An Adaptive Scheme for Compressed Video Steganography Using Temporal and Spatial Features of the Video Signal

Jafar Mansouri, Morteza Khademi

Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Received 5 September 2008; accepted 13 July 2009

ABSTRACT: Steganography is the art and science of data hiding. This article proposes an adaptive steganographic algorithm that embeds secret data in a compressed video stream using temporal and spatial features of the video signal and human visual system characteristics. Qualified-DCT coefficients of I-VOP and motion vectors of P-VOP and B-VOP are used for spatial and temporal features of the video, respectively. Embedded-data are extracted without using the original video and there is no need for full decompression. Experimental results demonstrate that the proposed algorithm has high imperceptibility and capacity. Furthermore, the bit rate remains approximately constant. © 2009 Wiley Periodicals, Inc. Int J Imaging Syst Technol, 19, 306–315, 2009; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.20207

Key words: covert communication; data hiding; MPEG-4; video steganography

I. INTRODUCTION

The rapid development of the Internet and communication systems in the past decade has enabled users to send digital data over network conveniently. However, transmission of data on an open network is not secure, and data can be easily tampered by unauthorized users. Consequently, protecting data during transmission is an important task. Although cryptography techniques can be used for this purpose, they are not secure enough because encryption can provide secure delivery of digital content, but when the content is decrypted, encryption no longer provides any protection. To solve this problem, data hiding techniques were proposed and have been considered widely in various fields like covert communication, copyright protection, and broadcast monitoring.

Data hiding techniques embed some data in digital media, which are named host or cover media, such as audio, image, and video, without introducing perceptual distortion (Wu and Liu, 2003a). These techniques consist of two main branches; digital watermarking and steganography. Watermarking is usually used to protect intellectual property rights of multimedia contents by hiding information such as copyright information robustly in digital products, whereas in steganography, the main goal is to convey data secretly by concealing the existence of communication. Steganography has been widely dealt with in covert communication applications.

Although there are similarities between these two techniques, some characteristics of steganography differentiate it from watermarking. Steganographic methods generally rely on the assumption that the existence of the covert communication is unknown to third parties and they are mainly used in secret point-to-point communication between trusting parties. On the other hand, in watermarking, the existence of information is not unknown to third parties (Hartung and Kutter, 1999; Amin et al., 2003; Lou and Sung, 2004). Another difference is that, generally speaking, steganographic methods are not robust, i.e., the hidden information cannot be recovered after data manipulation. However, watermarking, as opposed to steganography, has the additional feature of robustness against attacks. Even if the existence of the hidden information is known, it is difficult for an attacker to destroy the embedded data (Hartung and Kutter, 1999; Briassouli and Strintzis, 2004a,b; Stanescu et al., 2007). Unlike watermarking for copyright protection, attacks such as intentional geometric distortion and normal signal processing against the stego-media (the media in which data have been embedded) are usually not of major concern in steganography since the aim is covert communication. In the steganography scenario, the warden is considered passive. She/he simply inspects contents transmitted between Alice and Bob and discards those which arouse suspicion. In watermarking, the adversary wants to attack the watermarked media to remove or destroy the watermark. Furthermore, in steganography, the host signal is not considered to be of value to the two communicators. In contrast, in digital watermarking, the host signal is considered to be valuable to at least one of the communicators (Cox et al., 2005).

This article considers data embedding in videos. As a video can be viewed as a sequence of still images, data embedding in images seems very similar to videos. There are some papers whose data embedding algorithms in images have been extended to videos (Langelaar and Lagendijk, 2001; Shang, 2007) which prove this fact. However, there are several differences between data hiding in images and videos, where the first important difference is the size of the host media. Since videos contain more samples (i.e., the

Correspondence to: J. Mansouri; e-mail: jafar.mansouri@gmail.com

number of pixels or the number of transform domain coefficients), a video has higher capacity than a still image and more data can be embedded in the video. Also, there are some characteristics in videos which cannot be found in images as perceptual redundancy in videos due to their temporal features. These characteristics can be exploited in data embedding. On the other hand, as a really higher amount of data must be embedded, in the case of video sequences, there are more demanding constraints on real-time effectiveness of the system. As more samples are available in videos, the modification necessary to embed secret data will be less visible; so there is no need to apply complicated methods (Pan et al., 2004).

There are numerous papers about data embedding in videos. Some of them presented schemes in raw videos (Wu and Liu, 2003b; Chan et al., 2005; Carli et al., 2006). For example, Chan et al. (2005) have presented a hybrid method in which data embedding is based on scene change analysis and obtains assistance from audio to increase the embedding capacity. Embedding data in raw video is time-consuming because in this classical approach, a compressed video stream is first decompressed into standard video, data are then embedded in the video signal, and finally, the modified video is recompressed. This technique requires fully decompressing and recompressing the video stream, a procedure that takes a lot of computer processing time, thereby making it computationally very intensive. Also, the embedder has no knowledge of how the video will be recompressed and cannot make knowledgeable decision based on the compression parameters.

In many papers, data have been hidden in compressed domain. Some of these papers considered data hiding in DCT coefficients (Badura and Rymaszewski, 2007; Hu et al., 2007; Stanescu et al., 2007; Zhang et al., 2007). For example, Stanescu et al. (2007) presented a steganographic method using spatial characteristic of the video to embed data in I-frames of the video stream. In their algorithm, for each I-frame, each DCT coefficient, which is above the threshold, its least-significant bit is set to the secret bit. Badura and Rymaszewski (2007) have proposed a steganographic method in which data are hidden in each block of I-frame so that bit rate is not increased and also takes help from audio to increase the capacity. Hu et al. (2007) have presented an algorithm that embeds one bit in each qualified intrablock in H.264 bitstream by modifying intra-4×4 prediction modes (I4-modes) based on mapping between I4-mode and secret bits.

Few papers have utilized temporal feature of the video. Xu et al. (2006) have proposed a steganographic method that uses temporal features of the video for data hiding. In their method, motion vectors with large magnitude are selected for data hiding. Then, the phase angle of these motion vectors is calculated; for the acute angle, data are hidden in the horizontal component of the motion vector and for the obtuse angle, data are embedded in the vertical component. In each group of pictures (GOP), control information is embedded in I-frame for facilitating data extraction. Fang and Chang (2006) have presented another method by which data are hidden in the video using motion vector phase of the macroblock in the interframes.

Two important parameters for evaluating the performance of a steganographic system are capacity and imperceptibility (Liu et al., 2005; Fridrich and Soukal, 2006; Fridrich and Lisonek, 2007; Chang et al., 2008). Capacity refers to the amount of data that can be hidden in the cover medium so that no perceptible distortion is introduced. Imperceptibility or transparency represents the invisibility of the hidden data in the cover media without degrading the perceptual quality by data embedding. Security is the other parameter in the steganographic systems, which refers to an unauthorized person's inability to detect hidden data (Provos and Honyman, 2003). Previous work in data hiding field cared little about capacity and had low embedding capacity. In those methods, only one of the spatial or temporal features of the video was taken into consideration. Also in many of them, all the frames in the video sequence were not used for data embedding. These factors have led to the low capacity of their algorithm.

The goal of this article is to propose a steganographic method to covert communication as in military application and to increase the capacity while preserving acceptable imperceptibility. Secret data are embedded in a compressed video bitstream adaptively using temporal and spatial features of the video signal with the consideration of the human visual system characteristics. In this method, for each I-VOP, secret data are embedded in some AC coefficients of the blocks with high spatial changes. For each P-VOP and B-VOP, secret bits are embedded in horizontal and vertical components of motion vectors with large magnitude, which represent high temporal changes. Experimental results indicate that this algorithm has high visual quality and embedding capacity. Furthermore, the bit rate remains nearly constant.

In this article, a brief review of MPEG-4 is presented in Section II. Section III describes the embedding algorithm, and Section IV describes the extraction algorithm. Section V presents experimental results followed by the conclusion in Section VI.

II. MPEG-4 VIDEO COMPRESSION

MPEG-4 Visual (Part 2 of ISO/IEC 14496) (Information Technology, 2003; Richardson, 2003) is a large document that covers a very wide range of functionalities, all related to the coding and representation of visual information. MPEG-4 Visual is an improvement on the popular MPEG-2 standard both in terms of compression efficiency (better compression for the same visual quality) and flexibility (enabling a much wider range of applications). It reaches this improvement in two main ways: by making use of more advanced compression algorithms and by providing an extensive set of ''tools'' for coding and manipulating digital media.

Perhaps, the most fundamental shift in the MPEG-4 standard has been toward object-based or content-based coding, where a video scene can be handled as a set of objects. It encodes the visual information as objects, which include natural video (real-world), synthetic video (computer-generated visual objects such as meshes and animated human faces and bodies), and still texture. This article is limited to the natural video and ignores synthetic video and still texture.

A natural video object (VO) in MPEG-4 is an area of the video scene that can occupy an arbitrarily shaped region and can exist for an arbitrary length of time. The introduction of the VO concept allows more flexible options for video coding. An instance of a VO at a particular point in time is a video object plane (VOP). Generally, three kinds of VOP are defined in MPEG-4: I-VOP, P-VOP, and B-VOP. I-VOPs are coded without reference to other VOPs. Compression is achieved by reducing spatial redundancy, but not temporal redundancy. P-VOPs use the previous encoded I-VOP or P-VOP (reference VOP) for motion compensation. The macroblock, corresponding to a 16×16-pixel region of a VOP, is the basic unit for motion compensated prediction in MPEG-4. A macroblock in this VOP to be coded is matched with a set of 16×16-pixel regions of the reference VOP. The selected best matching region in the

reference VOP is subtracted from the current macroblock to produce a residual macroblock that is transformed, encoded, and transmitted together with a motion vector. This motion vector describes the position of the best matching region (relative to the current macroblock position). By reducing spatial and temporal redundancy, P-VOPs offer increased compression compared with I-VOPs. B-VOPs use the previous and next I-VOPs or P-VOPs for motion compensation and offer the highest degree of compression.

In MPEG-4, the texture information is encoded using hybrid DPCM/DCT compression algorithm similar to that used in MPEG-1 and MPEG-2. This algorithm uses motion compensation to reduce interframe redundancy and the DCT to compact the energy in every 8×8 block of the image into a few coefficients. The algorithm then adaptively quantizes the DCT coefficients to achieve the desired bit rate. Huffman codes are used by the algorithm to encode the quantized DCT coefficients, the motion vectors, and most control parameters to reduce the statistical redundancies in the data. All coded information is assembled in a bitstream.

MPEG-4 Visual provides its coding functions through a combination of tools. A tool is a subset of coding functions to support a specific feature, for example, spatial and temporal scalability. However, a specific application does not require all of the tools available in the MPEG-4 Visual framework. Therefore, to simplify the design of the decoders, the standard describes a series of profiles, recommended sets, or groupings of tools for particular types of applications.

The Simple Profile was introduced in the first version of the MPEG-4 Visual standard. It rapidly became popular with developers because of its improved efficiency compared with previous standards (such as MPEG-1 and MPEG-2), and the ease of integrating it into existing video applications that use rectangular video frames. The Advanced Simple Profile was incorporated into a later version of the standard with tools added to the Simple Profile to support improved compression efficiency. The Advanced Simple Profile includes all capabilities of Simple Profile; furthermore, it includes additional capabilities such as B-VOPs, quarter-pixel motion compensation, and alternate quantizer. However, Advanced Simple Profile does not support arbitrary-shaped objects, scalability, and sprite coding.

Here, the discussion in this article is limited to the steganography of natural video sequences that are compressed based on Advanced Simple Profile.

III. EMBEDDING ALGORITHM

A compressed video stream is mainly composed of DCT coefficients, motion vectors, and other information (header information, etc.). In the proposed scheme, the secret data are embedded in MPEG-4 bitstream by modifying DCT coefficients and motion vectors. For video degradation to be invisible, data embedding is performed adaptively and is based on the human visual system and local characteristics of the video signal. It should be mentioned that in the proposed algorithm the color components do not change. Also, each VOP is the entire frame but the algorithm can be extended to VOP with an arbitrary shape.

A. Embedding Data in DCT Coefficients. Each I-VOP in MPEG-4 is usually divided into 8×8 DCT blocks. The DC component of DCT of an image block represents the average energy of that block and AC coefficients represent the intensity changes. In case a block has high variance (a textured block or a block containing edge elements), the magnitude of AC coefficients is large and when the block has low variance and contains almost uniform areas, the magnitude of AC coefficients is small. In other words, the AC coefficients represent the intensity of spatial changes of the block. Because of reducing the sensitivity of the human visual system in regions with high luminance intensity variations (Wolfgang et al., 1999), this characteristic can be utilized in steganography and secret data can be embedded in edge pixels or textured area so that degradation in video quality is not perceptible. The details of data embedding algorithm in DCT coefficients of an I-VOP are explained as follows:

1. For each I-VOP, quantized DCT coefficients are extracted from the bitstream.
2. For each 8×8 DCT block, sum of the square of quantized AC coefficients is calculated to select blocks with high intensity changes as follows:

$$S = \sum_{m=1}^{63} |\mathrm{AC}_m|^2 \tag{1}$$

   where AC_m is mth quantized AC coefficient.
3. S is compared with a threshold, T_1:

$$B_n = \begin{cases} 1 & \text{if} \quad S \geq T_1 \\ 0 & \text{if} \quad S < T_1 \end{cases} \tag{2}$$

   If B_n is one, it indicates that nth block is a highly textured area or includes edge(s). As a result, the spatial changes of that block are high and that block can be used for data embedding. If B_n is zero, nth block is not used for embedding.
4. For each block with $S > T_1$, eight bits of secret data are embedded in eight quantized DCT coefficients. For adding security, these eight coefficients are determined by a secret key that is known between the embedder and the extractor. For example, $i = \{9, 14, 16, 17, 24, 29, 31, 40\}$ can be a secret string which determines the quantized AC coefficients into which secret bits are embedded. Embedding is according to the following rule:

$$\begin{aligned} &\text{if } \mathrm{mod}(\mathrm{AC}_i, 2) = \mathrm{data}(k) \text{ then } \mathrm{AC}'_i = \mathrm{AC}_i \\ &\text{if } \mathrm{mod}(\mathrm{AC}_i, 2) \neq \mathrm{data}(k) \text{ then } \mathrm{AC}'_i = \mathrm{AC}_i + \mathrm{sign}(\mathrm{AC})_{\mathrm{i}} \end{aligned} \tag{3}$$

   where AC'_i is the selected ith quantized AC coefficient after data embedding, sign(.) is the sign function, and data(k) is the kth secret bit to be embedded. For example, for nth block with $S > T_1$, (k)th secret bit is embedded in AC_9, ($k + 1$)th secret bit is embedded in AC_{14}. To make it difficult for the third party to detect this string, this secret string can be changed after some repetition with the embedder's and the extractor's knowledge.

It is observed that the magnitude of modification of quantized DCT coefficients is maximally one, which is minimal and does not produce visible distortion in the video quality. The reason why $\mathrm{sign}(\mathrm{AC}_i)$ is added to AC_i is that after modification, the sum of square of quantized AC coefficients always remains more than the

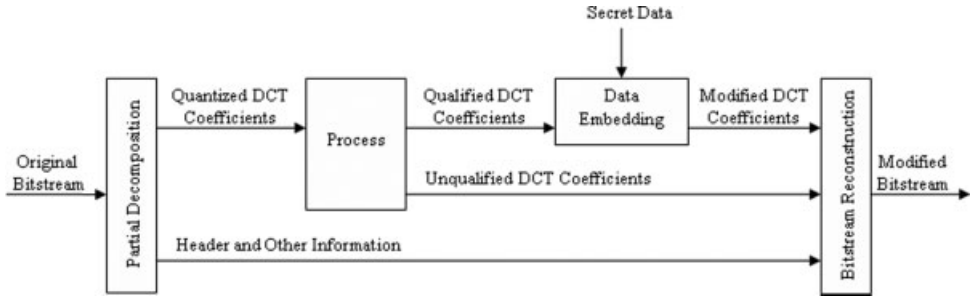

Figure 1. Block diagram of data embedding in DCT coefficients for I-VOP.

threshold. Therefore, in data extraction, all blocks which contain secret data are selected and no data will be missed. Moreover, blocks which do not contain secret data are not selected and detection error does not occur. Furthermore, in this case, there is no need to have the original host video for extracting hidden data. Figure 1 shows the block diagram of data embedding in quantized DCT coefficients.

B. Embedding Data in Motion Vectors. Motion vectors in P-VOP and B-VOP can be utilized for data hiding. Since human visual system is less sensitive to distortion in regions that are temporally near to features of high-luminance intensity (Reid et al., 1997), this feature can be utilized for data hiding. In the proposed method, data are not embedded in all motion vectors but only in motion vectors with a magnitude above a threshold. Larger magnitude illustrates faster temporal changes and less visible degradation due to data hiding. The details of data embedding in motion vectors of P-VOP and B-VOP are as follows:

1. For each P-VOP and B-VOP, motion vectors are extracted from the bitstream.
2. The magnitude of each motion vector is calculated as follows:

$$|\mathrm{MV}_j| = \sqrt{H_j^2 + V_j^2} \tag{4}$$

where MV_j is the motion vector of the jth macroblock, and H_j and V_j are horizontal and vertical components of the MV_j, respectively.

3. This magnitude is compared with a threshold, $\tilde{T}_2$:

$$\mathrm{AMV}_j = \begin{cases} 1 & \text{if} \quad |\mathrm{MV}_j| \geq \tilde{T}_2 \\ 0 & \text{if} \quad |\mathrm{MV}_j| < \tilde{T}_2 \end{cases} \tag{5}$$

If AMV_j is 1, it means that jth motion vector satisfies the requirement and can be used for data embedding; otherwise, it is not used for data embedding. To increase the speed of algorithm, instead of magnitude of the motion vector, its square is used and the threshold is also changed to $T_2 = \tilde{T}_2^2$. Along with this, for each motion vector, one operation (square root) is removed and this causes the speed improvement. So, the formula (5) changes to the following formula:

$$\mathrm{AMV}_j = \begin{cases} 1 & \text{if} \quad |\mathrm{MV}_j|^2 \geq T_2 \\ 0 & \text{if} \quad |\mathrm{MV}_j|^2 < T_2 \end{cases} \tag{6}$$

4. For each qualified motion vector, two secret bits are embedded in horizontal and vertical components. The order by which the first bit is embedded in horizontal or vertical component is determined by a known rule between the embedder and the extractor. As the experiments were performed with MPEG-4 Advanced Simple Profile in which motion estimation was carried out with quarter pixel accuracy, the algorithm for data hiding in the horizontal component is as follows (Suppose that the first bit is embedded in the horizontal component):

$$\begin{aligned} &\text{if } \mathrm{mod}(4 * H_j, 2) = \mathrm{data}(k) \text{ then } H_j' = H_j \\ &\text{if } \mathrm{mod}(4 * H_j, 2) \neq \mathrm{data}(k) \text{ then } H_j' = H_j + \mathrm{sign}(H_j) * 0.25 \end{aligned} \tag{7}$$

and for the vertical component it is as follows:

$$\begin{aligned} &\text{if } \mathrm{mod}(4 * V_j, 2) = \mathrm{data}(k+1) \text{ then } V_j' = V_j \\ &\text{if } \mathrm{mod}(4 * V_j, 2) \neq \mathrm{data}(k+1) \text{ then } V_j' = V_j + \mathrm{sign}(V_j) * 0.25 \end{aligned} \tag{8}$$

where H_j' and V_j' are horizontal and vertical components after data embedding and data(k) is the kth secret bit to be embedded.

As the magnitude of the motion vector is large and the magnitude of modification is maximally 0.25, which is the minimum possible value of changes, the introduced distortion does not have noticeable

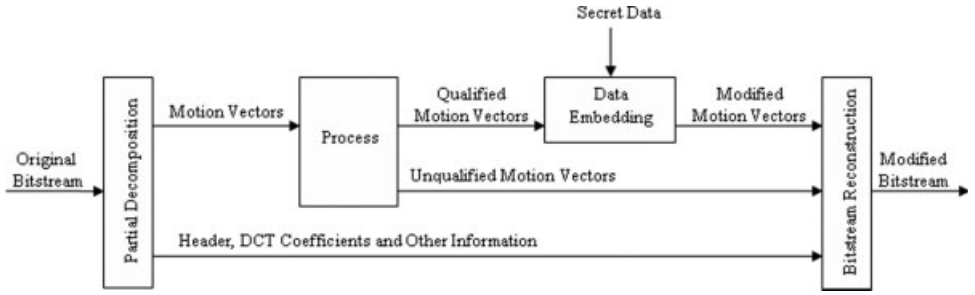

Figure 2. Block diagram of data embedding in motion vectors for P-VOP and B-VOP.

effect on the video quality. The reason why $\text{sign}(H_j) * 0.25$ is added to H_j, and also $\text{sign}(V_j) * 0.25$ is added to V_j, is that after modification, the magnitude of the motion vector always remains more than the threshold. So, in data extraction, just motion vectors which contain secret data are selected and no data will be missed. Moreover, motion vectors which do not contain secret data are not selected and detection error does not occur. Also in this case, there is no need to have the original host video to extract hidden data. Figure 2 shows the block diagram of data embedding in motion vectors.

IV. EXTRACTION ALGORITHM

Data extraction is in accordance with data embedding and detection error does not occur. Embedded data are extracted without using the original host video. The data extraction does not need full decompression that improves the algorithm speed.

A. Data Extraction from DCT Coefficients. Extraction of secret data from DCT coefficients is carried out as follows:

1. For each I-VOP, quantized DCT coefficients are obtained from the bitstream.
2. For each 8×8 DCT block, the sum of square of quantized AC coefficients is calculated:

$$S' = \sum_{m=1}^{63} |\text{AC}'_m|^2 \tag{9}$$

3. S' is compared with the T_1 threshold:

$$B'_n = \begin{cases} 1 & \text{if} \quad S' \geq T_1 \\ 0 & \text{if} \quad S' < T_1 \end{cases} \tag{10}$$

4. If B'_n is 1, the quantized AC coefficients containing secret data are determined using the secret key (that is known between the embedder and the extractor) and secret bits are obtained as follows:

$$\text{data}(k) = \text{mod}(\text{AC}'_i, 2) \tag{11}$$

where ''i'' is determined by the secret key. If B'_n is zero, the nth block does not contain secret data.

B. Data Extraction from Motion Vectors. The stages of data extraction from motion vectors are as follows:

1. For each P-VOP and B-VOP, motion vectors are obtained from the bitstream.
2. For each motion vector, the square of its magnitude is calculated as follows:

$$|\text{MV}'_j|^2 = H'^2_j + V'^2_j \tag{12}$$

3. This value is compared with the T_2 threshold:

$$\text{AMV}'_j = \begin{cases} 1 & \text{if} \quad |\text{MV}'_j|^2 \geq T_2 \\ 0 & \text{if} \quad |\text{MV}'_j|^2 < T_2 \end{cases} \tag{13}$$

4. If AMV'_j is 1, it means the jth motion vector contains secret data and secret bits can be extracted as follows:

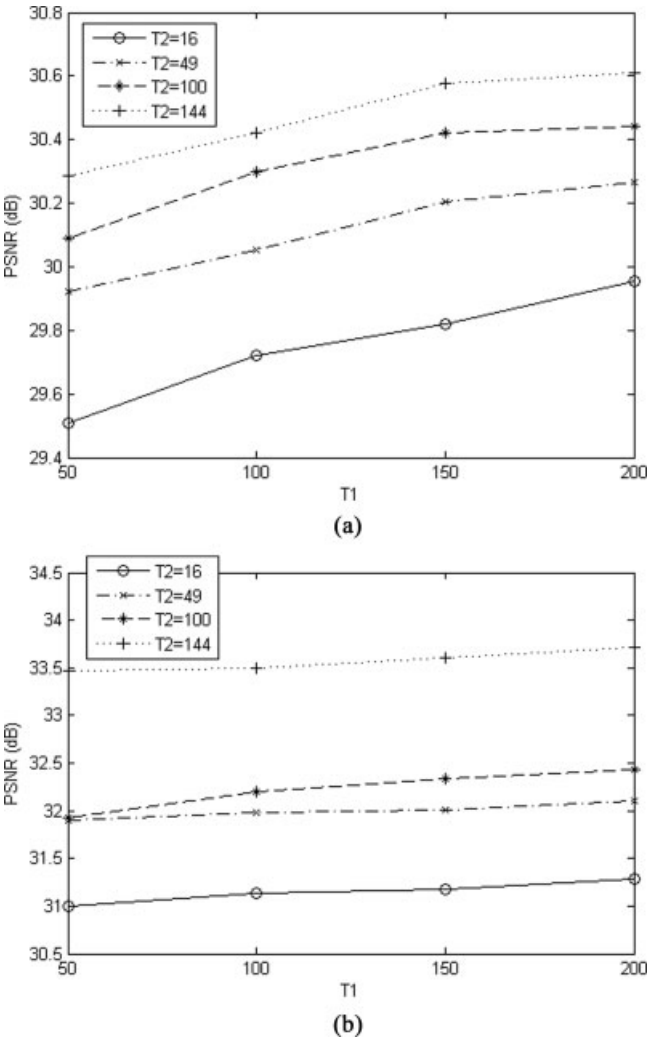

Figure 3. PSNR (dB) for GOV for different thresholds (a) For Bus sequence at 768 kbps. PSNR for original compressed video is 32.1965 dB. (b) For Stefan sequence at 2 Mbps. PSNR for original compressed video is 37.8584 dB.

$$\begin{aligned} \text{data}(k) &= \text{mod}(4 * H'_j, 2) \\ \text{data}(k+1) &= \text{mod}(4 * V'_j, 2) \end{aligned} \tag{14}$$

The order by which the first bit is extracted from horizontal or vertical component is determined by a known rule between the embedder and the extractor. In the earlier formula, it is assumed that the first bit is extracted from the horizontal component. If AMV'_j is zero, the jth motion vector does not contain secret data.

V. EXPERIMENTAL RESULTS

A. Test Conditions. Several experiments were performed to evaluate the performance of the proposed steganographic algorithm. A 256 gray image was used as secret data for the experiments; this image was first decomposed into eight binary images or bit planes. To increase the security, the order of the bit planes and the position

Table I. PSNR (dB) before embedding (oPSNR), after embedding (sPSNR) and decrease of PSNR (dPSNR) for I-VOP, P-VOP, B-VOP, and GOV for different sequences at four bit rates.

Sequence	Bit Rate (kbps)	I-VOP			P-VOP			B-VOP			GOV		
		oPSNR	sPSNR	dPSNR	oPSNR	sPSNR	dPSNR	oPSNR	sPSNR	dPSNR	oPSNR	sPSNR	dPSNR
Bus	384	28.9879	28.6939	−0.2940	28.6370	27.8412	−0.7958	27.5287	26.8483	−0.6804	27.9273	27.2504	−0.6769
	768	33.3780	32.5889	−0.7891	33.2480	31.2335	−2.0145	31.5807	29.7085	−1.8722	32.1965	30.4201	−1.7764
	1500	38.1771	36.4998	−1.6773	37.1069	33.1111	−3.9958	35.1910	31.9196	−3.2714	36.0091	32.7552	−3.2539
	2000	40.2235	38.6190	−1.6045	38.7638	33.7307	−5.0331	36.7003	32.6975	−4.0028	37.6183	33.6561	−3.9622
Flower garden	384	25.4118	25.3159	−0.0959	26.0729	25.9153	−0.1576	24.9991	24.8867	−0.1124	25.3020	25.1796	−0.1224
	768	29.9082	29.4165	−0.4917	30.4403	30.1201	−0.3202	29.0071	28.7075	−0.2996	29.4592	29.1316	−0.3276
	1500	35.0786	33.4565	−1.6221	34.8399	34.2695	−0.5704	33.3298	32.7665	−0.5633	33.9021	33.2100	−0.6921
	2000	37.6110	36.0188	−1.5922	36.5051	35.9402	−0.5649	35.1631	34.5011	−0.6620	35.7789	35.0286	−0.7503
Foreman	384	36.3641	35.3395	−1.0246	35.9213	35.4893	−0.4320	35.0891	34.6733	−0.4158	35.4034	34.9328	−0.4706
	768	39.1723	37.9154	−1.2569	38.9868	38.5309	−0.4559	37.7381	37.3013	−0.4368	38.2099	37.6701	−0.5398
	1500	42.7783	41.4536	−1.3247	41.5513	40.9345	−0.6168	40.1308	39.6324	−0.4984	40.7894	40.1634	−0.6260
	2000	46.6026	45.0724	−1.5302	42.6429	41.9534	−0.6895	40.9407	40.4950	−0.4457	42.0287	41.3943	−0.6344
Stefan	384	29.4945	29.0955	−0.3990	29.3775	28.5283	−0.8492	28.4781	27.8037	−0.6744	28.7876	28.0925	−0.6951
	768	33.2713	32.2490	−1.0223	33.4593	29.2842	−4.1751	32.5109	29.5651	−2.9458	32.8298	29.8198	−3.0100
	1500	38.2116	36.3368	−1.8748	36.9548	30.5383	−6.4165	35.7182	31.2232	−4.4950	36.3142	31.6725	−4.6417
	2000	41.0808	38.8062	−2.2746	38.4501	30.9144	−7.5357	37.0322	31.6632	−5.3690	37.8584	32.3406	−5.5178
Average	384	30.0646	29.6112	−0.4534	30.0022	29.4435	−0.5587	29.0238	28.5530	−0.4708	29.3551	28.8638	−0.4913
	768	33.9324	33.0425	−0.8900	34.0336	32.2922	−1.7414	32.7092	31.3206	−1.3886	33.1739	31.7604	−1.4135
	1500	38.5614	36.9367	−1.6247	37.6132	34.7134	−2.8998	36.0925	33.8854	−2.2071	36.7537	34.4503	−2.3034
	2000	41.3795	39.6291	−1.7504	39.0905	35.6347	−3.4558	37.4591	34.8392	−2.6199	38.3211	35.6049	−2.7162

of each bit in the bit plane were changed according to a known rule between the embedder and the extractor. Then, each bit plane was converted into a 1D stream and these streams were used for embedding. The proposed algorithm was tested with the standard color video sequences: Flower Garden, Stefan, Bus, and Foreman. All sequences were encoded with MPEG-4 Advanced Simple Profile in CIF format (352 × 288 pixels) at the frame rate 15 frames/s at 384 kbps, 768 kbps, 1.5 Mbps, and 2 Mbps. The motion vectors were supported under quarter-pixel accuracy and each VOP is the entire frame. Each group of VOPs (GOV) included one I-VOP, three P-VOPs,and eight B-VOPs, (IBBPBBPBBPBB).

It should be noted that if secret bits are embedded in DC and low frequency coefficients, this greatly affects the video quality and introduces high distortion. On the other hand, in the process of compression, especially for low bit rates, usually most of high-frequency coefficients become zero. As a result, embedding data in these frequencies may alter these coefficients and cause the bit rate to increase and this can draw the attention of the third party to steganography. Therefore, appropriate frequency range should be selected for embedding data in DCT coefficients. The experiments showed that the suitable frequency range was between 9th and 48th of DCT coefficients, for out of this range either noticeable distortion would occur or the bit rate would increase significantly.

It should be mentioned that when T_1 and T_2 decrease, more bits are embedded but more degradation is introduced. On the other hand, when the thresholds are increased, less modification is applied to the bitstream. Therefore, less distortion is introduced, while the capacity is reduced. For consideration of the impact of T_1 and T_2 on the video quality, Figure 3 shows the values of PSNR (dB) on the average for GOV for different thresholds for Bus sequence at 768 kbps and for Stefan sequence at 2 Mbps. According to the figure, for $T_1 = 150$ and $T_2 = 100$ PSNR reduction is not so much and these thresholds can be suitable choices.

In the following subsections, the imperceptibility, capacity, changes of the bit rate, and the security of the proposed steganographic method are discussed. It should be noted that, as it has been pointed out in the Introduction Section, steganography does not consider attacks.

B. Imperceptibility. Peak signal-to-noise ratio (PSNR) was used for evaluating the distortion introduced by data embedding. Table I illustrates PSNR (dB) of the luminance components of the compressed video without embedded data (oPSNR) and with embedded data (sPSNR), at four bit rates and with the thresholds $T_1 = 150$ and $T_2 = 100$ for four sequences and on the average. In this table, for each type of VOP (at each bit rate), PSNR is the average of PSNRs of that type of VOP in the sequence. Also for GOV, PSNR is the average of PSNRs for GOVs in the sequence, and dPSNR indicates the decrease of PSNR due to data embedding. According to the table, this value is small. For each sequence, if the bit rate increases, more redundancy happens and more bits will be embedded. So, dPSNR and video quality degradation increase. Furthermore, dPSNR is larger for P-VOP and B-VOP than I-VOP because distortion in I-VOP is only due to data embedding while for P-VOP and B-VOP it is due to both data embedding and prediction from previous I-VOP or P-VOP in which data have been embedded. Although PSNR for P-VOP is more than PSNR for B-VOP after embedding, dPSNR is smaller for B-VOP because fewer bits are embedded in B-VOP in comparison with P-VOP. Generally, the most dPSNR belongs to Stefan and Bus sequences, because more bits are embedded in them. All of these had good quality and were free of visible artifact after embedding.

Figure 4 shows three I-VOP, P-VOP, and B-VOP of Flower Garden sequence at 384 kbps before and after data embedding. It is observed that imperceptibility is high and the original VOPs do not differ significantly from VOPs after steganography, reducing the detection probability, and leaving the observer unaware of the embedded data. In the algorithm, drift compensation was not performed because the decrease of PSNR was not so much. Also since regions with high temporal or spatial changes are selected and

Figure 4. VOPs of Flower Garden sequence at 384 kbps. (a) I-VOP before embedding. (b) I-VOP after embedding. (c) B-VOP before embedding. (d) B-VOP after embedding. (e) P-VOP before embedding. (f) P-VOP after embedding. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

human visual system has low sensitivity in these regions, embedding data in these regions causes high imperceptibility and does not provoke suspicion of the warden. All of the stego-videos had good quality and were free of visible artifact after embedding. Nevertheless, if more bits are required to be embedded (with small thresholds), drift compensation (e.g., Hartung and Girod's method (1998)) might be required.

C. Capacity. Table II illustrates the number of embedded bits in each sequence at four bit rates (with $T_1 = 150$ and $T_2 = 100$). More bits are embedded in Stefan and Bus sequences because of having more textured and edgy regions and motion vectors with large amplitude (they have more temporal and spatial changes). Moreover, with increasing the bit rate, redundancy is increased and more bits are embedded in each sequence.

The proposed method has high embedding capacity because it utilizes both temporal and spatial features of the video signal according to the human visual system features for data embedding. Also, all frames are used in this process. However, in the previous methods (Fang and Chang, 2006; Xu et al., 2006; Badura and Rymaszewski, 2007; Hu et al., 2007; Stanescu et al., 2007; Zhang et al., 2007) only one of the temporal and spatial features and/or

Table II. Number of embedded bits in I-VOP, P-VOP, B-VOP, and GOV for four bit rates.

Sequence	Bit Rate (kbps)	I-VOP	P-VOP	B-VOP	GOV
Bus	384	888	618	240	4662
	768	3048	690	334	7790
	1500	8016	690	258	12,150
	2000	9712	690	232	13,638
Flower garden	384	872	182	186	2906
	768	3296	166	182	5250
	1500	7904	166	160	9682
	2000	8888	166	144	10,538
Foreman	384	2336	98	104	3462
	768	4096	174	110	5498
	1500	7040	174	86	8250
	2000	9344	174	82	10,522
Stefan	384	1152	326	234	4002
	768	4576	596	168	7708
	1500	8328	596	92	10,852
	2000	8824	596	78	11,236
Average	384	1312	306	191	3758
	768	3754	407	199	6562
	1500	7822	407	149	10,234
	2000	9192	407	134	11,484

some frames are used for data embedding. These reasons cause less capacity for those methods than the proposed method.

Figures 5 and 6 show capacity versus peak signal-to-noise ratio (PSNR) comparing the proposed method with the methods of Xu et al. (2006) and Stanescu et al. (2007), respectively. The method of Xu et al. uses temporal changes of the video and data are hidden in motion vectors. The method of Stanescu et al. uses spatial changes of the video and data are hidden in DCT coefficients of I-frames. It is worth mentioning that the measured points have been obtained for different thresholds at different bit rates. In order for PSNRs for VOPs of I, P, B, and GOV in the compared methods to be nearly in the same range, different thresholds are chosen. Consequently, the corresponding capacities could be comparable. At high bit rates, the method of Stanescu et al. has more capacity for Flower Garden and Foreman sequences. These sequences have low spatial and especially low temporal changes; and at high bit rates, there are more nonzero DCT coefficients. These reasons lead to more capacity for the method of Stanescu et al. than the proposed method at high-bit rates. With the exception of Flower Garden and Foreman sequences at high bit rates, the proposed method has more capacity in comparison with two other methods. The proposed method utilizes both temporal and spatial features of the video signal in accordance with

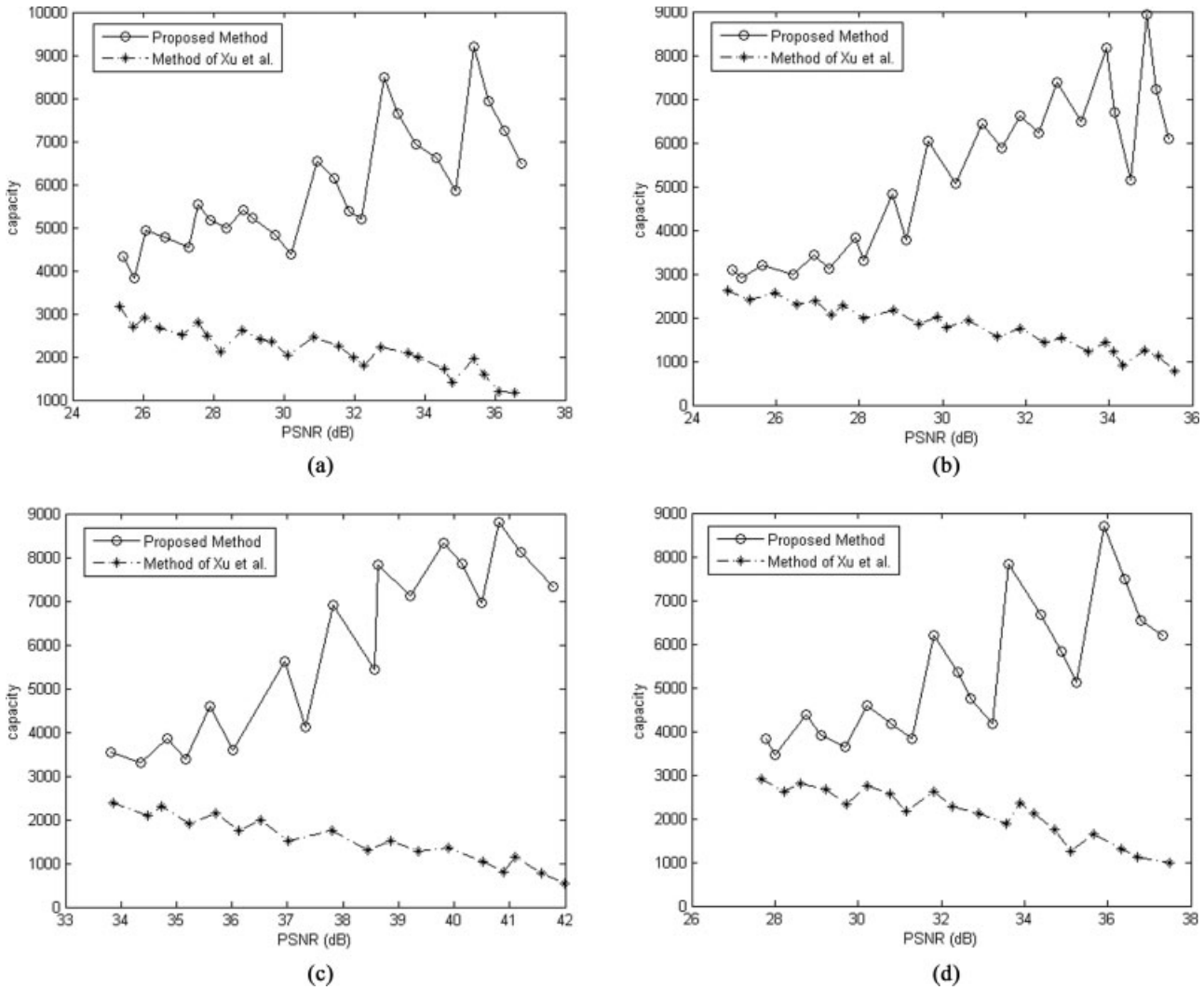

Figure 5. Capacity versus PSNR for the proposed method and method of Xu et al. for the sequences of (a) Bus. (b) Flower Garden. (c) Foreman. (d) Stefan.

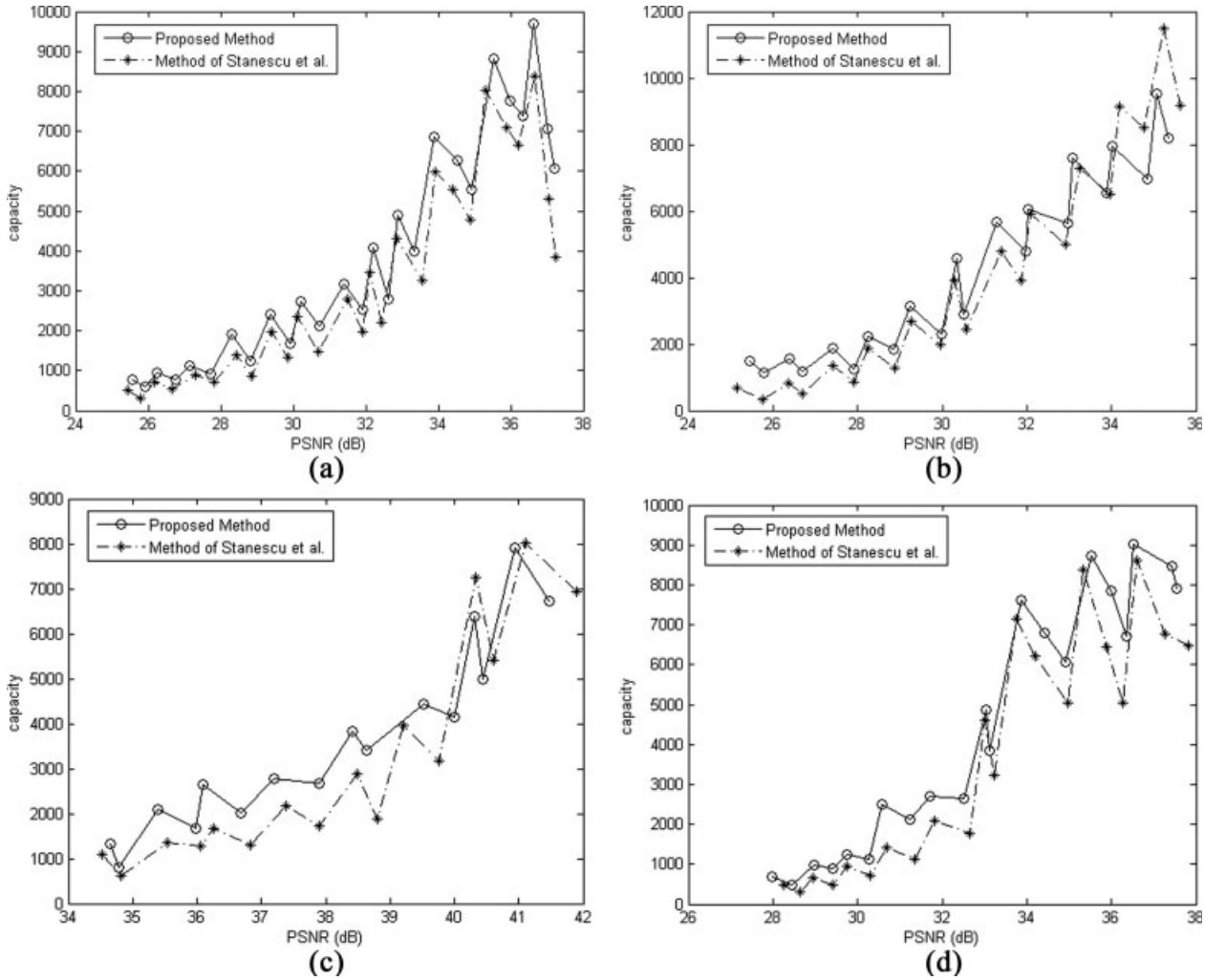

Figure 6. Capacity versus PSNR for the proposed method and method of Stanescu et al. for the sequences of (a) Bus. (b) Flower Garden. (c) Foreman. (d) Stefan.

the human visual system features for data embedding. Moreover, all frames are used in this process. These reasons lead to more capacity for the proposed method than the other two methods.

D. Changes of the Bit Rate. Table III shows changes of the bit rate due to data embedding. The embedding process changes the bit rate of the compressed video by maximally 0.59%. So the bit rate is approximately constant and there is no need of a bit rate controller; this is another advantage of the proposed method.

E. Security. Two stages are required to break a steganographic system, as follows (Zollner et al., 1998; Lou and Sung, 2004): first, a warden must discover that steganography has been used; second, the warden must manage to read the embedded message. Two cases are considered for evaluating the security.

In the first case, the warden is assumed to have only the video containing secret bits (stego-only case). A stego-video must not contain any artifacts associated with data embedding because a warden could easily utilize those artifacts to detect the secret data. If the hidden data are not advertised, a casual warden will be unaware of its existence and therefore will not attempt to read it. The proposed method embeds data based on the local characteristics of the video and human visual system, and the stego-video does not differ significantly from the cover-video. Hence, the stego-video maintains high fidelity and does not arouse suspicion of using steganography.

In the second case, the warden has also access to the original compressed video (known-cover case). Obviously, the warden can compare and detect the difference between a cover-video and a stego-video if she/he has access to both videos. Steganography is not secure against the known-cover case. However, if the difference contains random signals, then, it is difficult for the warden to break

Table III. Changes of the bit rate due to data embedding (%).

Sequence	384 kbps	768 kbps	1.5 Mbps	2 Mbps
Bus	0.0466	−0.0227	−0.0127	0.0946
Flower garden	−0.0166	0.1449	0.0090	0.0339
Foreman	0.1276	0.0270	0.1476	−0.2130
Stefan	−0.0463	0.1105	−0.1619	−0.5926
Average	0.0278	0.0649	−0.0045	−0.1693

the system. In the proposed method, before embedding, the order of the bit planes and position of each bit in the bit plane are changed according to a known rule between the embedder and the extractor. Moreover, embedding data in AC coefficients is performed according to a secret key. Without knowing these, the warden cannot read the original data. Even by comparing and subtracting, the warden may think that the extracted data are noise. If it is necessary, combining the steganographic method with traditional cryptography will increase security.

VI. CONCLUSIONS

A method for video steganography to covert communication was proposed in this article. Secret data were embedded in a compressed video bitstream adaptively using temporal and spatial features of the video signal. Embedding was performed with the consideration of the human visual system characteristics. In this method, for each I-VOP, the blocks with high spatial changes were selected and secret data were embedded in some AC coefficients. For P-VOP and B-VOP, secret bits were embedded in horizontal and vertical components of motion vectors with large magnitude which represented high temporal changes. The method did not require the original video signal or bitstream for data extraction. The algorithm was performed for different bit rates and experimental results indicated that this algorithm had high visual quality and embedding capacity. Furthermore, the bit rate remained nearly constant without using a bit rate controller and this was another advantage of the proposed method. Although this algorithm was carried out with Advanced Simple Profile in which each VOP was the entire frame, this algorithm could be extended to other profiles of MPEG-4 and to VOP with an arbitrary shape with minor modification.

REFERENCES

M.M. Amin, M. Salleh, S. Ibrahim, M.R. Katmin, and M.Z. Shamsuddin, Information hiding using steganography, IEEE International Conference on Telecommunication Technology, 2003, pp. 21–25.

S. Badura and S. Rymaszewski, Transform domain steganography in DVD video and audio content, IEEE International Workshop on Image Systems and Techniques, 2007, pp. 1–5.

A. Briassouli and M.G. Strintzis, Locally optimum nonlinearities for DCT watermarking detection, IEEE Trans Image Process 13 (2004), 1604–1617.

A. Briassouli and M.G. Strintzis, Optimal watermark detection under quantization in the transform domain, IEEE Trans Circuits Syst Video Technol 14 (2004), 1308–1319.

M. Carli, P. Campisi, and A. Neri, Data hiding driven by perceptual features for secure communications, IEEE International Conference on Systems, Mobile Communications, Learning Technologies and Networking, 2006, pp. 85–89.

P.W. Chan, M.R. Lyu, and R.T. Chin, A novel scheme for hybrid digital video watermarking: Approach, evaluation and experimentation, IEEE Trans Circuits Syst Video Technol 15 (2005), 1638–1649.

C.-C. Chang, C.-C. Lin, and Y.-H. Chen, Reversible data-embedding scheme using differences between original and predicted pixel values, IET Trans Inf Security 2 (2008), 35–46.

I.J. Cox, T. Kalker, G. Pakura, and M. Scheel, Information transmission and steganography, IEEE International Workshop on Digital Watermarking, 2005, pp. 15–29.

D.-Y. Fang and L.-W. Chang, Data hiding for digital video with phase of motion vector, IEEE International Symposium on Circuits and Systems, 2006, pp. 1422–1425.

J. Fridrich and P. Lisonek, Grid colorings in steganography, IEEE Trans Inf Theory 53 (2007), 1547–1549.

J. Fridrich and D. Soukal, Matrix embedding for large payloads, IEEE Trans Inf Forensics Security 1 (2006), 390–395.

F. Hartung and B. Girod, Watermarking of uncompressed and compressed video, Signal Process 66 (1998), 283–301.

F. Hartung and M. Kutter, Multimedia watermarking techniques, Proc IEEE 87 (1999), 1079–1107.

Y. Hu, C. Zhang, and Y. Su, Information hiding based on intra prediction modes for H. 264/AVC, IEEE International Conference on Multimedia and Expo, 2007, pp. 1231–1234.

Information Technology, Coding of Audio-Visual Objects, Part 2: Visual, International Organization for Standardization, 2003, ISO/IEC 14496-2.

G. Langelaar and R. Lagendijk, Optimal differential energy watermarking of DCT encoded images and video, IEEE Trans Image Process 10 (2001), 148–158.

G. Liu, Y. Dai, J. Wang, Z. Wang, and S. Lian, Image hiding by non-uniform generalized LSB and dynamic programming, IEEE International Workshop on Multimedia Signal Process, 2005, pp. 1–4.

D.-C. Lou and C.-H. Sung, A steganographic scheme for secure communications based on the chaos and Euler theorem, IEEE Trans Multimedia 6 (2004), 501–509.

J.-S. Pan, H.-C. Huang, and L.C. Jain, Intelligent watermarking techniques, World Scientific, Singapore, 2004.

N. Provos and P. Honyman, Hide and seek: An introduction to steganography, IEEE Security Privacy 1 (2003), 32–44.

M.M. Reid, R.J. Millar, and N.D. Black, Second-generation image coding, ACM Comput Survey 29 (1997), 3–29.

I.E.G. Richardson, H. 264 and MPEG-4 video compression, Wiley, Chichester, 2003.

Y. Shang, A new invertible data hiding in compressed videos or images, IEEE International Conference on Natural Computation, 2007, pp. 576–580.

D. Stanescu, M. Stratulat, B. Ciubotaro, D. Chiciudean, R. Cioarga, and M. Micea, Embedding data in video stream using steganography, IEEE International Symposium on Applied Computational Intelligence and Informatics, 2007, pp. 241–244.

R.B. Wolfgang, C.I. Podilchuk, and E.J. Delp, Perceptual watermarks for digital images and video, Proc IEEE (Special Issue on Identification and Protection of Multimedia Information) 87 (1999), 1108–1126.

M. Wu and B. Liu, Data hiding in image and video: Part I-Fundamental issues and solutions, IEEE Trans Image Process 12 (2003), 685–695.

M. Wu and B. Liu, Data hiding in image and video: Part II-Designs and applications, IEEE Trans Image Process 12 (2003), 696–710.

C. Xu, X. Ping, and T. Zhang, Steganography in compressed video stream, IEEE International Conference on Innovative Computing, Information and Control, 2006, pp. 269–272.

J. Zhang, A.T.S. Ho, G. Qiu, and P. Marziliano, Robust video watermarking of H. 264/AVC, IEEE Trans Circuits Syst II: Express Briefs 54(2007), 205–209.

J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, Modeling the security of steganographic systems, IEEE International Second Workshop on Information Hiding, 1998, pp. 344–354.