A Data Hiding Algorithm for H.264/AVC Video Streams Without Intra-Frame Distortion Drift

Xiaojing Ma, Zhitang Li, Hao Tu, and Bochao Zhang

***Abstract*—Intra-frame distortion drift is a big problem of data hiding in H.264/AVC video streams. Based on a thorough investigation of this problem, a novel readable data-hiding algorithm, which can embed data into the quantized discrete cosine transform (DCT) coefficients of I frames without bringing any intra-frame distortion drift into the H.264/advanced video coding (AVC) video host, is presented in this paper. We exploit several paired-coefficients of a 4 × 4 DCT block to accumulate the embedding induced distortion. The directions of intra-frame prediction are utilized to avert the distortion drift. It is proved analytically and shown experimentally that the proposed algorithm can achieve high embedding capacity and low visual distortion. Performance comparisons with other existing schemes are provided to demonstrate the superiority of the proposed scheme.**

***Index Terms*—Data hiding, H.264/advanced video coding (AVC), intra-frame distortion drift.**

I. Introduction

THE BOOMING of powerful sharing/transmission tools of digital video contents has created an urgent need to explore the algorithms of hiding data into the digital videos. H.264/advanced video coding (AVC) is the latest standard for video compression with high compression efficiency; furthermore, it is well adapted for network transmission. H.264/AVC is poised to replace the existing video coding standards [1].

The existing H.264 video data hiding schemes can be classified into two types according to their extraction approaches: detectable algorithm, which inserts a code that can only be detected, and readable algorithm, which embeds a message that can be read. The algorithms of the former type can authenticate the existence of embedded content, such as the video watermarking algorithms described in [2] and [3], and the algorithms of the latter type can extract the embedded content [4]–[9]. Hiding information into the digital video streams can be applied for many purposes, such as copyright protection, covert communication, and error concealment. However, only the algorithms of the latter type can be used in covert communication and error concealment.

In [2], the authors employ a human visual model based on 4 × 4 block discrete cosine transform (DCT) to embed the watermarks into quantized alternating current (AC) DCT coefficients of the luminance residual blocks. In [3], the authors proposed a algorithm which embeds watermarks into the quantized direct current (DC) DCT coefficients of the luminance residual blocks; the watermarks is added by an intra-frame error drift compensation signal before being embedded. Besides, the embedding strength varies according to the texture-masking-based perceptual model. The algorithms presented in [2] and [3] detect the existence of the watermarks with the knowledge of the contents of the watermarks.

The authors in [4] embed watermark bit to the sign bit of the trailing ones in context adaptive variable length coding and it yields no bit-rate change. But it would suffer from the subjective visual quality degradation due to the error propagation of the intra-prediction used in H.264/AVC. In [5], the proposed method takes advantage of the different intra-prediction block sizes to hide data. The authors in [9] embed watermarks into the motion vectors. However, the algorithms presented in [4], [5], and [9] are all fragile to re-encoding and re-quantization attacks.

The method proposed in [6] embeds the robust watermark into DCT domain for copyright protection and the fragile watermark into motion vectors for authentication, respectively. The robust watermark is embedded into one of the quantized AC coefficients in high frequency along the diagonal positions. In [7], the authors embed 2-D 8 bit grayscale image watermarks in the compressed domain for copyright protection. After preprocessing, one bit of the bi-polar vector watermark is embedded into the sign of one middle-frequency coefficient in the diagonal position in a 4 × 4 DCT block. In [8], the quantized AC DCT coefficient with the embedded watermark within a macro-block is determined by a key that is generated by using features of the macro-block itself. The algorithms in [6]–[8] embed the information into the DCT coefficients of I frames, and are more robust than the algorithms in [4], [5], and [9]. However, the algorithms in [6]–[8] have not considered the intra-frame distortion drift, and get low embedding capacity for the high embedding induced distortion.

In conclusion, all the existing robust algorithms embed the information into the DCT coefficients of I frames, but they have not handled the intra-frame distortion drift except [3]. However, the algorithm in [3] is detectable and non-blind.

Manuscript received November 18, 2009; revised March 2, 2010, accepted April 25, 2010. Date of publication August 30, 2010; date of current version October 8, 2010. This work was supported in part by the National Natural Science Foundation of China, under Grant 60573120, and in part by the National High Technology Research and Development Program of China (863 Program), under Grant 2007AA01Z420. This paper was recommended by Associate Editor D. S. Turaga.

X. Ma, H. Tu, and B. Zhang are with the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China.

Z. Li is with the Network Center, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: leeying@mail.hust.edu.cn).

Digital Object Identifier 10.1109/TCSVT.2010.2070950

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 1. Predicted and reference pixels in 4 × 4 luma block.

Performance comparison between [3] and our algorithm is provided in Section V-B.

The goal of this paper is to present a readable data hiding algorithm that embed the information into the quantized coefficients of the 4 × 4 luminance DCT blocks in I frames. Based on the analysis of the relationship between the DCT coefficients and the distortion of the pixel values used in intra-frame prediction, several paired-coefficients are found. For each paired-coefficient, one is used for embedding and the other is adjusted to fix the distortion on few pixels of the 4×4 block. Then, the presented algorithm chooses the blocks for embedding according to the intra-frame prediction modes of their adjacent blocks to make sure that the distortion will not propagate to its neighboring blocks. The experimental results show that the proposed scheme can effectively eliminate (or totally avert) intra-frame distortion drift and get higher visual quality. Besides, the embedding capacity of our data hiding algorithm is much higher than others since we exploit more DCT coefficients of a 4 × 4 luminance block for embedding. In addition, the proposed system is fast and blind. In other words, the proposed algorithm can be applied to real-time applications, and the embedded message can be retrieved without knowledge of the original video host.

The structure of this paper is organized as follows. Section II introduces the intra-frame prediction of H.264/AVC. Section III introduces the intra-frame distortion drift and explains how to avoid it. The data hiding algorithm is presented in Section IV. Section V gives the experimental results followed by conclusions in Section VI.

II. Intra-Frame Prediction

Intra-frame prediction is used in H.264 to reduce the spatial redundancies of video sequences. It is a major departure from the previous coding standards. A prediction block is predicted based on previously encoded adjacent blocks [1]. As illustrated in Fig. 1, the samples $a - p$ of the prediction block are calculated based on the encoded samples $A - M$ using a prediction formula corresponding to the selected optimal prediction mode.

At H.264 encoder, a real block $B_{4\times4}$ is subtracted from its prediction block $P_{4\times4}$ to get the residual block $R_{4\times4}$. It is the residual block that undergoes the following "usual" process of transformation, quantization, and entropy coding in H.264 encoder. At the decoder, the residual block will be added to the prediction block to reconstruct the video contents.

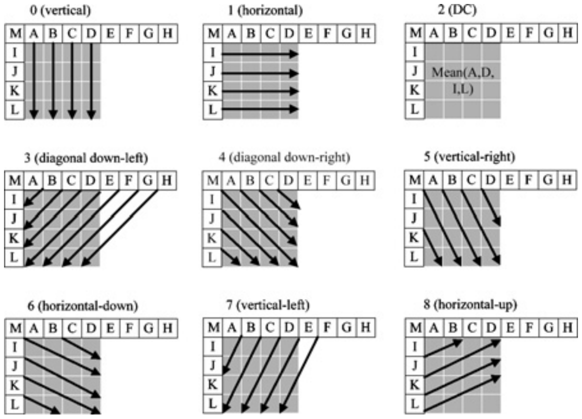


Fig. 2. 4 × 4 luma block prediction modes.

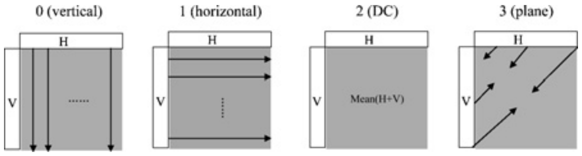


Fig. 3. 16 × 16 luma block prediction modes.

Intra-frame prediction algorithm used in H.264 provides nine optional prediction modes for each 4×4 luminance block, four modes for each 16 × 16 luminance block. Figs. 2 and 3 describe these prediction modes [1].

Each sample of current prediction block is calculated by the samples $A - M$.

III. Intra-Frame Distortion Drift Prevention

A. Embedding Procedure Analysis

The information is embedded into the quantized luminance DCT coefficients. The algorithm only embeds data into 4 × 4 luminance blocks, because the human eyes are less sensitive to the brightness and 16 × 16 luminance block often varies smoothly.

The integer discrete cosine transform (ICT) which is developed from the DCT is used in H.264/AVC standard. The ICT transform based on 4 × 4 blocks is shown in (1). $Y_{4\times4}$ is the matrix of unscaled DCT coefficients corresponding to the residual block $R_{4\times4}$. The "core" ICT transform between Y and R can be expressed in a matrix form (1) as follows:

$$Y = C_f R C_f^T \tag{1}$$

where

$$C_f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}.$$

Then, the post-scaling and quantization process is given in (2) as follows:

$$\widetilde{Y} = round[Y. \times PF./Qstep] \tag{2}$$

where $Qstep$ is the quantizer step size which is determined by QP (QP is the quantization parameter) and PF is the quantization matrix as follows:

$$qbits = 15 + floor(QP/6)$$

$$PF = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix} \quad a = \frac{1}{2} \; b = \sqrt{\frac{2}{5}}.$$

In this paper, $A = round[B]$ means $A_{ij} = round(B_{ij})$, where A and B are two matrices

$$Y' = \widetilde{Y}. \times Qstep. \times PF. \times 64 \tag{3}$$

$$\begin{aligned} R' &= round[(C_i^T Y' C_i)./64] \\ &= round[C_i^T(\widetilde{Y}. \times Qstep. \times PF)C_i] \end{aligned} \tag{4}$$

where

$$C_i^T = \begin{pmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{pmatrix}.$$

Then, the quantized coefficients undergo the following process in encoder such as entropy encoding.

At the decoder, after the re-scaling step as depicted in (3), the inverse ICT and the post-scaling step as described in (4), we can get the residual block R'.

In data hiding algorithms, the information is embedded into the quantized luminance DCT coefficients as in (5), where $\Delta_{4\times4}$ is the error matrix added to the quantized DCT coefficient matrix $\widetilde{Y}_{4\times4}$ by data hiding, $\Delta = (\alpha_{ij})_{4\times4}$, as follows:

$$\widetilde{Y}' = \widetilde{Y} + \Delta. \tag{5}$$

After the re-scaling step as depicted in (6), the inverse ICT and the post-scaling step as described in (7) of the decoder, we can get the residual block after embedding R'' as follows:

$$\begin{aligned} Y' &= \widetilde{Y}'. \times Qstep. \times PF. \times 64 \\ &= \widetilde{Y}. \times Qstep. \times PF. \times 64 \\ &\quad +\Delta. \times Qstep. \times PF. \times 64 \end{aligned} \tag{6}$$

$$\begin{aligned} R'' &= round[(C_i^T Y' C_i)./64] \\ &= round[C_i^T(\widetilde{Y}. \times Qstep. \times PF)C_i \\ &\quad +C_i^T(\Delta. \times Qstep. \times PF)C_i]. \end{aligned} \tag{7}$$

The deviation of the pixel luminance value between the original block and the one after embedding is $E_{4\times4}$, where $E_{4\times4} = (e_{ij})_{4\times4}$, which can be calculated according to (8) as follows:

$$\begin{aligned} E &= R'' - R' \\ &= round[C_i^T(\widetilde{Y}. \times Qstep. \times PF)C_i \\ &\quad +C_i^T(\Delta. \times Qstep. \times PF)C_i] \\ &\quad -round[C_i^T(\widetilde{Y}. \times Qstep. \times PF)C_i]. \end{aligned} \tag{8}$$

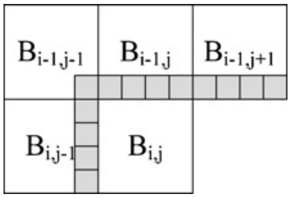


Fig. 4. Intra-frame prediction block $B_{i,j}$ and the adjacent encoded blocks.

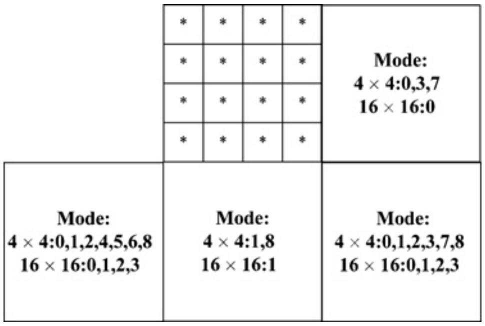


Fig. 5. Constraint conditions of the adjacent blocks of the chosen block.

Using $A_{4\times4} = (a_{ij})_{4\times4}$ to express $C_i^T(\widetilde{Y}. \times Qstep. \times PF)C_i$ and $B_{4\times4} = (b_{ij})_{4\times4}$ to express $C_i^T(\Delta. \times Qstep. \times PF)C_i$, we have $e_{ij} = round(b_{ij}) - 1$ or $round(b_{ij})$ or $round(b_{ij}) + 1$. Especially if $b_{ij} = 0$, then $e_{ij} = 0$.

B. Intra-Frame Distortion Drift

We focus on the intra-frame distortion drift since the embedded data is embedded only into I frames. As illustrated in Fig. 4, we assume that current prediction block is $B_{i,j}$. In intra-frame prediction, each sample of $B_{i,j}$ is the sum of the predicted value and the residual value. Since the predicted value is calculated by the samples that are gray in Fig. 4, the embedding induced deviation in blocks $B_{i-1,j-1}$, $B_{i,j-1}$, $B_{i-1,j}$, and $B_{i-1,j+1}$ would propagate to $B_{i,j}$.

In brief, the embedding induced distortion of a block (especially of the samples at the bottom line and on the right-most row) would propagate to other blocks by intra-frame prediction. This visual distortion would accumulate from the upper left to the lower right. This is defined as intra-frame distortion drift.

C. Drift Prevention

The 4×4 block on the right of the current block is defined as *right-block*. The 4×4 block under the current block is defined as *under-block*. The 4×4 block on the left of the *under-block* is defined as *under-left-block*. The 4×4 block on the right of the under-block is defined as *under-right-block*.

Condition 1 (Right-Mode$\in$ $(0, 3, 7)_{4\times4} \cup \{0\}_{16\times16}$*)*: The *right-block* has the prediction modes 0, 3, or 7 as shown in Fig. 2, or it is the sub-block of the 16×16 block with prediction mode 0 as shown in Fig. 3.

If a macro block meets *Condition 1*, the pixel values of its samples on the right-most column will not be used in the intra-frame prediction of its *right-block*, i.e., the embedding induced distortion will not propagate to its *right-block*.

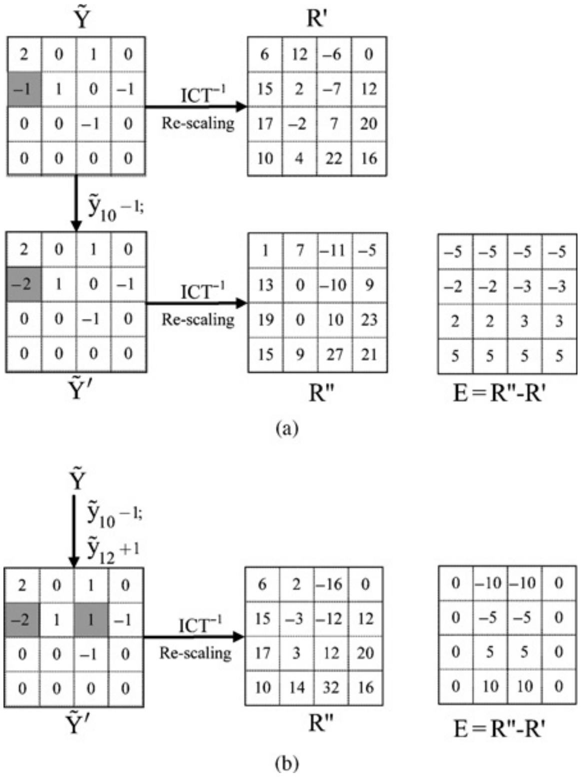


Fig. 6. Real 4 × 4 block from Akiyo. (a) $\tilde{Y}_{10} - 1$. (b) $\tilde{Y}_{10} - 1$, $\tilde{Y}_{12} + 1$.

Condition 2 (Under-Left-Mode$\in$ $\{0, 1, 2, 4, 5, 6, 8\}_{4\times4} \cup \{0, 1, 2, 3\}_{16\times16}$ *and Under-Mode*$\in$ $\{1, 8\}_{4\times4} \cup \{1\}_{16\times16}$*)*: The *under-left-block* has the prediction modes 0, 1, 2, 4, 5, 6, or 8, or the *under-left-block* is the sub-block of the 16 × 16 block. Simultaneously, the *under-block* has the prediction modes 1 or 8, or the *under-block* is the sub-block of the 16 × 16 block with prediction mode 1.

If a macro block meets *Condition 2*, the pixel values of its samples on the lowest row will not be used in the intra-frame prediction of its *under-block* and *under-left-block*, i.e., the embedding induced distortion will not propagate to its *under-block* and *under-left-block*.

Condition 3 (Under-Right-Mode$\in$ $\{0, 1, 2, 3, 7, 8\}_{4\times4} \cup \{0, 1, 2, 3\}_{16\times16}$*)*: The *under-right-block* has the prediction modes 0, 1, 2, 3, 7, or 8, or the *under-right-block* is the sub-block of the 16 × 16 block.

If a macro block meets *Condition 3*, the sample in the bottom right-hand corner will not be used in the intra-frame prediction of its *under-right-block*, i.e., the embedding induced distortion will not propagate to its *under-right-block*.

To prevent the distortion drift, the current block must meet *Condition 1*, *Condition 2*, and *Condition 3* as depicted in Fig. 5. Under such conditions, all the pixel values of current block will not be used in the following intra-frame prediction. Thus, the embedding distortion would not propagate to other blocks.

The number of such macro blocks in 14 test video sequences is provided in Fig. 9 of Section V. It can be concluded from the experimental results that there are few (less than 110 per I frame) macro blocks which meet all the three conditions.

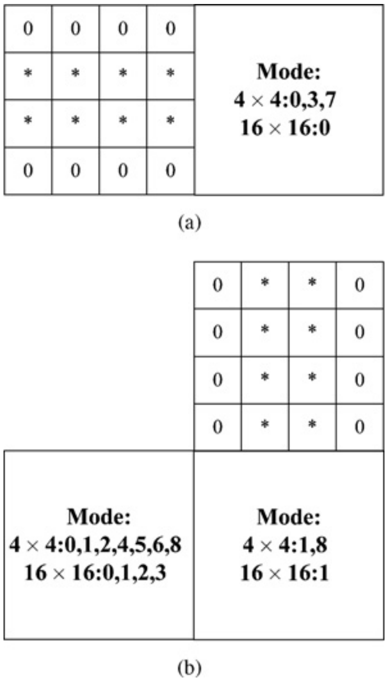


Fig. 7. Constraint conditions of the adjacent blocks of the chosen block. (a) Condition 1. (b) Condition 2.

D. Paired-Coefficients

A consensus in traditional algorithms for hiding data into images is "the less number of modification to the DCT coefficients, the less amount of distortion in the image" [10]. However, it is not correct for H.264 video streams because of the intra-frame distortion drift. We propose to hide data into the paired-coefficients.

The paired-coefficients can be defined as two quantized DCT coefficients (C_1, C_2), of which data is embedded into C_1, and C_2 is changed accordingly to make the deviation accumulated to the two columns or two rows in the middle of the matrix E. There are 12 paired-coefficients in H.264/AVC as follows:

$$(\tilde{Y}_{20}, \tilde{Y}_{22}), (\tilde{Y}_{02}, \tilde{Y}_{22}), (\tilde{Y}_{01}, \tilde{Y}_{21}), (\tilde{Y}_{10}, \tilde{Y}_{12})$$
$$(\tilde{Y}_{03}, \tilde{Y}_{23}), (\tilde{Y}_{30}, \tilde{Y}_{32}), (\tilde{Y}_{22}, \tilde{Y}_{20}), (\tilde{Y}_{22}, \tilde{Y}_{02})$$
$$(\tilde{Y}_{21}, \tilde{Y}_{01}), (\tilde{Y}_{12}, \tilde{Y}_{10}), (\tilde{Y}_{23}, \tilde{Y}_{03}), (\tilde{Y}_{32}, \tilde{Y}_{30})$$

$$\begin{aligned} \textit{Horizontal set } (HS) \quad &= \{(\tilde{Y}_{22}, \tilde{Y}_{20}), (\tilde{Y}_{02}, \tilde{Y}_{22}), (\tilde{Y}_{03}, \tilde{Y}_{23}) \\ &\quad (\tilde{Y}_{23}, \tilde{Y}_{03}), (\tilde{Y}_{21}, \tilde{Y}_{01}), (\tilde{Y}_{01}, \tilde{Y}_{21})\}. \end{aligned}$$

All the paired-coefficients of *HS* can accumulate the distortion to the two rows in the middle of the matrix E.

$$\begin{aligned} \textit{Vertical set } (VS) \quad &= \{(\tilde{Y}_{22}, \tilde{Y}_{20}), (\tilde{Y}_{20}, \tilde{Y}_{22}), (\tilde{Y}_{30}, \tilde{Y}_{32}) \\ &\quad (\tilde{Y}_{32}, \tilde{Y}_{30}), (\tilde{Y}_{12}, \tilde{Y}_{10}), (\tilde{Y}_{10}, \tilde{Y}_{12})\}. \end{aligned}$$

Similarly, the distortion is accumulated to the two columns in the middle of the matrix E for all the paired-coefficients of *VS*.

We give an example $(\tilde{Y}_{03}, \tilde{Y}_{23})$ to explain this more clearly. Assume that $\tilde{Y}_{03}$ is added by t caused by the embedding procedure, that is

$$\Delta = \begin{pmatrix} 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} \frac{abtQstep}{4} & -\frac{abtQstep}{2} & \frac{abtQstep}{2} & -\frac{abtQstep}{4} \\ \frac{abtQstep}{4} & -\frac{abtQstep}{2} & \frac{abtQstep}{2} & -\frac{abtQstep}{4} \\ \frac{abtQstep}{4} & -\frac{abtQstep}{2} & \frac{abtQstep}{2} & -\frac{abtQstep}{4} \\ \frac{abtQstep}{4} & -\frac{abtQstep}{2} & \frac{abtQstep}{2} & -\frac{abtQstep}{4} \end{pmatrix}.$$

However, if we subtract $\widetilde{Y}_{23}$ by t accordingly, that is

$$\Delta = \begin{pmatrix} 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{abtQstep}{2} & -abtQstep & abtQstep & -\frac{abtQstep}{2} \\ \frac{abtQstep}{2} & -abtQstep & abtQstep & -\frac{abtQstep}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The embedding induced distortion is accumulated to the two rows in the middle of the matrix, and the first and last rows of matrix B are zeroes. So the first and last rows of matrix E are zeroes according to (8).

Furthermore, we give a real 4×4 block as an example to justify paired-coefficients $(\widetilde{Y}_{10}, \widetilde{Y}_{12})$ in Fig. 6. This residual block is the 5th 4×4 block from the 72th macro block in the 7th I frame of *Akiyo*, and it undergoes the procedure in the real world. At the encoder, the residual block R converts to $\widetilde{Y}$ after ICT, post-scaling, and quantization used a fixed quantization step parameter of **28**. At the decode process, after the re-scaling and inverse ICT for $\widetilde{Y}$, we get the residual block R'.

Assume that $\widetilde{Y}_{10}$ is subtracted by one for data hiding as depicted in Fig. 6(a); after the re-scaling and inverse ICT, we get the residual block R'', and the error matrix E has non-zero values on all the positions.

Assume that $\widetilde{Y}_{10}$ is subtracted by one while $\widetilde{Y}_{12}$ is added by one correspondingly as depicted in Fig. 6(b); after the re-scaling and inverse ICT, we get the residual block R'', and the error matrix E has a zero column on the rightmost.

The pixel values of the two rows in the middle will not be used in the following intra-frame prediction if the current block meets *Condition 1* as shown in Fig. 7(a). The pixel values of the two columns in the middle will not be used in the following intra-frame prediction if the current block meets *Condition 2* as shown in Fig. 7(b). Thus, if a block is chosen according to the intra-frame prediction modes of its adjacent blocks, as described in Fig. 7, to make sure that the embedding induced distortion of current block do not propagate to the other blocks, the distortion drift can be totally avoided.

The comparison of the number of the macro blocks from 14 H.264 test video sequences which meet the specific conditions is provided in Fig. 9 of Section V. The macro blocks which meet *Condition 1* or *Condition 2* are almost nine times as many as the macro blocks which meet all the three conditions. Figs. 10 and 11 present the experimental results of distortion of H.264 video sequences with data hiding into the paired-coefficients (without intra-frame distortion drift) or non-paired-coefficients (with intra-frame distortion drift). For the sake of convenience, these results will be described in detail in Section V.

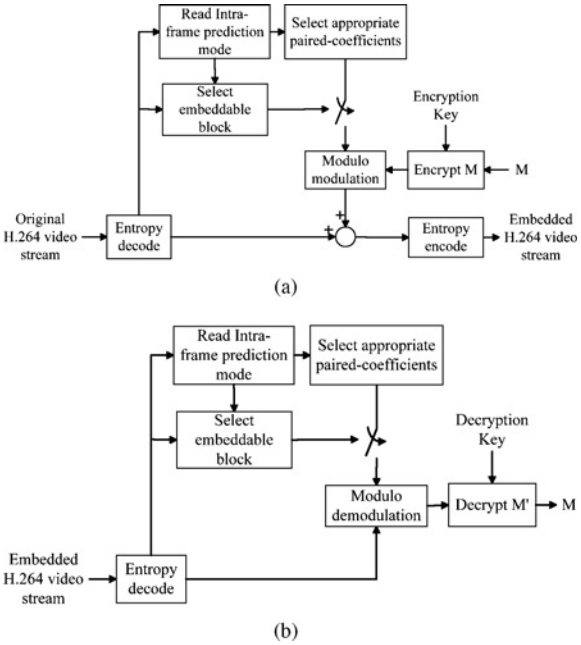

Read Intra-frame prediction mode
Select appropriate paired-coefficients
Select embeddable block
Encryption Key
Modulo modulation
Encrypt M
M
Original H.264 video stream
Entropy decode
+
+
Entropy encode
Embedded H.264 video stream
(a)
Read Intra-frame prediction mode
Select appropriate paired-coefficients
Select embeddable block
Decryption Key
Modulo demodulation
Decrypt M'
M
Embedded H.264 video stream
Entropy decode
(b)

Fig. 8. Proposed data hiding scheme diagram. (a) Embedding. (b) Extraction.

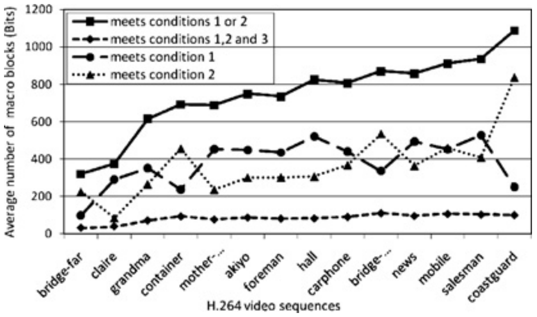


Fig. 9. Number of macro blocks which meet the specific conditions.

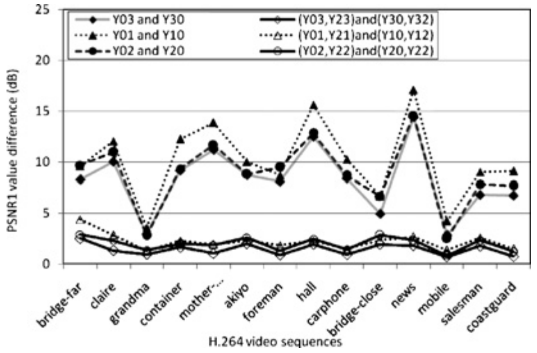


Fig. 10. Comparison of the average PSNR1 differences of H.264 video. Sequences with data hiding into the paired-coefficients or non-paired-coefficients $(\exists \widetilde{Y}_{ij} \neq 0)$.

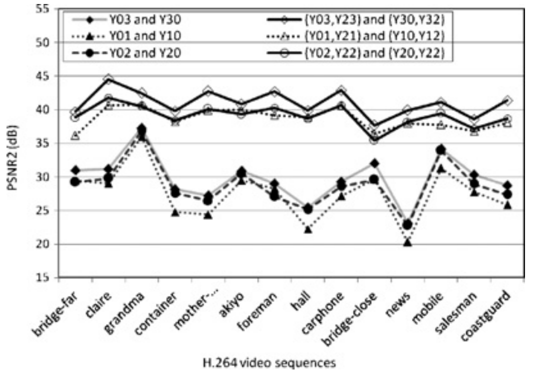


Fig. 11. Comparison of the average PSNR2 of H.264 video sequences with data hiding into the paired-coefficients or non-paired-coefficients ($\exists \widetilde{Y}_{ij} \neq 0$).

IV. Proposed Scheme

The proposed scheme hides information in H.264/AVC bit stream. Fig. 8 depicts the scheme of our proposed algorithm. The embedded information M is encrypted into M'. We assume that $M' = m_1 m_2, \ldots, m_n, m_i \in \{0, 1\}$.

A. Generic Data Hiding Procedure

First, the original video is entropy decoded to get the intra-frame prediction modes and quantized DCT coefficients. Then, the 4×4 luminance DCT blocks with large residuals and the appropriate paired-coefficients for embedding are selected. The encrypted message is embedded into appropriate paired-coefficients based on modulo modulation. Then, all the quantized DCT coefficients are entropy encoded to get the target embedded video.

To make the embedding procedure more clear, two paired-coefficients $(\widetilde{Y}_{03}, \widetilde{Y}_{23})$ and $(\widetilde{Y}_{30}, \widetilde{Y}_{32})$ are exploited for data hiding as an example.

1) *Select embeddable block:* The block is selected according to its quantized DCT coefficients, such as whether there exist non-zero coefficients in current block ($\exists \widetilde{Y}_{ij} \neq 0$) or $|\widetilde{Y}_{00}| >$ threshold (threshold = 0, 1, 2, 3, ...).
2) *Select appropriate paired-coefficients:* If current block meets *Condition 1*, embed 1 bit into $(\widetilde{Y}_{03}, \widetilde{Y}_{23})$ as described in *Modulation Method*.

   If current block meets *Condition 2*, embed 1 bit into $(\widetilde{Y}_{30}, \widetilde{Y}_{32})$ as described in *Modulation Method*.

   Otherwise go to Step 1 to select the next embeddable block.

Modulation Method: Assume that the selected paired-coefficient is (C_1, C_2).

1) If the embedded bit is 1, C_1 and C_2 are modified as follows.

   Case 1: $C_1 mod 2 \neq 0$ and $C_1 \geq 0$, then $C_1 = C_1 - 1, C_2 = C_2 + 1$.

   Case 2: $C_1 mod 2 \neq 0$ and $C_1 < 0$, then $C_1 = C_1 + 1, C_2 = C_2 - 1$.

   Case 3: $C_1 mod 2 = 0$, then $C_1 = C_1, C_2 = C_2$.
2) If the embedded bit is 0, C_1 and C_2 are modified as follows.

   Case 1: $C_1 mod 2 = 0$ and $C_1 \geq 0$, then $C_1 = C_1 - 1, C_2 = C_2 + 1$.

   Case 2: $C_1 mod 2 = 0$ and $C_1 < 0$, then $C_1 = C_1 + 1, C_2 = C_2 - 1$.

   Case 3: $C_1 mod 2 \neq 0$, then $C_1 = C_1, C_2 = C_2$.

B. Hidden Data Extraction Without Original Video Host

The retrieval of hidden information is simple and fast. After entropy decoding of the H.264/AVC video, we choose the embeddable blocks and extract the hidden data M' as follows:

$$m_i = \begin{cases} 1, & \text{if } \widetilde{Y}_{03} \text{ mod 2=1 and current block meets Condition 1} \\ 0, & \text{if } \widetilde{Y}_{03} \text{ mod 2=0 and current block meets Condition 1} \end{cases}$$

$$m_{i+1} = \begin{cases} 1, & \text{if } \widetilde{Y}_{30} \text{ mod 2=1 and current block meets Condition 2} \\ 0, & \text{if } \widetilde{Y}_{30} \text{ mod 2=0 and current block meets Condition 2.} \end{cases}$$

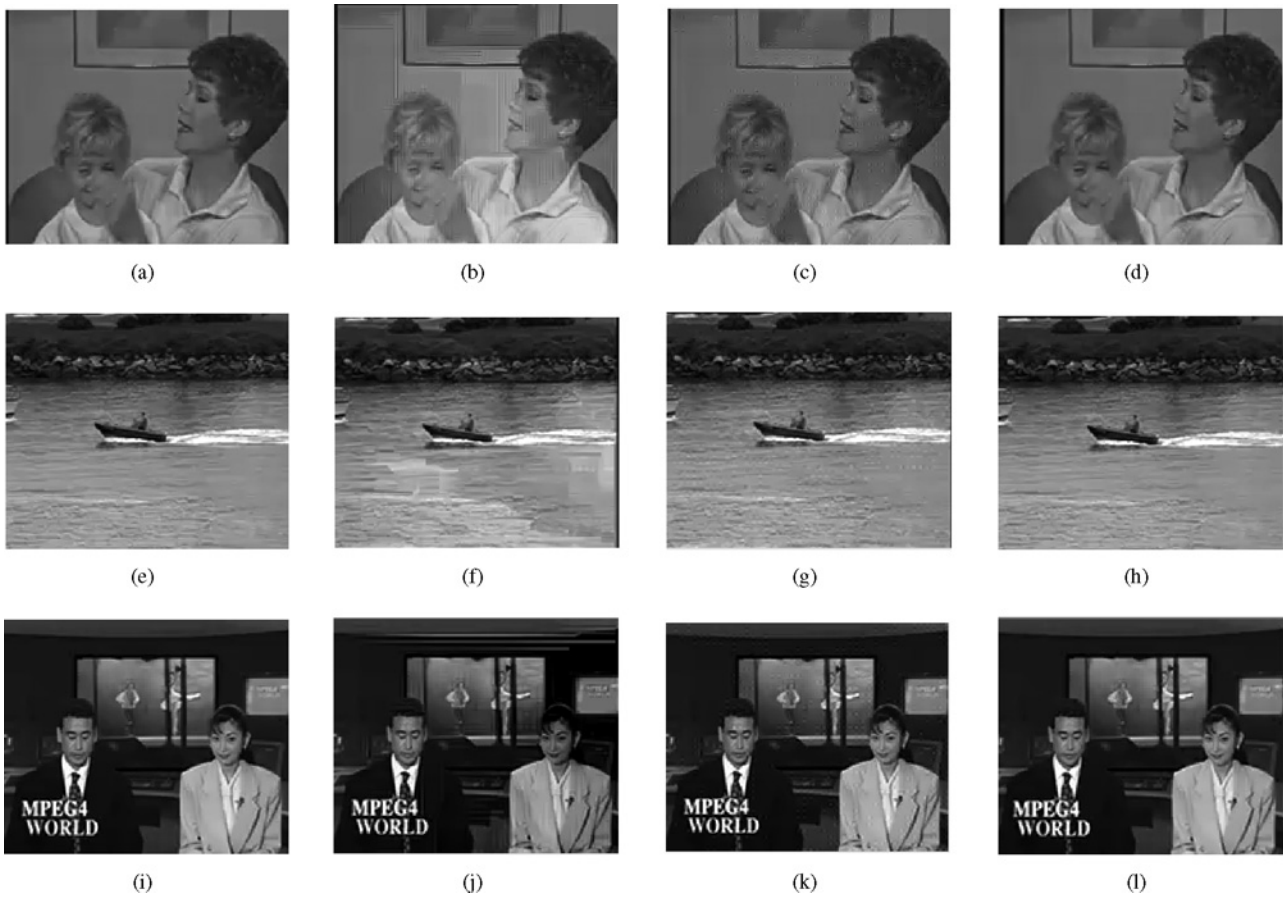
Then M' is decrypted to get the embedded information M. The algorithm has low computational complexity for the entropy decoding is very fast.

In this paper, the block is selected based on $\widetilde{Y}_{00}$, because the un-quantized coefficient Y_{00} is the average of the residual block R according to the 4×4 DCT basis functions, and there are two reasons for the usage of criterion $\exists \widetilde{Y}_{ij} \neq 0$. One reason is that embedding data into a block of all zero coefficients would bring significant visual distortion. The other reason is that "there is a coded block pattern parameter in H.264/AVC that indicates which blocks within a macro block contain non-zero coefficients, if the algorithm select a block of all zero coefficients to be embeddable, the parameter does not let the change take place," as the author said in [8].

V. Experimental Results and Discussions

The proposed algorithm has been implemented in the H.264/AVC reference software version JM16.0. The H.264 encoder used a fixed quantization step parameter of 28 for all I frames. The 14 video sequences (quarter common intermediate format, QCIF, 176×144) *Bridge-far*, *Claire*, *Grandma*, *Container*, *Mother-daughter*, *Akiyo*, *Foreman*, *Hall*, *Carphone*, *Bridge-close*, *News*, *Mobile*, *Salesman*, and *Coastguard* are used to be test samples. The standard video sequences are encoded into 300 frames at 30 frame/s and with an intra-period of 15 (group of picture IBPBPBPBPBPBPBPBPBPBPBPBPBPBPB). Each sequence has ten I frames. Three paired-coefficients $\{(\widetilde{Y}_{03}, \widetilde{Y}_{23}), (\widetilde{Y}_{30}, \widetilde{Y}_{32})\}$, $\{(\widetilde{Y}_{01}, \widetilde{Y}_{21}), (\widetilde{Y}_{10}, \widetilde{Y}_{12})\}$, and $\{(\widetilde{Y}_{02}, \widetilde{Y}_{22}), (\widetilde{Y}_{20}, \widetilde{Y}_{22})\}$ are exploited for data hiding, respectively.

The peak signal-to-noise ratio (PSNR) value "*PSNR1*" in this paper is calculated compared to the original frames of the corresponding YUV files. The discrepancy between the luminance average *PSNR1* value before and after data hiding is defined as *PSNR1 difference*. The PSNR value "*PSNR2*" in this paper is calculated compared to the decoded videos of the corresponding H.264 files. Besides, the PSNR values in this paper are the average of all the frames (including I, B, and P frames) of a video sequence.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 12. Original and embedded frames of *Mother-daughter* (frame 60), *Coastguard* (frame 0), and *News* (frame 30) with data hiding with and without intra-frame distortion drift. (See Table I for the captions of sub-figures.)

A. Comparison of Data Hiding with and without Intra-Frame Distortion Drift

To compare the perceptual quality of our proposed algorithm with data hiding into the paired-coefficients (without intra-frame distortion drift) and non-paired-coefficients (with intra-frame distortion drift), Fig. 10 gives the experimental results of *PSNR1 difference* between the original and embedded H.264 videos of 14 H.264 video sequences with data hiding into the paired-coefficients or single coefficient, and Fig. 11 provides the *PSNR2* values. We present the results for six different implementations. [In these experiments, a block is considered to be embeddable if there exists non-zero coefficients in it ($\exists \widetilde{Y}_{ij} \neq 0$).]

In the first case, the data is hidden only into the luminance coefficients $\widetilde{Y}_{03}$ and $\widetilde{Y}_{30}$. The average *PSNR1* difference is 8.21 dB. The average *PSNR2* values of each video sequence are less than 37.29 dB, and the average *PSNR2* value of these 14 sequences is 29.87 dB. In the second case, the data is hidden into the paired-coefficients ($\widetilde{Y}_{03}, \widetilde{Y}_{23}$) and ($\widetilde{Y}_{30}, \widetilde{Y}_{32}$) as described in Section IV. The average *PSNR1* difference is 1.46 dB. The average *PSNR2* values of each video sequence are more than 37.67 dB, and the average *PSNR2* value of these 14 sequences is 41.01 dB.

Similarly, in the third case, the data is hidden only into the luminance coefficients $\widetilde{Y}_{01}$ and $\widetilde{Y}_{10}$. The average *PSNR1* difference is 10.14 dB. The average *PSNR2* values of each video sequence are less than 35.97 dB, and the average *PSNR2* value of these 14 sequences is 27.55 dB. In the fourth case, the data is hidden into the paired-coefficients ($\widetilde{Y}_{01}, \widetilde{Y}_{21}$) and ($\widetilde{Y}_{10}, \widetilde{Y}_{12}$). The average *PSNR1* difference is 2.24 dB. The average *PSNR2* values of each video sequence are more than 36.17 dB, and the average *PSNR2* value of these 14 sequences is 38.67 dB.

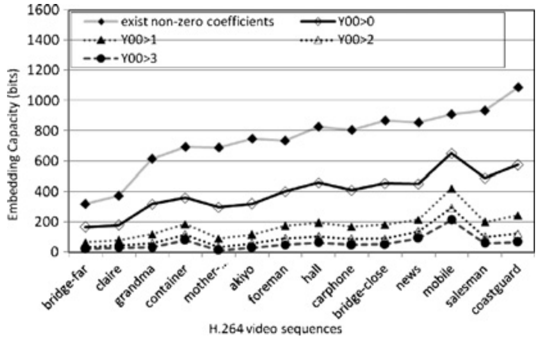


Fig. 13. Average embedding capacity of H.264 video sequences.

In the fifth case, the data is hidden only into the luminance coefficients $\widetilde{Y}_{02}$ and $\widetilde{Y}_{20}$. The average *PSNR1* difference is 8.82 dB. The average *PSNR2* values of each video sequence are less than 36.81 dB, and the average *PSNR2* value of these 14 sequences is 28.84 dB. In the sixth case, the data is hidden into the paired-coefficients ($\widetilde{Y}_{02}, \widetilde{Y}_{22}$) and ($\widetilde{Y}_{20}, \widetilde{Y}_{22}$).

TABLE I

COMPARISON FOR DATA HIDING WITH AND WITHOUT INTRA-FRAME DISTORTION DRIFT IN FIG. 12

Sequences /Parameters		Embedded Coefficients	Block Selection	DPSNR1 (dB)	PSNR2 (dB)
Mother-daughter	(a)	–	–	0	0
	(b)	$\tilde{Y}_{30},\ \tilde{Y}_{03}$	$\exists\tilde{Y}_{ij} \neq 0$	16.2	22.33
	(c)	$(\tilde{Y}_{03},\ \tilde{Y}_{23}), (\tilde{Y}_{30},\ \tilde{Y}_{32})$	$\exists\tilde{Y}_{ij} \neq 0$	2.27	39.38
	(d)	$(\tilde{Y}_{03},\ \tilde{Y}_{23}), (\tilde{Y}_{30},\ \tilde{Y}_{32})$	$\lvert\tilde{Y}_{00}\rvert > 0$	1.15	42.79
Coast-guard	(e)	–	–	0	0
	(f)	$\tilde{Y}_{10}, \tilde{Y}_{01}$	$\exists\tilde{Y}_{ij} \neq 0$	10.83	27.44
	(g)	$(\tilde{Y}_{01}, Y_{21}), (\tilde{Y}_{10}, Y_{12})$	$\exists\tilde{Y}_{ij} \neq 0$	2.88	35.5
	(h)	$(\tilde{Y}_{01}, Y_{21}), (\tilde{Y}_{10}, Y_{12})$	$\lvert\tilde{Y}_{00}\rvert > 0$	0.97	40.75
News	(i)	–	–	0	0
	(j)	$\tilde{Y}_{20}, \tilde{Y}_{02}$	$\exists\tilde{Y}_{ij} \neq 0$	14.6	21.69
	(k)	$(\tilde{Y}_{02}, \tilde{Y}_{22}), (\tilde{Y}_{20}, \tilde{Y}_{22})$	$\exists\tilde{Y}_{ij} \neq 0$	2.73	38.31
	(l)	$(\tilde{Y}_{02}, \tilde{Y}_{22}), (\tilde{Y}_{20}, \tilde{Y}_{22})$	$\lvert\tilde{Y}_{00}\rvert > 0$	1.63	41.21

TABLE II

EMBEDDING PERFORMANCE COMPARISON BETWEEN OUR SCHEME AND [2] AND [3]

Sequences		Algorithm in [2]	Algorithm in [3]	Our Scheme ($\exists\tilde{Y}_{ij} \neq 0$)
Readable/Detectable		Detectable	Detectable	Readable
Carphone	Capacity (bits)	507	500	806
	PSNR2 (dB)	38.32	41.24	42.9
	Bit-rate increase (%)	4.38	3.28	3.24
Foreman	Capacity (bits)	591	597	735
	PSNR2 (dB)	37.28	40.18	42.72
	Bit-rate increase (%)	4.62	2.64	3.16
Mobile	Capacity (bits)	1225	1219	910
	PSNR2 (dB)	34.2	37.22	41.12
	Bit-rate increase (%)	2.17	0.68	0.88
Salesman	Capacity (bits)	736	750	936
	PSNR2 (dB)	36.69	39.57	38.58
	Bit-rate increase (%)	4.43	2.49	7.17

The average *PSNR1* difference is 1.99 dB. The average *PSNR2* values of each video sequence are more than 35.41 dB, and the average *PSNR2* value of these 14 sequences is 39.10 dB.

Furthermore, frames from standard video sequences *Mother-daughter*, *Coastguard*, and *News* (QCIF, 176×144) are shown in Fig. 12. The coefficients for embedding, the block selection conditions, *PSNR1 difference*, and *PSNR2* corresponding to Fig. 12(a)–(l) are provided in Table I. Frames (a), (e), and (i) are the original frames, frames (b), (f), and (j) are the embedded frames with intra-frame distortion drift, and frames (c), (d), (g), (h), (k), and (l) are the embedded frames with data hiding into paired-coefficients.

The visual distortion of frames (b), (f), and (j) are obvious and intolerable for data hiding algorithms. There is a massive distortion in the middle of frame (b), including the two persons' faces and clothes. On the bottom right of frame (f), the distortion is on the surface of water, and there are several horizontal stripes on the upper part of frame (j).

In the experiments to get I frames (c), (g), and (k), a block is considered to be embeddable if there exist non-zero coefficients in it ($\exists\widetilde{Y}_{ij} \neq 0$); whereas, in the experiments to get I frames (d), (h), and (l), a block is considered to be embeddable if its DC coefficient is non-zero ($|\widetilde{Y}_{00}| > 0$). These frames have little distortion and achieve good invisibility compared to those original frames (a), (e), and (i).

It can be concluded from the results that embedding data into paired-coefficients without intra-frame distortion drift will get much better visual quality.

B. Data Hiding Performance

Figs. 13 and 14 show the data hiding performances of our algorithm.

The embedding capacity of a video sequence is the average number of bits embedded in one I frame of the ten I frames of that sequence. Fig. 13 presents the comparison of the embedding capacity with five different embeddable block selection conditions. The embedding capacity of a specific H.264 video sequence is only related to the embeddable block selection conditions; it has nothing to do with which paired-coefficients group of those three is used for data hiding. Data hiding with stricter block selection condition will get lower embedding capacity, better invisibility, and lower bit-rate increase.

We compared the *PSNR1 difference*, the average *PSNR2* value, and the bit-rate increase for each video sequence with different paired-coefficients groups and different embeddable block selection conditions in Fig. 14. The three frames (a), (b), and (c) depict the *PSNR1 difference*, the average *PSNR2* value, and the bit-rate increase for each video sequence with data hiding into $(\widetilde{Y}_{03}, \widetilde{Y}_{23})$ and $(\widetilde{Y}_{30}, \widetilde{Y}_{32})$. Similarly, the three frames (d), (e), and (f) provide the *PSNR1 difference*, the average *PSNR2* value, and the bit-rate increase with data hiding into $(\widetilde{Y}_{01}, \widetilde{Y}_{21})$ and $(\widetilde{Y}_{10}, \widetilde{Y}_{12})$. The three frames (g), (h), and (i) show the embedding performances of our algorithm with data hiding into $(\widetilde{Y}_{02}, \widetilde{Y}_{22})$ and $(\widetilde{Y}_{20}, \widetilde{Y}_{22})$.

Data hiding into paired-coefficients $(\widetilde{Y}_{03}, \widetilde{Y}_{23})$ and $(\widetilde{Y}_{30}, \widetilde{Y}_{32})$ will get better visual quality and higher bit-rate increase than hiding data into $(\widetilde{Y}_{02}, \widetilde{Y}_{22})$, $(\widetilde{Y}_{20}, \widetilde{Y}_{22})$, and $(\widetilde{Y}_{01}, \widetilde{Y}_{21})$, $(\widetilde{Y}_{10}, \widetilde{Y}_{12})$. Data hiding into paired coefficients $(\widetilde{Y}_{02}, \widetilde{Y}_{22})$ and $(\widetilde{Y}_{20}, \widetilde{Y}_{22})$ will get better visual quality and higher bit-rate increase than hiding data into $(\widetilde{Y}_{01}, \widetilde{Y}_{21})$ and $(\widetilde{Y}_{10}, \widetilde{Y}_{12})$.

Also, Table II gives the embedding performance comparison between the proposed scheme with data hiding into $(\widetilde{Y}_{03}, \widetilde{Y}_{23})$ and $(\widetilde{Y}_{30}, \widetilde{Y}_{32})$ and the algorithms in [2] and [3]. [2] and [3] are *detectable* algorithms which can detect the existence of the embedded data with the knowledge of the embedded contents. Our algorithm is *readable* which can read the embedded information without the knowledge of the embedded contents. The results proved that our method often performs better than those in [2] and [3], both in embedding capacity and *PSNR2*. But the bit-rate increase of our scheme and [2] are more significant than the algorithm in [3]. It is caused by high embedding capacity and the more modified DCT coefficients of our scheme. In our future work, we will combine the single-coefficient data hiding algorithm with the proposed scheme to decrease the bit-rate increase.

C. Discussions

In this paper, we focus on the intra-frame distortion drift; the embedding induced intra-frame distortion, although not propagate to the neighboring blocks, will propagate to the

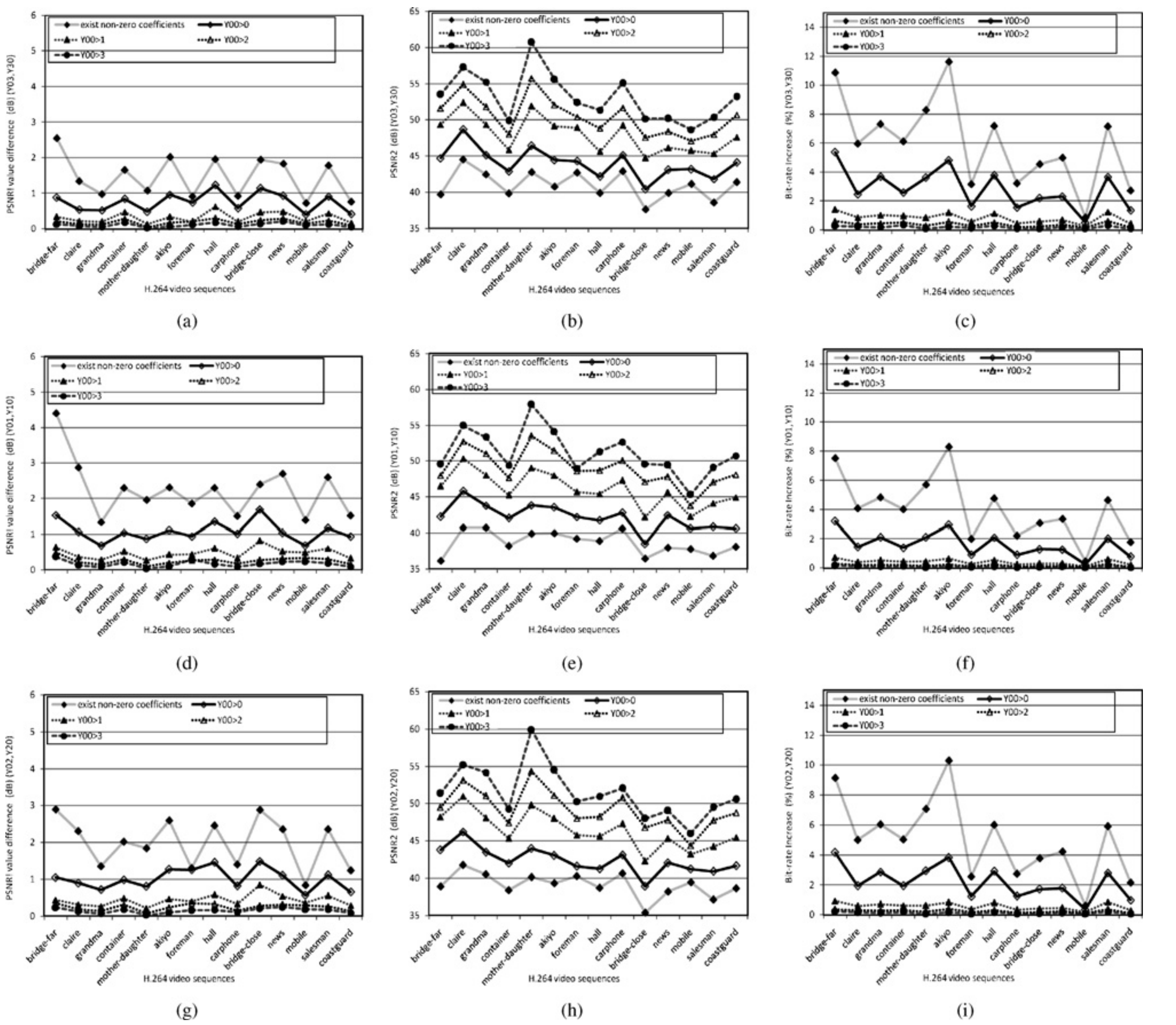


Fig. 14. Embedding performance of proposed algorithm. (a), (d), (g) PSNR1 difference of 14 video sequences with embedding into corresponding paired-coefficients. (b), (e), (h) PSNR2 of 14 video sequences with embedding into corresponding paired-coefficients. (c), (f), (i) Bit-rate increase of 14 video sequences with embedding into corresponding paired-coefficients.

following P or B frames. In order to evaluate the visual distortion of B and P frames, all the PSNR values in this paper are the average of all the frames (including I, B, and P frames). Besides, Fig. 15 gives the *PSNR1* and *PSNR2* of the first 50 frames of *Claire* with data hiding into $(\widetilde{Y}_{01}, \widetilde{Y}_{21})$ and $(\widetilde{Y}_{10}, \widetilde{Y}_{12})$ with and without intra-frame distortion drift (intra-period = 15, block selection condition is $\exists \widetilde{Y}_{ij} \neq 0$).

Table III gives the embedding capacity and visual distortion for the eight video sequences with different intra-period. The standard video sequences are encoded into 10, 60, 300, 300 frames at 30 frame/s with intra-period 1 (all intra), 3, 6, 15, and 30, respectively. Each sequence with intra-period 1, 3, 6, and 15 has ten I frames and each sequence with intra-period 30 has five I frames (because there are only 300 frames in most of the standard YUV test video sequences). The *Capacity* is the average number of bits embedded into one I frame, and *DPSNR1* is the *PSNR1* value difference of the tested sequence before and after embedding. There is no significant difference in the embedding capacity and visual distortion for a video compressed with different intra-period.

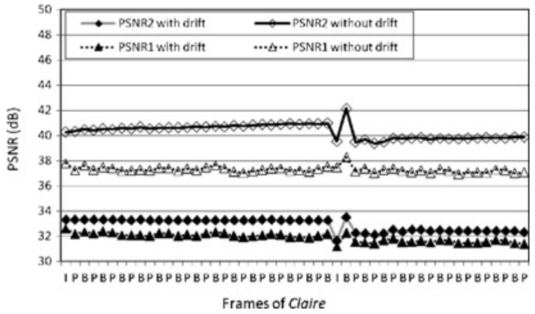


Fig. 15. Visual quality of the first 50 frames of *Claire*.

The proposed data hiding procedure includes entropy decoding, data embedding, and entropy encoding, and the embedded data retrieval procedure includes entropy decoding and data

TABLE III
EMBEDDING PERFORMANCES COMPARISON WITH DIFFERENT INTRA-PERIOD

Sequences		All Intra	Period = 3	Period = 6	Period = 15	Period = 30
Container	Capacity (bits)	671	674	673	692	698
	DPSNR1 (dB)	1.85	1.83	1.83	1.66	1.5
	PSNR2 (dB)	39.54	39.87	39.57	39.81	40.06
Mother-daughter	Capacity (bits)	682	682	695	689	671
	DPSNR1 (dB)	2.22	1.63	1.34	1.07	1.02
	PSNR2 (dB)	39.56	41.33	41.94	42.75	42.97
Foreman	Capacity (bits)	706	702	705	735	724
	DPSNR1 (dB)	1.97	1.08	0.92	0.91	1.47
	PSNR2 (dB)	39.39	42.01	42.65	42.72	41.64
Carphone	Capacity (bits)	798	784	761	806	814
	DPSNR1 (dB)	2.45	1.55	1.29	0.93	0.92
	PSNR2 (dB)	38.95	41.11	42.01	42.9	43.07
News	Capacity (bits)	901	893	878	856	860
	DPSNR1 (dB)	2.6	2.31	2.14	1.83	1.59
	PSNR2 (dB)	38.73	39.12	39.35	39.91	40.51
Mobile	Capacity (bits)	931	899	920	910	917
	DPSNR1 (dB)	1.58	0.67	0.75	0.72	1.69
	PSNR2 (dB)	38.25	41.45	40.82	41.12	42.2
Salesman	Capacity (bits)	919	945	917	936	936
	DPSNR1 (dB)	2.23	1.95	1.84	1.78	1.69
	PSNR2 (dB)	37.69	38.44	38.65	38.58	38.81
Coastguard	Capacity (bits)	1084	1111	1122	1088	1080
	DPSNR1 (dB)	1.91	1.36	0.94	0.76	0.55
	PSNR2 (dB)	37.43	38.68	40.4	40.4	42.6

extraction. Both procedures are simple and fast. All the experiments were run on a machine with 2.16 GHz processor and 2 GB RAM. The data hiding procedure for an I frame of the sequence *Coastguard* and *Bridge-far* cost 188 ms and 125 ms on average, respectively.

The proposed algorithm is a *readable* data hiding algorithm which aims to retrieve the embedded message exactly without the knowledge of the embedded contents. The embeddable (or extractable) blocks are selected based on the intra-frame prediction modes and the absolute value of their DC coefficients, if the video processing operation (such as recompression or re-quantization) changes the prediction modes or the absolute value of their DC coefficients and makes an embedded block un-extractable, the embedded bit will not be retrieved. The embedded message encryption and decryption will make this extraction bit error propagate to the other bits of the embedded message. However, the proposed algorithm embeds data into the quantized DCT coefficients of I frames, and the embedded bits will probably survive from video-processing operations. Thus, we can extend the algorithm by, such as, segmenting the embedded message and embedding repeatedly into different paired-coefficients, blocks, and I frames. In our future work, the embedded message will be preprocessed to be structured before embedding; we will combine the proposed algorithm with the secret sharing algorithms or error correcting coding technique to make it more robust to the video-processing operation.

VI. CONCLUSION

In this paper, we have investigated the problem of the intra-frame distortion drift induced by data hiding in H.264/AVC video streams. A readable data-hiding scheme is presented. In order to prevent the distortion propagation, we defined paired-coefficients and exploited the intra-frame prediction modes. Compared with other data-hiding methods on H.264/AVC video streams, the proposed algorithm can achieve high embedding capacity and low visual distortion.

ACKNOWLEDGMENT

The authors would sincerely like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] I. E. G. Richardson, *H. 264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. Chichester, U.K.: Wiley, 2003.

[2] M. Noorkami and R. M. Mersereau, “A framework for robust watermarking of H.264-encoded video with controllable detection performance,” *IEEE Trans. Inform. Forensics Security*, vol. 2, no. 1, pp. 14–23, Mar. 2007.

[3] X. Gong and H.-M. Lu, “Towards fast and robust watermarking scheme for H.264 video,” in *Proc. 10th IEEE ISM*, Dec. 2008, pp. 649–653.

[4] S. Kim, Y. Hong, and C. Won, “Data hiding on H.264/AVC compressed video,” *Image Anal. Recog.*, vol. 4633/2007, pp. 698–707, 2007.

[5] S. Kapotas, E. Varsaki, and A. Skodras, “Data hiding in H. 264 encoded video sequences,” in *Proc. IEEE 9th Workshop MMSP*, Oct. 2007, pp. 373–376.

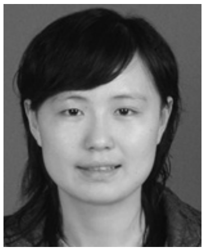
[6] G. Qiu, P. Marziliano, A. Ho, D. He, and Q. Sun, “A hybrid watermarking scheme for H.264/AVC video,” in *Proc. 17th ICPR*, Aug. 2004, pp. 865–868.

[7] J. Zhang, A. T. S. Ho, and G. Qiu, “Robust video watermarking of H.264/AVC,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 54, no. 2, pp. 205–209, Feb. 2007.

[8] M. Noorkami and R. Mersereau, “Compressed-domain video watermarking for H.264,” in *Proc. IEEE ICIP*, vol. 2. Sep. 2005, pp. 11–14.

[9] C.-V. Nguyen, D. Tay, and G. Deng, “A fast watermarking system for H.264/AVC video,” in *Proc. IEEE APCCAS*, Dec. 2006, pp. 81–84.

[10] R. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. 11th Int. Workshop IH*, Jun. 2009, pp. 48–58.

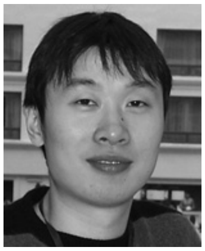


Xiaojing Ma was born in Jinzhai, China, in 1983. She received the B.E. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2003. She is currently pursuing the Ph.D. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan.

She has been a Software Engineer with Capital On-Line Communications Inc., Beijing, China. She has published several papers in the area of multimedia security. Her current research interests include cryptography, network security, and multimedia security.

Zhitang Li was born in Jianli, China, in 1951. He received the M.E. and Ph.D. degrees in computer architecture from the Huazhong University of Science and Technology, Wuhan, China, in 1987 and 1992, respectively.

He is currently the Director of the China Education and Research Network in Central China. He has been the Vice President with the Department of Computer Science and Technology, Huazhong University of Science and Technology. He has published more than 100 papers in the areas of computer architecture, network security, and P2P networks. His current research interests include computer architecture, network security, and P2P networks.



Hao Tu was born in Wuhan, China, in 1977. He received the B.S. and Ph.D. degrees in computer science and technology from the Huazhong University of Science and Technology, Wuhan, in 1999 and 2008.

He is currently a Lecturer with the Network Center, Huazhong University of Science and Technology. His current research interests include computer and network security and multimedia forensics.

Bochao Zhang was born in Hubei, China, in 1987. He received the B.S. degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2009. He is currently working toward a Masters degree with Huazhong University of Science and Technology.

His current research interests include information hiding and video technology.