17th ICEE 2009
Iranian Conference on Electrical Engineering
www.icee.ir/2009
IUST

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

Jafar Mansouri*, and Morteza Khademi**
*PhD student, Department of Electrical Engineering, Ferdowsi University of Mashhad,
jafar.mansouri@gmail.com
**Associate Professor, Department of Electrical Engineering, Ferdowsi University of Mashhad,
khademi@um.ac.ir

Abstract: *This paper 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 is 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. Also the bit rate remains approximately constant.*

Keywords: Covert communication, data hiding, MPEG-4, video steganography.

1. Introduction

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 [1]. These techniques contain 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 while 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 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 [2-4]. Another difference is that, generally, steganographic methods are not robust, i.e., the hidden information can not be recovered after data manipulation. However, watermarking, as opposed to steganography, has the additional feature of robustness against attacks. [3, 5-7]. 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 [8].

There are numerous papers about data embedding in videos. Some of them presented schemes in raw videos [9]. But embedding data in raw video is time-consuming because a compressed video stream is first decompressed into standard video, data is then embedded in the video signal, and lastly, the modified video is recompressed. This method requires fully decompressing and recompressing the video stream, a procedure that needs a lot of computer processing time.

In many papers data has been hidden in compressed domain. Some of these papers considered data hiding in DCT coefficients [5,10]. For example, Stanescu *et al.* [5] 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 a threshold its least-significant bit is set to the secret bit. Hu *et al*. [10] have presented an algorithm that embeds one bit in each qualified intra-block in H.264 bit stream.

Few papers have utilized temporal feature of the video [11,12]. Xu *et al.* [12] 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 is hidden in the horizontal component of the motion vector and for the obtuse angle, data is embedded in the vertical component.

Two important parameters in evaluating the performance of a steganographic system are capacity and imperceptibility [13-16]. 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 feature of steganography, which refers to an unauthorized person's inability to detect hidden data [17].

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 of the frames in the video sequence are not used for data embedding. These factors have led to the low capacity of their algorithm. The goal of this paper is to propose a steganographic method for covert communication, in order to increase the capacity while preserving acceptable imperceptibility. Secret data is 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. Experimental results indicate that this algorithm has high visual quality and embedding capacity. Moreover, the bit rate remains nearly constant.

2. 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 is embedded in MPEG-4 bitstream by modifying DCT coefficients and motion vectors. In order 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 (Video Object Plane) is the entire frame. For more information about MPEG-4 it can be referred to [18].

2.1 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 [19], this characteristic can be utilized in steganography and 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 in order to select blocks with high intensity changes as follows:

$$S = \sum_{m=1}^{63} | AC_m |^2 \qquad (1)$$

where AC_m is m-th quantized AC coefficient.

3. S is compared with a threshold, T_1 :

$$B_n = \begin{cases} 1 & if \quad S >= T_1 \\ 0 & if \quad S < T_1 \end{cases} \qquad (2)$$

If B_n is one, it indicates n-th 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, n-th 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:

$$\begin{aligned} &if \quad \mathrm{mod}(AC_i,2) = data(k) \quad then \quad AC_i' = AC_i \\ &if \quad \mathrm{mod}(AC_i,2) \neq data(k) \quad then \quad AC_i' = AC_i + sign(AC_i) \end{aligned} \qquad (3)$$

where AC_i' is the selected i-th quantized AC coefficient after data embedding, $sign(.)$ is the sign function and $data(k)$ is the k-th secret bit to be embedded. For example, for n-th 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 be 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 $sign(AC_i)$ is added to AC_i is that after modification, the sum of square of quantized AC coefficients always remains more than the threshold. Also for blocks without secret data, sum of the square of quantized AC coefficients is less than the threshold T_1. Therefore the same threshold can be used in data embedding and extraction. 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. Also in this case, there is no need to the original host video for extracting hidden data. Fig. 1 shows the block diagram of data embedding in quantized DCT coefficients.

2.2 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 [20], this feature can be utilized for data hiding. In the proposed method, data is 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

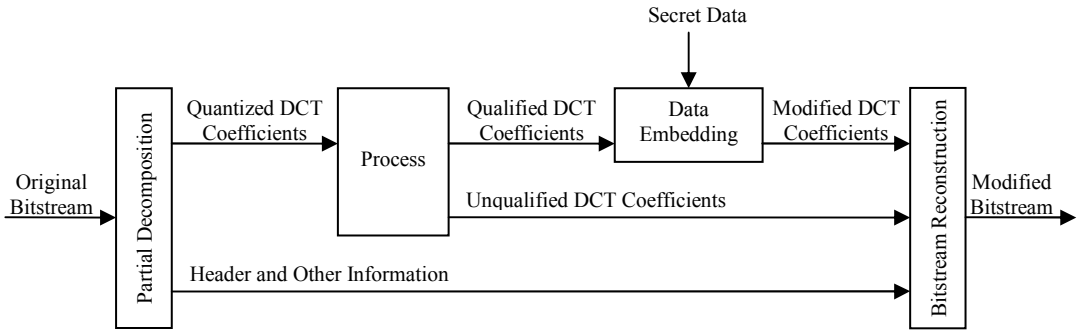


Fig. 1: Block diagram of data embedding in DCT coefficients for I-VOP.

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:

$$|MV_j| = \sqrt{H_j^2 + V_j^2} \tag{4}$$

where MV_j is the motion vector of the j-th 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, $\tilde{T}_2$:

$$AMV_j = \begin{cases} 1 & if \quad |MV_j| >= \tilde{T}_2 \\ 0 & if \quad |MV_j| < \tilde{T}_2 \end{cases} \tag{5}$$

If AMV_j is one, it means that j-th motion vector satisfies the requirement and can be used for data embedding; otherwise, it is not used for data embedding. In order to increase the speed of algorithm, instead of magnitude of the motion vector, its square is used and the threshold is also changed to $T_2 = \tilde{T}_2^2$. 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:

$$AMV_j = \begin{cases} 1 & if \quad |MV_j|^2 >= T_2 \\ 0 & if \quad |MV_j|^2 < T_2 \end{cases} \tag{6}$$

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. Since the experiments were performed with MPEG-4 Advanced Simple Profile, in which motion estimation is 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):

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

and for the vertical component it is as follows:

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

where $H_j^{'}$ and $V_j^{'}$ are horizontal and vertical components after data embedding and $data(k)$ is the k-th secret bit to be embedded.

Since 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 effect on the video quality. The reason $sign(H_j)*0.25$ is added to H_j, and also $sign(V_j)*0.25$ is added to V_j, is that after modification, the magnitude of the motion vector always remains more than the threshold. Also motion vectors without secret data have magnitude less than the threshold T_2. Therefore the same threshold T_2 can be used in data embedding and extraction. 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 the original host video in order to extract hidden data. Fig. 2 shows the block diagram of data embedding in motion vectors.

3. Extraction Algorithm

Embedded data is extracted without using the original host video. Furthermore, data extraction does not need full decompression.

3.1 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:

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

3. S' is compared with the T_1 threshold:

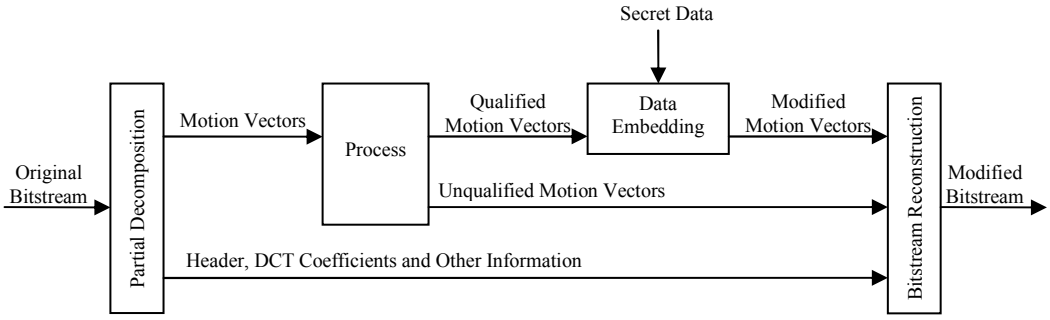


Fig. 2: Block diagram of data embedding in motion vectors for P-VOP and B-VOP.

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

4. If B'_n is one, 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:

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

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

3.2 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:

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

3. This value is compared with the T_2 threshold:

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

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

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

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 above formula it is assumed that the first bit is extracted from the horizontal component. If AMV'_j is zero, the j-th motion vector does not contain secret data.

4. Experimental Results

4.1 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 8 binary images or bit planes. To increase the security, the order of the bit planes and the position 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 one-dimensional stream and these streams were used for embedding. The 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 768 kbps. The motion vectors were supported under quarter-pixel accuracy and each VOP is the entire frame. Each group of VOPs (GOV) includes one I-VOP, three P-VOPs and eight B-VOPs, (IBBPBBPBBPBB).

For embedding data in DCT coefficients, 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 make changes in these coefficients and cause the bit rate to increase and this can draw the attention of the third party to steganography. The experiments showed that the suitable frequency range is between 9^{th} and 48^{th} of DCT coefficients, for out of this range either noticeable distortion occurred or the bit rate increased significantly.

It should be mentioned 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. According to the experiments, for $T_1 = 150$ and $T_2 = 100$ PSNR reduction is not so much and these thresholds can be suitable choices. It should be noted as it has been pointed out in the Introduction Section, steganography does not consider attacks.

4.2 Imperceptibility

Peak signal-to-noise ratio (PSNR) is the criterion that is usually used for evaluating the distortion introduced by data embedding [16]. Table I illustrates PSNR (dB) of the luminance components of the compressed video without embedded data (oPSNR) and with embedded data (sPSNR), 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, 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. Also dPSNR indicates the decrease of PSNR due to data embedding. According to the table, this value is small. 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 has 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. The most dPSNR belonged to Stefan and Bus sequences because more bits were embedded in them. But all of these had good quality and were free of visible artifact after embedding.

Fig. 3 shows a B-VOP of Bus sequence at 768 kbps before and after data embedding. It is observed that imperceptibility is high and the original VOP do not differ significantly from VOP after steganography, reducing the detection probability and leaving the observer unaware of the embedded data. Because regions with high temporal or spatial changes are selected and human visual system has low sensitivity in these regions. Also for other experiments, videos after data embedding did not differ significantly from the original videos from the viewpoint of human visual system.

4.3 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 were 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).

(a) (b)

Fig. 3: (a) Original B-VOP. (b) B-VOP after data embedding.

TABLE II: Number of embedded bits in I-VOP, P-VOP, B-VOP and GOV at 768 kbps.

Sequence	I-VOP	P-VOP	B-VOP	GOV
Bus	3048	690	334	7790
Flower Garden	3296	166	182	5250
Foreman	4096	174	110	5498
Stefan	4576	596	168	7708
Average	3754	407	199	6562

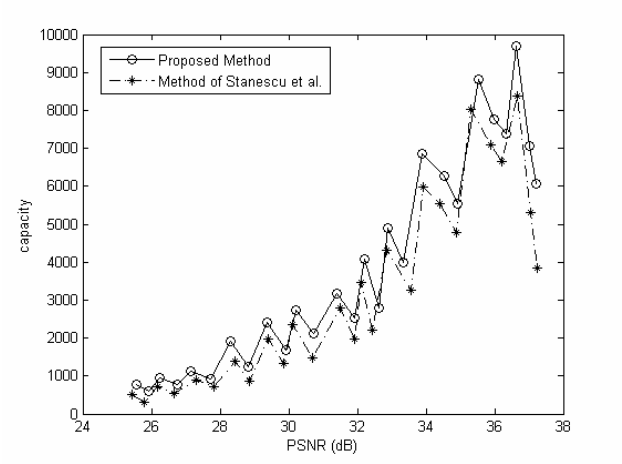
Fig. 4 show capacity vs. peak signal-to-noise ratio (PSNR) comparing the proposed method with the method of Stanescu *et al.* [5]. The method of Stanescu et al. uses spatial changes of the video and data is hidden in DCT coefficients of I-frames. It is worth mentioning that the measured points have been 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. The proposed method has more capacity in comparison with the other method for the same PSNR because the proposed method 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 Stanescu' s method only the spatial feature of the video and some frames are used for data embedding.

4.4 Security

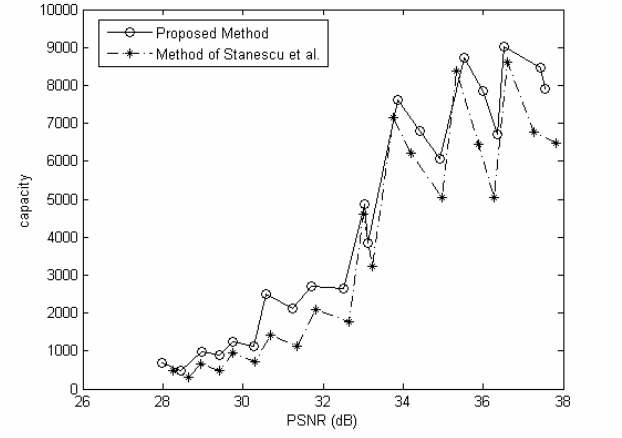
For adding security, secret bits are embedded in DCT coefficients and motion vectors using secret keys that are known between the embedder and the extractor.

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 768 kbps.

Sequence	I-VOP			P-VOP			B-VOP			GOV		
	oPSNR	**sPSNR**	**dPSNR**	**oPSNR**	**sPSNR**	**dPSNR**	**oPSNR**	**sPSNR**	**dPSNR**	**oPSNR**	**sPSNR**	**dPSNR**
Bus	33.3780	32.5889	-0.7891	33.2480	31.2335	-2.0145	31.5807	29.7085	-1.8722	32.1965	30.4201	-1.7764
Flower Garden	29.9082	29.4165	-0.4917	30.4403	30.1201	-0.3202	29.0071	28.7075	-0.2996	29.4592	29.1316	-0.3276
Foreman	39.1723	37.9154	-1.2569	38.9868	38.5309	-0.4559	37.7381	37.3013	-0.4368	38.2099	37.6701	-0.5398
Stefan	33.2713	32.2490	-1.0223	33.4593	29.2842	-4.1751	32.5109	29.5651	-2.9458	32.8298	29.8198	-3.0100
Average	33.9324	33.0425	-0.8900	34.0336	32.2922	-1.7414	32.7092	31.3206	-1.3886	33.1739	31.7604	-1.4135



(a)



(b)

Fig. 4: Capacity vs. PSNR for the proposed method and method of Stanescu et al. for the sequences of (a) Bus (b) Stefan.

4.5 Changes of the Bit Rate

Table III shows changes of the bit rate due to data embedding. The bit rate after data embedding was approximately constant and there is no need to a bit rate controller. This is another advantage of the proposed method.

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

Sequence	Bus	Flower Garden	Foreman	Stefan	Average
Changes of the bit	-0.0227	0.1449	0.0270	0.1105	0.0649

5. Conclusion

A method for video steganography for covert communication was proposed in this paper. Secret data was embedded in a compressed video bitstream adaptively using temporal and spatial features of the video signal and human visual system characteristics. In this method, for each I-VOP, the blocks with high spatial changes were selected and secret data was 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 represent high temporal changes. The method did not require the original video signal or bitstream for data extraction. Experimental results indicated that this algorithm has high visual quality and embedding capacity. Furthermore, the bit rate remained nearly constant without using a bit rate controller and this is another advantage of the proposed method.

References

[1] M. Wu and B. Liu, "Data hiding in image and video: Part I—Fundamental issues and solutions," *IEEE. Trans. Image Processing*, vol. 12, no. 6, pp. 685–695, Jun. 2003.

[2] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin, and M. Z. Shamsuddin, "Information hiding using steganography," in *Proc. IEEE Int. Conf. Telecommunication Technology*, Shah Alam, Jan. 2003, pp. 21–25.

[3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp. 1079–1107, Jul. 1999.

[4] D.-C. Lou and C.-H. Sung, "A steganographic scheme for secure communications based on the chaos and Euler theorem," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp 501–509, Jun. 2004.

[5] D. Stanescu, M. Stratulat, B. Ciubotaro, D. Chiciudean, R. Cioarga and M. Micea, "Embedding data in video stream using steganography," in *Proc. Int. Symp. Applied Computational Intelligence and Informatics*, Timisoara, May 2007, pp. 241–244.

[6] A. Briassouli and M. G. Strintzis, "Locally optimum nonlinearities for DCT watermarking detection," *IEEE Trans. Image Processing*, vol. 13, no.12, pp. 1604–1617, Dec. 2004.

[7] A. Briassouli and M. G. Strintzis, "Optimal watermark detection under quantization in the transform domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 12, pp. 1308–1319, Dec. 2004.

[8] I. J. Cox, T. Kalker, G. Pakura and M. Scheel, "Information transmission and steganography," in *Proc. Int. Workshop Digital Watermarking*, vol. 3710, Siena, Sep. 2005, pp. 15–29.

[9] 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.

[10] Y. Hu, C. Zhang and Y. Su, "Information hiding based on intra prediction modes for H.264/AVC," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2007, pp. 1231–1234.

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

[12] C. Xu, X. Ping and T. Zhang, "Steganography in compressed video stream," in *Proc. IEEE Int. Conf. Innovative Computing, Information and Control*, Beijing, vol. 1, Aug. 2006, pp. 269–272.

[13] G. Liu, Y. Dai, J. Wang, Z. Wang and S. Lian, "Image hiding by non-uniform generalized LSB and dynamic programming," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, Oct. 2005, pp. 1–4.

[14] J. Fridrich and P. Lisonek, "Grid colorings in steganography," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1547–1549, Apr. 2007.

[15] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 390–395, Sep. 2006.

[16] C.-C. Chang, C.-C. Lin, and Y.-H. Chen, "Reversible data-embedding scheme using differences between original and predicted pixel values," *IET Trans Information Security*, vol. 2, no. 2, pp. 35–46, Jun. 2008.

[17] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography," *IEEE Security Privacy Mag.*, vol.1, no. 3, pp. 32–44, May 2003.

[18] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia.* Chichester: John Wiley & Sons, 2003.

[19] 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)*, vol. 87, no. 7, pp. 1108–1126, Jul. 1999.

[20] M. M. Reid, R. J. Millar and N. D. Black, "Second-generation image coding: An overview," *ACM computing survey*, vol. 29, pp. 3–29, Mar. 1997.