A Robust Error Resilient Approach for Data Hiding in MPEG Video Files Using Multivariate Regression and Flexible Macroblock Ordering

Vinoth.M.M*, Mr.S.Rinesh**

*PG-Scholar,
** Assistant Professor,
Department of computer science and engineering,
Karpagam University, Coimbatore
*vinomm.27@gmail.com,
** rin.iimmba@gmail.com

Abstract—This paper proposes two data hiding approaches using compressed MPEG video. In the first approach, the quantization scale of a Constant Bit Rate (CBR) video is either incremented or decremented according to the underlying message bit. A second-order multivariate regression is used to associate macroblock-level features with the hidden message bit. The decoder makes use of this regression model to predict the message bits. However, the message payload is restricted to one bit per macroblock. The second approach of our work for both CBR and variable bit rate (VBR) coding and achieves a message payload of 3 bits per macroblock. The Flexible Macroblock Ordering (FMO) was used to allocate macroblocks to slice groups according to the content of the message. In existing network delivery of compressed video, packets may be lost if the channel is unreliable. Such losses tend to occur in burst. We can enhance our work to robustness of the existing work against packet losses in video steganalysis methods. We propose a robust error resilient approach for MPEG video transmission over internet. In this work, we develop an error resilient video encoding approach to help error concealment at the decoder. We introduce a new block shuffling scheme to isolate erroneous blocks caused by packet losses. And we apply data hiding to add additional protection for motion vectors. The existing solutions are superior in terms of message payload while causing less distortion and compression overhead and the proposed solution reduces the packet loss during transmission.

Index Terms - Data hiding, flexible macroblock ordering, MPEG coding, multivariate regression, steganography, Error spreading shuffling

I. INTRODUCTION

Data hiding techniques can be used to embed a secret message into a compressed video bit stream for copyright protection, access control, content annotation and transaction tracking. Such data hiding techniques can also be used for other purposes. For instance, [1] used data hiding techniques to assess the quality of compressed video in the absence of the original reference. The quality is estimated based on computing the degradations of the extracted hidden message. The authors of [2] used data hiding to enable real time scene change detection in compressed video. The information is hidden using the motion compensation block sizes of an H.264/AVC video. Data hiding is also used for error detection and concealment in applications of video transmission.

Examples of data hiding using DCT coefficients include the use the parity of the quantized coefficients to hide a message [3]. Additionally, [4] utilized zero-length codes to insert a dummy value at certain locations to indicate message bits.

Examples of using MVs for data hiding include, where phase angles of MVs are used to hide messages. The work in [5] and [6], on the other hand, proposed solutions for using the magnitude of MVs for data hiding. More specifically, [6] uses the least significant bit of both components of candidate motion vectors to embed a secret message. The candidate motion vectors are selected based on the prediction error of the underlying macroblock. MVs associated with high prediction errors are chosen. A prediction error threshold is computed per frame and transmitted in the video bit stream to guide the decoder in recognizing the MVs that carry bits of the secret message.

The quantization scale is also used for data hiding; a recent publication in [7] proposed to divide the quantization scale of a macroblock by a certain factor. The factor is multiplied by all ac coefficients in the corresponding macroblock. The procedure is referred to as promoting and exiting a macroblock. If a message bit to hide is equal to zero, then such a procedure is followed, otherwise no action is taken.

From a syntax viewpoint, since a relatively large number of prediction modes and block sizes are available in H.264/AVC, it has been proposed to use these variants to hide message bits. Likewise, the work in [8] proposed the use of intra prediction modes to hide

message bits. It was shown that 1 bit can be hidden in each candidate 4X4 intrablock. Additionally, the work in utilized the block types and modes of intracoded blocks of H.264/AVC to hide message bits.

Data hiding can also be applied prior to compression. For example, [9] introduced a method that is robust to heavy JPEG compression. It is also possible to hide data in the wavelet domain as reported in [10]. In such an approach, significant wavelet coefficients are identified and used for embedding a message payload. Lastly, hiding of data can also be applied in the compressed domain. For example, the work in [11] proposed hiding messages in the compressed H.264/AVC I-frames without the introduction of drift distortion.

Steganalysis, on the other hand, is the process of detecting the presence of hidden messages in multimedia. Steganalysis can be applied to digital images and to digital video as reported in [12] and [13], respectively. Existing work on video-based steganography takes such analysis into account and tries to maintain the statistics of carrier before and after message hiding. For example, the work in proposed a sub histogram preserving approach for quantization modulation using matrix encoding.

This paper is organized as follows. Section II introduces message hiding using quantization scale modulation and multivariate regression.

II. MESSAGE HIDING USING QUANTIZATION SCALE MODULATION

To hide a message using quantization scale modulation, the message is first converted into a binary stream of bits. During the MPEG encoding of individual macroblocks, the message bits are read one at a time. For each coded macroblock, the quantization scale is either incremented or decremented based on the corresponding message bit. Clearly, if the original quantization scale was either the lowest or largest allowable values then no modification is applied. This simple process of hiding a message bit in a macroblock is illustrated. Although the message hiding procedure is straightforward, nonetheless, the question that remains is how to extract the message from the bitstream. This problem can be solved by extracting macroblock-level feature variables during the encoding process. Once the whole message is hidden, we end up with a feature matrix and a message vector. We will then treat the feature matrix as predictors and the message bits as a response variable and use multivariate regression to compute a prediction model. Once computed, the prediction model can be used to predict the message bit hidden in a given macroblock based on its feature variables. We elaborate on the extraction of macroblock features from an MPEG-2 video, consequently, we formulate the message extraction as a regression problem.

In the existing system the two novel solutions for data hiding are obtained. In the first solution, the message bits are hidden by modifying the quantization scale of MPEG video coded with constant bit rates. Features are extracted from individual macroblocks and a second-order regression model is computed. The decoder uses the regression model to predict the content of the hidden message based on macroblock-level feature variables. In the second solution, both constant and variable bit rate coding are supported. The solution utilizes the flexible macroblock ordering (FMO) feature of H.264/AVC video for message hiding and extraction. It is shown that both solutions can hide messages at an average payload of around 10 and 30 kb/s, respectively. Therefore, the applications of such solutions are not restricted to copyright protection where few bits are hidden per frame. Rather, the existing solutions can be used for other applications such as content annotation, transaction tracking, error detection and error concealment.

A. Macroblock Level Features Variables

The following feature variables are extracted or computed from a MPEG-2 video stream for each coded macroblock.

- The first feature is the virtual buffer discrepancy from uniform distribution model. This discrepancy is computed

$$d_j^t = d_0^t + B_{j-1} - \left(\frac{T_{t*}(j-1)}{\#MBs}\right)$$

Where the subscript j indicates a macroblock index, #MB sindicates the total number of macroblocks in a video frame and indicates the frame type; I, P, or B.

- The second feature is the spatial activity of the underlying macroblock. This activity is computed from the four original (i.e., noncoded) luminance blocks of the current macroblock. It is computed using

$$act_j = 1 + min(v_{b1}, v_{b2}, v_{b3}, v_{b4})$$

Where the subscript indicates a macroblock index. The variables $v_{b1}, v_{b2}, v_{b3}, v_{b4}$ indicate the spatial variance of each luminance block in a frame-based coding.

- The third feature is the actual quantization scale of the current macroblock. This scale is available from the macroblock header in the video bit stream.

B. Message Prediction

The message prediction problem is formulated using a second-order multivariate regression. The response variable in this case is the message binary bits denoted by the vector. As mentioned previously, each macroblock has three feature variables, consequently, the predictors or the feature vectors of macroblocks are arranged into one matrix which is referred to as the feature matrix. This matrix is denoted by X as shown in

$$X = \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ \vdots & \vdots & \vdots \\ a_{1,n} & a_{2,n} & a_{3,n} \end{bmatrix}$$

The subscripts of the matrix elements $a_{j,i}(j = 1 \dots 3, i = 1 \dots n)$ indicate the index of feature variables and the number of macroblocks, respectively.

C. Message Extraction

To extract the hidden message from a coded video, the feature variables of each macroblock are computed and/or extracted from the bitstream. The feature vectors are consequently arranged into a feature matrix and expanded to the second order, resulting in matrix P. The feature matrix is multiplied by the model weights α^{opt} to generate the predicted hidden message $\hat{m}$ as follows:

$$\hat{m} = P * \alpha^{opt}$$

The process of message hiding and prediction is summarized. Notice that the feature extraction and polynomial expansion steps are repeated at both stages of message hiding and prediction. As such, the feature vector need not be transmitted with the bit stream.

III. MESSAGE HIDING USING FLEXIBLE MACROBLOCK ORDERING(FMO)

One of the limitations of the quantization scale modulation solution of the previous section is related to the message payload where only one message bit can be hidden per macroblock. This section introduces a second solution that benefits from a higher message bitrate through the use of FMO of the H.264/AVC video coding standard.

In this work we make use of the explicit assignment of macroblocks to slice groups to hide messages in the video stream. Since macroblocks can be arbitrary assigned to slice groups, we propose to use the slice group ID of individual macroblocks as an indication of message bits. Assume for instance that two slice groups are used, the allocation of a macroblock to slice group 0 indicates a message bit of 0 and the allocation of macroblock to slice group 1 indicates a message bit of 1. Hence, one message bit per macroblock can be carried. Furthermore, since the H.264/AVC standard allows for a maximum of eight slice groups per picture then two or three message bits can be carried per macroblock as elaborated in Table.

TABLE I BLOCK SIZE

Number of slice groups	Potential message bits/MB	Message bits/MB
2	0,1	1
4	00,01,10,11	2
8	000,001,010,011, 100,101,110,111	3

In general, to hide a message into the H.264/AVC bit stream, the message is first read into chunks of bits, where is 1, 2, or 3 according to the values in Table. If macroblocks are coded per picture, then message bits can be used to allocate the macroblocks to slice groups. The process of message hiding is illustrated in above figure.

To hide a message into the H.264/AVC bit stream, the message is first read into chunks of bits, where is 1, 2, or 3 according to the values in Table. If m macroblocks are coded per picture,

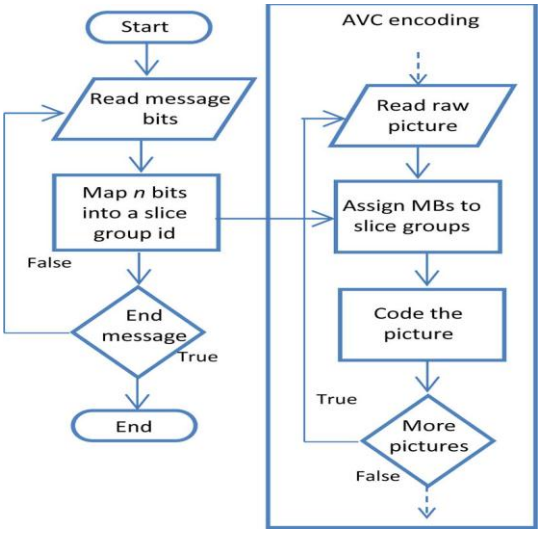


Figure 1. The process of message hiding.

Then m×n message bits can be used to allocate the macroblocks to slice groups. The process of message hiding is illustrated in above figure.

Message extraction

To extract the message bits, each time a picture is decoded, the macroblock to slice group mapping syntax structure is used to read message bits and append them to the extracted message. The process of message extracting is illustrated.

Block Shuffling in source coding can be analogue to bit (block) interleaving in channel coding, which aims to break burst error into random bit errors. The difference lies at decoder: source decoder makes use of the remaining natural. Several shuffling patterns redundancy to recover random block error, while channel decoder uses explicit Forward Error Correction (FEC) code to recover the random bit error. Because the structure of MPEG video usually leads bit error to block error, and there is always some redundancy left in the coded video, source block shuffling can be more effective than channel interleaving in dealing with burst error.

The new shuffling pattern proposed in this paper disperses the errors over a wide area of the image, thus lowering the probability that the connected important blocks are lost at the same time. We shall call this shuffling pattern \ Error Spreading Shuffling (ESS)", which can more spread errors and minimize the overhead in compression. The basic idea is that each transmitted packet will include blocks running through all the columns and rows of blocks, and that every column and row of blocks will have the same probability to be lost. One such pattern is shown in Figure.

1	8	3	5	2	7	4	6
6	1	8	3	5	2	7	4
4	6	1	8	3	5	2	7
7	4	6	1	8	3	5	2

Figure 1. Error spreading shuffling (ESS)

This pattern can be generated as follows. Suppose the image has N rows and M columns of Blocks/macroblocks, and assume M > N. We label the (i; i) blocks as \1" starting with the top left corner. After the N \1" blocks, we label as \2" all the (i, i + N) blocks, starting from the (1; N+1) block. When the labeling reaches the right extreme of the image, it wraps around to the left side. When the label \k" reaches the bottom row, say the (N; j) block, a new label \k +1" begins with the (1; j +1) block.

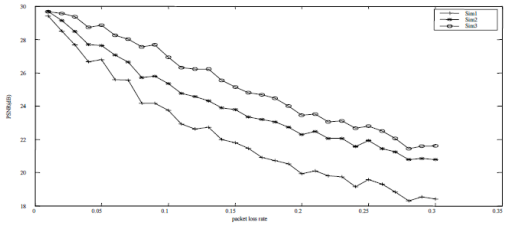
If the (1; j +1) block has already labeled then \k + 1" is assigned to the (1; j0) block, where j0 is the median of j + 1 and the column index of its greater nearest labeled neighbor. If it also has been labeled, then assign \k + 1" to the (1; j"), where j00 is the median of j + 2 and the column index of its greater nearest labeled neighbor. Continue with this manner until all the blocks have been labeled. This kind of shuffling can isolate error blocks under packet loss. We have noticed that fixed shuffling pattern can cause artifacts in fixed pattern, which can be annoyed for long video streams. In practice, mixing shuffling pattern of different direction can be employed to reduce the artifacts.

First, it is observed that the bit rate increases for Sim2 over Sim1 by 0.35% and for Sim3 by 1.02%. It is thus clear that our algorithm decreases the compression ratio negligibly.

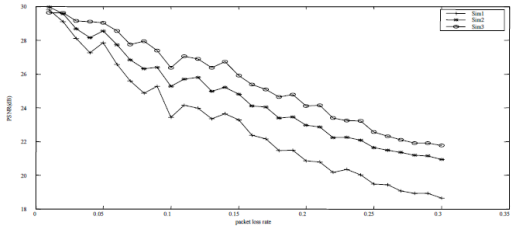
Second, we show in Figure 5 the average PSNR as a function of Pl from simulations using 5 different channel conditions. We simulate a random loss case and a burst case where average burst length is 2, which is typical for the internet transmission.

IV. CONCLUSION

In this paper, can enhance our work to robustness of the existing work against packet losses in video steganalysis methods. A robust error resilient approach for MPEG video transmission over internet. In this work, we develop an error resilient video encoding approach to help error concealment at the decoder. It introduce a new block shuffling scheme to isolate erroneous blocks caused by packet losses. And we apply data hiding to add additional protection for motion vectors. The existing solutions are superior in terms of message payload while causing less distortion and compression overhead and the proposed solution reduces the packet loss during transmission. In this approach can achieve a reasonable quality for packet loss efficiently. In proposed takes use of the structure and syntax of the video, and it is useful in combating packet losses.

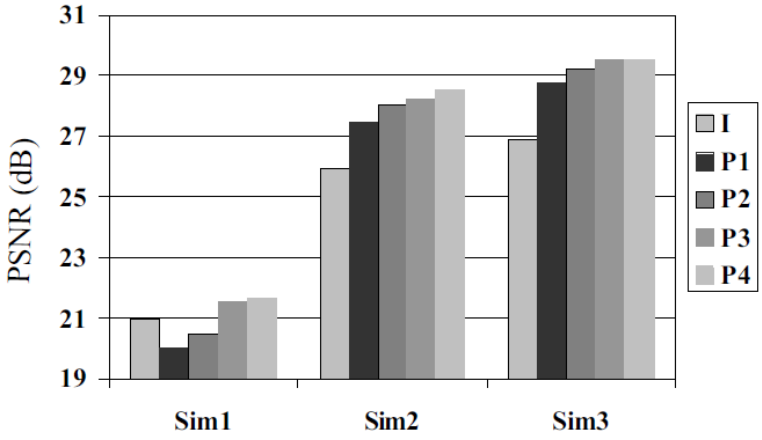


(a) random loss



(b) $L_b = 2$

Figure 1. Average PSNR of five channel realizations using 3 different methods (Sim1, Sim2, Sim3)



(c)PSNR of Concealed anchor I frame and following P frames

Figure 2. Comparison of various methods. For (a)and (b): upper left - Original frame; upper right - Sim1; lower left - Sim2; lower right - Sim3

REFERENCES

[1] Peng Yin, Min Wu and Bede Liu. "Robust error-resilient approach for MPEG video transmission over Internet", *Proc. SPIE 4671, Visual Communications and Image Processing* 2002, 103 (January 7, 2002); doi:10.1117/12.453035.

[2] M. Carli, M. Farais, E. D. Gelasca, R. Tedesco, and A. Neri, "Quality assessment using data hiding on perceptually important areas," in *Proc .IEEE Int. Conf. Image Processing, ICIP*, Sep. 2005, pp. III-1200-3–III-1200-3.

[3] S. Kapotas and A. Skodras, "A new data hiding scheme for scenechange detection in H.264 encoded video sequences," in *Proc. IEEEInt. Conf. Multimedia Expo ICME*, Jun. 2008, pp. 277–280.

[4] Y. Li, H.-X. Chen, and Y. Zhao, "A new method of data hiding basedon H.264 encoded video sequences," in *Proc. IEEE Int. Conf. Signal Processing, ICSP*, Oct. 2010, pp. 1833–1836.

[5] K. Nakajima, K. Tanaka, T. Matsuoka, and Y. Nakajima, "Rewritabledata embedding on MPEG coded data domain," in *Proc. IEEE Int.Conf. Multimedia and Expo, ICME,* Jul. 2005, pp. 682–685.

[6] C. Xu, X. Ping, and T. Zhang, "Steganography in compressed videostream," in *Proc. Int. Conf. Innovative Computing, Information and Control, ICICIC'06*, 2006, vol. II, pp. 803–806.

[7] H. A. Aly, "Data hiding in motion vectors of compressed video basedon their associated prediction error," *IEEE Trans. Inform. Forensics Security*, vol. 6, no. 1, pp. 14–18, Mar. 2011.

[8] K. Wong, K. Tanaka, K. Takagi, and Y.Nakajima, "Complete video quality-preserving data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, Oct. 2009.

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

[10] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I.El-Khalil, "'Print and Scan' resilient data hiding in images," *IEEE Trans. Inform. Forensics Security*, vol. 1, no. 4, pp. 464–478, Dec.2006.

[11] X.-P. Zhang, K. Li, and X. Wang, "A novel look-up table design method for data hiding with reduced distortion," *IEEE Trans. CircuitsSyst. Video Technol.*, vol. 8, no. 6, pp. 769–776, Jun. 2008.

[12] M. Xiaojing, L. Zhitang, T. Hao, and Z. Bochao, "A data hiding algorithmfor H.264/AVC video streams without intra-frame distortion drift," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp.1320–1330, Oct. 2010.

[13] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inform. Forensics Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[14] U. Budhia, D. Kundur, and T. Zourntos, "Digital video steganalysis exploiting statistical visibility in the temporal domain," *IEEE Trans. Inform. Forensics Security*, vol. 1, no. 4, pp. 502–516, Dec. 2006.

[15] Mohamed Elsadig Eltahir, Laiha Mat Kiah, B.B.Zaidan and A.A.Zaidan," High Rate Video Streaming Steganography", *International Conference on Information Management and Engineering (ICIME09),* Session 10, P.P 550-553, 2009, Kuala Lumpur, Malaysia.

AUTHORS

First Author – Vinoth.M.M , PG-Scholar, Karpagam University, Coimbatore,vinomm.27@gmail.com.
Second Author – S.Rinesh M.E., (Ph.D)., MBA., Assistant Professor, Karpagam University, Coimbatore, rin.iimmba@gmail.com.