M. Rama Koteswara Rao, Bhookya Nageswararao Naik / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue4, July-August 2012, pp.1577-1582

A robust secured datahiding technique in videos and associated estimated artifacts

M. RAMA KOTESWARA RAO
M.Tech Student Scholar, DECS,
Dept of Electronics and Communication Engineering,
Nalanda Institute of Engineering and technology,
Sattenapalli (M); Guntur (Dt); A.P, India.

BHOOKYA NAGESWARARAO NAIK
M.Tech , Asst Professor
Dept of Electronics and Communication Engineering,
Nalanda Institute of Engineering and technology,
Sattenapalli (M); Guntur (Dt); A.P, India.

Abstract

New activities have been recently launched in order to challenge the H.264 standard. Several improvements of this standard are already known, however the targeted 50% bit rate saving for equivalent quality is not yet achieved. In this context, a previous work proposes to use data hiding techniques to reduce the signaling information resulting from an improvement of Inter-coding. The main idea is to hide the indices into appropriately transform coefficients. To minimize the prediction errors, the modification is performed via a rate-distortion optimization. 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 steganalys is, 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 macro block 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 bit stream 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 and to compare with the .avi file format.

Keywords- **Data hiding, motion vectors, Motion Picture Expert Group (MPEG), steganography.**

I. INTRODUCTION

The widespread of the Internet and World Wide Web has changed the way digital data is

handled. The easy access of images, musical documents and movies has modified the development of data hiding, by placing emphasis on copyright protection, content-based authentication, tamper proofing, annotation and covert communication. Data hiding deals with the ability of embedding data into a digital cover with a minimum amount of perceivable degradation, i.e., the
embedded data is invisible or inaudible to a human observer. Data hiding consists of two sets of data, namely the cover medium and the embedding data, which is called the message. The digital medium or the message can be text, audio, picture or video depending on the size of the message and the capacity of the cover.

Early video data hiding approaches were proposing still image watermarking techniques extended to video by hiding the message in each frame independently [1]. Methods such as spread spectrum are used, where the basic idea is to distribute the message over a wide range of frequencies of the host data. Transform domain is generally preferred for hiding data since, for the same robustness as for the spatial domain, the result is more pleasant to the Human Visual System (HVS). For this purpose the DFT (Discrete Fourier Transform), the DCT (Discrete Cosine Transform), and the DWT (Discrete Wavelet Transform) domains are usually employed [2-4].

Recent video data hiding techniques are focused on the characteristics generated by video compressing standards. Motion vector based schemes have been proposed for MPEG algorithms [5]. Motion vectors are calculated by the video encoder in order to remove the temporal redundancies between frames. In these methods the original motion vector is replaced by another locally optimal motion vector to embed data. Only few data hiding algorithms considering the properties of H.264 standard [8-10] have recently appeared in the open literature. In [8] a subset of the 4×4 DCT coefficients are modified in order to achieve a robust watermarking algorithm for H.264. In [6] the blind algorithm for copyright protection is based on the intra prediction mode of the H.264 video coding standard.

The well established H.264/AVC video coding standard has various motion-compensation units in sizes of 16×16, 16×8, 8×16, 8×8, and sub8×8 . For sub8×8, there are further four sub-partitions of sub8×8, sub8×4, sub4×8, and sub4×4. In this paper we propose a new data hiding scheme, which takes advantage of the different block sizes used by the H.264 encoder during the inter prediction, in order to hide the desirable data. The message can be extracted directly from the encoded stream without knowing the original host video. This method is best suited for content-based authentication and covert communication applications.

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. 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.

The methods is mainly 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.

II. BACKGROUNDS AND SOME NOTATIONS

The, Overview of the lossy video compression to define our notation and evaluation metrics. At the encoder, the intra-predicted (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 bi-directionally predicted (B)-frames. In the commonly used Motion Picture Expert Group (MPEG-2) standard , 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 macro blocks B**ij** of size b x 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 d(x) for the spatial variables **x=(x,y)** in the underlying image. The search window is constrained by assigning limited n-bits for **d** ; in other words, both

$$d^x \text{ and } d^y \in [-2^{n-1} - 1, 2^{n-1} - 1]$$

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 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 **d** does not represent the true motion in the video then the compensated frame **P** using **(x+d(x))** must be associated with a prediction error **E(x)=(P-*P)(x)*** in order to be able to reconstruct **P=*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 **(d,E)** will be different and the data size **D** of the compressed error **E** will be different. The motion vectors **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 **(d, Ê)** applies motion compensation to form $\tilde{P}$ or $\tilde{B}$ and decompresses Ê to obtain a reconstructed **Er.** Since **E** and **Er** are different by the effect of the quantization, then the decoder in unable to reconstruct identically but it alternatively reconstructs **Pr=P+Er** The reconstruction quality is usually measured by the mean squared error **P-Pr** represented as peak signal-to-noise ratio (PSNR), and we denote it by $\Re$.

III. PROBLEM DEFINITION

Data hiding in motion vectors at the encoder replaces the regular pair **(d, Ê)** due to tampering the motion vectors, to become $(\mathbf{d}^h, \tilde{E}^h)$ where the superscript 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=(d,E)*** to produce the composite signal $s = (\mathbf{d}^h, E^h)$ The composite signal is subject to video lossy compression to become $y = (\mathbf{d}^h, E^h)$ The message should survive the video lossy compression and can be identically extracted from . 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** .

The selection of the CMV is the key difference between different methods. For instance [2] and [3] choose the CMV based on their magnitude $\bar{\mathbf{d}} = \{\mathbf{d} : \|\mathbf{d}\| <$ 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 macro block 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 macro blocks of high prediction error; and

2) there are motion vectors whose magnitude is small but their associated macro block prediction error is high.

These observations stimulated our proposal to rely directly on the associated macro block prediction error, such that we choose *our CMV associated with macro blocks 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

$$\bar{\mathbf{d}} = \{\mathbf{d} : 10\log_{10}(b^2/\textstyle\sum_{\mathfrak{B}_{i,j}} E(\mathbf{x})) \le \tau\}$$

In this direction, we choose the CMV based on the pair (**d,E**) and not **d** alone However, this incurs the difficulty that **E** is lossy compressed and what we have at the decoder after decompression is actually **Er.**

IV. PROPOSED CONCEPT

Our data-hiding algorithm is applied at the encoder side, uses the regular pair $(\mathbf{d}, E)$ 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 in the GOP as shown in Algorithm 1. The secret message is organized as a bit stream $m(k), 0 < k < K$ message length. A subset of **d** is selected to be the CMV . The selection of (line 6 of Algorithm 1) is performed if their associated macro block $\mathfrak{B}_{i,j}$ prediction error measured in PSNR is below an initial threshold value . The least significant bit (LSB) of both components , are replaced by bits of the message. After data embedding (lines 7 to 13 of Algorithm 1), we validate the used value of 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

$$(10\log_{10}(b^2/\textstyle\sum_{\mathfrak{B}_{i,j}} E_r^h(\mathbf{x})) < \tau_{\text{key}})$$

can be identified by the data extractor for the given value of τ_{key}. If any macro block associated with fails to maintain the criterion (line 5 of Algorithm 2), then 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 macro blocks 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 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 will decrease the payload and vice versa, thus Algorithm 1 tries to find the maximum feasible for each frame[9].

V. ALGORITHMS

Algorithm 1: Data Hiding in GOP

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

Algorithm 2: Validate τ

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

The decoder receives the pair $(d^h, \tilde{E}^h)$ and it can decode without loss and decompresses to obtain a lossy reconstructed version E^h_r. During normal operation for viewing the video, the decoder is able to reconstruct P^h_r or B^h_r by suitable compensation from reference frame(s) using $(\mathbf{x} + \mathbf{d}^h(\mathbf{x}))$ and adding 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. 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 bit stream.

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^h_r, and identify the candidate motion vectors: $\bar{\mathbf{d}}_{i,j}(\mathbf{x}) = \{\mathbf{d}^h_{i,j}(\mathbf{x}) : 10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E^h_r(\mathbf{x})) \leq \tau_{\text{key}}\}$
  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

VI. MATLAB 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: macro block 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 two standard test sequences: policemen and bulb video sequence, which are shown in Figure 1 All the policemen and bulb video sequence have a frame size of 352 x 288 which corresponds to 396 macro blocks $\mathfrak{B}$ per frame

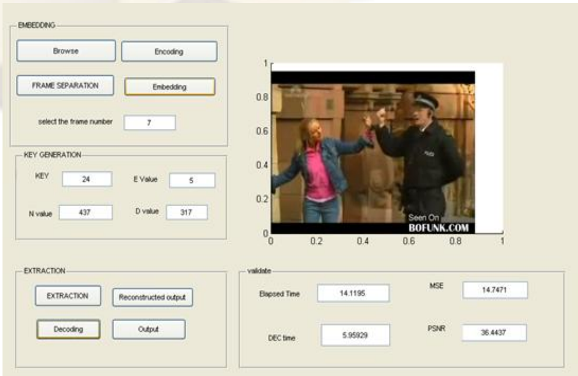

Figure 1 : A policemen test sequence

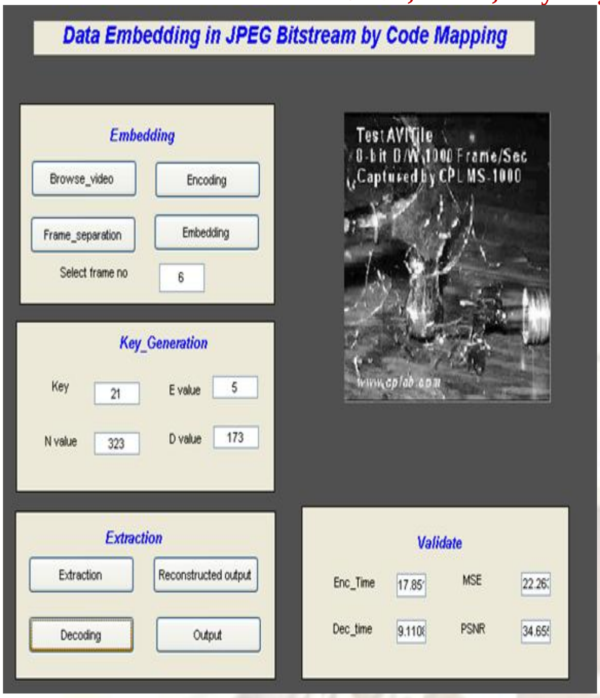

Figure 2 : A bulb video test sequence

The sequences policemen have high motion dynamics, while the bulb video sequence have moderate motion, 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 **Er** 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 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 $-\Delta\mathfrak{R}/t$ 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 $10\log_{10}(b^2/\sum_{\mathfrak{B}_{i,j}} E_r(\mathbf{x})) \le \tau\}$ that selects those $\mathfrak{B}_{i,j}$ whose prediction error is high and refrain from tampering those associated with low error.

VII.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 macro blocks 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 macro blocks corresponding to the CMV is done such that the candidates will be identically identified by the decoder even after these macro blocks 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 and same criteria of MPEG file format is compared with the .avi file format is done.

References

[1]. Hussein A. Aly "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error" IEEE Trans On Information Forensics And Security, Vol. 6, No. 1, Mar 2011.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.
[8] 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.
[9] *Generic Coding of Moving Pictures and Associated Audio Information: Video, 2 Edition*, SO/IEC13818-2, 2000.
[10] B. Chen and G. W. Wornell, "Quantization index modulation for digital watermarking a and information embedding of multimedia," *J. VLSI Signal Process.*, vol. 27, pp. 7–33, 2001.