Journal of Engineering Sciences, Assiut University, Vol. 34, No. 4, pp. 1265-1282, July 2006

INFORMATION HIDING IN VIDEO

Yousef B. Mahdy and Mohammed A. Atiea
Computer Science Department, Faculty of Computer and Information, Assiut University, Egypt.

(Received March 28, 2006. Accepted May 22, 2006)

ABSTRACT– *This paper proposes a video information-embedding scheme in which the embedded information is reconstructed without knowing the original host video. The proposed method presents high rate of information embedding and is robust to motion compensated coding, such as MPEG-II and MPEG-III. The characteristic of the proposed scheme is to use pixels contractive relation to assist motion compensated coding process. Information is embedded in video frames using the block DCT. Embedded frames are then coded using MPEG-III. After that the information will be extracted from video with high efficiency. Experimental results have indicated that the method is robust against MPEG encoding and re-encoding.*

KEYWORDS: *Information Hiding, Motion Compensated Coding, Contractive Relation.*

1. INTRODUCTION

Steganography is the art of hiding the existence of a message, the word Steganography comes from the Greek words steganos (secret) and graphy (writing) [1]. An example could be a letter written with two different inks, when the letter is submerged in water, one of the inks dissolves while the other remains on the letter, thus revealing the secret message. The original message on the letter is just a cover to hide the existence of the secret message, so we can say steganography is hiding the existence of communication. One famous example of early steganography is that of Herodotus who shaved the head of one of his slaves and tattooed a message on his head. After his hair had re-grown, he was sent to deliver the message to instigate a revolt against the Persians. Steganography has a wide range of forms, from hiding messages in the soles of shoes to hiding messages in musical scores [5, 6].

Steganography is sometimes confused with cryptography. Cryptography is the art of concealing the contents of a message (encryption) whereas Steganography is the art of hiding the existence of a message. The message can be any type of digital information including a simple text file, a JPEG image, or any other type of file. Much of the available software that embeds steganographic content into a host file often employs cryptography as well. This greatly adds to the complexity and difficulty of retrieving concealed content.

One of the earliest examples of cryptography was used by Julius Caesar [5], when he sent military messages to his armies. Perhaps since that time, people have also tried to decode encrypted messages. Allies in World War II were able to break a secret German code called Enigma. This discovery enabled Allied forces to locate and sink many German U-boats. Moreover, they were able to obtain advanced information about German military operations that was critical to the campaign in Europe. Similar code breaking abilities also allowed the United States Navy to intercept the Japanese fleet in one of the most decisive battles in the Pacific the Battle of Midway. These are just a few examples of how cryptographic technology has played an important role in history [6].

Recently, more and more people communicate with each other by surfing on the Internet. However, it is not very secure when we transmit information through Internet. Everyone can peek, copy even alter our information easily in this wide-open environment. Thus, People don’t want to transmit the important information without any protection in the public network unless a secure channel is provided for the transmission. The cryptography technique can protect the message content from a peeper. But the cryptography technique will cause the message content to be meaningless random codes. It is easy to guess something important in the transmitted information even the receiver do not know what is inside. They may cut, hack or break these meaningless random codes. Therefore, information-hiding technique is needed to help in solving the problem of transmitting important data in an absolute secure channel. Thus, it is not only difficult to decrypt the data, but also difficult for attackers to detect the hidden data.

People usually hide data or information into one medium. It can be a text article, image, music, or video. The medium that hides data is named stego-medium. It can keep the confidential data secretly, and call the original medium that does not hide data the cover medium. It is difficult for the unauthorized people to detect hidden data from a stego-medium. They can use it to hide important data. Then, when they transmit the medium, the peepers will not find the secret data in it. Because the peepers will not find the difference between the original medium and the medium already altered. They will think that the transmitted medium is an unimportant data or a data that they don’t want to collect or peep [8].

Information hiding technique can hide information or data in a meaning medium. Other people cannot easily detect what hidden in the original meaningful medium. If some one hide important information in common media channels and transmit it on the Internet, others will think that the message sent is unimportant. Only the authorized receivers can know the secret and extract it. They can combine these two techniques (i.e., transmission and hiding) to construct a secure and secret transmitting channel [9].
In digital image (video frames) model, there are two major models. One is spatial domain; image is created by many pixels. The other is frequency domain, in which pixel values will be transformed to frequency parameters by mathematical algorithm. In general, most people use the frequency domain to compress the image. There are many information-hiding researches both in spatial domain and frequency domain image. In spatial domain, it alters pixel values directly in memory [10-12]. Generally, in spatial domain, people can have a higher hiding capacity [13, 14]. Besides, the information hiding process of hiding and extraction are easy and quick [15, 16]. But in most of the cases, they will have problem when operated with lossy compression.

While the stego-medium is compressed by lossy compression methods, the receiver cannot extract the complete data correctly [17, 18]. If user loses one of the hiding bits, they will not apprehend or identify the hidden contents.

One of the widely used information hiding methods is LSB (least significant bit) [19] method. This method changes the last bit of each pixel bit to hide the data, because the last bit of pixel changes little pixel's color, the image quality would not be affected much. In fact, Human vision will not be aware of the difference of image.

In frequency domain [20, 21], people will employ the feature of frequency parameters to hide data [22]. The methods in this domain can fight against more attacks and raise the robustness [23, 24]. In frequency domain, good image quality can be retained. But it will lose some embedded data, after performing the lossy compression process to the stego-medium [25, 26]. Thus, users cannot hide text information in it [27]. Besides, the process of compression and hiding has a higher complexity than spatial domain [28]. The information hiding technique common methods are based on discrete cosine transformation [29, 30] and discrete wavelet transformation.

In this paper, new method based on DCT (discrete cosine transformation), which can be compatible with Motion Compensated Coding. The proposed method has the features of the spatial and frequency domains, using the proposed method for information hiding, the hiding and extraction process will be more efficient.

The remainder of the paper is organized as follows: Background is summarized in Section 2. The proposed method presented in detail in Section 3 and some of the experiments and the results are given in Section 4. Conclusions are given in Section 5.

2. BACKGROUND

2.1 YCrCb Color Space And Image Compression

There are an infinite number of possible color spaces instead of the common RGB (Red, Green and Blue) system most commonly used in computer graphics. Many of these other color spaces are derived by applying linear functions of RGB.

The human visual system has much less dynamic range (acuity) for spatial variation in color than for brightness (luminance), in other words, they are acutely aware of changes in the brightness of details than of small changes in hue. So rather than saving as RGB it can be more efficient to encode luminance in one channel and color information (that has had the luminance contribution removed) in two other channels.

The two color channels can be encoded with less bandwidth by using a number of techniques, predominantly by reducing the precision, or as will be discussed here, reducing the spatial resolution.

Since green dominates the luminance channel it makes sense to base the other two chrominance channels on luminance subtracted red and blue. Such luminance, red chrominance and blue chrominance systems are generally referred to as Y, Cr, and Cb. In what follows they will generally be referred this way or simply as YCC, the relation between RGB and YCC as shown in **Fig. 1**.

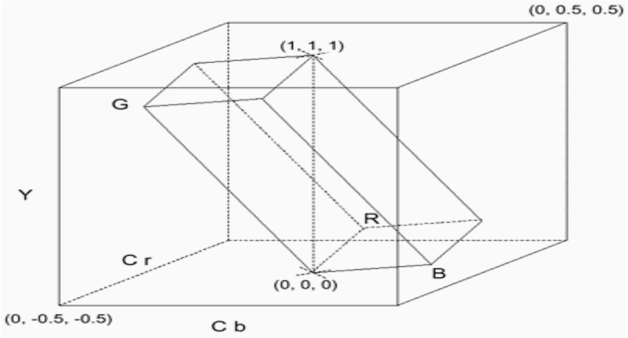


Figure. 1: Relation between RGB and YCC color space.

The YCC color space conversion equation used follows that used by TIFF and JPEG:

$$Y = 0.2989\ R + 0.5866\ G + 0.1145\ B \quad (1)$$

$$Cb = -0.1687\ R - 0.3312\ G + 0.5000\ B \quad (2)$$

$$Cr = 0.5000\ R - 0.4183\ G - 0.0816\ B \quad (3)$$

RGB values are normally on the scale of 0 to 1, or since they are stored as unsigned single bytes, 0 to 255. The resulting luminance value is also on the scale of 0 to 255, the chrominance values need 127.5 added to them so they can be saved in an unsigned byte. Of course when the YCC values are returned back into RGB, then 127.5 must be first subtracted from the two chrominance values. The reverse transform is:

$$R = Y + 1.4022\ Cr \quad (4)$$

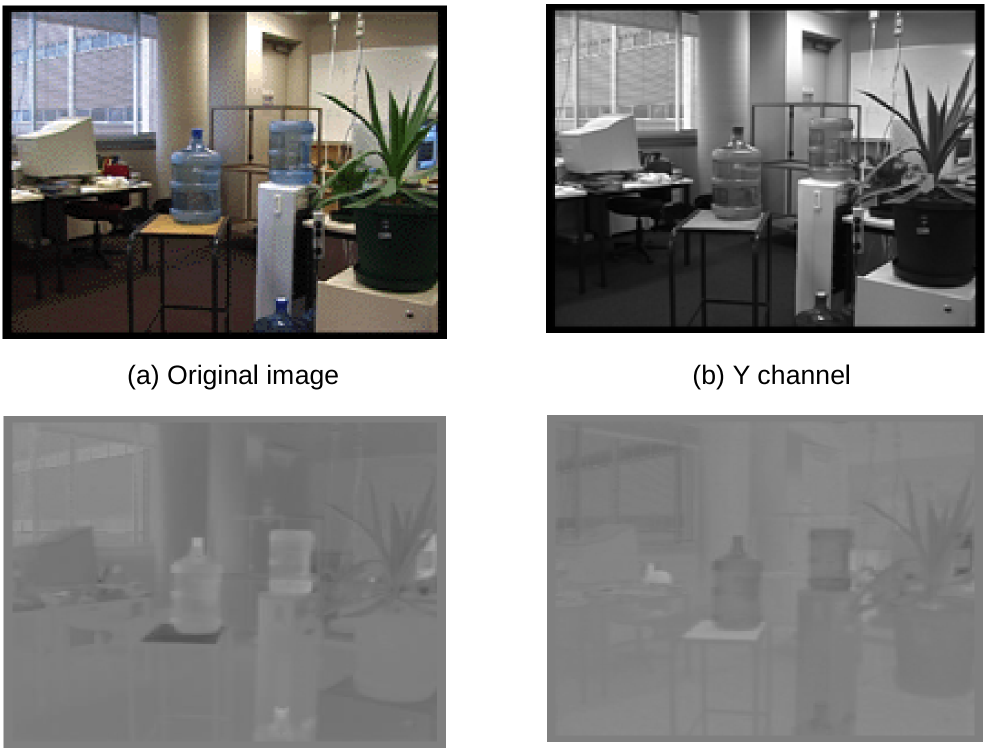
$$G = Y - 0.3456\ Cb - 0.7145\ Cr \quad (5)$$

$$B = Y + 1.7710\ Cb \quad (6)$$

The full chrominance range of [-0.5, +0.5] is mapped into a larger color space than supported by RGB. The above equations can yield RGB values outside the 0 to 255 (or 0 to 1) range, these typically relate to very light or dark colors. The RGB values should be clipped so they lie within the allowed range after the transform from YCC to RGB is calculated.

Figure 2a shows original image that was captured from a digital movie camera. The result is shown in **Fig. 2b** from a 1 byte Y channel, 4x4 sub samples Cr and Cb channels as shown in **Fig. 2c** and **Fig. 2d**.

In most image compression technique after color space conversion, each pixel is represented as (Y, Cb and Cr) because human visual system (HVS) is most sensitive to Y component. So encode Y component with full resolution. But HVS is less sensitive to Cb Cr components, so sub sample Cb Cr components. By doing so data can be reduced without affecting visual quality from person view.



(a) Original image

(b) Y channel

(c) Cr channel

(d) Cb channel

Figure 2: (a) original image, (b) Y channel, (c) Cr channel, (d) Cb channel.

2.2 The Discrete Cosine Transform

The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies [33]. The DCT has the property that most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. For this reason, the DCT is often used in image compression applications. For example, the DCT is at the heart of the international standard lossy image compression algorithm known as JPEG [34].

The two-dimensional DCT of an M-by-N matrix A is defined as follows:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \, Cos \frac{\pi(2m+1)p}{2M} Cos \frac{\pi(2n+1)q}{2N} \qquad (7)$$

Where

$$\begin{cases} 0 \le \mathrm{p} \le \mathrm{M} - 1 \\ 0 \le q \le N - 1 \end{cases} \qquad (8)$$

And

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & p=0 \\ \sqrt{\frac{2}{M}} & 1 \le p \le M-1 \end{cases} \tag{9}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & q=0 \\ \sqrt{\frac{2}{N}} & 1 \le q \le N-1 \end{cases} \tag{10}$$

The DCT is an invertible transform, and its inverse is given by:

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} Cos \frac{\pi(2m+1)p}{2M} Cos \frac{\pi(2n+1)q}{2N} \tag{11}$$

Where

$$\begin{cases} 0 \le \mathrm{m} \le \mathrm{M} - 1 \\ 0 \le n \le N - 1 \end{cases} \tag{12}$$

And

$$\begin{cases} 0 \le \mathrm{p} \le \mathrm{M} - 1 \\ 0 \le q \le N - 1 \end{cases} \tag{13}$$

2.3 MPEG Video Codec

MPEG uses 8 by 8 (8x8) DCT. By using this transform they can convert an 8x8 pixel block to another 8x8 block. In general most of the energy (value) is concentrated to the top-left corner. After quantizing (Quantization is to reduce accuracy of numbers to small value, so they can use less bits to represent a large number) the transformed matrix, most data in this matrix may be zero, then using zigzag order scan and run length coding can achieve a high compression ratio.

In MPEG-2, a matrix called quantizer Q [i, j] as shown in **Fig. 3** is used to define quantization step.

The quantization equation is given by:

$$Xq\,[i, j] = Round\,(X\,[i, j] \,/\, Q\,[i, j]) \tag{14}$$

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Figure 3: Intra Quantizer Matrix.

3. THE PROPOSED TECHNIQUE

The proposed technique embeds secret information in the frequency domain into Y component of each frame, in order to minimize the color distortion in the embedded video. Embedding secret information into the Cb and Cr components may result in undesirable color alterations, the flow diagram of embedding process is as shown in **Fig. 4**.

During the embedding, the process splits the cover-frame (video frame used as the media to hide the secret message in) into 8×8 pixel blocks, each block hide exactly four secret message bits, and each message bit needs two DCT coefficients.

The embedding process starts with selecting a block {bi}, which will be used to code the four message bits. Let Bi = DCT {bi} be the DCT-transformed image block. Embedding process uses eight DCT coefficients denoted by (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7) and (x_8, y_8).

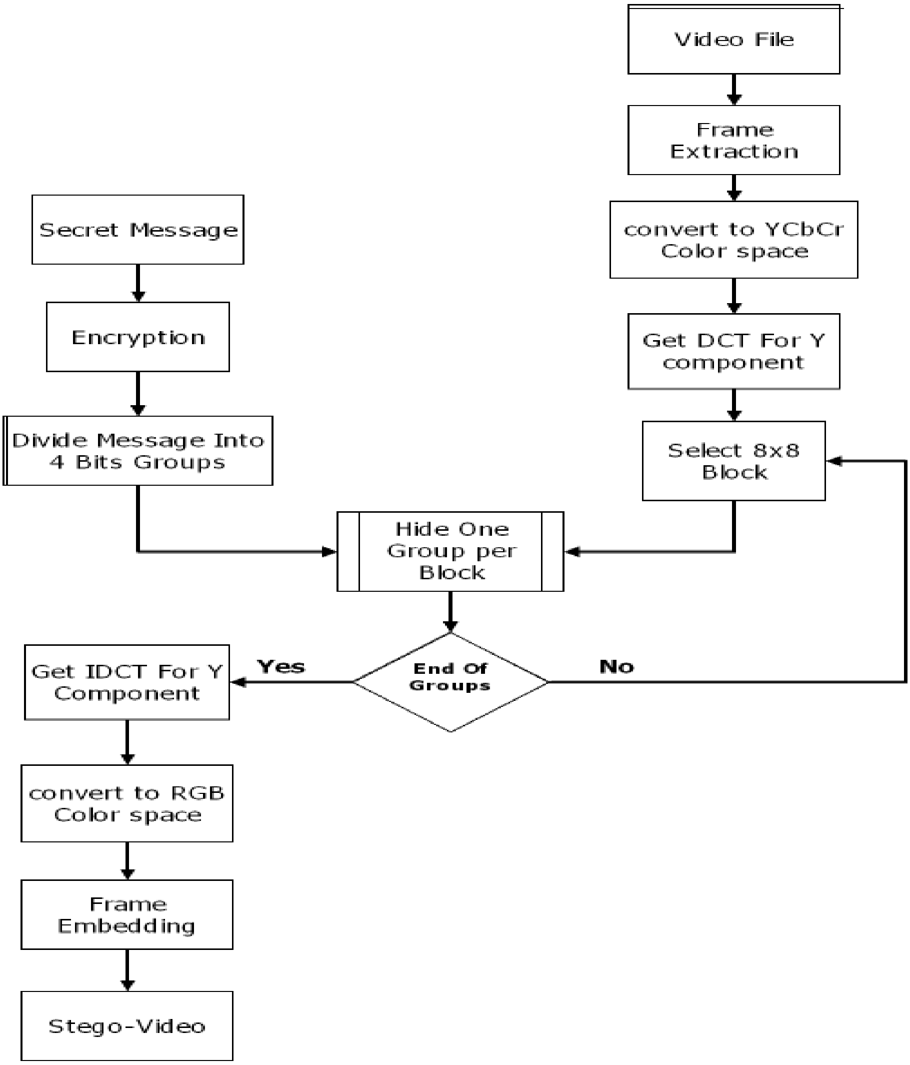


Figure 4: The flow diagram of embedding process.

The eight coefficients should correspond to cosine functions with middle frequencies; this ensures that the information is stored in significant parts of the signal (hence the embedded information will not be completely damaged by MPEG encoding). Furthermore, assume that the embedding process will not degenerate the cover heavily, because it is widely believed that DCT coefficients of middle frequencies have similar magnitudes [31]. Since the proposed system should be robust against MPEG encoding, then choose the DCT coefficients in such a way that the quantization values associated with them in the MPEG encoding algorithm are equal, since in the quantization process both coefficients are divided by the same quantization values. Their relative size will therefore only be affected in the rounding step. According to **Fig. 3**, the coefficients {(1,2), (2,1}}, {(2,4), (1,5)}, {(5,5), (4,6)} and {(5,1), (4,2)} are good candidates, the flow diagram of extracting process is as shown in **Fig. 5**.

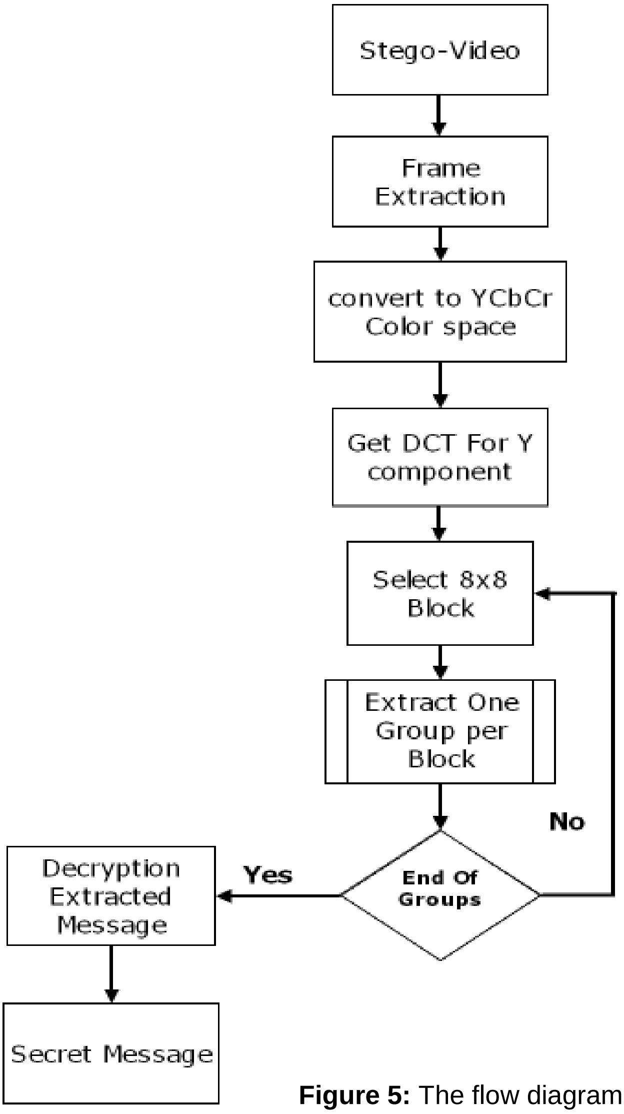


Figure 5: The flow diagram of extracting process.

The following parameters are defined:

1- M (J) is the bit number J of message data (secret information).

2- Z is number of bits in message data.

3- T denotes the robustness factor.

4- IDCT is the inverse of DCT.

Since the MPEG encoding can (in the quantization step) affect the relative sizes of the coefficients, the algorithm ensures that $|Bi(x_n,y_n) - Bi(x_m,y_m)| > T$ for some $T > 0$, by adding one to the larger value of $Bi(x_n,y_n)$ and $Bi(x_m,y_m)$ and subtracting one from the smaller value of $Bi(x_n,y_n)$ and $Bi(x_m,y_m)$ and repeat that until the condition becomes true (where n=1,3,5,7 and m=2,4,6,8). The higher T is, the more robust the algorithm will be against MPEG encoding, however, at the expense of image quality.

3.1. Embedding Algorithm

The steps in the embedding algorithm are as follows:

Step-1: For J=0 to Z-1 do steps from **step-2** to **step-13**.

Step-2: Choose one block bi.

Step-3: Let Bi = DCT {bi}.

Step-4: Adjust eight Coefficients such that $|Bi(x_1,y_1) - Bi(x_2,y_2)| > T$, $|Bi(x_3,y_3) - Bi(x_4,y_4)| > T$, $|Bi(x_5,y_5) - Bi(x_6,y_6)| > T$ and $|Bi(x_7,y_7) - Bi(x_8,y_8)| > T$.

Step-5: If M(J)= 0 check if $Bi(x_1,y_1) > Bi(x_2,y_2)$ then swap $Bi(x_1,y_1)$ and $Bi(x_2,y_2)$, else if M(J)=1 check if $Bi(x_1,y_1) < Bi(x_2,y_2)$ then swap $Bi(x_1,y_1)$ and $Bi(x_2,y_2)$.

Step-6: Increment J by one.

Step-7: If M(J)= 0 check if $Bi(x_3,y_3) > Bi(x_4,y_4)$ then swap $Bi(x_3,y_3)$ and $Bi(x_4,y_4)$, else if M(J)=1 check if $Bi(x_3,y_3) < Bi(x_4,y_4)$ then swap $Bi(x_3,y_3)$ and $Bi(x_4,y_4)$.

Step-8: Increment J by one.

Step-9: If M(J)= 0 check if $Bi(x_5,y_5) > Bi(x_6,y_6)$ then swap $Bi(x_5,y_5)$ and $Bi(x_6,y_6)$, else if M(J)=1 check if $Bi(x_5,y_5) < Bi(x_6,y_6)$ then swap $Bi(x_5,y_5)$ and $Bi(x_6,y_6)$.

Step-10: Increment J by one.

Step-11:If M(J)= 0 check if $Bi(x_7,y_7) > Bi(x_8,y_8)$ then swap $Bi(x_7,y_7)$ and $Bi(x_8,y_8)$, else if M(J)=1 check if $Bi(x_7,y_7) < Bi(x_8,y_8)$ then swap $Bi(x_7,y_7)$ and $Bi(x_8,y_8)$.

Step-12: Increment J by one.

Step-13: Pi=IDCT {Bi}

Step-14: After the end of for loop create stego-media out of all Pi(s).

As an example for hiding one bit take T=20, $Bi(x_1,y_1)=38$ and $Bi(x_2,y_2)=56$ where 38 and 56 are DCT values. After step-4 $Bi(x_1,y_1)=37$ and $Bi(x_2,y_2)=57$, now assume secret bit =1 so after step-5 $Bi(x_1,y_1)=57$ and $Bi(x_2,y_2)=37$. The steps for hiding one bit is as shown in **Fig. 6**.

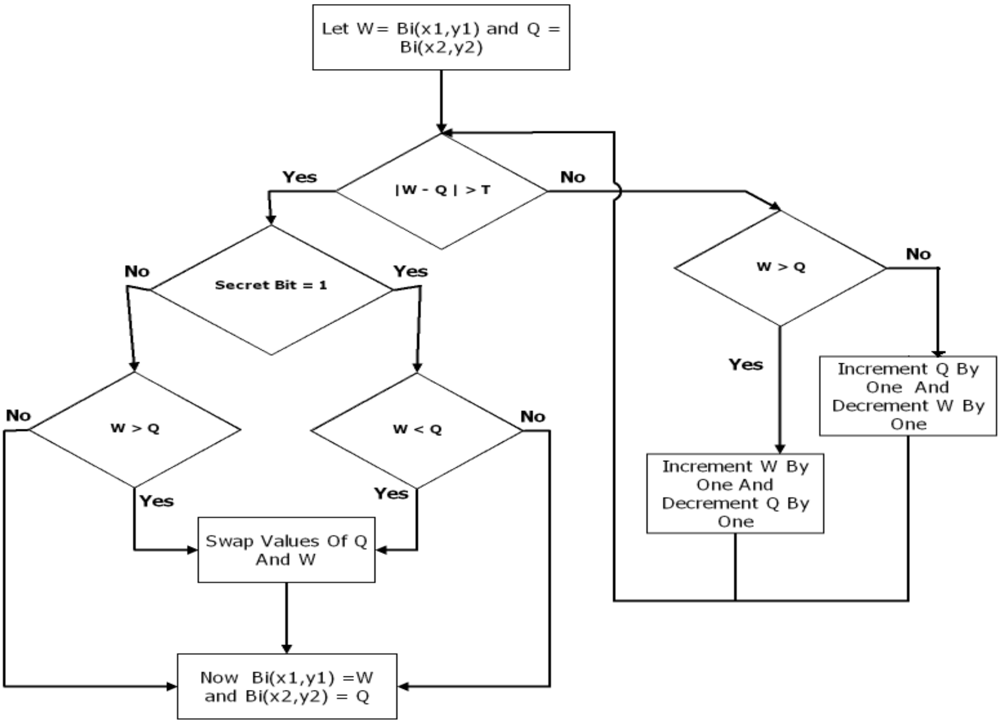

Let W= Bi(x1,y1) and Q = Bi(x2,y2)
|W - Q | > T
Yes
No
Secret Bit = 1
No
Yes
W > Q
No
Yes
W < Q
No
Yes
Swap Values Of Q And W
Now Bi(x1,y1) =W and Bi(x2,y2) = Q
W > Q
No
Yes
Increment Q By One And Decrement W By One
Increment W By One And Decrement Q By One

Figure 6: Steps for hide one bit.

3.2. Extraction Algorithm

The steps in the extraction algorithm are as follows:

Step-1: For J=0 to Z-1 do steps from **step-2** to **step-11.**

Step-2: Choose one cover-block Pi.

Step-3: Let Bi = DCT {Pi}.

Step-4: If $Bi(x_1,y_1) > Bi(x_2,y_2)$ then M(J)= 1 else M(J)= 0.

Step-5: Increment J by one.

Step-6: If $Bi(x_3,y_3) > Bi(x_4,y_4)$ then M(J)= 1 else M(J)= 0.

Step-7: Increment J by one.

Step-8: If $Bi(x_5,y_5) > Bi(x_6,y_6)$ then M(J)= 1 else M(J)= 0.

Step-9 : Increment J by one.

Step-10: If Bi(x_7,y_7) > Bi(x_8,y_8) then M(J)= 1 else M(J)= 0.

Step-11: Increment J by one.

Step-12: Integrate all the bits M (J) extracted by later Steps, and make up the embedded data.

Now as an example of extracting one bit after MPEG the encoder finds that the values of Bi(x_1,y_1)=45 and Bi(x_2,y_2)=19 so after step-4 of extraction algorithm finds that the hidden bit equals one. The steps for extracting one bit is shown in **Fig. 7**.

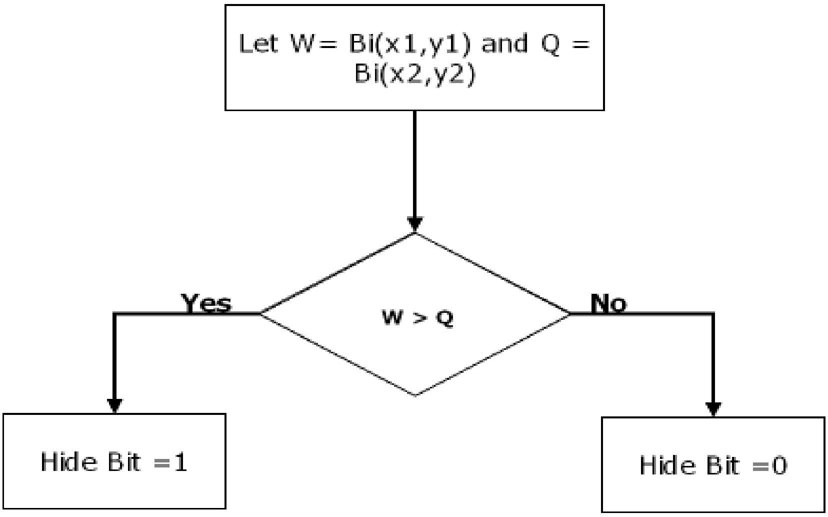


Figure 7: Steps for extracting one bit.

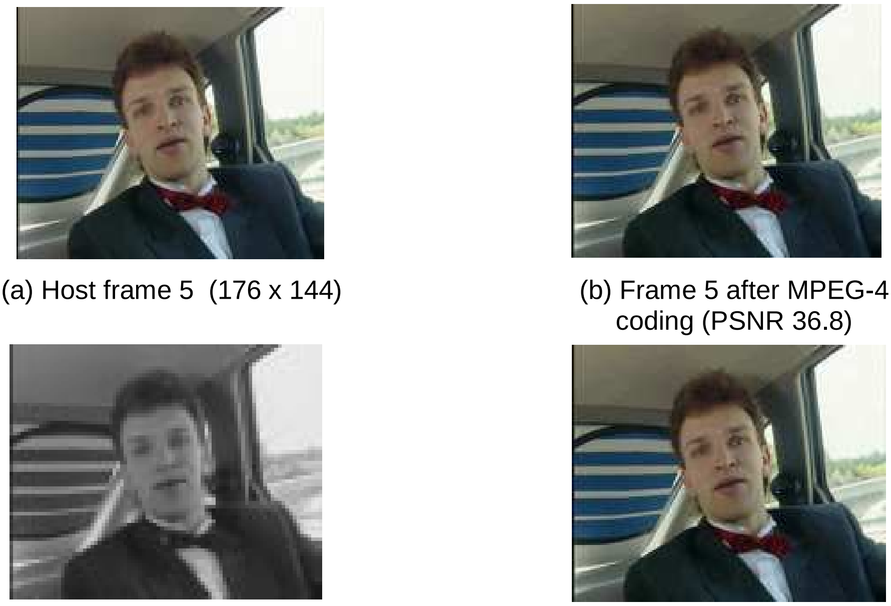
4. EXPERIMENTAL RESULTS

Extensive simulations are carried out to prove that the proposed technique is effective under different conditions. The proposed algorithm is evaluated using 30 frames of the “Car” video file. Each frame is of size 176 x 144.

Figure 8 shows samples of the test images. A host video frame is as shown in **Fig. 8a**, it's Y component **Fig. 8c** and host frame after MPEG-4 coding is shown in **Fig. 8b** with PSNR=36.8 (MPEG encoding reduced frame quality).

After embedding 88 bits in frame 5, uncompressed (without MPEG encoding) stego-frame is shown in **Fig. 8d** with PSNR=44.4, evaluated LSB hiding method with frame 5 with 88 bits embedded in it PSNR=85.39, LSB hiding method comes with high frame quality more than proposed technique but not robust MPEG encoding.

Most current video data embedding techniques are designed for authentication or copyright protection (video watermarking) instead of hiding information. Thus, most video data embedding systems can only embed a few bytes of a company logo or other types of identification information. On the other hand, the size of the private information in the proposed method is far beyond several bytes even after MPEG-4 coded, usually more than 6000 bits per 352x288 frame two times more than in [3].



(a) Host frame 5 (176 x 144)

(b) Frame 5 after MPEG-4 coding (PSNR 36.8)

c) The Y component of host frame 5

(d) Uncompressed frame 5 with 88 bits embedded in it (PSNR 44.4)

Figure 8: (a) host frame from the car video sequence, uncompressed.
(b) Host frame after MPEG-4 encoding (12 frames/second), (PSNR 36.8).
(c) The Y component of car video frame 5.
(d) Frame 5 without MPEG encoding with 88 bits embedded in it (PSNR 44.4).

With use of Peak Signal to Noise Ratio (PSNR) value to evaluate the quality of video frames after embedding process. PSNR value is widely used on the evaluation of difference between the processed image and its original image, if the value is bigger than 30db; it means that it is difficult for human vision to detect the difference between these two images.

$$MSE = \left(\frac{1}{m \times n}\right)\sum_{i=0}^{i<n}\sum_{j=0}^{j<m}\left(c'_{ij} - c_{ij}\right)^2 \tag{15}$$

$$PSNR = 10 \log_{10} \frac{(255)^2}{MSE} db \tag{16}$$

The results were obtained using T=10, T=15 and T=40. **Table 1** shows the Bit Error Rate (BER) of embedded data versus the length of embedded data after performing MPEG encoding once for T=10 where PSNR=42, **Table 2** shows results for T=15 where PSNR =37, **Table 3** shows the same for T=40 but PSNR = 24.2 and **Table 4** shows two MPEG encoding iterations for T=40. The BER for each frame is simply the number of error bits divided by the length of embedded data.

Figure 9 shows the relationship between length of embedded data and PSNR for proposed method (T=15) and the technique used in [2], PSNR decrease with increasing length of embedded data as a result, **Fig. 10** shows the relationship between BER and robustness factor; BER decrease with increasing of robustness factor, of course the more robust the algorithm (high T value) is the more resistance against MPEG encoding will be, however, at the expense of image quality(small PSNR value).

This means that frame quality depends on two factors:

1-Length of embedded data.

2-Robustness factor (T).

Increasing the length of embedded data reduces frame quality and also increasing robustness factor reduces frame quality. For hiding text file inside video file the robustness factor must increase because any bit error will change the ASCII character, but for hiding image file in video file there is no need to increase robustness factor because bit error will give some noise in image file and this is pinsignificant change for image file. Change between the two factors depends on the type of the secret data.

Table 1: BER versus Length of embedded data per frame after performing MPEG encoding for the proposed (T=10) as well as the techniques presented in [2] and [4].

Length of embedded data Per frame (bit)	Proposed Technique T=10	Technique used in [2]	Technique used in [4]
	BER for all frames (%) (PSNR=42)	BER for all frames (%)	BER for all frames (%)
16	11.621	20.625	27.708
24	16.217	21.388	28.333
32	20.911	23.229	30.000
40	22.754	24.167	31.333

Table 2: BER versus Length of embedded data per frame after performing MPEG encoding for the proposed (T=15) as well as the techniques presented in [2] and [4].

Length of embedded data Per frame (bit)	Proposed Technique T=15	Technique used in [2]	Technique used in [4]
	BER for all frames (%) (PSNR=37)	BER for all frames (%)	BER for all frames (%)
16	10.718	20.625	27.708
24	12.572	21.388	28.333
32	17.185	23.229	30.000
40	20.652	24.167	31.333

Table 3: BER versus Length of embedded data per frame after performing MPEG encoding once for the proposed (T=40) as well as the techniques presented in [2, 4].

Length of embedded data Per frame (bit)	Proposed Technique T=40	Technique used in [2]	Technique used in [4]
	BER for all frames (%) (PSNR=24.2)	BER for all frames (%)	BER for all frames (%)
2	0.000	20.000	20.000
4	0.000	18.333	26.667
6	0.000	22.777	30.000
8	0.000	22.083	26.667
10	0.000	23.000	26.000
12	0.000	21.111	26.667
14	0.000	20.714	27.857
16	0.000	20.625	27.708
18	0.000	20.740	26.852
20	0.000	20.833	27.000
22	0.000	20.454	27.122
24	0.000	21.388	28.333
26	0.000	22.949	28.718
28	0.000	23.214	28.333
30	0.000	22.667	29.444
32	0.000	23.299	30.000
34	0.000	23.137	30.294
36	0.000	23.148	30.370
38	0.000	22.544	31.228
40	0.000	24.167	31.333

Table 4: BER versus Length of embedded data per frame after performing MPEG encoding twice for the proposed (T=40) as well as the techniques presented in [2, 4].

Length of embedded data Per frame (bit)	Proposed Technique T=40	Technique used in [2]	Technique used in [4]
	BER for all frames (%) (PSNR=24.2)	BER for all frames (%)	BER for all frames (%)
16	0.182	21.458	29.167
24	0.044	22.500	29.444
32	0.166	24.063	30.000
40	0.260	24.750	31.000

As can be seen from the tables, proposed method robust to MPEG encoding and BER increases if MPEG re-encoding is performed.

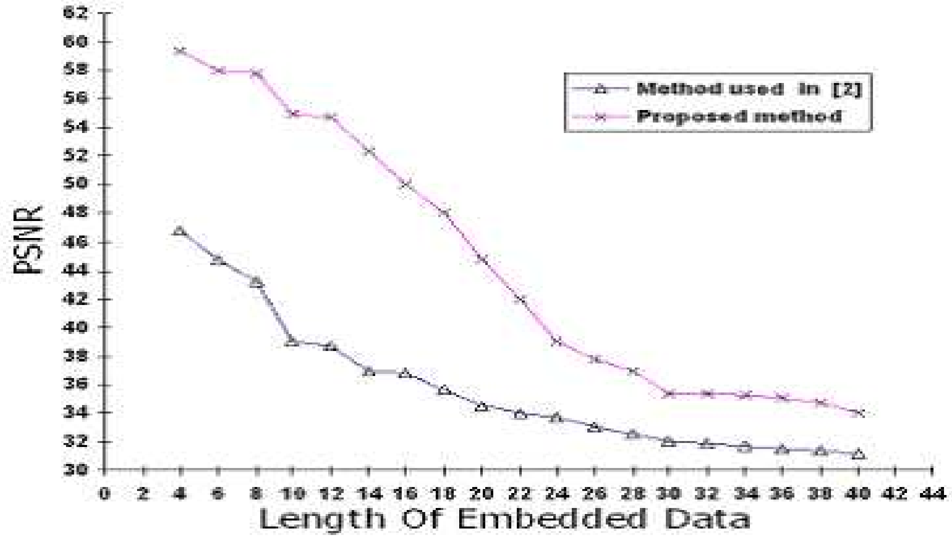


Figure 9: The relationship between PSNR and length of embedded data (T=15).

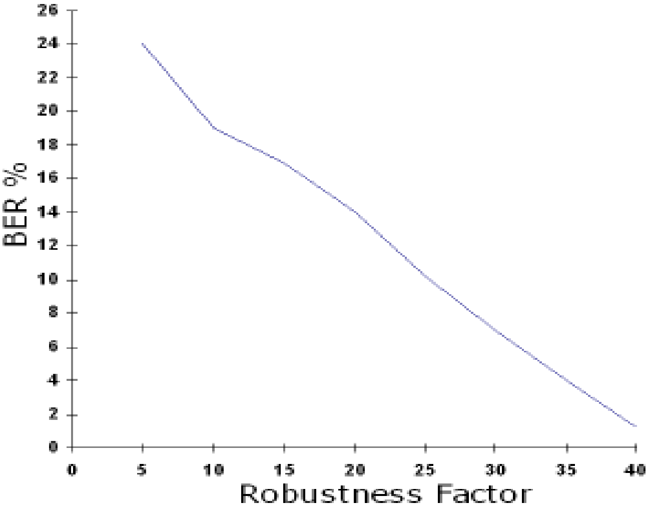


Figure 10: The relationship between BER and T (robustness factor).

5. CONCLUSION

The proposed technique is successfully able to extract embedded information from each frame without using the original video. Besides, the embedded data is robust against MPEG encoding and re-encoding with minimal perceptual distortion. The proposed technique enables high rate of information embedding more than 6000 bits per 352x288 frame that is two times more than the proposed technique in [3], and the information extracted with high efficiency more than what produced by both proposed techniques in [2,4].

REFERENCES

[1] G., Derrick, Data watermarking steganography and watermarking of digital data, Computer Law & Security Report, 17 (2), (2001), 101-104

[2] S. N. Merchant, A. Harchandani, S. Dua, H. Donde, and I. Sunesara, “Watermarking of Video Data Using Integer-to-Integer Discrete Wavelet Transform”, in Proceedings of the IEEE TENCON 2003: Conference on Convergent Technologies for Asia-Pacific Region, (2003), pp. 939- 943.

[3] W. Zhang, S. S. Cheung, and M. Chen, "Hiding Privacy Information in Video Surveillance System", Proceedings of ICIP (2005), Genova, Italy, Sep. 11-14

[4] Hisashi Inoue, Akio Miyazaki, Takashi Araki, and Takashi Katsura, “A Digital Watermark Method Using the Wavelet Transform for Video Data,” IEICE Trans. Fundamentals, vol. E83-A, no.1, January (2000).

[5] D., Gruhl, W., Bender, Information hiding to foil the casual counterfeiter, Second International Workshop on Information Hiding, IH'98, Portland, Oregon, 1-15, (1998).

[6] D. Kahn, The history of steganography, Information Hiding: First International Workshop, Lecture Notes in Computer Science, 1174, 1-6, 1996

[7] M.V., Dijk, on a special class of broadcast channels with condential Messages, IEEE Transactions on Information Theory, 43 (2), 712-714, (1997).

[8] Bender, W., Gruhl, D., Morimoto, N. and Lu, A., “Techniques for Data Hiding,” IBM System Journal, Vol. 35(3&4), pp. 313 336 (1996).

[9] Pfitzmann, B., “Information Hiding Terminology,” Proc. of First Internet Workshop on Information Hiding, Cambridge, UK, pp. 347 350 (1996).

[10] Lee, Y. K. and Chen, L. H., “An Adaptive Image Steganographic Model Based on Minimum-Error Replacement,” Proc. of the Ninth National Conference on Information Security, Taichung, Taiwan, pp. 8 15 (1999).

[11] Liu, J. C. and Chen, S. Y., “Fast two-layer Image Watermarking Without Resorting to Original Image and Watermark,” Proc. of Joint Conference of International Computer Symposium, Taiwan, R.O.C, pp. 231238 (2000).

[12] Yu, P. T., Tsai, H. H. and Kin, J. S. “Digital Watermarking Based on Neural Networks for Color Images,” Signal Processing, Vol. 81, pp. 663 671 (2001).

[13] Moulin, P. and Mihcak, M. K., “A Framework for Evaluating the Data-Hiding Capacity of Image Sources,” IEEE Trans. on Image Processing, Vol. 11(9), pp. 1029 1042 (2002).

[14] Thien, C. C. and Lin, J. C., “A Simple and High-hiding Capacity Method for Hiding Digit-by-digit Data in Images Based on Modulus Function,” Pattern Recogni- tion, Vol. 36, pp. 2875 2881 (2003).

[15] Noda, H., Spaulding J., Shirazi, M. N. and Kawaguchi, E., “Application of Bit-Plane Decomposition Steganography to JPEG2000 Encoded Images,” IEEE Signal Processing Letters, Vol. 9(12), pp. 410 413 (2002).

[16] Wu, D. C. and Tai, W. S., “A Steganographic Method for Images by Pixel-Value Differencing,” Pattern Recognition Letters, Vol. 24, pp. 1613 1626 (2003).

[17] Chang, C. C. and Tseng, H. W., “A Steganographic Method for Digital Images using Side Match,” Pattern Recognition, Vol. 25, pp. 1431 1437 (2004).

[18] Kutter, M., Jordan, F. and Bossen, F., “Digital Signature of Color Image using Amplitude Modulation,” J. Electron. Imaging, Vol. 7(2), pp. 326 332 (1998).

[19] Chang, C. K. and Cheng, L. M., “Hiding Data in Images by Simple LSB Substitution,” Pattern Recognition, Vol. 37, pp. 469 474 (2004).

[20] Chen, T. S., Chang, C. C. and Hwang, M. S., “A Virtual Image Cryptosystem Based upon Vector Quantization,” IEEE Transactions on Image Processing, Vol. 7(10), pp. 1485 1488 (1998).

[21] Gray, R. M. and Neuhoff, D. L., “Quantization,” IEEE Trans. Information Theory, Vol. 44(6), pp. 2325 2383 (1998).

[22] Hsu, C. T. and Wu, J. L., “Hidden Digital Watermarks in images,” IEEE Transactions on Image Processing, Vol. 8(1), pp. 58 68 (1999).

[23] Chen, L. H. and Chang, H. M., “Two New Methods for Visible Digital Watermarking,” Proc. of the Ninth National Conference of Information Security, Taichung, Taiwan, R.O.C., pp. 23 26 (1999).

[24] Cox, I. J., Kilian J., Leighton T. and Shamoon T., “Secure Spread Spectrum Watermarking for Multimedia,” IEEE Transactions on Image Processing, Vol. 6(12), pp. 1673 1687 (1997).

[25] Langelaar, G. C., Lagendijk, R. L. and Biemond J., “Removing Spatial Spread Spectrum Watermarks by Non-linear Filtering,” In 9th European Signal Pro- cessing Conference (EUSIPCO’98), Island of Rhodes, Greece, pp. 2281 2284 (1998).

[26] Wang, Y. and Pearmain A., “Blind Image Data Hiding Based on Self Reference,” Pattern Recognition, Vol.25, pp. 1681 1689 (2004).

[27] Chung, K. L., Shen, C. H. and Chang, L. C., “A Novel SVD- and VQ-based Image Hiding Scheme,” Pattern Recognition Letters, Vol. 22, pp. 1051 1058 (2001).

[28] Swanson, M. D., Xu B. and Tewfik, A. H., “Robust Data Hiding for Images,” In 7th Digital Signal Processing Workshop (DSP96), Loen, Norway, pp. 37 40 (1996).

[29] Hsu, C. T. and Wu, J. L., “DCT-Based Watermarking for Video,” IEEE Transactions on Consumer Electron- ics, Vol. 44(1), pp. 206 215 (1998).

[30] Langelaar, G. C. and Lagendijk, R. L., “Optimal Differential Energy Watermarking of DCT Encoded Images and Video,” IEEE Transactions on Image Pro- cessing, Vol. 10(1), pp. 148 158 (2001).

[31] Smoot, S., and L. A. Rowe, "DCT Coefficient Distributions," in Proceedings of the SPIE 2657, Human Vision and Electronic Imaging, 1996, pp. 403–411.

[32] MPEG Software Simulation Group, http://www.mpeg.org/MSSG/ (12-12-2005)

[33] http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html (15-12-2005).

[34] Pennebaker, W. B. and Mitchell, J. L., “JPEG: Still Image Data Compression Standard,” New York: Van Nostrand Reinhold, (1993).

إخفاء المعلومات في ملفات الفيديو

في خضم التقدم السريع في عالم تكنولوجيا المعلومات ظهرت الحاجة إلى تقدم وسائل الحماية للمعلومات حيث أصبح تسرب المعلومات والاحتيال للحصول عليها باختراق أنظمة الحاسبات أمراً متاح للجميع ومن هنا برزت الحاجة لنظم إخفاء المعلومات حيث يتم إخفاء المعلومات السرية بداخل احد الوسائط المتعددة كملفات الفيديو بحيث تبدوا كمجرد ملفات فيديو عادية كغيرها من الملفات عديمة الأهمية بالنسبة لمن يطمع في الحصول على المعلومات السرية دون أن يتبادر إليه شك في احتواء هذه الملفات على معلومات هامة وفى هذا البحث تقديم لطريقة جديدة لإخفاء المعلومات داخل ملفات الفيديو وتقدم هذه الطريقة مساحة اكبر لإخفاء المعلومات مع قدرة المعلومات المخفية على مقاومة الضغط الشائع **"MPEG "** لملفات الفيديو.