Protection of Privacy-sensitive Contents in Surveillance Videos Using WebM Video Features[†]

Hsin-Hsiang Tseng[1] and Wen-Hsiang Tsai[1, 2, *]

[1] Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan
itsummervar@gmail.com
[2] Department of Information Communication, Asia University, Taiwan
whtsai@cis.nctu.edu.tw

Abstract. Privacy protection is a critical issue in video surveillance because nowadays video cameras existing everywhere might monitor spaces of individuals and violate protection of personal privacy. Based on the data hiding approach, a new method for protection of privacy-sensitive contents in surveillance videos of the recently-developed WebM format is proposed. With skillful uses of certain special features of the WebM video, techniques for removing privacy-sensitive contents from a given WebM video, embedding the removed contents into the same video imperceptibly, and extracting the hidden contents later to recover the original privacy-sensitive contents are proposed. Experimental results showing the feasibility of the proposed method are also included.

Keywords: data hiding, privacy protection, video surveillance, WebM video.

1 Introduction

Privacy protection is an important issue in video surveillance. Since video cameras exist everywhere in our environments nowadays, which conduct monitoring of public or private spaces for long time periods, it might happen that information of individuals' activities is videotaped, leading to infringement upon personal privacy. Hence, it is necessary occasionally to hide the privacy-violating parts of surveillance video contents to avoid legal disputes or to protect personal privacy from being misused. It is so desirable to propose methods to solve this issue of privacy protection in videos.

Several methods have been proposed for this purpose. Dufaux et al. [1] proposed a method to scramble regions in videos containing personal information; the resulting scene in the video remains visible, but the privacy-sensitive information is not identifiable. A method was proposed by Meuel et al. [2] to protect faces in surveillance videos; any visible information of faces in a video is deleted and embedded in the same video, allowing later reconstructions of the faces when needed. Zhang et al. [3] presented a method to protect authorized persons appearing in videos, which conducts both removal and embedding of the shapes of concerned persons in videos. Also proposed by Yu et al. [4] is a method for privacy protection by controlling the disclosure of individuals' visual information in videos using visual abstraction operations like silhouette generation and transparency enhancement.

About data hiding via videos, many techniques have been proposed in the past

[†] This work was supported in part by the NSC, Taiwan under Grant No. 100-2631-H-009-001 and in part by the Ministry of Education, Taiwan under the 5-year Project of "Aiming for the Top University" from 2011 through 2015.

[*] To whom all correspondence should be sent.

decade [5-8]. By data hiding, information can be transmitted covertly or kept securely for various applications. Hu et al. [5] proposed a method for hiding data in H.264/AVC videos based on the idea of modifying 4×4 intra-prediction modes to encode hidden bits. Only intra-coded macroblocks are used to hide data. Hussein [6] proposed a method for embedding data in motion vectors based on their associated prediction errors. Yang and Bourbakis [7] proposed a method for embedding data in the DCT coefficients by means of vector quantization. Kapotas et al. [8] proposed a method for embedding data into encoded video sequences by modulating the partition size; only inter-coded macroblocks are used for embedding information.

In this study, we propose a new method to deal with the problem of privacy protection in WebM videos. By the method, an authorized user is allowed to specify a region R identically-located in each frame of an input *cover video* V. The region R, called the *protection region* hereafter, presumably contains the privacy-sensitive content C_i in each frame F_i of V. The privacy-sensitive contents of all the frames of V are collected sequentially to form a *privacy-sensitive data set* E. Then, a process of replacing the privacy-sensitive content C_i with an identical *background image portion* B in each video frame is conducted automatically. Also, the privacy-sensitive data set E is hidden into the same video to produce a *privacy-protected video* V_p. Thereafter, whenever the privacy-sensitive contents need be recovered, the hidden data in V_p can be extracted to reconstruct the original video. The main contributions of this study are skillful uses of special features of the WebM video format for *removals*, *embeddings*, and *recoveries* of privacy-sensitive contents to achieve the purpose of privacy protection in video surveillance using WebM videos.

In the remainder of this paper, the proposed data hiding technique for embedding privacy-sensitive contents and the corresponding data extraction technique are introduced first in Section 2 after a brief review of the WebM video format is given. Proposed techniques for removing and recovering the privacy-sensitive contents in WebM videos are described in Section 3. In Section 4, some experimental results are presented, followed by conclusions in the last section.

2 Data Hiding in WebM Videos

2.1 Brief Review of WebM Video Format

WebM is an open media file format designed for the web whose openness was offered by Google Inc. in May 2010 [10]. Each WebM file consists of video streams compressed with the VP8 video codec and audio streams compressed with the Vorbis audio codec. The VP8 video codec works exclusively with an 8-bit YUV 4:2:0 image format, each 8-bit chroma pixel in the two chroma color spaces (U and V) corresponds to a 2×2 block of 8-bit luma pixels in the luma color space (Y).

Also, each frame in a WebM video is decomposed into an array of macroblocks. And each macroblock is a square array of pixels whose Y dimensions are 16×16 and whose U and V dimensions are 8×8. The macroblock-level data in a compressed frame are processed in a raster-scan order. Each macroblock is further decomposed into four 4×4 subblocks. So each macroblock has sixteen Y subblocks, four U subblocks, and four V subblocks. These three types of subblocks, when composed together respectively, are called the *Y*, *U*, and *V components* of the macroblock henceforth. So, the *Y* component is a 16×16 array of 8-bit luma pixels, and the U and V components are both 8×8 arrays of chroma pixels. Fig. 1 illustrates one of the 4×4 subblocks of a macroblock.

The VP8 video codec transforms pixels in the spatial domain into coefficients in the frequency domain by the discrete cosine transform (DCT) and the Walsh-Hadamard transform (WHT) at the 4×4 resolution. The DCT is used for the sixteen Y, four U, and four V subblocks and the WHT is used to encode a 4×4 array comprising the average intensities of the sixteen Y subblocks of a macroblock. These average intensities are, up to a constant normalization factor, nothing more than the zeroth DCT coefficients of the Y subblocks. The VP8 video codec considers this 4×4 array as a second-order subblock, called the Y2 subblock.

Furthermore, two frame types are used in the VP8 codec, namely, *intra-frame* and *inter-frame*. Intra-frames, also called *key frames* or *I-frames*, are decoded without reference to other frames in a sequence; and inter-frames, also called *prediction frames* or *P-frames*, are encoded with reference to prior frames, including the recent key frame.

2.2 Ideas of Proposed Data Hiding Method

The pixel values in each 4×4 subblock of a WebM video frame are converted by the DCT into frequency-domain coefficients, and the energy of the coefficient signals is "clumped" at the left-upper corner of the subblock. In addition, after the coefficients are quantized according to an adaptive level and put into a zigzag order as illustrated by Fig. 2, near-zero coefficients will usually appear in the *positive-sloped diagonal* of the subblock as shown by the example illustrated in Fig. 3, where the positive-sloped diagonal means the four red squares in Fig. 3. Furthermore, human eyes have lower sensitivity on high-frequency signals and chrominance than on low-frequency signals and luminance [9]. Accordingly, it is proposed is this study to define 16 data patterns to replace the DCT coefficients in the positive-sloped diagonal of the 4×4 subblock to achieve imperceptible data hiding.

In addition, we pre-compute the PSNR resulting from data hiding for each macroblock, and if the computed value is too large (larger than a pre-selected threshold), then data hiding is abandoned. It is in this way that the proposed method maintains good video quality in the data hiding result. To mark which macroblock has been used for data hiding, we use a feature of the WebM video, namely, *region-of-interest* (ROI) *map*. Each macroblock has its own map index. Such an index is also encoded into the bitstream by tree coding in the compressed video data. Taking advantage of this feature, we use the map index as a *data embedding marker* to label each of those macroblocks whose coefficients have been modified for data hiding.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Fig. 1. A subblock with yellow coefficients composing a positive-sloped diagonal line.

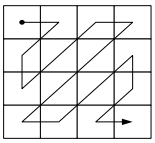


Fig. 2. Zigzag scan order of coefficients in a subblock of WebM video.

81	20	6	-2
11	4	0	0
1	0	0	0
0	0	0	0

Fig. 3. A subblock obtained after DCT and coefficient quantization with red coefficients composing a diagonal line.

2.2 Embedding of Message Data into WebM Videos

In this section, we will describe the detailed algorithm of the proposed method for hiding privacy-sensitive contents into cover videos by changing the frequency coefficients into pre-defined data patterns. At first, the aforementioned 16 data patterns

for use in the proposed algorithm are defined. For this, let the notations N and 0 denote the meanings "non-zero" and "zero," respectively. Then, a *data pattern* DP_i (i = 0 to 15) is defined as a 4×4 block with its positive-sloped diagonal being filled with four symbols $S_3S_2S_1S_0$ of N's and 0's, which correspond to the binary value $b_3b_2b_1b_0$ of i in the following way: if $b_j = 0$, then $S_j = 0$; and if $b_j = 1$, then, $S_j = N$, where j = 0, 1, 2, 3. Fig. 4 illustrates all the 16 data patterns DP_0 through DP_{15}. In each of the 4×4 data patterns, the top-rightmost square contains S_0 and the bottom-leftmost one S_3, and so on. For example, when i = 3, the corresponding binary value is 0011_2 and the defined data pattern DP_3 is a 4×4 block with its positive-sloped diagonal being filled with the four symbols $S_3S_2S_1S_0$ = 00NN. And when i =10, the corresponding binary value is 1010_2 and the data pattern DP_{10} has its positive-sloped diagonal being filled with $S_3S_2S_1S_0$ = N0N0.

Each of the data patterns DP_i is used in this study to embed a message data item i into a 4×4 frequency-coefficient subblock SB in a video frame by the following *match-and-replace rule* where the four elements of the positive-sloped diagonal of SB are denoted as $B_3B_2B_1B_0$, and those corresponding ones of DP_i as $S_3S_2S_1S_0$, respectively:

if $S_j = 0$, *then replace* B_j *by 0; if* $S_j = N$, *then*
if $B_j \neq 0$, *keep* B_j *unchanged; otherwise, set* $B_j = 1$. (1)

Reversely, when the message data item i already embedded in SB is to be extracted, the following *match-and-extract rule* is conducted:

if $B_j = 0$, *then extract* S_j *to be 0; if* $B_j \neq 0$, *then extract* S_j *to be N.* (2)

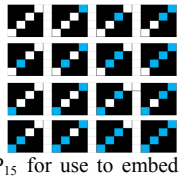


Fig. 4. Sixteen data patterns DP_0 through DP_{15} for use to embed message data 0 through 15, respectively, where the small white squares means 0's and the blue ones mean N's (non-zeros).

In the following algorithm, given a cover WebM video V, we assume that a protection region R has been selected for V and the privacy-sensitive contents of all the video frames in R have been removed (using Algorithm 3 described later in Section 3.2) and collected as a privacy-sensitive data set E, resulting in a *non*-privacy-sensitive cover WebM video which we denote by V_o. We regard E to be associated with V_o. The technique we propose to remove the privacy-sensitive content from the protection region in each prediction frame will be described later in the next section.

Algorithm 1: ***embedding privacy-sensitive contents into a non-privacy-sensitive cover WebM video.***

Input*:* a non-privacy-sensitive cover WebM video V_o and its associated privacy-sensitive data set E of V_o, a secret key K, a random-number generating function f, and a threshold value T.

Output: a privacy-protected video V_p with E being embedded in it.

Steps.

Step 1. (*Randomizing the privacy-sensitive data set E*) Transform data set E in a character form into a binary string B, use key K and function f to randomize string B, and divide the result into a sequence A of 4-bit segments.

Step 2. (*Generating a random sequence for later uses in randomizing generated data patterns*) Use K and f to generate a sequence Q of random numbers.

Step 3. (*Embedding the random data sequence A*) Take sequentially an *unprocessed* macroblock *MB* from the prediction frames of V_o, and perform the following steps to embed the data of sequence A into *MB*.

3.1 (*Generating data patterns which encode the data to be embedded*) Take sequentially eight *unprocessed* 4-bit elements $A_1, A_2, \ldots, A_8$ from A; and for each A_j with binary value i, generate the corresponding 4×4 data pattern DP_{i_j} (e.g., if $A_j = i_j = 1001_2 = 9_{10}$, then generate DP_9), where $j = 1, 2, \ldots, 8$.

3.2 (*Randomizing the generated data patterns*) Take sequentially eight *unprocessed* elements, $N_1, N_2, \ldots, N_8$, from sequence Q; and for each N_j with $j = 1, 2, \ldots, 8$, combine it with DP_{i_j} by the exclusive-OR operator $\oplus$ to yield a new data pattern $DP_{i_j}' = DP_{i_j} \oplus N_j$.

3.3 (*Saving the original macroblock content*) Save the original content of *MB* into a temporary macroblock MB_{temp}.

3.4 (*Embedding the generated random data string DP_{i_j}'*) Denote the eight subblocks of the chroma channels, four in the U channel and the other four in the V channel, of macroblock *MB* by $SB_1, SB_2, \ldots, SB_8$, and for each SB_{i_j} with $j = 1, 2, \ldots, 8$, conduct the match-and-replace rule described previously by (1) to embed DP_{i_j}' into SB_j, resulting in a *modified* macroblock *MB'*.

3.5 (*Computing the distortion in the modified macroblock MB'*) Denote the U and V components of *MB* as MB_U and MB_V, respectively, and those of *MB'* as MB_U' and MB_V', respectively; compute the average peak signal-to-noise ratios (PSNRs), $PSNR_U$ and $PSNR_V$, of MB_U' and MB_V' with respect to MB_U and MB_V, respectively, as well as the average PSNR, $PSNR_{avg}$, of $PSNR_U$ and $PSNR_V$ as $PSNR_{avg} = (PSNR_U + PSNR_V)/2$ for *MB'* with respect to *MB*.

3.6 (*Checking the data embeddability of macroblock MB*) If $PSNR_{avg}$ is smaller than the input pre-selected threshold value T, then regard *MB* as *data-embeddable* by setting the ROI map index mi_{ROI} of *MB'* to be 1; else, keep the default value, which is 0, of mi_{ROI} unchanged, meaning that *MB* is non-data-embeddable and resume the original content of *MB* to be those saved in the temporary macroblock MB_{temp}.

Step 4. (*Ending*) Repeat Step 3 until the entire content of sequence A is embedded, take the final version of video V_o as the desired privacy-protected video V_p, and exit.

2.3 Extraction of Embedded Data from WebM Videos

The proposed process for extracting privacy-sensitive contents from a privacy-protected video is described as an algorithm in the following, which essentially is a reverse version of Algorithm 1 proposed previously for privacy-sensitive content embedding.

Algorithm 2: ***extraction of the privacy-sensitive contents from a privacy-protected WebM video.***

Input: a privacy-protected WebM video V_p; and the secret key K and the random number generating function f used in Algorithm 1.

Output: a privacy-sensitive data set E including the privacy-sensitive contents of all frames in the original video V within a privacy region R.

Steps.

Step 1. Use key K and function f to generate a sequence Q of random numbers and take the first eight random numbers from Q, denoted as N_1 through N_8.

Step 2. For each macroblock MB_k in the prediction frames of V_p with its ROI map index $MI_{ROI} = 1$, perform the following steps to extract data embedded in it.

1.1 Take out the eight subblocks SB_1, SB_2, …, SB_8 of the chroma channels, four in the U channel and the other four in the V channel, and for each SB_j with $j = 1, 2, \ldots, 8$, conduct the match-and-extract rule described by (2) above to extract a data pattern DP_{i_j}', and combine it with N_j by the exclusive-OR operator $\oplus$ to yield a new data pattern $DP_{i_j} = DP_{i_j} \oplus N_i$.

1.2 Transform each DP_{i_j} with $j = 1, 2, \ldots, 8$ into a 4-bit segment A_j with its decimal value equal to i_j, and concatenate all A_j sequentially to form a binary string B_k.

Step 3. Concatenate all the binary strings B_k's obtained above to form a binary string B'.

Step 4. Use key K and function f to de-randomize B' to get another binary string B, and transform B into a character form as the desired output data set E.

3 Protection of Privacy-sensitive Contents

3.1 Idea of Proposed Method

Like other codecs, the VP8 video codec has a process to find the best prediction block in blocks. A *motion vector* is used to indicate the location of the best prediction block. The difference between the best prediction block and the currently-processed block is converted by the DCT into a set of *frequency coefficients*. Motion vectors and frequency coefficients are used in the decoding process to decode corresponding blocks, and they together are called the *decoding information* hereafter.

A video can be decoded correctly based on the decoding information generated during the encoding process. The idea we propose to protect the privacy-sensitive content C in a selected protection region R specified by a user is to set the decoding information of C to be some *pre-defined* values. In this way, the privacy-sensitive video content can be removed and replaced by the background image. The decoding information of C is then embedded into the video. If the privacy-sensitive content of C need be recovered, the embedded decoding information of C may be extracted and used to conduct the recovery work.

A critical issue to overcome here is how to replace the privacy-sensitive content C with the background image without causing negative visible effects. In order to remove the privacy-sensitive content C in the user-specified protection region R, we have to replace an *encoded macroblock* in the encoding process, but this will cause a *reference problem* here, which occurs when the encoded macroblock is used as a reference to encode other macroblocks during the encoding process. This problem, if not solved, will cause errors in the decoding result. Fig. 5 shows an example of the reference problem.

In this study, we propose the use of the *golden reference frame*, which is a feature of the WebM video, to solve this reference problem. It is provided for the VP8 video codec to store a video frame from an arbitrary point in the past. The VP8 encoder could use such a type of frame to maintain a copy of the background image when there are objects moving in the foreground part; by using the golden reference frame, the foreground part can be easily and cheaply reconstructed when a foreground object moves away. And this is just what we need for dealing surveillance videos in this study because a surveillance video often comes from monitoring a fixed area for a long time, and the background

image is usually still with no moving object included.

Another problem encountered here is caused by the use of the intra-coded macroblock, which is a type of encoded macroblock used by the VP8 video codec. With the intra prediction mode, an intra-coded macroblock does not use any reference frame and appear in the last video frame, a golden reference frame, or an alternative reference frame [10]. Therefore, any modification of the frequency coefficients in an intra-coded macroblock to remove the privacy-sensitive content will result in a grey color macroblock rather than the background image. Fig. 6 shows an example of this problematic phenomenon. To solve this problem, we choose to enforce the VP8 encoder to use the inter prediction mode when displaying the privacy-protected video.

Fig. 5. An example of errors caused by the reference problem.

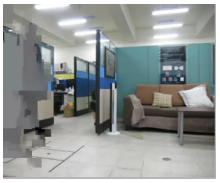


Fig. 6. Modifying coefficients in an intra-coded macroblock yields grey macroblocks.

3.2 Process for Removing Privacy-Sensitive Contents

The proposed process for removing privacy-sensitive contents as described in the following is applied to the prediction frames of an input WebM video. After a protection region R in which privacy-sensitive contents, if there is any, should be removed is specified by the user, the motion vectors and frequency coefficients of the currently-processed macroblock within R are all set to be zero. Also, we assume that the first video frame is a background image which is taken to be a golden reference frame. The values of the original motion vectors and frequency coefficients of the macroblocks of all frames within R are grouped sequentially to form a data set E and hidden into the prediction frames in the input video using algorithm 1, as mentioned previously.

Algorithm 3: ***removing the privacy-sensitive contents in a specified protection region.***

Input: a WebM video V and a pre-specified protection region R.

Output: a non-privacy-sensitive WebM video V_o with the privacy-sensitive contents in R of all the frames removed and collected as a privacy-sensitive data set E.

Steps.

Step 1. Take the first frame of video V as the golden reference frame, and set it as the reference frame for each prediction frame F in V.

Step 2. Restrict the encoder to use inter prediction modes during the prediction step.

Step 3. For each prediction frame F in V, perform the following steps.

3.1 Detect motions in region R of F by checking the prediction mode of each macroblock within R; and if there exists any macroblock with its prediction mode other than ZEROMV (meaning "no motion exists") [10], then set a *motion flag* f_m to be 1; otherwise, set f_m to be 0.

3.2 If f_m = 1, then for each macroblock MB within R of F, perform the following steps.

(1) Record all the sixteen coefficients of the Y2 subblock into E and set all the sixteen coefficients of this subblock to be zero.

(2) Record the DC coefficient of each subblock of the chroma channels (including the U channel and the V channel) into E and set all the coefficients of these subblocks to be zero.
(3) Record, according to the zigzag scan order as shown in Fig. 7, the first seven coefficients of each subblock of the luma color channel (the Y channel) into E and set all the coefficients of these subblocks to be zero.
(4) Record the index of MB and the index of F into E.
(5) Record the motion vector of MB into E and set this vector to be zero.

Step 4. Repeat Step 3 until all frames of V have been processed, take the final versions of V and E as the desired outputs V_o and E, respectively, and exit.

Some considerations involved in designing the above algorithm are reviewed here. Considering the capacities of the proposed hiding data method, we cannot record all the coefficients in a macroblock which will be used to recover privacy-sensitive contents. Therefore, we conduct some tests in order to decide which coefficients of color channels should be recorded. There exists a type of 4×4 second-order subblock in the WebM video called Y2, as mentioned previously, which records the DC coefficients of all the sixteen Y subblocks. If we lose the coefficients of the Y2 subblock, we cannot recover the contents in this macroblock. Fig. 7 shows an example of such cases. Therefore, we record all the coefficients in the Y2 subblock in order to make sure we can recover the privacy-sensitive contents. In addition, because the VP8 video codec uses the zigzag scan order, which is shown in Fig. 2, to encode subblocks, after some experimental tests we decide to record, according to the zigzag scan order, the first seven coefficients of each Y subblock, as described in Step 3.2(3).

(a) (b)

Fig. 7. Comparison between original image and image with coefficients of Y2 subblocks lost. (a) Original image. (b) Image with coefficients of Y2 subblocks in a region lost.

3.3 Process for Recovering Privacy-Sensitive Contents

Once the privacy-sensitive contents in an input privacy-protected video need be recovered, the recovery information, namely, the privacy-sensitive data set E, extracted by Algorithm 2 may be used to recover the original privacy-sensitive contents. If the extracted data are correct, we can know accordingly the positions of the protected regions in the frames of the input privacy protected video and the original privacy-sensitive contents.

There are two phases in the proposed process for recovery of privacy information in a privacy-protected video. The first is to extract the recovery information of the protected region in the privacy-protected video by Algorithm 2. The second is to replace the contents of the protected regions with the recovery information. A detailed algorithm for the second phase is described in the following.

Algorithm 4: ***recovering the privacy-sensitive contents of a privacy-protected WebM video.***

Input: a privacy-protected WebM video V_p, and a privacy-sensitive data set E of the protected regions in V.
Output: a WebM video V' with the privacy-sensitive contents recovered.
Steps.
Step 1. Set a *recovery flag* f_r initially to be 1.
Step 2. Take sequentially an unprocessed prediction frame F from V_p and perform the following steps.
  2.1 Take an unprocessed macroblock MB from F and perform the following steps.
    (1) Extract sequentially a set of unextracted recovery information from E, including a frame index i_f, a macroblock index i_{mb}, a motion vector MV, and a set FC of frequency coefficients.
    (2) If the frame index of F is equal to i_f and the macroblock index of MB is equal to i_{mb}, then replace the motion vector in MB by MV, and replace the set of frequency coefficients in MB by the respective ones in FC.
  2.2 Repeat Step 2.1 until the macroblocks in F are exhausted
Step 3. Repeat Step 2 until the prediction frames in V_p are exhausted (i.e., until reaching the end of V_p).

4 Experimental Results

Four clips of surveillance videos are used in part of our experiments using the previously-proposed algorithms (Algorithms 1 through 4) conducted in a sequence as illustrated in Fig. 8. The first one is acquired by a camera monitoring an aisle of Engineering Building 5 in National Chiao Tung University. Four representative original frames of the video clip are shown in Fig. 9(a). The purpose of surveillance was to monitor activities around the aisle, but it was hoped that the personal information appearing in the window of the second floor would not be revealed. Therefore, we utilized the proposed process for removing privacy-sensitive contents (Algorithm 3) to conceal such personal information. Furthermore, we embedded the information into the resulting video to yield a privacy-protected video by Algorithm 1. The four frames of the privacy-protected video yielded by Algorithm 1 and corresponding to those of Fig. 9(a) are shown in Fig. 9(b). Finally, the four corresponding frames of the recovered video yielded by the proposed processes for recovering the privacy-sensitive contents (Algorithms) are shown in Fig. 9(c). A comparison between an original frame and the corresponding recovered one is shown in Fig. 10. The average PSNR of the recovered image part in the protection region with respect to the original one is 35.73.

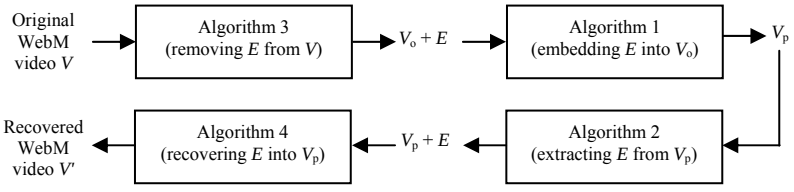
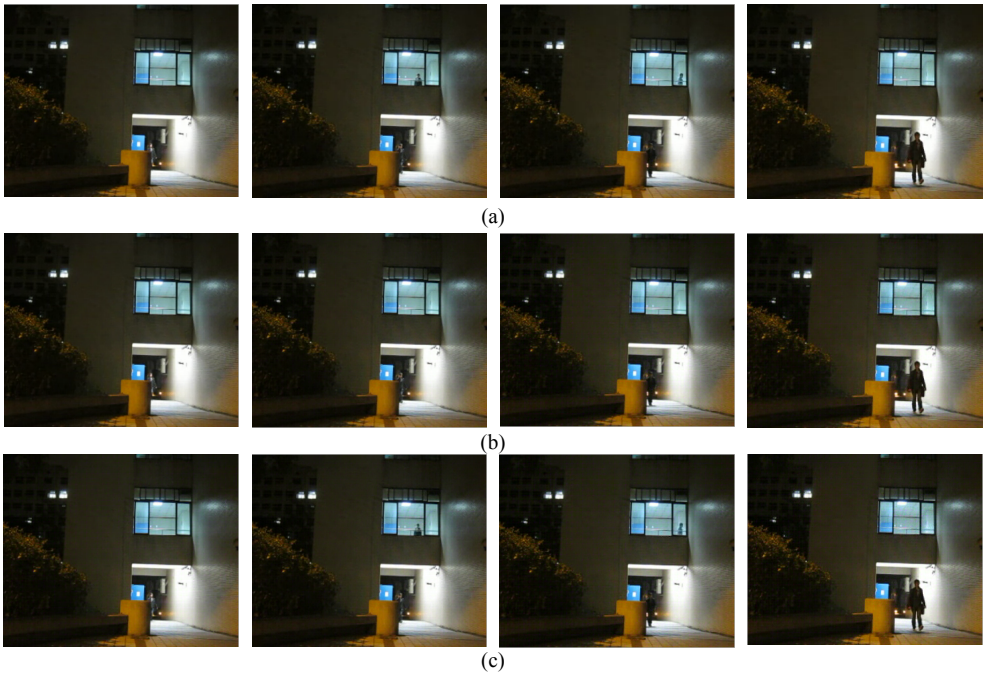


Fig. 8. Illustration of sequence of conducting proposed processes (Algorithms 3, 1, 2, 4).

Another surveillance video used in our experiments came from monitoring the Computer Vision Lab at National Chiao Tung University. A comparison between an original frame and the corresponding recovered one is shown in Fig. 11. The average PSNR of the recovered image part with respect to the original one is 30.372, which together with the previous one (35.73) indicate that the qualities of the recovered video frames are good for practical applications. More PSNR values of other experimental results show the same conclusion.

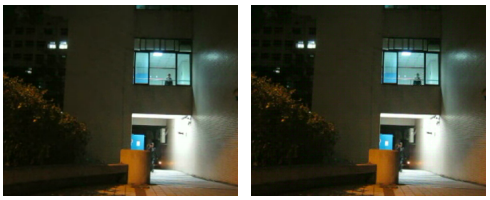


(a)

(b)

(c)

Fig. 9. Representative original and processed video frames. (a) Four original video frames. (b) Four corresponding frames in the privacy-protected video. (c) Four representative frames of recovered video.



(a) (b)

Fig. 10. Comparison between original and corresponding recovered images. Average PSNR of recovered area is 35.73. (a) Original image. (b) Recovered image.

(a) (b)

Fig. 11. Comparison between original and corresponding recovered images. Average PSNR of recovered area is 30.37. (a) Original image. (b) Recovered image.

5 Conclusions

For privacy protection in surveillance videos, a data-hiding method using WebM video features has been proposed. A user can specify a protection region in an input WebM

video, and the privacy-sensitive contents in the region in all frames can be removed to protect personal privacy. The problem of yielding unacceptable decoding errors in the resulting video due to such content removals have been solved by assigning a background image as the golden reference frame of the WebM format for use by the video encoder. The removed contents can be embedded into the video frames imperceptibly by modifying the DCT frequency coefficients of the prediction frames of the chroma channels in the compression result, according to a set of predefined data patterns, to encode the removed data. The ROI map index of the WebM format is used to indicate frames where data have been embedded and should be extracted later for recovering the original video. The recovered videos still have good qualities as shown by experimental results. Future studies may be directed to applying the proposed data hiding techniques for other purposes like video authentication and covert communication, etc.

References

1. F. Dufaux, T. Ebrahimi, and S. A. Emitall, "Smart video Surveillance System Preserving Privacy," *Proc. of SPIE Image & Video Communication & Processing*, San Jose, CA, USA, vol. 5685, pp. 54-63, Jan. 2005.
2. P. Meuel, M. Chaumont, and W. Puech "Data Hiding in H. 264 Video for Lossless Reconstruction of Region of Interest," *Proc. of European Signal Processing Conf.*, Poznań, Poland, pp. 120-124, Sept. 2007.
3. W. Zhang, S.-C. S. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," *Proc. of IEEE Int'l Conf. on Image Processing,* Genova, Italy, vol. 3, pp. 868-871, Sept. 2005.
4. X. Yu, K. Chinomi, T. Koshimizu, N. Nitta, Y. Ito, and N. Babaguchi, "Privacy protecting visual processing for secure video surveillance," *Proc. of IEEE Int'l Conf. on Image Processing*, Los Alamitos, CA, USA, pp. 1672-1675, Oct. 2008.
5. Y. Hu, et al., "Information hiding based on intra prediction modes for H.264/AVC," *Proc. of IEEE Int'l Conf. on Multimedia and Expo*, Beijing, China, pp. 1231-1234, Jul., 2007.
6. Hussein A. Aly, "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error," *IEEE Trans. on Information Forensics & Security*, vol. 6, pp. 14-18, March, 2011.
7. M. Yang and N. Bourbakis, "A High Bitrate Information Hiding Algorithm for Digital Video Content under H.264/AVC Compression," *Proc. of IEEE Int'l Conf. on Image Processing Midwest Symp. on Circuits & Systems, Cincinnati*, OH, USA, vol. 2, pp. 935- 938, Aug., 2005.
8. S. K. Kapotas, et al., "Data hiding in H.264 encoded video sequences," *Proc. of Int'l Workshop on Multimedia Signal Processing*, Chania, Crete, Greece, pp. 373-376, Oct., 2007.
9. S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt (2001). *Vision and Video: Models and Applications*, Springer, USA, 2001.
10. K. John. (2010). *The WebM project*. [Online]. Available: http://www.webmproject.org/