Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error

Hussein A. Aly, *Member, IEEE*

***Abstract*—This paper deals with data hiding in compressed video. Unlike data hiding in images and raw video which operates on the images themselves in the spatial or transformed domain which are vulnerable to steganalysis, we target the motion vectors used to encode and reconstruct both the forward predictive (P)-frame and bidirectional (B)-frames in compressed video. The choice of candidate subset of these motion vectors are based on their associated macroblock prediction error, which is different from the approaches based on the motion vector attributes such as the magnitude and phase angle, etc. A greedy adaptive threshold is searched for every frame to achieve robustness while maintaining a low prediction error level. The secret message bitstream is embedded in the least significant bit of both components of the candidate motion vectors. The method is implemented and tested for hiding data in natural sequences of multiple groups of pictures and the results are evaluated. The evaluation is based on two criteria: minimum distortion to the reconstructed video and minimum overhead on the compressed video size. Based on the aforementioned criteria, the proposed method is found to perform well and is compared to a motion vector attribute-based method from the literature.**

***Index Terms*—Data hiding, motion vectors, Motion Picture Expert Group (MPEG), steganography.**

I. Introduction

DATA hiding [1] and watermarking in digital images and raw video have wide literature. This paper targets the internal dynamics of video compression, specifically the motion estimation stage. We have chosen this stage because its contents are processed internally during the video encoding/decoding which makes it hard to be detected by image steganalysis methods and is lossless coded, thus it is not prone to quantization distortions. In the literature, most work applied on data hiding in motion vectors relies on changing the motion vectors based on their attributes such as their magnitude, phase angle, etc. In [2] and [3], the data bits of the message are hidden in some of the motion vectors whose magnitude is above a predefined threshold, and are called candidate motion vectors (CMVs). A single bit is hidden in the least significant bit of the larger component of each CMV. In [4], the data is encoded as a region where the motion estimation is only allowed to generate motion vectors in that specified region. Using the variable macroblock sizes $(16 \times 16,\ 16 \times 8,\ 8 \times 16,\ 8 \times 8)$ of H.264, the authors in [5] used every 2 bits from the message bitstream to select one of the four sizes for the motion estimation process. The authors in [6] and [7] embed the data in video using the phase angle between two consecutive CMV. These CMV are selected based on the magnitude of the motion vectors as in [2]. The message bitstream is encoded as phase angle difference in sectors between CMV. The block matching is constrained to search within the selected sector for a magnitude to be larger than the predefined threshold.

The methods in [2]–[7] focused on finding a direct reversible way to identify the CMV at the decoder and thus relied on the attributes of the motion vectors. In this paper, we take a different approach directed towards achieving a minimum distortion to the prediction error and the data size overhead. This approach is based on the associated prediction error and we are faced by the difficulty of dealing with the nonlinear quantization process; thus we use an adaptive threshold as discussed in Section IV.

The rest of the paper is organized as follows: in Section II we overview the terms of video compression and decompression. The problem definition is given in Section III along with the evaluation criteria used in the paper. Our proposed method is given in Section IV followed by the results and analyses in Section V. Finally, the paper is concluded in Section VI.

II. Background and Notations

In this section, we overview lossy video compression to define our notation and evaluation metrics. At the encoder, the intrapredicted (I)-frame is encoded using regular image compression techniques similar to JPEG but with different quantization table and step; hence the decoder can reconstruct it independently. The I-frame is used as a reference frame for encoding a group of forward motion-compensated prediction (P)- or bidirectionally predicted (B)-frames. In the commonly used Motion Picture Expert Group (MPEG-2) standard [8], the video is ordered into groups of pictures (GOPs) whose frames can be encoded in the sequence: [I,B,B,P,B,B,P,B,B]. The temporal redundancy between frames is exploited using block-based motion estimation that is applied on macroblocks $\mathfrak{B}_{ij}$ of size $b \times b$ in P or B and searched in target frame(s). Generally, the motion field in video compression is assumed to be translational with horizontal component d^x and vertical component d^y and denoted in vector form by $\mathbf{d}(\mathbf{x})$ for the spatial variables $\mathbf{x} = (x, y)$ in the underlying image. The search window is constrained by assigning limited n-bits for $\mathbf{d}$; in other words, both d^x and $d^y \in [-2^{n-1} - 1, 2^{n-1} - 1]$, which corresponds to $[-2^{n-2} - 1/2, 2^{n-2} - 1/2]$ pixels if the motion vectors are computed with half-pixel accuracy. An exhaustive search in the window of size $b + 2^n \times b + 2^n$ can be done to find the optimal

Manuscript received June 11, 2010; revised September 11, 2010; accepted October 24, 2010. Date of publication November 01, 2010; date of current version February 16, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wenjun Zeng.

The author is with the Military Technical College (MTC), Ministry of Defense, Cairo, Egypt (e-mail: haly@ieee.org; h.aly@alumni.uottawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2090520

1556-6013/$26.00 © 2010 IEEE

Authorized licensed use limited to: Tunku Abdul Rahman University of Management & Technology. Downloaded on February 22,2023 at 09:15:50 UTC from IEEE Xplore. Restrictions apply.

motion vector satisfying the search criterion which needs many computations, or suboptimal motion vectors can be obtained using expeditious methods such as three steps search, etc.; this is based on the video encoding device processing power, the required compression ratio, and the reconstruction quality. Since $\mathbf{d}$ does not represent the true motion in the video then the compensated frame $\tilde{P}$ using $(\mathbf{x}+\mathbf{d}(\mathbf{x}))$ must be associated with a prediction error $E(\mathbf{x}) = (P - \tilde{P})(\mathbf{x})$ in order to be able to reconstruct $P = \tilde{P} + E$ with minimum distortion at the decoder in case of a P-frame. Similar operation is done for the B-frame but with the average of both the forward compensation from a previous reference frame and backward compensation from a next reference frame. E is of the size of an image and is thus lossy compressed using JPEG compression reducing its data size. The lossy compression quantization stage is a nonlinear process and thus for every motion estimation method, the pair $(\mathbf{d}, E)$ will be different and the data size $\mathfrak{D}$ of the compressed error $\tilde{E}$ will be different. The motion vectors $\mathbf{d}$ are lossless coded and thus become an attractive place to hide a message that can be blindly extracted by a special decoder.

The decoder receives the pair $(\mathbf{d}, \tilde{E})$, applies motion compensation to form $\tilde{P}$ or $\tilde{B}$, and decompresses $\tilde{E}$ to obtain a reconstructed E_r. Since E and E_r are different by the effect of the quantization, then the decoder in unable to reconstruct P identically but it alternatively reconstructs $P_r = \tilde{P} + E_r$. The reconstruction quality is usually measured by the mean squared error $P - P_r$, represented as peak signal-to-noise ratio (PSNR), and we denote it by $\mathfrak{R}$.

III. Problem Definition

Data hiding in motion vectors at the encoder replaces the regular pair $(\mathbf{d}, \tilde{E})$, due to tampering the motion vectors, to become $(\mathbf{d}^h, \tilde{E}^h)$, where the superscript h denotes hiding. We define data hiding in motion vectors of compressed video in the context of super-channel [9]; the secret message m is hidden in the host video signal $x = (\mathbf{d}, E)$ to produce the composite signal $s = (\mathbf{d}^h, E^h)$. The composite signal is subject to video lossy compression to become $y = (\mathbf{d}^h, \tilde{E}^h)$. The message m should survive the video lossy compression and can be identically extracted from y. This robustness constrain should have low distortion effect on the reconstructed video as well as low effect on the data size (bit rate). Given that m can be identically extracted, in this paper, we use two metrics to evaluate data-hiding algorithms in compressed video which are:

1) increase in data size: $\Delta\mathfrak{D} = \mathfrak{D}(\tilde{E}^h) - \mathfrak{D}(\tilde{E})$ representing the overhead price paid for the embedded data;
2) drop in the reconstruction quality: this reconstruction is with quality loss than that without data hiding and is denoted by $\Delta\mathfrak{R}$ and expressed as the PSNR difference which is as well that of the quantity of the relative error $P - P_r^h / P - P_r$ and $B - B_r^h / B - B_r$ for P- and B-frame, respectively.

Our objective is to provide a good data-hiding algorithm that should maintain $\Delta\mathfrak{D}$ and $\Delta\mathfrak{R}$ as close to zero as possible for a given data payload. The payload should be robust to the video compression, specifically the quantization step applied to the prediction error E.

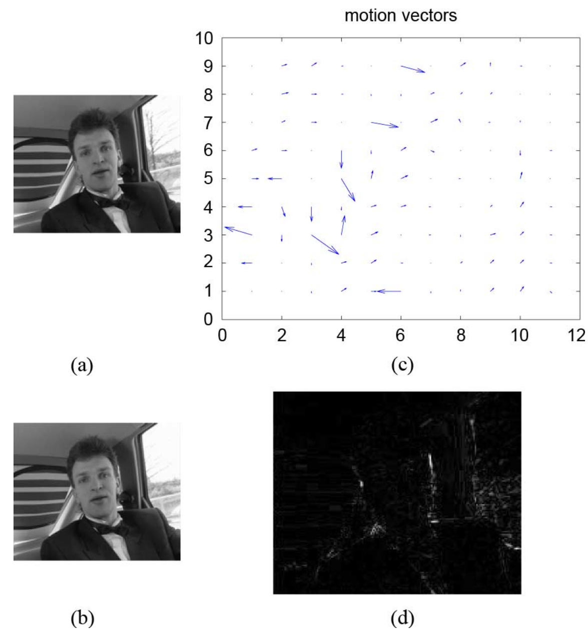

Fig. 1. Sample from the Car-phone sequence. (a) Reference frame I. (b) Predictive frame P. (c) Motion vectors $\mathbf{d}$ and (d) the associated prediction error E_r.

The selection of the CMV is the key difference between different methods. For instance [2] and [3] choose the CMV based on their magnitude $\bar{\mathbf{d}} = \{\mathbf{d} : \|\mathbf{d}\| < \text{threshold}\}$. On the other hand [6] and [7] rely on the magnitude and the phase between consecutive motion vectors. Their idea is that motion vectors with large magnitude are less likely to represent the real underlying motion accurately and thus their associated macroblock prediction error E is expected to be large. Tampering these CMVs will not affect the reconstruction quality that much. Analyzing this relation, we found it not to be usually correct as shown in Fig. 1 for a sample from the car-phone sequence:

1) not all motion vectors with large magnitude are associated with macroblocks of high prediction error; and
2) there are motion vectors whose magnitude is small but their associated macroblock prediction error is high.

These observations stimulated our proposal to rely directly on the associated macroblock prediction error, such that we choose *our CMV associated with macroblocks of high prediction error*. If we tamper with these CMVs, then we will not have poor effect on the video reconstruction quality. Since PSNR is a reciprocal of the mean squared error (mse), then our selection criteria in this paper can be thought of as $\bar{\mathbf{d}} = \{\mathbf{d} : 10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E(\mathbf{x})) \le \tau\}$. In this direction, we choose the CMV based on the pair $(\mathbf{d}, E)$ and not $\mathbf{d}$ alone. However, this incurs the difficulty that E is lossy compressed and what we have at the decoder after decompression is actually E_r. We deal with this problem using adaptive threshold τ as discussed in Section IV.

IV. Proposed Method

Our data-hiding algorithm is applied at the encoder side, uses the regular pair $(\mathbf{d}, \tilde{E})$ produced, tampers d to become d^h, and thus replaces them by the pair $(\mathbf{d}^h, \tilde{E}^h)$ for each P and B-frame

Authorized licensed use limited to: Tunku Abdul Rahman University of Management & Technology. Downloaded on February 22,2023 at 09:15:50 UTC from IEEE Xplore. Restrictions apply.

in the GOP as shown in Algorithm 1. The secret message is organized as a bitstream $m(k),\ 0 < k \leq K$: message length. A subset of $\mathbf{d}$ is selected to be the CMV $\bar{\mathbf{d}}$. The selection of $\bar{\mathbf{d}}$ (line 6 of Algorithm 1) is performed if their associated macroblock $\mathfrak{B}_{i,j}$ prediction error measured in PSNR is below an initial threshold value $\tau_{\max}$. The least significant bit (LSB) of both components $\bar{d}^x$, $\bar{d}^y$ are replaced by bits of the message. After data embedding (lines 7 to 13 of Algorithm 1), we validate the used value of τ by calling Algorithm 2. The algorithm tests the robustness of the hidden message to the quantization effect of the JPEG compression. For the prediction error E^h, it performs the compression by the encoder followed by the decompression performed by the decoder (lines 1 and 2 of Algorithm 2). If the reconstructed prediction error E_r^h maintains the same criterion $(10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r^h(\mathbf{x})) < \tau_{\text{key}})$, then $\bar{\mathbf{d}}$ can be identified by the data extractor for the given value of τ_{key}. If any macroblock associated with $\bar{\mathbf{d}}$ fails to maintain the criterion (line 5 of Algorithm 2), then $\bar{\mathbf{d}}$ will not be identified by the data extractor and the message will not be extracted correctly. Hence, we propose to use an adaptive threshold by iteratively decrementing $\tau_{\max}$ by 1 decibel (dB) for this frame until either the criterion is satisfied for all macroblocks or the stopping value $\tau_{\min}$ is reached for which we embed no data in this frame (line 19 in Algorithm 1). Since the threshold used for each frame τ_{key} is different, we hide their eight values for that GOP in the I-frame using any robust image data-hiding technique or sending them on a separate channel based on the application. Decreasing τ_{key} will decrease the payload and vice versa, thus Algorithm 1 tries to find the maximum feasible τ_{key} for each frame.

Algorithm 1 Data Hiding in GOP

Input: message bitstream m, $\text{GOP}(d, \tilde{E})$, k, $\tau_{\max}$, $\tau_{\min}$
Output: Data embedded in the Encoded GOP $(d^h, \tilde{E}^h)$
1 **foreach** P and B-frame in the GOP **do**
2 initialize $\tau_{\text{key}} = \tau_{\max}$;
3 Simulate the decoder: decompress $\tilde{E}$ to obtain E_r;
4 **repeat**
5 set $\mathbf{d}^h = \mathbf{d}$;
6 Obtain the candidate motion vectors: $\bar{\mathbf{d}}_{i,j}(\mathbf{x}) = \{\mathbf{d}_{i,j}(\mathbf{x}) : 10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r(\mathbf{x})) \leq \tau_{\text{key}}\}$;
7 **while** $(k \leq K)$ & $\forall(i,j) \in \bar{\mathbf{d}}_{i,j}(\mathbf{x})$ **do**
8 replace the least significant bit $\text{LSB}(\bar{d}^x_{i,j}) = m(k)$, $\text{LSB}(\bar{d}^y_{i,j}) = m(k+1)$;
9 $k = k + 2$;
10 **if** B-frame **then**
11 replace for the backward compensation motion vectors the least significant bit $\text{LSB}(\bar{d}^x_{i,j}) = m(k)$, $\text{LSB}(\bar{d}^y_{i,j}) = m(k+1)$;
12 $k = k + 2$;
13 **end**
14 $\mathbf{d}^h_{i,j} = \bar{\mathbf{d}}_{i,j}$;
15 **end**
16 Compute associated $E^h(\mathbf{x})$ by suitable compensation using $(\mathbf{x} + \mathbf{d}^h(\mathbf{x}))$;
17 $[\text{KeyFound}, \tau_{\text{key}}] \longleftarrow \text{validate}\ \tau(E^h, \tau_{\text{key}}, \bar{d})$;
18 **until** KeyFound or $\tau_{\text{key}} = \tau_{\min}$;
19 **if** not KeyFound **then**
20 $\tau_{\text{key}} = -1$
21 **end**
22 Hide τ_{key} in I-frame or send on a separate channel;
23 **end**

Algorithm 2 Validate τ

Input: E^h, τ_{key}, $\bar{d}$
Output: KeyFound, τ_{key}
1 Compress E^h using JPEG compression to produce $\tilde{E}^h$;
2 Decompress $\tilde{E}^h$ to obtain lossy E_r^h;
3 set KeyFound=True;
4 **while** KeyFound & $(i,j) \in \bar{\mathbf{d}}_{i,j}(\mathbf{x})$ **do**
5 **if** $10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r^h(\mathbf{x})) > \tau_{\text{key}}$ **then**
6 KeyFound = False;
7 decrement τ_{key};
8 **end**
9 **end**

The decoder receives the pair $(\mathbf{d}^h, \tilde{E}^h)$ and it can decode $\mathbf{d}^h$ without loss and decompresses $\tilde{E}^h$ to obtain a lossy reconstructed version E_r^h. During normal operation for viewing the video, the decoder is able to reconstruct P_r^h or B_r^h by suitable compensation from reference frame(s) using $(\mathbf{x} + \mathbf{d}^h(\mathbf{x}))$ and adding E_r^h to it. Acting as a new kind of motion estimation, Algorithm 1 will have two effects on the new compressed video: change in data size and reconstruction quality which are thoroughly analyzed in Section V. The data extractor operates to extract the hidden message as a special decoder and our proposal is straightforward, as shown in Algorithm 3. After data extraction from the consecutive GOPs the hidden message m is reconstructed back by concatenation of the extracted bitstream.

Algorithm 3 Data Extraction

Input: GOP $(d^h, \tilde{E}^h)$, k
Output: message bitstream m
Extract the thresholds τ_{key} for all frames in GOP from I-frame or use them from other channel;
foreach P & B frame in the GOP **do**
Decompress $\tilde{E}^h$ to obtain E_r^h, and identify the candidate motion vectors: $\bar{\mathbf{d}}_{i,j}(\mathbf{x}) = \{\mathbf{d}^h_{i,j}(\mathbf{x}) : 10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r^h(\mathbf{x})) \leq \tau_{\text{key}}\}$
foreach $(i,j) \in \bar{\mathbf{d}}_{i,j}(\mathbf{x})$ **do**
Extract 2 message bits $m(k) = \text{LSB}(\bar{d}^x_{i,j})$, $m(k+1) = \text{LSB}(\bar{d}^y_{i,j})$;
$k = k + 2$;
if B-frame **then**
Extract from backward compensation motion vectors 2 message bits $m(k) = \text{LSB}(\bar{d}^x_{i,j})$, $m(k+1) = \text{LSB}(\bar{d}^y_{i,j})$;
$k = k + 2$;
end
end
end

Authorized licensed use limited to: Tunku Abdul Rahman University of Management & Technology. Downloaded on February 22,2023 at 09:15:50 UTC from IEEE Xplore. Restrictions apply.

(a) (b) (c) (d) (e) (f)

Fig. 2. First image of the test sequences. (a) Car-phone; (b) Foreman; (c) Football; (d) Coastguard; (e) Mobile; and (f) Garden.

TABLE I
QUANTITATIVE PERFORMANCE MEASURES $\Delta\mathfrak{D}$, $\Delta\mathfrak{R}$ OF OUR METHOD AND THAT IN [3] FOR DIFFERENT SEQUENCES

Experiment setup		**Proposed method**			**method in [3]**		
Test sequence	motion estimation search method	total payload (Bytes)	average $\Delta\mathfrak{R}$ per frame (dB)	total $\Delta\mathfrak{D}$ (Bytes)	total (Bytes) payload for threshold T (pixels)	average $\Delta\mathfrak{R}$ per frame (dB)	total $\Delta\mathfrak{D}$ (Bytes)
Car-phone (99$\mathfrak{B}$/frame)	3-steps	4811	-0.325	10645	3995 (T=0.5)	-0.560	28148
(270 frames = 30 GOP)	Exhaustive	4438	-0.404	5761	4032 (T=0.5)	-0.662	23211
Foreman (396$\mathfrak{B}$/frame)	3-steps	6577	-0.244	30379	5722 (T=1.5)	-0.346	86818
(135 frames = 17 GOP)	Exhaustive	5695	-0.220	34974	5329 (T=1.5)	-0.369	99721
Football (330$\mathfrak{B}$/frame)	3-steps	9592	-0.269	32011	6211 (T=0.5)	-0.317	57077
(117 frames = 13 GOP)	Exhaustive	9460	-0.290	37541	6250 (T=0.5)	-0.341	58743
Coastguard (396$\mathfrak{B}$/frame)	3-steps	17911	-0.219	46600	18259 (T=1.0)	-0.516	156457
(270 frames = 30 GOP)	Exhaustive	18020	-0.267	41281	18327 (T=1.0)	-0.524	171505
Mobile (396$\mathfrak{B}$/frame)	3-steps	19921	-0.517	196941	19683 (T=0.5)	-0.945	392576
(270 frames = 30 GOP)	Exhaustive	19845	-0.595	244727	19708 (T=0.5)	-1.059	455802
Flower-garden (396$\mathfrak{B}$/frame)	3-steps	20078	-0.471	150620	18229 (T=0.5)	-0.646	265381
(252 frames = 28 GOP)	Exhaustive	19785	-0.547	160077	18303 (T=0.5)	-0.701	284086

V. RESULTS

We implemented the hiding and extraction Algorithms 1, 2, and 3 and integrated them to the MPEG-2 encoder and decoder operation. The parameters of our experiments, presented in this section, are: macroblock size $b = 16$, motion vector representation bits $n = 5$. We used both the fast three-steps and exhaustive search motion estimation algorithms with half pixel accuracy. Each test video sequence is organized into consecutive GOP organized as [I,B,B,P,B,B,P,B,B]. The compression to the I-frame and the prediction error of the P- and B-frames are implemented using JPEG compression with a quality factor 75, 70, and 30, respectively. We tested our algorithms on six standard test sequences: car-phone, foreman, coastguard, football, flower-garden, and mobile sequence which are all shown in Fig. 2. All the foreman, coastguard, flower-garden, and mobile sequence have a frame size of 352×288 which corresponds to 396 macroblocks $\mathfrak{B}$ per frame. The number of macroblocks per frame and the total number of frames for each sequence are given in the first column of Table I. The motion estimation method used is given in the second column of Table I. Thus we performed 12 different experiments on natural video sequences with different levels of motion. The sequences football, mobile, and flower-garden have high motion dynamics, while the coastguard sequence have moderate motion, and finally the foreman and car-phone sequences have an almost static background with human face motion at the foreground. Our algorithm may hide a maximum of $2 \times \mathfrak{B}/8$ bytes per P-frame and $4 \times \mathfrak{B}/8$ per B-frame. Analyzing the PSNR values of the prediction error E_r for all sequences, we set $\tau_{\max} = 60$, $\tau_{\min} = 20$ dB for P-frames, and $\tau_{\max} = 40$, $\tau_{\min} = 15$ dB for B-frames. We evaluated our algorithm and compared it to an attribute-based method [3] which is dependent on a threshold T of the magnitude of the motion vectors. We have chosen the threshold T for [3] that produces the closest total number of embedded bytes (payload) to that of our algorithm for the whole test sequence. The payload for both methods and the associated threshold T in values of pixels for [3] are shown in the first column of each method's results in Table I. For each sequence we calculated the average over all frames for the drop in PSNR $\Delta\mathfrak{R}$ which indicates the quality degradation of the reconstructed video in effect to the hiding; $\Delta\mathfrak{R}$ are shown in the second columns for both methods' results. Finally, the data size increase due to hiding the data is measured for each frame and the total data size increase for all frames are given in the third column of the results of both methods. Analyzing the results in Table I, we find that for approximately the same payload, our hiding method produces less distortion to the video as $-\Delta\mathfrak{R}/$total payload is smaller than that in [3] and generally the distortion is less than 0.6 dB which is nearly invisible. The effect on the data size increase is less than that in [3] which is accounted for our hiding criteria $10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r(\mathbf{x})) \leq \tau\}$ that selects those $\mathfrak{B}_{i,j}$ whose prediction error is high and refrain from tampering those associated with low error.

Another comparison that we present is the distribution of the hidden data between the P- and B-frames. According to the GOP organization among I-, P-, and B-frames and the fact that a B-frame has double the number of motion vectors than a P-frame, we approximately have 85% of the motion vectors associated with B-frames and only 15% with P-frames. The motion vectors of the B-frames can hold a bigger share of the payload than those of the P-frames. However, tampering the motion vectors of B-frames will have more distortive effect on the

Authorized licensed use limited to: Tunku Abdul Rahman University of Management & Technology. Downloaded on February 22,2023 at 09:15:50 UTC from IEEE Xplore. Restrictions apply.

TABLE II
PERCENTAGE OF HIDDEN DATA PAYLOAD IN THE P- AND B-FRAMES USING OUR METHOD AND THAT IN [3] FOR DIFFERENT SEQUENCES USING EXHAUSTIVE SEARCH

Test Sequence	Proposed method		method in [3]	
	P-frames	B-frame	P-frame	B-frame
Car-phone	27%	73%	15%	85%
Foreman	36%	64%	18%	82%
Football	22%	78%	14%	86%
Coastguard	23%	77%	15%	85%
Mobile	21%	79%	15%	85%
Flower-garden	21%	79%	15%	85%

video than that by P-frames, as the B-frames are bidirectionally optimized to minimize the prediction error. We counted the percentage of hidden bytes in both the P- and B-frames for the six test sequences using exhaustive search motion estimation for our method and that in [3], where the total size of the payload is as given in Table I. The results are presented in Table II and we observe that our method tends to utilize more motion vectors from the P-frames to increase their share above the 15% fair share percentage utilized by [3]. Less usage of the motion vectors of the B-frames is an additional reason that accounts for the enhanced results of our method in both aspects of the data size and the quality of the reconstructed video.

VI. CONCLUSION

We proposed a new data-hiding method in the motion vectors of MPEG-2 compressed video. Unlike most data-hiding methods in the motion vectors that rely their selection on attributes of the motion vectors, we chose a different approach that selects those motion vectors whose associated macroblocks prediction error is high (low PSNR) to be the candidates for hiding a bit in each of their horizontal and vertical components. A greedy search for the suitable value of the threshold to be used for choosing the macroblocks corresponding to the CMV is done such that the candidates will be identically identified by the decoder even after these macroblocks have been lossy compressed. The embedding and extraction algorithms are implemented and integrated to the MPEG-2 encoder/decoder and the results are evaluated based on two metrics: quality distortion to the reconstructed video and data size increase of the compressed video. The method is compared to another one from the literature that relies on a motion vector attribute. The proposed method is found to have lower distortion to the quality of the video and lower data size increase. Future work will be directed towards increasing the size of the embedded payload while maintaining the robustness and low distortions.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[2] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," in *Proc. XIV Symp. Computer Graphics and Image Processing*, Oct. 2001, pp. 179–182.

[3] C. Xu, X. Ping, and T. Zhang, "Steganography in compressed video stream," in *Proc. Int. Conf. Innovative Computing, Information and Control (ICICIC'06)*, 2006, vol. II, pp. 803–806.

[4] P. Wang, Z. Zheng, and J. Ying, "A novel video watermark technique in motion vectors," in *Int. Conf. Audio, Language and Image Processing (ICALIP)*, Jul. 2008, pp. 1555–1559.

[5] S. K. Kapotas, E. E. Varsaki, and A. N. Skodras, "Data hiding in H.264 encoded video sequences," in *IEEE 9th Workshop on Multimedia Signal Processing (MMSP07)*, Oct. 2007, pp. 373–376.

[6] D.-Y. Fang and L.-W. Chang, "Data hiding for digital video with phase of motion vector," in *Proc. Int. Symp. Circuits and Systems (ISCAS)*, 2006, pp. 1422–1425.

[7] X. He and Z. Luo, "A novel steganographic algorithm based on the motion vector phase," in *Proc. Int. Conf. Comp. Sc. and Software Eng.*, 2008, pp. 822–825.

[8] *Generic Coding of Moving Pictures and Associated Audio Information: Video, 2 Edition*, SO/IEC13818-2, 2000.

[9] B. Chen and G. W. Wornell, "Quantization index modulation for digital watermarking aand information embedding of multimedia," *J. VLSI Signal Process.*, vol. 27, pp. 7–33, 2001.

Hussein A. Aly (M'00) received the B.Eng. degree (honours) in computer engineering and the M.Eng. degree in electrical engineering from the Military Technical College (MTC), Cairo, Egypt, in 1993 and 1997, respectively, and the PhD. degree in electrical engineering from the University of Ottawa, Canada, in 2004.

He works for the MTC, Ministry of Defense, Egypt. He is currently serving as the head of the computer department. His research interests are in multidimensional digital signal processing theory for image and video sampling and interpolation, modeling physical imaging process and the applications of PDEs to image processing, embedded systems for video and digital light processing, multisensor satellite image fusion, image-based rendering, fast implementations on the graphics processing unit (GPU), and data hiding in digital video. While at the University of Ottawa he participated in a project on image magnification for the Royal Canadian Mounted Police (RCMP).

Dr. Aly is a member of the Egypt Engineers Syndicate.

Authorized licensed use limited to: Tunku Abdul Rahman University of Management & Technology. Downloaded on February 22,2023 at 09:15:50 UTC from IEEE Xplore. Restrictions apply.