Optimal Data Hiding in H.264/AVC Videos for Covert Communication[*]

Guan-Lin Huang (黃冠霖) and *Wen-Hsiang Tsai* (蔡文祥)
Dept. of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
E-mails: whtsai@cis.nctu.edu.tw, amaple0229@gmail.com

ABSTRACT

A method using an optimal data hiding technique for covert communication via H.264/AVC videos is proposed. An H.264/AVC video consists of a series of I and P image frames. We propose proper hiding techniques for macroblocks in I and P frames by utilizing their properties, respectively. We hide data into the intra-prediction mode of the I macroblock, and into the tree structured motion compensation of the P macroblock. Briefly, we modify the prediction mode and the variable partition size of the tree structured motion compensation to hide data and maintain imperceptibly of the hidden data. In addition, we also use the Lagrange optimization technique to optimize the changes yielded by the data hiding process. Experimental results show the feasibility of the proposed method.

Keywords: Covert communication, data hiding, intra-prediction mode, tree structured motion compensation, H.264/AVC, Lagrange optimization technique.

1. INTRODUCTION

Due to the growth of computer network and audio/video compression technologies, many applications of digital media emerge on the network. The transmission of secret information is one of such applications. Data hiding is a good solution to this kind of covert communication application. It is a technique for use to hide secret messages into certain *cover media* imperceivably. Videos are suitable for use as cover media because they are used widely and have large capacities for data embedding. So we try to use a kind of video, namely, H/264/AVC, as the cover media and propose accordingly an optimal data hiding method for covert communication in this study.

A lot of approaches related to hiding data into videos have been proposed [1-3]. Hu et al. [2] proposed a method for hiding data in H.264/AVC videos based on the use of intra-prediction modes which are employed for compressing video data. Kapotas et al. [3] proposed a method for embedding data in encoded video sequences, in which the technique of modulating the partition size of the blocks in the image frames was employed to hide secret data.

Traditionally, when applying video data hiding techniques for covert communication, the data hiding capacity and the imperceptibility of the hidden data are two major concerns. With the popularity of web applications, people give more and more attention to low bit rate videos. Therefore, a problem is how to hide data into videos and get optimal results which take data hiding capacity, imperceptibility, and low bit rating into consideration simultaneously.

Two macroblock types are used in the *baseline profile* of the H.264/AVC standard, namely, I macroblock and P macroblock. We propose optimal data hiding techniques for the two macroblock types, respectively, in this paper.

The method proposed for hiding data into I macroblocks is based on the use of the *intra-prediction mode*, which is a new coding scheme adopted in the H.264/AVC standard. An encoder selects the *best* prediction mode for each block by the use of a *Lagrangian cost function* [4] which minimizes the rate and the distortion in the H.264/AVC standard simultaneously. The function is formulated as follows:

$$J = \arg \min_{M_k \in \tau}(D(S_k, M_k) + \lambda R(S_k, M_k)) \qquad (1)$$

where

(1) τ denotes the set of all the nine prediction modes, i.e., $\tau = \{M_1, M_2, ..., M_9\}$;
(2) λ represents the Lagrange multiplier;
(3) S_k denotes the block being processed;
(4) D is a *distortion function* whose value is computed as the *sum of the squared differences (SSD)* between the reconstructed block S_k' and the original one S_k;
(5) R denotes the used bits for encoding the block S_k using the prediction mode M_k.

In our approach, the block S_k is fixed to be 4×4 which yields higher data embedding rates. Furthermore, we add the hiding capacity as a new parameter to the Lagrangian cost function

[*] This work was supported partially by the NSC project Advanced Technologies and Applications for Next Generation Information Networks (II) – Subproject 5: Network Security, No. 96-2752- E-009-006-PAE and partially by the NSC project No. 96-2422-H-009-001.

described by (1), resulting in:

$$J = \arg \min_{M_i \in \tau}(D(S,\ M_i) + \lambda R(S,\ M_i) - \gamma_1 \cdot N_i) \quad (2)$$

where the new parameter γ_1 is a multiplier for the hiding capacity N_i (in unit of bit) in the 4×4 block. By this function, we can get the best result as a tradeoff among the data hiding capacity, the bit rate, and the resulting distortion.

The idea of hiding data in the P macroblocks proposed in this paper is to modify the variable partition size of the tree structured motion compensation, which is a different feature of the H.264/AVC standard from earlier standards. Tree structured motion compensation is a method of partitioning macroblocks into motion compensated sub-blocks of varying sizes. The encoder selects the best partition size for each macroblock by a Lagrangian cost function described as follows:

$$J = \arg \min_{P_k \in \omega}(D(S, M_k) + \lambda R(S, M_k)) \quad (3)$$

where ω denotes the set of all alternative partition sizes, and P_k denotes the current partition size. Similarly, we add hiding capacity as a new parameter to the Lagrangian cost function, resulting in:

$$J = \arg \min_{P_i \in \omega}(D(S\ , M_i) + \lambda R(S\ , M_i) - \gamma_2 \cdot N_i) \quad (4)$$

where the new parameter γ_2 is the multiplier for hiding capacity (in unit of bit) in P macroblocks. By this formula, we can get the best result as a tradeoff among the data hiding capacity, the bit rate, and the resulting distortion.

In the remainder of this paper, the proposed method for hiding secret data into H.264/AVC videos is described in Section 2, and the corresponding data extraction method is stated in Section 3. In Section 4, several experimental results of applying the proposed method will be shown. Finally, some discussions and a summary will be made in the last section.

2. HIDING SECRET DATA INTO H.264/AVC VIDEOS

In this section, the proposed methods of hiding data into different types of macroblocks of H.264/AVC videos will be described.

2.1. Process for Hiding Data Optimally into I Macroblocks Based on Intra-Prediction Mode

We describe how we propose to hide secret data in I macroblocks optimally in a sense mentioned previously, based on the use of the nine prediction modes. Each 4×4 luma prediction mode in the I macroblock can be used to hide zero to four bits of data by this method, without influencing the degree of imperceptibility. In addition, we recode the number of bits so hidden in the highest-frequency quantized coefficients of the 4×4 block. We also use the user's secret key to encrypt the secret data to enhance the security. A detailed algorithm of the process is described in the following.

***Algorithm* 1**: optimal data hiding in I macroblocks
Input: an I macroblock I in the spatial domain , a user's key R, and a secret data file D.
Output: a stego-macroblock I'.
Steps:

```
1. For each character $D_i$ of the secret data $D$, perform the following steps.
   1.1 Compute the remainder $R'$ of dividing $R$ by 256.
   1.2 Transform each character $D_i$ of the secret data $D$ in the following way to form encrypted data $E$:
         $E_i = D_i \oplus R' \cdot$          (5)
2. For each luma 4×4 block $B$ of $I$, perform the following operations.
   2.1 For each luma 4×4 prediction mode $M_i$, perform intra-prediction, DCT-based transform, and quantization in the video coding process, and then match four bits of $E$, $E_3E_2E_1E_0$ with the 4-bit numeral value $I_0I_1I_2I_3$ the index $i$ of $M_i$ to obtain the number of bits $N_i$ which can be hidden in $M_i$ in the following way:
         if $E_3 = I_3$, set $N_i = 1$;
         if $E_3 = I_3$ and $E_2 = I_2$, set $N_i = 2$;
         if $E_3 = I_3$, $E_2 = I_2$ and $E_1 = I_1$, set $N_i = 3$;
         if all $E_j$ are equal to ;$I_j$;, set $N_i = 4$;
         otherwise, set $N_i = 0$.
   2.2 Replace the highest-frequency quantized coefficients $C$ by a new value according to the following rules:
         if $N_i = 0$, then set $C = 0$;
         if $N_i = 1$, then set $C = 1$;
         if $N_i = 2$, then set $C = -1$;          (6)
         if $N_i = 3$, then set $C = 2$;
         if $N_i = 4$, then set $C = -2$.
3. Select the best prediction mode according to Eq. (2), and decide the number of bits which can be hidden in this block. Take away from $E$ these bits.
4. Repeat the above steps to encode more bits in the remaining portion of $E$ until no more is left.
```

2.2. Process for Hiding Data Optimally into P Macroblocks Based on Tree Structured Motion Compensation

We hide secret data into P blocks based on variable partition sizes of 16×16 macroblocks. Each 16×16 P macroblock can be used to hide one or four bits of data by modifying the partition size.

In order to allow better choices of sizes to reduce rate-distortion, we encode hidden data by the partition size with multiple choices for 0 or 1 according to Table 1, in which two groups of sizes are used to encode 0 and 1, respectively. In addition, we use the user's secret key to encrypt secret data. A detailed algorithm of the process is described in the following.

***Algorithm* 2**: optimal data hiding in P macroblocks

Input: a P macroblock *P* in the spatial domain, a user's key *R*, a secret data file *D*, and the macroblock partition size *K*.

Output: a stego-macroblock *P'*.

Steps:

```
1. For each character $D_i$ of the secret data $D$, perform the following steps.
   1.1 Compute the remainder $R'$ of dividing $R$ by 256.
   1.2 Transform each character $D_i$ of the secret data $D$ according to Eq. (5) to form encrypted data $E$.
2. According to Table 3.1, hide one bit $e_1$ or four bits $e_j$ of $E$ into the macroblock partition according to the following rules for the macroblock partition size $K$.
   2.1 When the partition size K is 16×16:
       if $e_1 = 0$, then perform next partition size;
       if $e_1 = 1$, then perform Step 3.                          (7)
   2.2 When the partition size K is 8×16 or 16×8:
       if $e_1 = 0$, then perform Step 3;
       if $e_1 = 1$, then perform the next step.                   (8)
   2.3 When the partition size K is 8×8, split P into the four 8×8 sub-macroblocks $P_j$.
       For j = 1 to 4,
         i. when the partition size is 4×8 or 8×4:
            if $e_j = 0$, then perform Step 3;
            if $e_j = 1$, then perform the next step               (9)
         ii. when the partition size is 8×8 or 4×4:
            if $e_j = 0$, then perform the next step;
            if $e_j = 1$, then perform Step 3.                     (10)
3. According to Eq. (4), compute the best partition size, and decide the number of bits which can be hidden in this block. Take these bits away from $E$.
4. Repeat the above steps to encode more bits in the remaining portion of $E$ until no more is left.
```

Table 1 Relations between hidden data and partition sizes.

Partition size	Hidden data
16×16	1
8×16	0
16×8	0
8×8	1
4×8	0
8×4	0
4×4	1

3. EXTRACTION OF SECRET DATA FROM H.264/AVC VIDEOS

In this section, the proposed methods for extracting the hidden data from an input H.264/AVC stego-video will be described.

3.1. Process for Extraction of Data from I Macroblocks

The proposed data extraction process retrieves the adopted prediction modes of I macroblocks from the bitstream of the stego-video first. After entropy-decoding the stego-video, the quantized highest-frequency coefficient of each luma 4×4 block of I macroblocks is retrieved. Then we take these coefficients, the prediction modes and the user's key as input to the data extraction process for I macroblocks. A detailed algorithm is described in the following.

***Algorithm* 3:** data extraction from I macroblocks

Input: a quantized luma 4×4 block *I* of the I macroblock in the frequency domain, the prediction mode *M*, and a user's key *R*.

Output: an extracted data file *D*.

Steps:

```
1. Obtain the number of bits $N_i$ hidden in the mode $M$ from the highest-frequency coefficient $C$ of $I$ according to the following rules:
       if $C = 0$, then set $N_i = 0$;
       if $C = 1$, then set $N_i = 1$;
       if $C = -1$, then set $N_i = 2$;                            (11)
       if $C = 2$, then set $N_i = 3$;
       if $C = -2$, then set $N_i = 4$.
2. Extract the last $N_i$ bit(s) $e$ of $M$ as part of encrypted data $E$.
3. For every eight bits $E_i$ of $E$, perform the following steps.
   3.1 Compute the remainder $R'$ of dividing $R$ by 256.
   3.2 Compute the 8-bit code of each character $D_i$ of the secret data $D$ as follows and transform the obtained codes into original characters to get the embedded message:
       $D_i = E_i \oplus R'$ .                                     (12)
```

3.2. Process for Extraction of Data from P Macroblocks

The proposed data extraction process retrieves the macroblock partition sizes of the P macroblocks from the stego-video first; if the macroblock partition size is 8×8, we get the sub-macroblock partition size of P macroblocks further, and take the partition size and the user's key as input to the data extraction process for P macroblocks. A detailed algorithm is described in the following.

```
Algorithm 4: data extraction from P macroblocks.
Input: a macroblock partition size $P$ (and four sub-macroblock partition size $P_j$) and a user's key $R$.
Output: an extracted data file $D$.
Steps:
  1. Extract a bit $e_1$ or four bits $e_j$ as part of the encrypted data $E$ from the $P$ according to the following rules.
    1.1 When P is 16×16, 8×16, or 16×8, extract a bit $e_1$ from the macroblock partition size $P_1$ in the following way:
        if $P_1$ is 16×16, then set $e_1 = 1$;
        if $P_1$ is 8×16, 16×8 then set $e_1 = 0$.        (13)
    1.2 When P is 8×8, extract 4 bits $e_j$ from four sub-macroblocks partition size $P_j$ in the following way:
        for $j$ = 1 to 4:
          if $P_j$ is 8×8, 4×4, then set $e_j = 1$;
          if $P_j$ is 4×8, 8×4, then set $e_j = 0$.       (14)
  2. For every 8 bits $E_i$ of $E$, perform the following steps.
    2.1 Compute the remainder $R'$ of dividing $R$ by 256.
    2.2 Set each character $D_i$ of the secret data $D$ according to Equation (3.11).
```

4. EXPERIMENTAL RESULTS

The proposed optimal data hiding algorithms were integrated into the H.264 reference software JM12.4 [5]. The most important configuration parameters of the JM12.4 are shown in Table 2; other parameters are kept to retain their default values. Several video sequences, Foreman, Football, Mobile and Tempete, in CIF (352×288 pixels) format were used in our experiments.

Table 2 Configuration parameters.

Profile	Baseline
No. of frames to be coded	10
Period of I-pictures	5
RD Optimization	High complexity mode

The performance of the optimal data hiding algorithms was evaluated with the number of bits hidden (NBH), the peak-signal-to-noise-ratio increase in the Y color space (PSNRI), the bit rate increase (BRI) and the subjective perception testing by comparing the original video and the stego video frames with hidden secret data. The secret data with the size of 2262 bytes used in the experiments are shown in Figure 1. Four of ten frames of the input video are shown in Figure 2 and four of ten frames of the resulting stego-video are shown in Figure 3. The extracted data are shown in Figure 4. Finally the NBH, PSNRI and BRI values for several video sequences are shown in Table 3. From the above results and data, it can be observed that the proposed method can embed the secret data into H.264/AVC videos imperceptibly with light bit rate increasing.

Table 3 NBH, PSNRI and BRI values for several video sequences (γ_i,γ_p= 250).

	Football	**Foreman**	**Mobile**	**Tempete**
Average of PSNRI/NBH in I macroblocks(%)	-0.031761	-0.052796	-0.024492	-0.030125
Average of BRI/NBH in I macroblocks	7.668156	13.382799	4.941370	6.188712
Average of NBH in I macroblocks	1567	593	3650	2321
Average of PSNRI/NBH in P macroblocks(%)	-0.189027	-0.162934	-0.235225	-0.250322
Average of BRI/NBH in P macroblocks	27.370086	11.071856	17.341488	13.365207
Average of NBH in P macroblocks	1043	835	981	868

5. CONCLUSIONS

In this paper, we have proposed an optimal data hiding method that can be used to hide data into I macroblocks and P macroblocks. The optimal data hiding method not only considers hiding capacity of secret data and imperceptibility, but also considers bit rates. Therefore, the method is suitable for covert communication applications, especially when it is necessary to transmit an H.264/AVC video in a low bit rate network.

REFERENCES

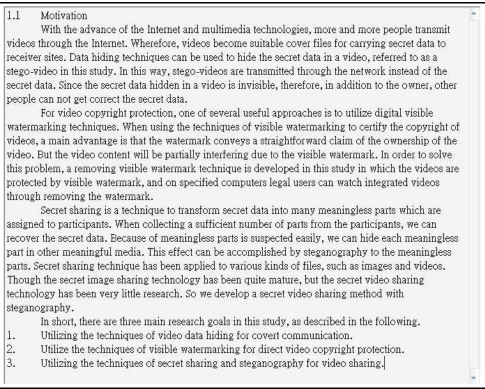
[1] M. Yang and N. Bourbakis, "A High Bitrate Information Hiding Algorithm for Digital Video Content under H.264/AVC Compression," *Proceedings of IEEE International Conference on Image Processing Midwest Symposium on Circuits and Systems*, Cincinnati, OH, USA, vol. 2, pp. 935-938, Aug., 2005.

[2] Y. Hu, et al., "Information hiding based on intra prediction modes for H.264/AVC," *Proceedings of IEEE International Conference on Multimedia and Expo*, Beijing, China, pp. 1231-1234, Jul., 2007.

[3] S. K. Kapotas, et al., "Data hiding in H.264 encoded video sequences," *Proceedings of International Workshop on Multimedia Signal Processing*, Chania, Crete, Greece, pp. 373-376, Oct., 2007.

[4] T. Wiegand et al., "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, pp. 688- 703, July, 2003.

[5] H.264/AVC JVT reference software JM12.4, http://iphome.hhi.de/suehring/tml/download/



1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. Wherefore, videos become suitable cover files for carrying secret data to receiver sites. Data hiding techniques can be used to hide the secret data in a video, referred to as a stego-video in this study. In this way, stego-videos are transmitted through the network instead of the secret data. Since the secret data hidden in a video is invisible, therefore, in addition to the owner, other people can not get correct the secret data.

For video copyright protection, one of several useful approaches is to utilize digital visible watermarking techniques. When using the techniques of visible watermarking to certify the copyright of videos, a main advantage is that the watermark conveys a straightforward claim of the ownership of the video. But the video content will be partially interfering due to the visible watermark. In order to solve this problem, a removing visible watermark technique is developed in this study in which the videos are protected by visible watermark, and on specified computers legal users can watch integrated videos through removing the watermark.

Secret sharing is a technique to transform secret data into many meaningless parts which are assigned to participants. When collecting a sufficient number of parts from the participants, we can recover the secret data. Because of meaningless parts is suspected easily, we can hide each meaningless part in other meaningful media. This effect can be accomplished by steganography to the meaningless parts. Secret sharing technique has been applied to various kinds of files, such as images and videos. Though the secret image sharing technology has been quite mature, but the secret video sharing technology has been very little research. So we develop a secret video sharing method with steganography.

In short, there are three main research goals in this study, as described in the following.

1. Utilizing the techniques of video data hiding for covert communication.
2. Utilize the techniques of visible watermarking for direct video copyright protection.
3. Utilizing the techniques of secret sharing and steganography for video sharing.

Figure 1 The secret data file.

(a)	(b)	(c)	(d)

Figure 2 Four frames of the original video (a) 4th frame (P frame). (b) 5th frame (P frame). (c) 6th frame (I frame). (d) 7th frame (P frame)

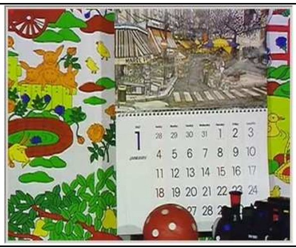
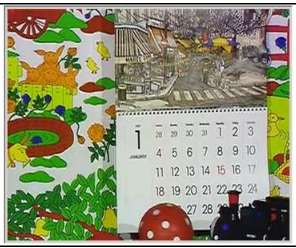
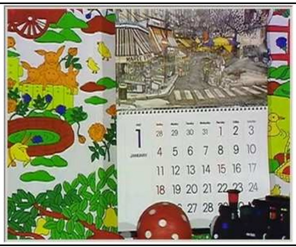
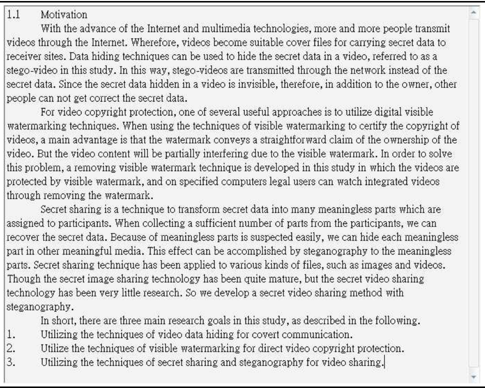
			
(a)	(b)	(c)	(d)

Figure 3 Four frames of the stego-video (a) 4th frame (P frame). (b) 5th frame (P frame). (c) 6th frame (I frame). (d) 7th frame (P frame)



1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. Wherefore, videos become suitable cover files for carrying secret data to receiver sites. Data hiding techniques can be used to hide the secret data in a video, referred to as a stego-video in this study. In this way, stego-videos are transmitted through the network instead of the secret data. Since the secret data hidden in a video is invisible, therefore, in addition to the owner, other people can not get correct the secret data.

For video copyright protection, one of several useful approaches is to utilize digital visible watermarking techniques. When using the techniques of visible watermarking to certify the copyright of videos, a main advantage is that the watermark conveys a straightforward claim of the ownership of the video. But the video content will be partially interfering due to the visible watermark. In order to solve this problem, a removing visible watermark technique is developed in this study in which the videos are protected by visible watermark, and on specified computers legal users can watch integrated videos through removing the watermark.

Secret sharing is a technique to transform secret data into many meaningless parts which are assigned to participants. When collecting a sufficient number of parts from the participants, we can recover the secret data. Because of meaningless parts is suspected easily, we can hide each meaningless part in other meaningful media. This effect can be accomplished by steganography to the meaningless parts. Secret sharing technique has been applied to various kinds of files, such as images and videos. Though the secret image sharing technology has been quite mature, but the secret video sharing technology has been very little research. So we develop a secret video sharing method with steganography.

In short, there are three main research goals in this study, as described in the following.

1. Utilizing the techniques of video data hiding for covert communication.
2. Utilize the techniques of visible watermarking for direct video copyright protection.
3. Utilizing the techniques of secret sharing and steganography for video sharing.

Figure 4 The extracted data file.