World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:6, 2011

Reversible Watermarking for H.264/AVC Videos

Yih-Chuan Lin and Jung-Hong Li

Abstract**—**In this paper, we propose a reversible watermarking scheme based on histogram shifting (HS) to embed watermark bits into the H.264/AVC standard videos by modifying the last nonzero level in the context adaptive variable length coding (CAVLC) domain. The proposed method collects all of the last nonzero coefficients (or called last level coefficient) of 4×4 sub-macro blocks in a macro block and utilizes predictions for the current last level from the neighbor block's last levels to embed watermark bits. The feature of the proposed method is low computational and has the ability of reversible recovery. The experimental results have demonstrated that our proposed scheme has acceptable degradation on video quality and output bit-rate for most test videos.

Keywords**—**Reversible data hiding, H.264/AVC standard, CAVLC, Histogram shifting

I. Introduction

DATA hiding is a signal process for multimedia to add some secret message into the context of multimedia file in a way without causing suspicion for the presence or content of the hidden message in perception. It is widely acceptable nowadays for applications on private conversation, data integrity detection and authentication of intellectual property.

Recently, many researchers concentrate on developing data hiding schemes in video compression domain, such as H.264/AVC standard [1] - [5] and [8]-[9], in view of its better compression performance. However, on the other side, the sophisticated encoding mechanisms in H.264/AVC encoder also make several restrictions for hiding data in the coded video sequences. From our observations, the restrictions include:

i. The hiding of a little data into the coded video sequence would result in a great deal of degradation on both the bit-rate and the reproduced video quality.
ii. It is a challenge task to make a cost effective trade-off between the bit-rate and video quality degradation when embedding your secret data into the coded bit-stream.
iii. In order to raise hiding capacity, we would sacrifice significantly bit-rate or video quality [4]. But this would obviously destroy the original application settings on coded bit-rate and reproduced quality made in the encoder.

Based upon these restrictions, the video watermarking for compression domain should be careful on the degradation of performance. Therefore, the reversible video watermarking method in compression domain is scarce than static image watermarking method due to design hardly and restriction mostly, too. However, there had two methods [8]-[9] which had applied the traditional reversible methods to H.264/AVC directly. In Lie et al [8], they utilize the difference expansion (DE) [6] to hide data into quantized discrete cosine transform (QDCT) coefficients for error concealment. The performances are capable of achieving peak signal-to-noise ratio (PSNR) improvement of up to 1.48dB, at a considerable bit-rate, when the packet loss rate is 20%. Chung et al [9] presented an efficient algorithm which utilizes the HS [7] through QDCT coefficients to embed motion vector (MV) for error concealment, but according to the HS rules, the algorithm should find the peak point and zero point to shift and embed data from embedding target. According to the CAVLC rule [10], when we want to move a large number of coefficients, the degradation of bit-rate or PSNR may increase seriously and the coded length is increasing due to the number of the coeff_tokens and run_befores increasing. For other examples in [2] and [3], their proposed methods need to change the number of coefficients, although the bit-rate is decreasing seriously, the degradation of video quality is raised. Therefore, we can know that changing the number of each CAVCL symbols may affect the distortion and coding length.

Although the degradation of Lie and Chung's methods after embedding is not discussed in their paper, it is well know which the QDCT coefficients have modified seriously by utilize DS and HS. If we can redesign or change the embedding target, the degradation of performance may reduce after hiding data into video. Thus, we propose a reversible watermarking method based on HS, and utilizing all of the last level for each 4×4 DCT block instead of all nonzero coefficients do the shifting in order to avoid modifying seriously. By The difference value actual last level ℓ_c and predicted last level $\hat{\ell}_c$. If the $\hat{\ell}_c$ is same with ℓ_c, we embed data into ℓ_c. Otherwise, we don't do anything. The $\hat{\ell}_c$ is from the previous coded DCT block, so it is easy to implement. The experimental results have demonstrated our proposed method can achieve reversible.

The remainder of this paper is organized as follows. Section 2 explains our proposed scheme in details, including the data embedding/extracting schemes and their pseudo-codes. In Section 3, the performance of our proposed scheme is presented. Finally, some conclusions are given in Section 4.

II. The Proposed Scheme

A. Overview of our method

The block diagram of our proposed method which includes encoder and decoder and embedding target are same with Lin [1]. The watermark embedding method is inserted into the H.264/AVC framework during the encoding process. Data is hided in DCT blocks before entropy coding. Our method takes rate distortion optimization (RDO) into account in order to get

Yih-Chuan Lin is with the National Formosa University, Yunlin, 63201 Taiwan (phone: +886-5-631-5576; fax: 886-5-633-0456; e-mail: lyc@ nfu.edu.tw).
Jung-Hong Li is with the National Formosa University, Yunlin, 63201 Taiwan (phone: +886-5-631-5596; fax: 886-5-633-0456; e-mail: jhl0258@gmail.com).

better coding performance while embedding the data into the video sequence [1], [4]. The data hiding method in our proposed method is done on luminance DCT blocks in both intra and inter modes for AC component, not considering for skip and IPCM (intra-frame pulse code modulation) block types. The DCT blocks of chrominance are not considered in our method. The extracting phase can be done in DCT blocks after entropy decoding and we need only to do extract on the luminance part of DCT blocks.

B. Embedding algorithm

In this sub-section, we will show the pseudo code for the embedding method and explain them in details. At firstly, we define the important parameter: predication error e, that means the difference of ℓ_c and $\hat{\ell}_c$, and is shown as (1)

$$predication\ error\ \ e = \left|\ell_c - \hat{\ell}_c\right| \quad (1)$$

Fig.1 illustrates that the relationship of $\hat{\ell}_c$ and ℓ_c, the position (0, 0) of ℓ_c is the $\hat{\ell}_c$ to the position of (0, 1) or (1, 0) firstly. The position (1, 1) is the average value of the position of (0, 1) and (1, 0), and the horizontal edge block is utilizing the left block of ℓ_c be the $\hat{\ell}_c$, the vertical edge block is utilizing the up block. According to the HS method, when the difference value of $\hat{\ell}_c$ and ℓ_c is zero, we as it are the peak point to hide data. Otherwise, we shift the value by adding one or subtracting one.

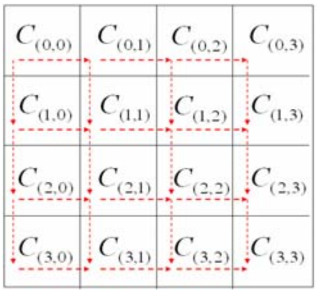


Fig.1 prediction pattern for the last level coefficients

The data is embedding into the difference which equals zero, and we can extract the data if the difference is equals zero or one in decoder. The embedding formula shows as (2), and w is watermark bit.

$$\ell_c' = \begin{cases} \ell_c + w, \ if\ e = 0\ and\ \ell_c > 0 \\ \ell_c - w, \ if\ e = 0\ and\ \ell_c < 0, where\ sign(x)\begin{cases} +1,\ if\ x > 0 \\ -1,\ if\ x < 0 \end{cases} \\ \ell_c + sign(\ell_c) \end{cases} \quad (2)$$

In Table I, we define the notations and the functions to be used in the pseudo codes. These functions often refer to the DCT block or the set of last levels to get the information of DCT block or coefficients.

One macro blocks have 16 4×4 DCT block, we will utilize each last level of 4×4 DCT block for embedding except the position (0, 0) of DCT block due to it doesn't have reference block to get predication last level. Therefore, the number of DCT blocks will be possess by embedding procedure is 15. At embedding procedure firstly, the process order of DCT block is left to right and top to bottom. If the ℓ_c is not equal to $\hat{\ell}_c$, we only shift the ℓ_c. Otherwise; we embed data into ℓ_c. The predication error e can be as peak point of HS to embed data, so most of last level which not satisfy the embedding condition should be shifted. After embedding procedure, the new last level of coded block will replaced original last level to provide later DCT block using. The detail of embedding algorithm can see the pseudo code.

TABLE I
THE SYMBOL AND FUNCTION EXPLANATION

Notations or Functions	Definition
MB	Macro block
MB'	Macro block after embedding
DCTB	A 4x4 DCT block
DCTB'	A 4x4 DCT block after embedding
W	Watermark bit, W = {0,1}
lastLevel	The last level in current block
e	Predication error
lastLevelTable	Record all last level of each 4x4 DCT block in current macro block
getPrelastLevel(DCTB, lastLevelTable)	Get the predication last level of current DCTB
getLastLevel(DCTB)	Get the last nonzero level from DCTB
setShifting(DCTB)	Shifting the last level from DCTB
setShiftBack(DCTB')	Shifting back the last level from DCTB
getDCTB(MB)	Get the 4x4 DCT block from the MB
getWatermarking(lastLevel)	Retrieve the embedded data
abs(Value)	Getting the absolute value for argument parameter value

THE PSEUDO CODE FOR EMBEDDING ALGORITHM

```
Embedding Algorithm
 Input: MB
 Output: MB'
Begin Embedding()
   if(getDCTB(MB) != NULL)
      PrelastLevel = getPrelastLevel(DCTB, lastLevelTable)
      lastLevel     = getLastLevel(DCTB)
      e = abs|lastLevel – prelastLevel|
      if(e == 0)
      {
         If(lastLevel>0)
         {
           lastLevel'  = lastLevel+w
         }
         else if(lastLevel<0)
         {
           lastLevel'  = lastLevel-w
         }
         end if
      }
      else if
      {
         setShifting(DCTB)
      }
      end if
   end if
   output MB';
 end Begin
```

C. Extracting algorithm

The pseudo code of extracting as shown in below is easier than the embedding phase. In one macro block for each DCT block, the extraction algorithm find the last level of DCT block and utilize it to comparison with other block. If the *e* is zero or one, this block represent that have embedded data and we can retrieve the data and restore. Otherwise, we only need to shift back to restore. Thus, we can achieve the object of reversible.

THE PSEUDO CODE FOR EXTRACTING ALGORITHM

```
Extracting Algorithm
Input: MB'
Output: MB, W
Begin Extracting()
 if(getDCTB(MB') != NULL)
     PrelastLevel' = getPrelastLevel(DCTB', lastLevelTable)
     lastLevel'    = getLastLevel(DCTB')
     e = abs|lastLevel' – prelastLevel'|
     if(e == 0)
     {
        W = 0
       output W
     }
     else if(e == 1)
     {
       lastLevel = lastLevel' -1
       W = 1
       output W
     }
     else
     {
       setShiftBack(DCTB')
     }
     end if
 end if
 output MB
end Begin
```

III. EXPERIMENTAL RESULTS

A. The experiment environment

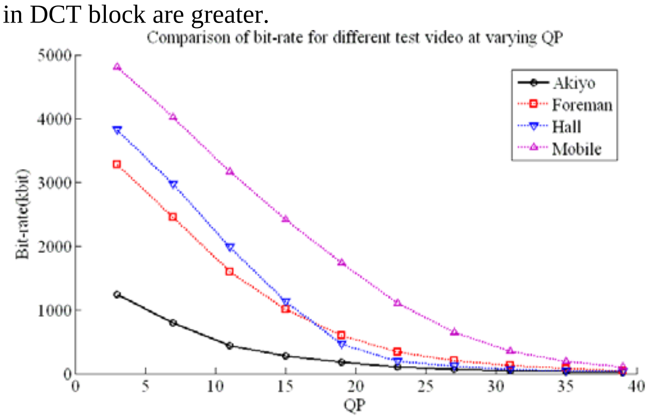
TABLE II
THE EXPERIMENTAL PARAMETERS FOR H.264/AVC CODEC.

Parameter	Information
Profile IDC	66(baseline)
Intra period	15(I-P-P-P)
Frames to be encoded	300
Motion Estimation scheme	Fast Full Search
Rate Control	Disable
Parameter	Information

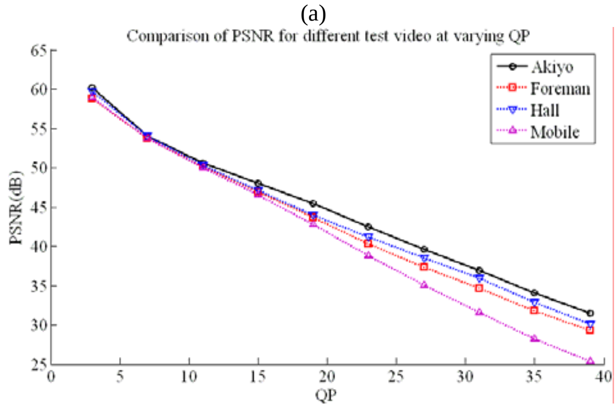
We adopt the H.264/AVC JM Reference software [11] as the platform to simulate our proposed method and comparison method. This sub-section presents the experiment parameters for our method in JM reference software. The version of JM 12.2 software is executed with the related environmental parameters shown in Table IV. In the experiments, four videos: "foreman", "akiyo", "hall" and "mobile" are used as test data set, and their format information and frame rate is QCIF and 30 fps. The secret data to be hided into the test videos is a bit stream which all 1.

B. The experiment results

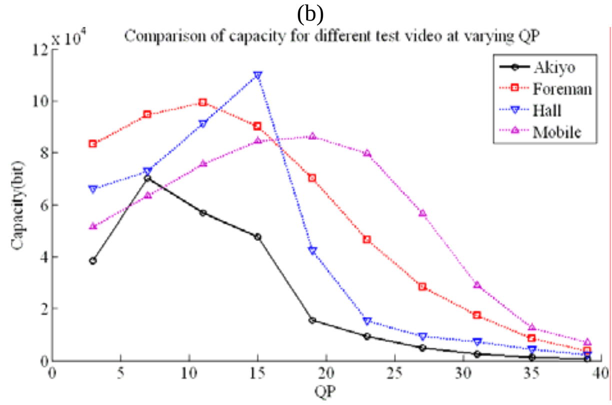
We demonstrate the experimental results and make an explanation in this sub-section. Fig.2 (a)-(c) illustrate that the performance of bit-rate, video quality and capacity, the degradation of PSNR is up to 2.5 dB and 0.5 dB in low and high QP and the degradation of bit-rate is up to 80 kbits and 20 kbits in low and high QP. Although proposed method can't provide good efficiency in video and bit-rate, our aim is to develop a reversible watermarking algorithm in H.264/AVC. Thus, the performance is minor important. The quantity of capacity is increasing with low QP increasing until medium QP, the predication error often is large at low QP due to the coefficients in DCT block are greater.



(a)



(b)



(c)

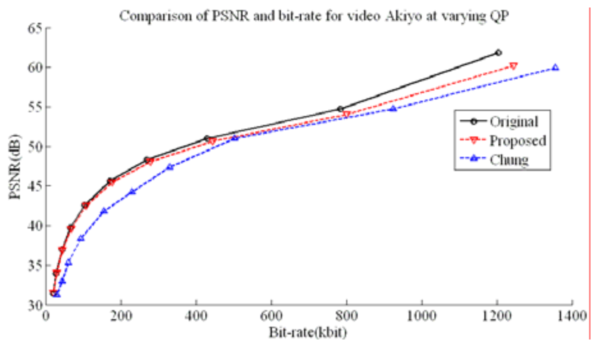
Fig. 2 Comparison of performance for different test video at varying QP. (a) Bit-rate, (b) PSNR, (c) Capacity

(a) Compressed (b) Watermarked

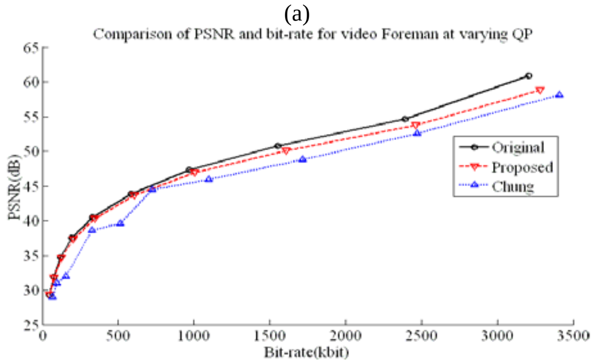
Fig.3 Comparison of video quality for akiyo (a) Compressed, PSNR=45.07 dB; (b) Watermarked, PSNR=44.729 dB

Fig.3 (a)-(b) illustrate the original video, compressed video, marked compressed video by our method and recovery video in 10th P frame of akiyo with medium QP 19. Because of H.264 Compression performance, the compressed video can keep high quality and low bit-rate. Comparison of quality between compressed and embedded, we can observe that the quality after embedding data is similar to compressed, and the PSNR is near.

C. Comparison with other method



(a)



(b)

Fig.4 Comparison in terms of PSNR vs bit-rate at varying QP. (a) akiyo; (b) foreman

We select Chung' method [9] which is same with our method is based on HS algorithm to comparison the performance. Their method is need high capacity to embed MV, so the feature of HS is just suitable for them to application in error concealment domain. To reduce quality degradation caused by embedding into DCT coefficients, the first three coefficients which included DC component are not considered in the embedding process. The remaining twelve coefficients still occupies a lot of high-energy, and utilizing them to shift for embedding data. This will still cause considerable distortion than general watermark methods.

We take akiyo and foreman as experimental target to compare with Chung. Fig.4 (a)-(b) illustrate that performance between original, our proposed method and Chung. We can found that our performance of video quality or bit-rate is greater than Chung, because our proposed method only modified the last level for each DCT block, the Chung's method shift the considerable quantities of coefficients to embedding data and it will caused more degradation in performance.

IV. CONCLUSIONS

In this paper, we propose a reversible data hiding algorithm that has considered the rate distortion performance for H.264/AVC standard. Although the method can not provide good efficiency in video quality and bit-rate, it is able to reach the reversible. At the same time, we understand that applying tradition reversible method to video compression domain directly is not suitable by comparison with Chung. In order to reduce the degradation of hiding modification, the information is hided in the last level symbols of CAVLC domain in H.264/AVC encoder.

REFERENCES

[1] Y. C. Lin and J. H. Li, "Data Hiding with Rate-Distortion Optimization on H.264AVC Video" , *in Pro. 23th IPPR Conf. on Computer Vision, Graphics, and Image Processing, CVGIP*. Aug. 15-17, 2010.

[2] S. D. Lin, C. Y. Chuang, and M.J. Chen, "A Watermarking Scheme by Changing the Number of Nonzero Coefficients in H.264/AVC Encoder", *in Pro. 22th IPPR Conference on Computer Vision, Graphics, and Image Processing, CVGIP*. Aug. 23-25, 2009.

[3] C. C. Wang, Y. C. Hsu, "A Fragile Watermarking Scheme for H.264/AVC Video Authentication," *Optical Engineering*, vol. 49, (2), 027003-1, Feb 2010.

[4] Z. Shahid, M. Chaumont and W. Puech, "Considering the Reconstruction Loop for Data Hiding of Intra and Inter Frames of H.264/AVC", *in Pro. European Signal Processing Conf.*, EUSIPCO 2009.

[5] X. Huang, Y. Abe, and I. Echizen, "Capacity Adaptive Synchronized Acoustic Steganography Scheme", *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 2, pp. 72-90, Apr, 2010.

[6] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp.890-896, 2003.

[7] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Trans. on Circuits and Systems Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.

[8] W. N. Lie, Tom C. I. Lin, D. C. Tsai, and G. S. Lin, "Error Resilient Coding Based on Reversible Data Embedding Technique for H.264/AVC Video," *in Proc. of IEEE International Conf. on Multimedia & Expo, ICME*, July 2005.

[9] K. L Chung, Y. H. Huang, P. C. Chang, and H. Y. M. Liao, "Reversible Data Hiding Based Approach for Intra Frame Error Concealment in H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 20, No. 11, pp. 1643-1647, November, 2010.

[10] Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression," John Willy & Sons Ltd, 2003.

[11] K. Sühring, H.264/AVC Reference Software Group [On-line]. Available: http://iphome.hhi.de/suehring/tml/, Joint Model 12.2 (JM12.2), Jan. 2009.