, 2012.

[79] Y. Liu, Z. Li, X. Ma, and J. Liu, "A Robust Data Hiding Algorithm for H. 264/AVC Video Streams," Journal of Systems and Software, vol. 86, pp. 2174-2183, 2013.

[80] N. Ke and Z. Weidong, "A video steganography scheme based on H. 264 bitstreams replaced," in *4th IEEE International Conference on Software Engineering and Service Science (ICSESS),* pp. 447-450, 2013.

[81] K. Liao, S. Lian, Z. Guo, and J. Wang, "Efficient information hiding in H. 264/AVC video coding," *Telecommunication Systems*, vol. 49, pp. 261-269, 2012.

[82] C.-S. Lu, J.-R. Chen, and K.-C. Fan, "Real-time frame-dependent video watermarking in VLC domain," *Signal Processing: Image Communication,* vol. 20, pp. 624-642, 2005.

[83] R. WANG, L. HU, and D. XU, "A Watermarking Algorithm Based on the CABAC Entropy Coding for H.264/AVC," *J. Comput. Inform. Syst.,* vol. 7, no. 6, pp. 2132–2141, 2011.

[84] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. Baik, "A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image," *Multimedia Tools and Applications,* vol. 75, pp. 14867-14893, 2016.

[85] R. Zhang, V. Sachnev, M. B. Botnan, H. J. Kim, and J. Heo, "An efficient embedder for BCH coding for Steganography," *IEEE Transactions on Information Theory,* vol. 58, pp. 7272-7279, 2012.

[86] A. Cheddad, J. Condell, K. Curran, and P. McKevitt, "Skin tone based Steganography in video files exploiting the YCbCr colour space," in *IEEE International Conference on Multimedia and Expo,* pp. 905-908, 2008.

[87] M. M. Sadek, A. S. Khalifa, and M. G. Mostafa, "Robust video steganography algorithm using adaptive skin-tone detection," *Multimedia Tools and Applications, vol. 76, pp. 3065-3085, 2017.*

[88] M. A. Alavianmehr, M. Rezaei, M. S. Helfroush, and A. Tashk, "A lossless data hiding scheme on video raw data robust against H.264/AVC compression," in *2nd*

International eConference on Computer and Knowledge Engineering (ICCKE), pp. 194-198, 2012.

[89] M. E. Eltahir, L. M. Kiah, and B. B. Zaidan, "High Rate Video Streaming Steganography," in *International Conference on Information Management and Engineering, ICIME '09*, pp. 550-553, 2009.

[90] K. Dasgupta, J. K. Mondal, and P. Dutta, "Optimized Video Steganography Using Genetic Algorithm (GA)," *Procedia Technology*, vol. 10, pp. 131-137, 2013.

[91] E. Kawaguchi and R. O. Eason, "Principles and applications of BPCS steganography," in *Photonics East (ISAM, VVDC, IEMB)*, pp. 464-473, 1999.

[92] S. Sun, "A New Information Hiding Method Based on Improved BPCS Steganography," *Advances in Multimedia*, vol. 2015, 2015.

[93] K. Patel, K. K. Rora, K. Singh, and S. Verma, "Lazy Wavelet Transform Based Steganography in Video," in *International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 497-500, 2013.

[94] J. Spaulding, H. Noda, M. N. Shirazi, and E. Kawaguchi, "BPCS steganography using EZW lossy compressed images," *Pattern Recognition Letters*, vol. 23, pp. 1579-1587, 2002.

[95] H. Noda, T. Furuta, M. Niimi, and E. Kawaguchi, "Application of BPCS steganography to wavelet compressed video," in *International Conference on Image Processing, ICIP'04*, pp. 2147-2150, 2004.

[96] A. Sarkar, U. Madhow, and B. S. Manjunath, "Matrix Embedding With Pseudorandom Coefficient Selection and Error Correction for Robust and Secure

Steganography," *IEEE Transactions on Information Forensics and Security,* vol. 5, pp. 225-239, 2010.

[97] C. Chin-Chen, T. D. Kieu, and C. Yung-Chen, "A High Payload Steganographic Scheme Based on (7, 4) Hamming Code for Digital Images," in *International Symposium on Electronic Commerce and Security*, pp. 16-21, 2008,.

[98] Y. Hoyoung, J. Jaehwan, J. Jihyuck, and P. In-Cheol, "Area-Efficient Multimode Encoding Architecture for Long BCH Codes," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 60, pp. 872-876, 2013.

[99] A. K. Jain, "*Fundamentals of digital image processing*," Prentice-Hall, Inc., 1989.

[100] W. B. Pennebaker and J. L. Mitchell, "*JPEG: Still image data compression standard*," Springer Science & Business Media, 1992.

[101] B. G. Vani and E. Prasad, "High Secure Image Steganography based on Hopfield Chaotic Neural Network and Wavelet Transforms," International Journal of Computer Science and Network Security (IJCSNS), vol. 13, p. 1, 2013.

[102] G. Prabakaran and R. Bhavani, "A modified secure digital image steganography based on Discrete Wavelet Transform," in *International Conference on Computing, Electronics and Electrical Technologies (ICCEET),* pp. 1096-1100, , 2012.

[103] O. S. Faragallah, "Efficient video watermarking based on singular value decomposition in the discrete wavelet transform domain," *AEU - International Journal of Electronics and Communications,* vol. 67, pp. 189-196, 2013.

[104] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 11, pp. 674-693, 1989.

[105] P. Viola and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision,* vol. 57, pp. 137-154, 2004.

[106] R. Isukapalli, A. Elgammal, and R. Greiner, "Learning a Dynamic Classification Method to Detect Faces and Identify Facial Expression," in *Analysis and Modelling of Faces and Gestures*. W. Zhao, S. Gong, and X. Tang, Eds., ed: Springer Berlin Heidelberg, vol. 3723, pp. 70-84, 2005.

[107] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR,* vol. 1, pp. I-511-I-518, 2001.

[108] E. Torres-Pereira, H. Martins-Gomes, A. Monteiro-Brito, and J. de Carvalho, "Hybrid Parallel Cascade Classifier Training for Object Detection," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. E. Bayro-Corrochano and E. Hancock, Eds., ed: Springer International Publishing, vol. 8827, pp. 810-817, 2014.

[109] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *International Journal of Computer Vision,* vol. 77, pp. 259-289, 2008.

[110] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *7th international joint conference on Artificial intelligence,* pp. 674-679, 1981.

[111] C. Tomasi and T. Kanade, "*Detection and tracking of point features,*" School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[112] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR '94*, pp. 593-600, 1994.

[113] H. Fassold, J. Rosner, P. Schallauer, and W. Bailer, "Realtime KLT feature point tracking for high definition video," *GraVisMa,* 2009.

[114] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A skin tone detection algorithm for an adaptive approach to steganography," *Signal Processing,* vol. 89, pp. 2465-2478, 2009.

[115] L. Guo-Shiang and T. Tung-Sheng, "A face tracking method using feature point tracking," in *International Conference on Information Security and Intelligence Control (ISIC)*, pp. 210-213, 2012.

[116] K. Muhammad, J. Ahmad, M. Sajjad, and S. W. Baik, "Visual saliency models for summarization of diagnostic hysteroscopy videos in healthcare systems," *SpringerPlus,* vol. 5, pp. 1-13, 2016.

[117] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR),* vol. 38, pp. 1-45, 2006.

[118] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 1-6, 2009.

[119] J. Ahmad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "Saliency-weighted graphs for efficient visual content description and their applications in real-time image retrieval systems," *Journal of Real-Time Image Processing*, pp. 1-17, 2015.

[120] K. Muhammad, J. Ahmad, M. Sajjad, and M. Zubair, "Secure Image Steganography using Cryptography and Image Transposition," *NED University Journal of Research*, vol. 12, pp. 81-91, 2015.

[121] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli, "New full-reference quality metrics based on HVS," in *proceedings of the second international workshop on video processing and quality metrics*, vol. 4, 2006.

[122] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, "On between-coefficient contrast masking of DCT basis functions," in *Proceedings of the Third International Workshop on Video Processing and Quality Metrics*, vol. 4, 2007.

[123] L. Tse-Hua and A. H. Tewfik, "A novel high-capacity data-embedding system," *IEEE Transactions on Image Processing*, vol. 15, pp. 2431-2440, 2006.

[124] Y. He, G. Yang, and N. Zhu, "A real-time dual watermarking algorithm of H.264/AVC video stream for Video-on-Demand service," *AEU - International Journal of Electronics and Communications*, vol. 66, pp. 305-312, 2012.

[125] J. Huang and Y. Q. Shi, "Reliable information bit hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 916-920, 2002.

[126] M. Barni, F. Bartolini, and N. Checcacci, "Watermarking of MPEG-4 video objects," *IEEE Transactions on Multimedia*, vol. 7, pp. 23-32, 2005.

[127] G. Li, Y. Ito, X. Yu, N. Nitta, and N. Babaguchi, "Recoverable privacy protection for video content distribution," *EURASIP Journal on Information Security,* vol. 2009, pp. 1-11, 2009.

[128] B. G. Mobasseri and M. P. Marcinak, "Watermarking of MPEG-2 video in compressed domain using VLC mapping," in *Proceedings of the 7th workshop on Multimedia and security,* pp. 91-94, 2005.

[129] S. Khupse and N. N. Patil, "An adaptive steganography technique for videos using Steganoflage," in *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 811-815, 2014.

[130] S. K. Moon and R. D. Raut, "Analysis of secured video steganography using computer forensics technique for enhance data security," in *IEEE Second International Conference on Image Information Processing (ICIIP)*, pp. 660-665, 2013.

[131] H. M. Kelash, O. F. Abdel Wahab, O. A. Elshakankiry, and H. S. El-sayed, "Hiding data in video sequences using steganography algorithms," in *International Conference on ICT Convergence (ICTC)*, pp. 353-358, 2013.

[132] R. Paul, A. K. Acharya, V. K. Yadav, and S. Batham, "Hiding large amount of data using a new approach of video steganography," in *Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, pp. 337-343, 2013.

[133] A. T. Bhole and R. Patel, "Steganography over video file using Random Byte Hiding and LSB technique," in *IEEE International Conference on Computational Intelligence & Computing Research (ICCIC)*, pp. 1-6, 2012.

[134] A. Hanafy, G. Salama, and Y. Z. Mohasseb, "A secure covert communication model based on video steganography," in *IEEE Military Communications Conference, MILCOM 2008*, pp. 1-6, 2008.

[135] D.-C. Lou and C.-H. Hu, "LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis," *Information Sciences,* vol. 188, pp. 346-358, 2012.

[136] G. R. Tadiparthi and T. Sueyoshi, "A novel steganographic algorithm using animations as cover," *Decision Support Systems*, vol. 45, pp. 937-948, 2008.

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

APPENDIX A: PUBLICATIONS

Journal Papers

1. R. J. Mstafa and K. M. Elleithy, "A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes," Multimedia Tools and Applications, vol. 75, pp. 10311-10333, 2016. (***Springer*** **IF=1.346**)
2. R. J. Mstafa and K. M. Elleithy, "Compressed and Raw Video Steganography Techniques: A Comprehensive Survey and Analysis," *Multimedia Tools and Applications,* pp. 1-38, 2017. (***Springer*** **IF=1.331**)
3. R. J. Mstafa and K. M. Elleithy, "An ECC/DCT-Based Robust Video Steganography Algorithm for Secure Data Communication," *Journal of Cyber Security and Mobility,* vol. 5, pp. 167-194, 2016. (***BkCI***)
4. R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC," *IEEE Access,* vol. 5, pp. 5354-5365, 2017. (***IEEE Access*** **IF=1.27**)

Conference Papers

1. R. J. Mstafa and K. M. Elleithy, “A highly secure video steganography using Hamming code (7, 4),” in IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-6, 2014.

2. R. J. Mstafa and K. M. Elleithy, “An Efficient Video Steganography Algorithm Based on BCH Codes,” in American Society for Engineering Education Conference (ASEE), pp. 1- 10, 2015.

3. R. J. Mstafa and K. M. Elleithy, “A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11),” in IEEE Wireless Telecommunications Symposium (WTS), pp. 1-8, 2015.

4. R. J. Mstafa and K. M. Elleithy, “A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes,” in IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-7, 2015.

5. R. J. Mstafa and K. M. Elleithy, “A New Video Steganography Algorithm Based on the Multiple Object Tracking and Hamming Codes,” in IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 335-340, 2015.

6. R. J. Mstafa and K. M. Elleithy, “A DCT-based Robust Video Steganographic Method Using BCH Error Correcting Codes,” in IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-6, 2016.

7. R. J. Mstafa and K. M. Elleithy, “A novel video steganography algorithm in DCT domain based on hamming and BCH codes,” in *IEEE 37th Sarnoff Symposium*, pp. 208-213, 2016.

8. R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, “Video Steganography Techniques: Taxonomy, Challenges, and Future Directions,” in IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-6, 2017.

Posters

1. R. J. Mstafa and K. M. Elleithy, "An adaptive Video Steganography Method Based on the Multiple Object Tracking and Hamming Codes," in ASEE, University of Massachusetts Lowell, MA, 2017.

2. R. J. Mstafa and K. M. Elleithy, "Efficient and Robust Video Steganography Algorithms for Secure Data Communication," in Connecticut Symposium on Microelectronics and Optoelectronics (CMOC), University of Connecticut, CT, 2017.