The Journal of Systems and Software 86 (2013) 604–614

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames

Tseng-Jung Lin[a], Kuo-Liang Chung[a,1], Po-Chun Chang[a], Yong-Huai Huang[b,*,2], Hong-Yuan Mark Liao[c], Chiung-Yao Fang[d]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC
[b] Institute of Computer and Communication Engineering and Department of Electronic Engineering, Jinwen University of Science and Technology, No. 99, An-Chung Road, Hsin-Tien Dist., New Taipei City 23154, Taiwan, ROC
[c] Institute of Information Science, Academia Sinica, No. 128, Academia Road, Section 2, Taipei 115, Taiwan, ROC
[d] Department of Computer Science and Information Engineering, National Taiwan Normal University, No. 162, Heping East Road Section 1, Taipei 106, Taiwan, ROC

ARTICLE INFO

Article history:
Received 16 October 2011
Received in revised form
20 September 2012
Accepted 8 October 2012
Available online 8 November 2012

Keywords:
Data hiding
DCT
Embedding capacity
Error propagation-free
H.264/AVC
Human visual effect
Intra prediction

ABSTRACT

Recently, Ma et al. proposed an efficient error propagation-free discrete cosine transform-based (DCT-based) data hiding algorithm that embeds data in H.264/AVC intra frames. In their algorithm, only 46% of the 4×4 luma blocks can be used to embed hidden bits. In this paper, we propose an improved error propagation-free DCT-based perturbation scheme that fully exploits the remaining 54% of luma blocks and thereby doubles the data hiding capacity of Ma et al.'s algorithm. Further, in order to preserve the visual quality and increase the embedding capacity of the embedded video sequences, a new set of sifted 4×4 luma blocks is considered in the proposed DCT-based perturbation scheme. The results of experiments on twenty-six test video sequences confirm the embedding capacity superiority of the proposed improved algorithm while keeping the similar human visual effect in terms of SSIM (structural similarity) index.

1. Introduction

Data hiding techniques are used in a variety of application domains, such as the authentication, identification, annotation, and copyright protection of digital data. Among the different types of such data, digital video has become one of the most popular media due to the rapid development of Internet technology. Because video files are very large, the H.264/AVC video compression standard (Richardson, 2003) is frequently used to compress video sequences. Several H.264/AVC-based data hiding algorithms have been developed (Chen et al., 2005; Gong and Lu, 2008; Kapotas et al., 2007; Kim et al., 2007; Nguyen et al., 2006; Noorkami and Mersereau, 2005, 2007; Qiu et al., 2004; Wong et al., 2006, 2009; Zhang et al., 2007); however, they suffer from the error propagation problem in intra frames, which degrades the video quality. To avoid this error propagation problem, Ma et al. (2010) proposed an efficient discrete cosine transform-based (DCT-based) data hiding algorithm. Later, Huo et al. (2011) presented an improved algorithm to enhance the quality of marked video sequences. Besides the quality issue, capacity is also very important. After running Ma et al.'s algorithm on twenty-six video sequences, we found that it only embeds hidden bits in 46% of the 4×4 luma blocks. The motivation of this research is to design an improved algorithm that fully utilizes all luma blocks in order to improve the embedding capacity.

Instead of perturbing three QDCT coefficient-pairs to embed three bits into each luma block (Ma et al., 2010), this paper allows one more coefficient-pair to embed four bits into each luma block. Contrasty to Ma et al.'s approach, we fully utilize the remaining 54% of all luma blocks. To further preserve the visual quality of the embedded video sequences, a new set of sifted 4×4 luma blocks, which can be used to embed hidden bits by perturbing the related QDCT coefficients, is considered in the proposed DCT-based perturbation scheme. Based on twenty-six test video sequences, experimental results confirm the embedding capacity superiority while keeping the similar human visual effect in terms of SSIM (structural similarity) index when compared with Ma et al.'s

* Corresponding author.
E-mail addresses: klchung01@gmail.com (K.-L. Chung), yonghuai@ms28.hinet.net (Y.-H. Huang).

[1] Supported by the National Council of Science of R.O.C. under the contract NSC98-2221-E-011-084-MY3.
[2] Supported by the National Science Council of R.O.C. under the contract NSC100-2221-E-228-007.

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.10.922

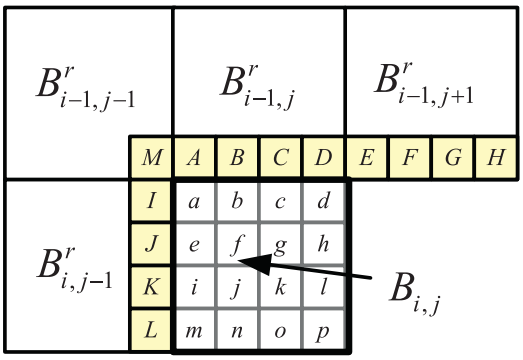

Fig. 1. The predicted pixels in a 4×4 luma block and the related reference pixels in the four encoded neighboring blocks.

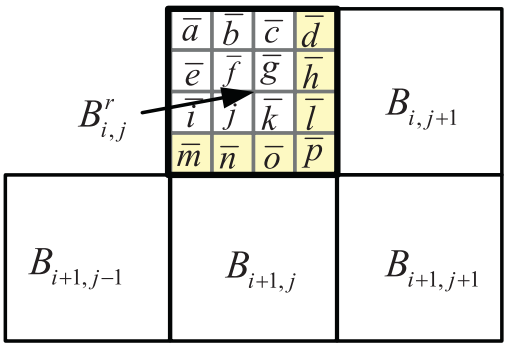

Fig. 3. An encoded 4×4 luma block and the four adjacent luma blocks affected by the encoding process.

algorithm. The remainder of this paper is organized as follows. In the next section, we highlight the capacity limitation in Ma et al.'s algorithm. In Section 3, the proposed algorithm is presented and a related theoretical analysis is given. In Section 4, we report the results of experiments. Section 5 contains some concluding remarks.

2. The capacity limitation problem in Ma et al.'s data hiding scheme

In this section, we briefly explain intra prediction under H.264/AVC, and describe how an existing approach defines five categories for all 4×4 luma blocks. It is well known that intra prediction plays an important role in reducing spatial redundancy among H.264/AVC intra frames. In H.264/AVC, each 16×16 macroblock (MB) can be encoded by 16×16 intra mode or 4×4 intra mode. Here we only consider embedding data into MBs encoded by 4×4 intra mode since MBs encoded by 16×16 intra mode are homogeneous and thus the human eyes are very sensitive to the change of luma values in 16×16 intra MBs. As shown in Fig. 1, $B_{i,j}$ is the current 4×4 luma block whose pixels are labeled from a to p, and its four adjacent luma blocks, $B_{i-1,j-1}$, $B_{i-1,j}$, $B_{i-1,j+1}$, and $B_{i,j-1}$, have been encoded. If the four blocks are decompressed, the four reconstructed blocks (denoted by $B^r_{i-1,j-1}$, $B^r_{i-1,j}$, $B^r_{i-1,j+1}$, and $B^r_{i,j-1}$, respectively) and the reference pixels labeled A to M can be utilized to predict $B_{i,j}$. Then, an intra mode decision scheme selects an optimal prediction mode from the nine modes and uses it to construct the predicted luma block $B^p_{i,j}$, as shown in Fig. 2. Next, based on $B^p_{i,j}$, the corresponding residual block $R^p_{i,j}$ is calculated by $R^p_{i,j} = B_{i,j} - B^p_{i,j}$, after which the DCT process and the quantization operation are implemented to encode $R^p_{i,j}$. For convenience, the quantized DCT (QDCT) coefficients of $R^p_{i,j}$ are denoted by $R^{QDCT}_{i,j}$. Before encoding the next luma block $B_{i,j+1}$, the encoded $B_{i,j}$, which is represented by $B^p_{i,j}$ and $R^{QDCT}_{i,j}$, must be decompressed. Then, the dequantization operation and the inverse DCT process are applied to obtain the reconstructed version of $R^{QDCT}_{i,j}$ denoted as $R^r_{i,j}$. Therefore, the reconstructed luma block, $B^r_{i,j}$, is derived by $B^r_{i,j} = B^p_{i,j} + R^r_{i,j}$.

Ma et al. (2010) embedded hidden bits in the encoded block $B_{i,j}$ by perturbing the coefficients of $R^{QDCT}_{i,j}$. As shown in Fig. 3, the seven pixels, $\overline{d}$, $\overline{h}$, $\overline{l}$, $\overline{m}$, $\overline{n}$, $\overline{o}$, and $\overline{p}$, in the reconstructed block $B^r_{i,j}$ are utilized in the intra prediction process. The four luma blocks affected by the above process are $B_{i,j+1}$, $B_{i+1,j-1}$, $B_{i+1,j}$, and $B_{i+1,j+1}$. To study the error propagation-free property, Ma et al. considered the error propagation problem caused by perturbing $R^{QDCT}_{i,j}$. For example,

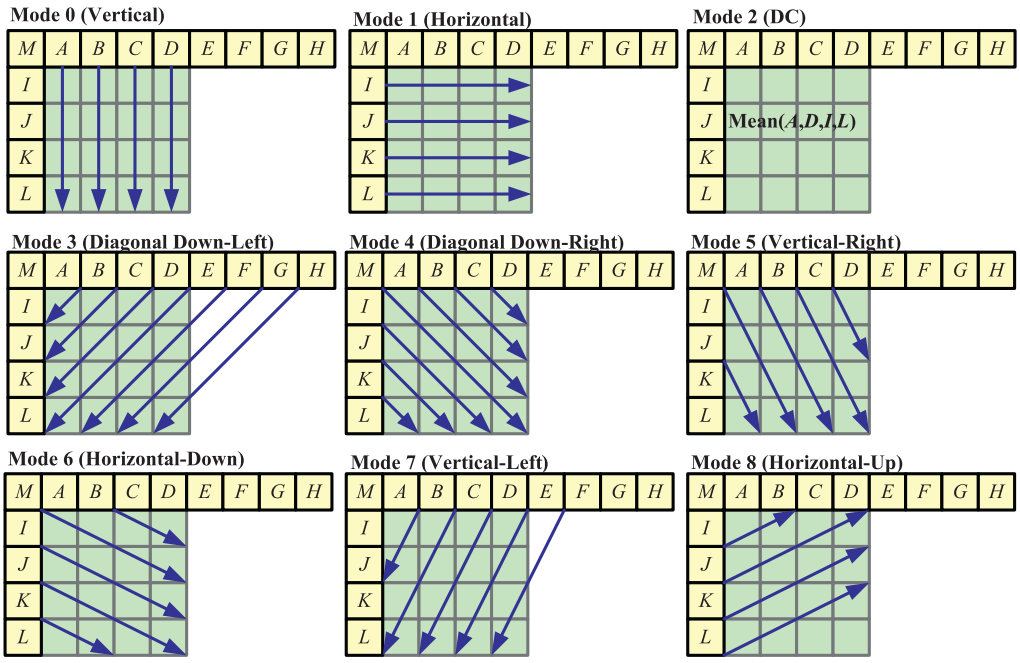

Fig. 2. Nine 4×4 luma prediction modes.

Table 1
Five categories of optimally selected modes of adjacent 4×4 luma blocks and the related reference pixels.

	$Case_1$	$Case_2$	$Case_3$	Reference pixels
Cat_1	TRUE	FALSE	X	$\overline{d}, \overline{h}, \overline{l}, \overline{p}$
Cat_2	FALSE	TRUE	X	$\overline{m}, \overline{n}, \overline{o}, \overline{p}$
Cat_3	FALSE	FALSE	FALSE	NIL
Cat_4	FALSE	FALSE	TRUE	$\overline{p}$
Cat_5	TRUE	TRUE	X	ALL

X: Don't care; NIL: No reference pixel; ALL: All seven related pixels.

assume that the selected prediction mode for $B_{i,j+1}$ is 1, as shown in Fig. 2, and that $\overline{d}$, $\overline{h}$, $\overline{l}$, and $\overline{p}$ are used to predict $B_{i,j+1}$. The error propagation problem will occur if the luma values of $\overline{d}$, $\overline{h}$, $\overline{l}$, and $\overline{p}$ are changed by perturbing some coefficients of $R_{i,j}^{QDCT}$.

To solve the error propagation problem, Ma et al. (2010) first classify the relations between the above-mentioned seven boundary pixels and the selected prediction modes of four adjacent blocks into three cases. For simplicity, let $M_{i,j+1}$, $M_{i+1,j-1}$, $M_{i+1,j}$, and $M_{i+1,j+1}$ denote the four sets of prediction modes selected for $B_{i,j+1}$, $B_{i+1,j-1}$, $B_{i+1,j}$, and $B_{i+1,j+1}$, respectively. The value of each mode is in the range 0–8. The three cases, $Case_1$, $Case_2$, and $Case_3$, are defined as follows. $Case_1$ occurs when the value of $M_{i,j+1}$ is in $\{1, 2, 4, 5, 6, 8\}$, which indicates that $\overline{d}$, $\overline{h}$, $\overline{l}$, and $\overline{p}$ are the reference pixels for predicting $B_{i,j+1}$. $Case_2$ occurs when the value of $M_{i+1,j}$ falls in $\{0, 2, 3, 4, 5, 6, 7\}$ and the value of $M_{i+1,j-1}$ is in $\{3, 7\}$. This indicates that $\overline{m}$, $\overline{n}$, $\overline{o}$, and $\overline{p}$ are the reference pixels for predicting $B_{i,j+1}$ and $B_{i+1,j}$. $Case_3$ occurs when the value of $M_{i+1,j+1}$ falls in $\{4, 5, 6\}$, which indicates that $\overline{p}$ is the reference pixel for predicting $B_{i+1,j+1}$. The five categories shown in Table 1 (Cat_1, Cat_2, Cat_3, Cat_4, and Cat_5) are used to characterize the relations between the reference pixels and the selected prediction modes of the four adjacent blocks. For example, for Cat_1, $Case_1$ is true and $Case_2$ is false, so the reference pixels $\overline{d}$, $\overline{h}$, $\overline{l}$, and $\overline{p}$ are used to predict $B_{i,j+1}$. In this category, if the perturbed coefficients of $R_{i,j}^{QDCT}$ do not affect the four reference pixels $\overline{d}$, $\overline{h}$, $\overline{l}$, and $\overline{p}$, there will be no error propagation. The same finding applies to the other four categories.

Actually, Ma et al. (2010) solved the error propagation problem in the first four categories in Table 1. In their scheme, each 4×4 luma block belonging to Cat_1, Cat_2, and Cat_4 contributes three-bits of embedding capacity. For each category, three specific QDCT coefficient-pairs can be identified and the three-bits embedding scheme can be implemented by perturbing the three corresponding coefficient-pairs. Since there are no reference pixels in Cat_3, Ma et al.'s scheme utilizes each QDCT coefficient in the corresponding 4×4 luma block; thus, each block contributes 16 bits of capacity. Interestingly, Ma et al. did not present an error propagation-free data hiding scheme for Cat_5. Based on six different quantization parameters (QPs), 18, 23, 28, 33, 38, and 43, we ran their algorithm on twenty-six sequences. Table 2 shows the percentage distribution of luma blocks corresponding to each category. In average, we observe that only 46% of the 4×4 luma blocks are error propagation-free because Ma et al.'s scheme does not consider Cat_5. As a result, 54% of the luma blocks are not error propagation-free. This observation motivates us to design a new embedding algorithm that ensures all luma blocks are error propagation-free. The proposed algorithm is described in detail in the next section.

Table 2
The percentage of 4×4 luma blocks in each category.

QP	Cat_1	Cat_2	Cat_3	Cat_4	Cat_5
18	23%	22%	4%	1%	50%
23	23%	23%	4%	1%	49%
28	22%	23%	4%	1%	50%
33	20%	24%	4%	1%	51%
38	17%	22%	3%	1%	57%
43	13%	17%	2%	1%	67%
Average	19%	22%	4%	1%	54%

3. The proposed high capacity data hiding algorithm

In this section, we propose a modified DCT-based perturbation scheme to improve the embedding capacity of Ma et al.'s algorithm. The scheme embeds one extra bit for each luma block belonging to Cat_1, Cat_2, Cat_3, and Cat_4; it embeds one hidden bit in each luma block belonging to Cat_5. This section consists of three subsections, namely, the proposed DCT-based perturbation scheme, the distortion analysis, and the proposed data hiding algorithm.

3.1. The proposed DCT-based perturbation scheme

According to the intra prediction of block $B_{i,j}$ in the encoding stage, the corresponding residual block $R_{i,j}^{p}$ is given by $R_{i,j}^{p} = B_{i,j} - B_{i,j}^{p}$, as mentioned in Section 2. Then, the 4×4 integer DCT and the quantization operation are applied to $R_{i,j}^{p}$ and the QDCT coefficient matrix of $R_{i,j}^{p}$, i.e. $R_{i,j}^{QDCT}$, is derived as follows:

$$R_{i,j}^{QDCT} = (C_f R_{i,j}^{p} C_f^{T}) \otimes \left(\frac{E_f}{Q}\right) = \begin{bmatrix} Y_{00} & Y_{01} & Y_{02} & Y_{03} \\ Y_{10} & Y_{11} & Y_{12} & Y_{13} \\ Y_{20} & Y_{21} & Y_{22} & Y_{23} \\ Y_{30} & Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \quad (1)$$

where $C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$, $E_f = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$, $a = \frac{1}{2}$, $b = \sqrt{\frac{2}{5}}$; the operator "$\otimes$" denotes an element-by-element product of two matrices; and Q is the step size of a quantizer determined by a quantization parameter (QP). In the decoding stage, the dequantization operation and the 4×4 integer inverse DCT are applied to $R_{i,j}^{QDCT}$ to obtain a residual block $R_{i,j}^{r}$ as follows:

$$R_{i,j}^{r} = C_r^{T}(R_{i,j}^{QDCT} \times Q \otimes E_r)C_r \quad (2)$$

where $C_r = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$ and $E_r = \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$.

From $R_{i,j}^{r}$, we derive the decoded luma block $B_{i,j}^{r}$ ($= B_{i,j}^{p} + R_{i,j}^{r}$). Next, we describe the proposed improved data hiding algorithm.

For the category Cat_1, we define the set of vertical coefficient-pairs in $R_{i,j}^{QDCT}$ to be $S_V = \{(Y_{10}, Y_{12}), (Y_{20}, Y_{22}), (Y_{30}, Y_{32}), (Y_{00}, Y_{02})\}$. The first three pairs are used in Ma et al.'s algorithm. We propose the fourth pair (Y_{00}, Y_{02}) to increase the embedding capacity.

Next, we demonstrate the correctness of error propagation-free assurance. Each coefficient-pair in S_V can embed one hidden bit by

perturbing the two corresponding QDCT coefficients. Hence, our scheme has a four-bit embedding capacity for each 4×4 luma block belonging to Cat_1, compared to the three-bit embedding capacity of Ma et al.'s scheme. If we want to embed a hidden bit in (Y_{00}, Y_{02}), we can perturb (Y_{00}, Y_{02}) to $(Y_{00}+1, Y_{02}-1)$ to obtain a perturbed $R_{i,j}^{QDCT}$, say $R_{i,j}^{QDCT'}$. The difference between $R_{i,j}^{QDCT'}$ and $R_{i,j}^{QDCT}$ is calculated as follows:

$$\Delta R_{i,j}^{QDCT} = R_{i,j}^{QDCT'} - R_{i,j}^{QDCT} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

Let $R_{i,j}^{r'}$ denote the perturbed residual block obtained by performing the 4×4 inverse DCT operation defined in Eq. (2) on $R_{i,j}^{QDCT'}$. Following the derivation in Ma et al. (2010), the difference between $R_{i,j}^{r'}$ and $R_{i,j}^{r}$ is calculated by

$$\Delta R_{i,j}^{r} = R_{i,j}^{r'} - R_{i,j}^{r} = C_r^T(\Delta R_{i,j}^{QDCT} \times Q \otimes E_r)C_r = Q \times \begin{bmatrix} 0 & 2a^2 & 2a^2 & 0 \\ 0 & 2a^2 & 2a^2 & 0 \\ 0 & 2a^2 & 2a^2 & 0 \\ 0 & 2a^2 & 2a^2 & 0 \end{bmatrix}. \quad (4)$$

Because $B_{i,j}^{r'} = B_{i,j}^{p} + R_{i,j}^{r} + \Delta R_{i,j}^{r}$ and the rightmost column vector in $\Delta R_{i,j}^{r}$ is a zero vector, the rightmost column vector of $B_{i,j}^{r'}$ will equal to that of $B_{i,j}^{r}$. It implies there is no error propagation since the value of the reference pixels in Cat_1 are not changed. The same rationale applies to the remaining three pairs, (Y_{10}, Y_{12}), (Y_{20}, Y_{22}) and (Y_{30}, Y_{32}) in S_V.

For Cat_2, besides the pairs, (Y_{01}, Y_{21}), (Y_{02}, Y_{22}), and (Y_{03}, Y_{23}) used in Ma et al.'s scheme, we add an extra pair (Y_{00},Y_{20}) to the set of horizontal QDCT coefficient-pairs S_H. Because the matrix $\Delta R_{i,j}^{r}$ in Cat_2 is the transpose of that in Cat_1 (see Eq. (4)), the error propagation-free characteristic is assured. Therefore, we can embed four hidden bits in each block belonging to Cat_2. Meanwhile, Cat_3 does not consider any reference pixels, so we can embed sixteen hidden bits in each 4×4 block. Cat_4 can be handled in the same way as Cat_1 or Cat_2. In our implementation, every block belonging to Cat_4 can be perturbed by the same way as the blocks in Cat_1, that is, one more bit can be embedded.

For Cat_5, we present a new perturbation technique to embed one hidden bit without error propagation. For each 4×4 luma block, the four suggested coefficients in $R_{i,j}^{QDCT}$ are denoted as the set $S_{Quad} = \{(Y_{00},Y_{02},Y_{20},Y_{22})\}$. The perturbation of S_{Quad} is realized by perturbing $(Y_{00},Y_{02},Y_{20},Y_{22})$ to $(Y_{00}+1, Y_{02}-1,Y_{20}-1, Y_{22}+1)$. The error propagation-free property of the perturbation technique used in the fifth category is proved by the following theorem.

Theorem 1. *For Cat_5, we can embed one hidden bit in each 4×4 luma block by perturbing $(Y_{00},Y_{02},Y_{20},Y_{22})$ to $(Y_{00}+1, Y_{02}-1,Y_{20}-1,Y_{22}+1)$.*

Proof. After perturbing $(Y_{00},Y_{02},Y_{20},Y_{22})$ in $R_{i,j}^{QDCT}$ to $(Y_{00}+1, Y_{02}-1,Y_{20}-1,Y_{22}+1)$ in $R_{i,j}^{QDCT'}$, the difference between $R_{i,j}^{QDCT'}$ and $R_{i,j}^{QDCT}$ is given by

$$\Delta R_{i,j}^{QDCT} = R_{i,j}^{QDCT'} - R_{i,j}^{QDCT} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

Then, the difference between the decoded residual block $R_{i,j}^{r}$ and the perturbed residual block $R_{i,j}^{r'}$ is calculated by

$$\Delta R_{i,j}^{r} = R_{i,j}^{r'} - R_{i,j}^{r} = C_r^T(\Delta R_{i,j}^{QDCT} \times Q \otimes E_r)C_r = Q \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4a^2 & 4a^2 & 0 \\ 0 & 4a^2 & 4a^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6)$$

Because $B_{i,j}^{r'} = B_{i,j}^{p} + R_{i,j}^{r'} + \Delta R_{i,j}^{r}$ and the rightmost column vector and the bottom row vector in $\Delta R_{i,j}^{r}$ are zero vectors, the corresponding vectors in the block $B_{i,j}^{r'}$ are equal to those in the reconstructed block $B_{i,j}^{r}$. It indicates that there is no error propagation since the reference pixels have not been changed. □

3.2. Distortion analysis

In this subsection, we analyze the distortion caused by the improved DCT-based perturbation scheme. For Cat_1, we consider the set with four vertical QDCT coefficient-pairs, denoted as $S_V = \{(Y_{00}, Y_{02}), (Y_{10}, Y_{12}), (Y_{20}, Y_{22}), (Y_{30}, Y_{32})\}$. After perturbing the suggested coefficient-pair (Y_{00}, Y_{02}), the resultant distortion is dependent on the difference $\Delta R_{i,j}^{r}$ (see Eq. (4)). We represent $\Delta R_{i,j}^{r}$ as

$$\Delta R_{i,j}^{r} = \begin{bmatrix} \Delta R_{00}^{r} & \Delta R_{01}^{r} & \Delta R_{02}^{r} & \Delta R_{03}^{r} \\ \Delta R_{10}^{r} & \Delta R_{11}^{r} & \Delta R_{12}^{r} & \Delta R_{13}^{r} \\ \Delta R_{20}^{r} & \Delta R_{21}^{r} & \Delta R_{22}^{r} & \Delta R_{23}^{r} \\ \Delta R_{30}^{r} & \Delta R_{31}^{r} & \Delta R_{32}^{r} & \Delta R_{33}^{r} \end{bmatrix}. \quad (7)$$

Based on Parseval's theorem (Papoulis, 1962), we can calculate the distortion of $\Delta R_{i,j}^{r}$ by the sum of square errors (SSE)

$$\begin{aligned} D(Y_{00}, Y_{02}) &= \sum_{m=0}^{3}\sum_{n=0}^{3} \Delta R_{mn}^{r2} = 8 \times (2a^2)^2 \times Q^2 \\ &= 32a^4Q^2 = 2Q^2, \end{aligned} \quad (8)$$

where $a = \frac{1}{2}$ and Q is the step size of a quantizer determined by a quantization parameter (QP). The distortion caused by perturbing the first three coefficient-pairs in S_V, i.e. (Y_{10}, Y_{12}),(Y_{20}, Y_{22}), and (Y_{30}, Y_{32}), can be derived in the same way. For simplicity, we take the pair (Y_{10}, Y_{12}) as an example to analyze the distortion after perturbing (Y_{10}, Y_{12}) to $(Y_{10}+1, Y_{12}-1)$. The difference between $R_{i,j}^{QDCT'}$ and $R_{i,j}^{QDCT}$ can be calculated as follows:

$$\Delta R_{i,j}^{QDCT} = R_{i,j}^{QDCT'} - R_{i,j}^{QDCT} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

Through further derivation, we have

$$\Delta R_{i,j}^{r} = R_{i,j}^{r'} - R_{i,j}^{r} = C_r^T(\Delta R_{i,j}^{QDCT} \times Q \otimes E_r)C_r = Q \times \begin{bmatrix} 0 & 2ab & 2ab & 0 \\ 0 & ab & ab & 0 \\ 0 & -ab & -ab & 0 \\ 0 & -2ab & -2ab & 0 \end{bmatrix}, \quad (10)$$

where $b = \sqrt{\frac{2}{5}}$. The distortion caused by perturbing (Y_{10}, Y_{12}) is calculated as follows:

$$D(Y_{10}, Y_{12}) = \sum_{m=0}^{3}\sum_{n=0}^{3}\Delta R_{mn}^{r2} = 4\times(2ab)^2 + 4\times(ab)^2\times Q^2 = 20\times(ab)^2\times Q^2 = 2Q^2. \quad (11)$$

By the same arguments, we have $D(Y_{20}, Y_{22}) = D(Y_{30}, Y_{32}) = 2Q^2$. Overall, the upper bound of the total distortion with respect to the first category is $8Q^2$ ($=D(Y_{00}, Y_{02}) + D(Y_{10}, Y_{12}) + D(Y_{20}, Y_{22}) + D(Y_{30}, Y_{32})$). According to the above distortion analysis, the upper bound of the total distortion with respect to the second and fourth categories is $8Q^2$. It is known that at most sixteen hidden bits can be embedded into the luma block in the third category, so the total distortion is bounded by $16Q^2$. For the fifth category, perturbing $(Y_{00}, Y_{02}, Y_{20}, Y_{22})$ to $(Y_{00}+1, Y_{02}-1, Y_{20}-1, Y_{22}+1)$ yields the following distortion:

$$D(Y_{00}, Y_{02}, Y_{20}, Y_{22}) = \sum_{m=0}^{3}\sum_{n=0}^{3}\Delta R r2_{mn}^{2} = 4\times(4a^2)^2 = 64t^2a^4Q^2 = 4Q^2. \quad (12)$$

Theorem 2. *The average distortion of the proposed data hiding scheme is bounded by* $6.16Q^2$*, where Q denotes the quantizer step size.*

Proof. From the probability distribution of the five categories shown in Table 2, we have P_{Cat_1}=0.24, P_{Cat_2}=0.19, P_{Cat_3}=0.04, P_{Cat_4}=0.01, and P_{Cat_5}=0.54, where P_{Cat_i} denotes the percentage of 4×4 luma blocks belonging to Cat_i, $1 \le i \le 5$. Therefore, the average distortion of our data hiding scheme is bounded by

$$D_{ours} = 8Q^2\times(P_{Cat_1} + P_{Cat_2} + P_{Cat_4}) + 16Q^2\times P_{Cat_3} + 4Q^2\times P_{Cat_5} = (8Q^2\times0.42 + 16Q^2\times0.04 + 4Q^2\times0.54) = 6.16Q^2. \quad (13)$$

For comparison, we analyze the distortion rate of Ma et al.'s algorithm (Ma et al., 2010). In the algorithm, three hidden bits are embedded in each 4×4 luma block belonging to Cat_1, Cat_2, and Cat_4; and sixteen hidden bits are embedded in one 4×4 luma block belonging to Cat_3. Moreover, if any of the luma blocks is in Cat5, they are discarded. Consequently, the average distortion of Ma et al.'s algorithm for hiding data in H.264/AVC intra frames is bounded by

$$D_{Ma} = [6Q^2\times(P_{Cat_1} + P_{Cat_2} + P_{Cat_4}) + 16Q^2\times P_{Cat_3} + 0\times P_{Cat_5}] = 6Q^2\times0.42 + 16Q^2\times0.04 + 0\times0.54 = 3.16Q^2. \quad (14)$$

□

3.3. The proposed data hiding algorithm

Although the proposed DCT-based perturbation scheme significantly improves the embedding capacity of Ma et al.'s algorithm, the above distortion analysis reveals that the proposed scheme suffers from the quality degradation problem. Besides preserving the high embedding capacity benefit, we still hope to preserve the visual quality in terms of SSIM, the proposed data hiding algorithm is performed in a new set of sifted 4×4 luma blocks and results in less visual quality degradation. Human visual system is much more sensitive to smooth regions than to textural regions, if a MB is encoded by 16×16 intra mode, we do not embed any hidden bits since the MB contains a smooth region; otherwise, the magnitudes of the sixteen QDCT coefficients of each 4×4 luma block are used to sift the textural luma blocks from smooth blocks by

$$\frac{1}{16}\sum_{m=0}^{3}\sum_{n=0}^{3}|Y'_{m,n}| \ge T \quad (15)$$

where $|Y'_{m,n}|$ denotes the absolute value of $Y'_{m,n}$ and T is the threshold of average QDCT coefficient magnitude. When a 4×4 block violates the condition of Eq. (15), the proposed data hiding algorithm discards the block; otherwise, the following embedding process is used to embed hidden bits leading to less visual quality degradation.

Once a luma block satisfies the condition of Eq. (15), we perturb the related QDCT coefficient-pairs in the sixteen 4×4 luma blocks. If the 4×4 target luma block $B_{i,j}$ falls in Cat_1, we check each pair (Y_{mn}, Y_{pq}) in S_V. If (Y_{mn}, Y_{pq}) is a zero coefficient-pair, we do not need to embed any hidden bits; otherwise, the perturbed pair (Y'_{mn}, Y'_{pq}) is calculated by

$$(Y'_{mn}, Y'_{pq}) = \begin{cases} (Y_{mn}+1, Y_{pq}-1), & \text{if}(Y_{mn}\text{ is even}, h=1, \text{ and } Y_{mn}\ge0) \\ & \text{or}\,(Y_{mn}\text{ is odd}, h=0, \text{ and } Y_{mn}\ge0) \\ (Y_{mn}-1, Y_{pq}+1), & \text{if}(Y_{mn}\text{ is even}, h=1, \text{ and } Y_{mn}<0) \\ & \text{or}\,(Y_{mn}\text{ is odd}, h=0, \text{ and } Y_{mn}<0) \\ (Y_{mn}, Y_{pq}), & \text{otherwise.} \end{cases} \quad (16)$$

The value of h is the bit to be embedded. Eq. (16) is called more three times to perturb the remaining three coefficient-pairs in S_V. When $B_{i,j}$ belongs to Cat_2 or Cat_4, Eq. (16) is called four times to embed the four corresponding hidden bits. When $B_{i,j}$ belongs to Cat_3, if a QDCT coefficient Y_{mn} is a zero coefficient, it's unnecessary to embed any hidden bits; otherwise, the perturbed coefficient Y'_{mn} is calculated by

$$Y'_{mn} = \begin{cases} Y_{mn}+1, & \text{if}(Y_{mn}\text{ is even}, h=1, \text{and } Y_{mn}\ge0) \\ & \text{or}\,(Y_{mn}\text{ is odd}, h=0, \text{ and } Y_{mn}\ge0) \\ Y_{mn}-1, & \text{if}(Y_{mn}\text{ is even}, h=1, \text{ and } Y_{mn}<0) \\ & \text{or}\,(Y_{mn}\text{ is odd}, h=0, \text{ and } Y_{mn}<0) \\ Y_{mn}, & \text{otherwise.} \end{cases} \quad (17)$$

Eq. (17) will be called sixteen times to perturb the sixteen QDCT coefficients in the target block. For Cat_5, if S_{Quad} is a zero quad-coefficient, we skip the embedding process of $B_{i,j}$; otherwise, the perturbed quad-coefficient $(Y'_{00}, Y'_{02}, Y'_{20}, Y'_{22})$ is calculated by

$$(Y'_{00}, Y'_{02}, Y'_{20}, Y'_{22}) = \begin{cases} (Y_{00}+1, Y_{02}-1, Y_{20}-1, Y_{22}+1), \\ \quad \text{if}(Y_{00}\text{ is even}, h=1, Y_{00}\ge0, \text{ and } Y_{22}\ge0) \\ \quad \text{or}\,(Y_{00}\text{ is odd}, h=0, Y_{00}\ge0, \text{ and } Y_{22}\ge0) \\ (Y_{00}-1, Y_{02}+1, Y_{20}+1, Y_{22}-1), \\ \quad \text{if}(Y_{00}\text{ is even}, h=1, Y_{00}<0, \text{ and } Y_{22}<0) \\ \quad \text{or}\,(Y_{00}\text{ is odd}, h=0, Y_{00}<0, \text{ and } Y_{22}<0) \\ (Y_{00}, Y_{02}, Y_{20}, Y_{22}), \quad \text{otherwise.} \end{cases} \quad (18)$$

According to the value derived by Eq. (18), one extra bit can be embedded into the luma block. Based on the above embedding process, after embedding hidden bits into suitable luma blocks, the embedded luma blocks still make the condition of Eq. (15) held.

The hidden data extraction process is performed in the H.264/AVC decoding stage. For each decoded 4×4 intra mode, if the condition of Eq. (15) is held, it indicates that some hidden bits

Table 3
Comparison of the embedding capacity of Ma et al.'s algorithm and the proposed algorithm in terms of the number of bits per 4×4 luma block.

QP	Ma et al.	Proposed	Improvement ratio
18	0.97	1.32	36%
23	0.69	0.94	36%
28	0.45	0.60	33%
33	0.24	0.31	29%
38	0.10	0.13	30%
43	0.03	0.04	33%
Average	0.41	0.56	33%

have been embedded in the luma block successfully. If the current decode luma block belongs to Cat_1 or Cat_2, we check each perturbed pair (Y'_{mn}, Y'_{pq}) in S_V or S_H. If it is a non-zero coefficient-pair, the hidden bit can be extracted by

$$\bar{h} = \begin{cases} 1, & \text{if } Y'_{mn} \text{ is odd} \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

The above extraction process is also applicable to Cat_3. If a decoded marked luma block belongs to Cat_4, we check each non-zero coefficient $Y'_{m,n}$ in that block and the hidden bit can be extracted by Eq. (19). In Cat_5, if S_{Quad} is a non-zero quad-coefficient, the hidden bit can be extracted by

$$\bar{h} = \begin{cases} 1, & \text{if } Y'_{00} \text{ is odd} \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

4. Experimental results

We conducted a number of experiments to compare the performance between Ma et al.'s algorithm and the proposed algorithm on the twenty-six test sequences consisting of fourteen QCIF sequences, namely Akiyo, Bridge-Close, Bridge-Far, Carphone, Claire, Coastguard, Container, Foreman, Grandma, Hall, Mobile, Mother–Daughter, News, and Salesman; seven CIF sequences, namely Carphone_1, Coastguard_1, Crew, Foreman_1, Highway, Mobile_1, and Mother–Daughter_1; three 4CIF sequences, namely City, Ice, and Soccer; one 720P sequence, namely, In-to-Tree, and one 1080P sequence, namely Tractor. All implementations were realized on an IBM compatible computer with an Intel Core 2 Due E7400 CPU 2.8 GHz and 2 GB RAM. The operating system was Microsoft Windows 7; the program development environment was Visual C++2005, and the implementation platform was JM 16.0. The size of GOP was chosen to be 30 and its structure was set to IBPBP. Six different QPs, 18, 23, 28, 33, 38, and 43, were considered by the encoder side for the above twenty-six sequences.

Table 3 compares the embedding capacity performance in terms of the number of bits per 4×4 luma block. We found that the average capacity improvement ratio of the proposed algorithm over Ma et al.'s algorithm was 33%. Table 4 shows the embedding capacity of these test sequences when QP is set to 28. It is observed that the embedding capacity improvement ratio is ranged from 27% to 46% since the selected prediction modes of 4×4 luma blocks are dependent on the content of test sequences. Table 5 illustrates the bitrate increment rate for different QPs. The average bitrate increment rates of Ma et al.'s algorithm and the proposed algorithm were 1.44% and 1.68%, respectively, and it indicates that the bitrate degradation is rather small. For saving space of the context, we omit the bitrate increment ratios of both algorithms for each test sequence.

The quality comparison for both algorithms is illustrated in Table 6. Besides using PSNR to measure the quality of the embedded video frames, the well-known SSIM index is adopted to measure

Table 4
Embedding capacity comparison between Ma et al.'s algorithm and the proposed algorithm for QP = 28.

Sequence	Ma et al.	Proposed	Improvement ratio
Akiyo	0.36	0.47	32%
Bridge-Close	0.58	0.76	31%
Bridge-Far	0.14	0.20	43%
Carphone	0.44	0.59	36%
Clair	0.19	0.26	37%
Coastguard	1.00	1.27	27%
Container	0.49	0.65	31%
Foreman	0.38	0.56	46%
Grandma	0.30	0.43	42%
Hall	0.55	0.74	33%
Mobile	1.16	1.59	37%
Mother–Daughter	0.27	0.36	35%
News	0.56	0.74	31%
Salesman	0.55	0.76	38%
Carphone_1	0.32	0.43	37%
Coastguard_1	0.96	1.25	30%
Crew	0.28	0.37	31%
Foreman_1	0.23	0.30	31%
Highway	0.31	0.43	39%
Mobile_1	0.11	0.14	34%
Mother–Daughter_1	1.08	1.44	34%
City	0.16	0.22	37%
Ice	0.68	0.89	30%
Soccer	0.12	0.16	34%
In-to-Tree	0.34	0.45	32%
Tractor	0.26	0.37	41%
Average	0.45	0.60	35%

Table 5
Bitrate increment rate comparison between Ma et al.'s algorithm and the proposed algorithm.

QP	Ma et al.	Proposed
18	0.24 %	0.65%
23	0.82 %	1.22%
28	1.71 %	2.06%
33	2.29 %	2.53%
38	2.19 %	2.28%
43	1.39 %	1.33%
Average	1.44 %	1.68%

the visual quality. In our experiments, the SSIM index (Wang et al., 2004) between the original and embedded frames I_X and I_Y is calculated by

$$\text{SSIM}(I_X, I_Y) = \frac{1}{N_B - 1} \sum_{i=0}^{N_B - 1} \frac{(2\mu_{B_{X_i}}\mu_{B_{Y_i}} + c_1) \times (2\sigma_{B_{X_i}B_{Y_i}} + c_2)}{(\mu^2_{B_{X_i}} + \mu^2_{B_{Y_i}} + c_1)(\sigma^2_{B_{X_i}} + \sigma^2_{B_{Y_i}} + c_2)} \quad (21)$$

where B_{X_i} and B_{Y_i} denote i-th 4×4 luma block in I_X and I_Y, respectively, and both two frames can be partitioned into N_B 4×4 luma blocks; $\mu_{B_{X_i}}$ ($\mu_{B_{Y_i}}$) and $\sigma_{B^2_{X_i}}$ ($\sigma^2_{B_{Y_i}}$) denotes the mean and variance of

Table 6
PSNR and SSIM comparison between Ma et al.'s algorithm and the proposed algorithm.

QP	Ma et al.		Proposed	
	PSNR	SSIM	PSNR	SSIM
18	42.66	0.994	42.21	0.993
23	39.22	0.987	38.74	0.986
28	36.05	0.976	35.62	0.974
33	32.94	0.956	32.65	0.955
38	30.07	0.924	29.91	0.923
43	27.53	0.878	27.49	0.878
Average	34.75	0.953	34.44	0.952

B_X (B_Y), respectively; $\sigma_{B_{X_i}B_{Y_i}}$ is the covariance of B_X and B_Y; c_1 and c_2 are defined by

$$c1 = (k_1 \times L)^2$$
$$c2 = (k_2 \times L)^2$$

where $L = 255$, $k_1 = 0.01$, and $k_2 = 0.03$.

When compared to Ma et al.'s algorithm, although the PSNR degradation of the proposed algorithm is 0.31 dBs (=34.75–34.44 dBs), the SSIM index is similar to that of Ma et al.'s algorithm and the proposed algorithm can improve 35% embedding capacity. The similar SSIM indexes of two concerned algorithms imply that the proposed algorithm improve the embedding capacity of Ma et al.'s algorithm without visual quality degradation. For QP = 28, when compared to Ma et al.'s algorithm, Table 7 demonstrates that PSNR degradation of the proposed algorithm is ranged from 0.06 dBs to 0.68 dBs, but the SSIM index is almost the same each other.

Figs. 4–9 are six image sets namely, Bridge-Close, Coastguard, Container, Crew, City, and Soccer sequences, which are used to evaluate the visual quality performance. In the first image set shown in Fig. 4, we compare the visual effect between Fig. 4(a) and each of Fig. 4(b)–(c), and observe that both concerned algorithms slightly degrade the visual quality when compared to H.264. The remaining five image sets shown in Figs. 5–9 also reveal the same visual quality observation. The decoding time performance of the concerned two algorithms is shown in Table 8 and it indicates that in average, each algorithm requires about 18.8 ms to decode each frame from the compressed video sequence. Table 8 also demonstrates that the decoding time performance of each algorithm satisfies real-time requirement. For example, in the popular NTSC and PAL standards, 30 and 40 ms are required for encoding each image frame.

Besides Ma et al.'s algorithm, we also implement Huo et al.'s algorithm (Huo et al., 2011), which mainly improves the quality of Ma et al.'s algorithm. Table 9 shows the concerned four performances of Huo et al.'s algorithm. When compared to Tables 3, 5, 6, and 8, Table 9 indicates that Huo et al.'s algorithm has smaller bitrate overhead, higher PSNR, and similar SSIM index when

Table 7
PSNR and SSIM comparison between Ma et al.'s algorithm and the proposed algorithm for QP = 28.

Sequence	Ma et al.		Proposed	
	PSNR	SSIM	PSNR	SSIM
Akiyo	37.32	0.990	36.77	0.989
Bridge-Close	34.01	0.965	33.55	0.963
Bridge-Far	37.58	0.965	37.20	0.964
Carphone	36.05	0.992	35.57	0.991
Clair	39.49	0.994	39.02	0.994
Coastguard	33.45	0.935	32.96	0.931
Container	35.25	0.958	34.72	0.957
Foreman	35.73	0.989	35.19	0.987
Grandma	36.17	0.982	35.67	0.980
Hall	35.70	0.990	35.04	0.988
Mobile	32.35	0.981	31.67	0.979
Mother–Daughter	36.47	0.985	36.41	0.985
News	35.74	0.988	35.12	0.987
Salesman	34.67	0.976	34.08	0.974
Carphone_1	36.69	0.992	36.36	0.991
Coastguard_1	33.77	0.944	33.24	0.940
Crew	36.98	0.980	36.75	0.980
Foreman_1	36.14	0.989	35.81	0.988
Highway	38.08	0.968	37.91	0.968
Mobile_1	33.07	0.986	32.43	0.984
Mother–Daughter_1	38.62	0.990	38.23	0.990
City	34.46	0.972	33.98	0.970
Ice	40.05	0.985	39.76	0.984
Soccer	36.53	0.963	36.23	0.961
In-to-Tree	34.91	0.939	34.59	0.937
Tractor	38.12	0.985	37.84	0.984
Average	36.05	0.976	35.62	0.975

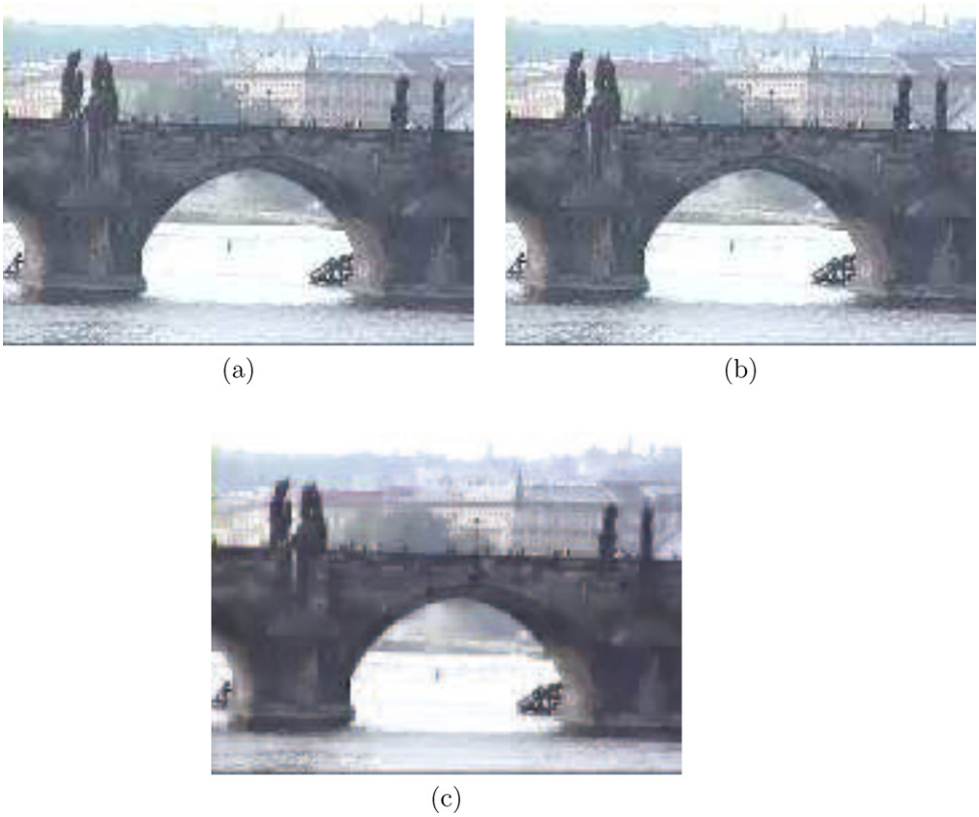

Fig. 4. Visual effect of the resultant intra frames in Bridge-Close sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

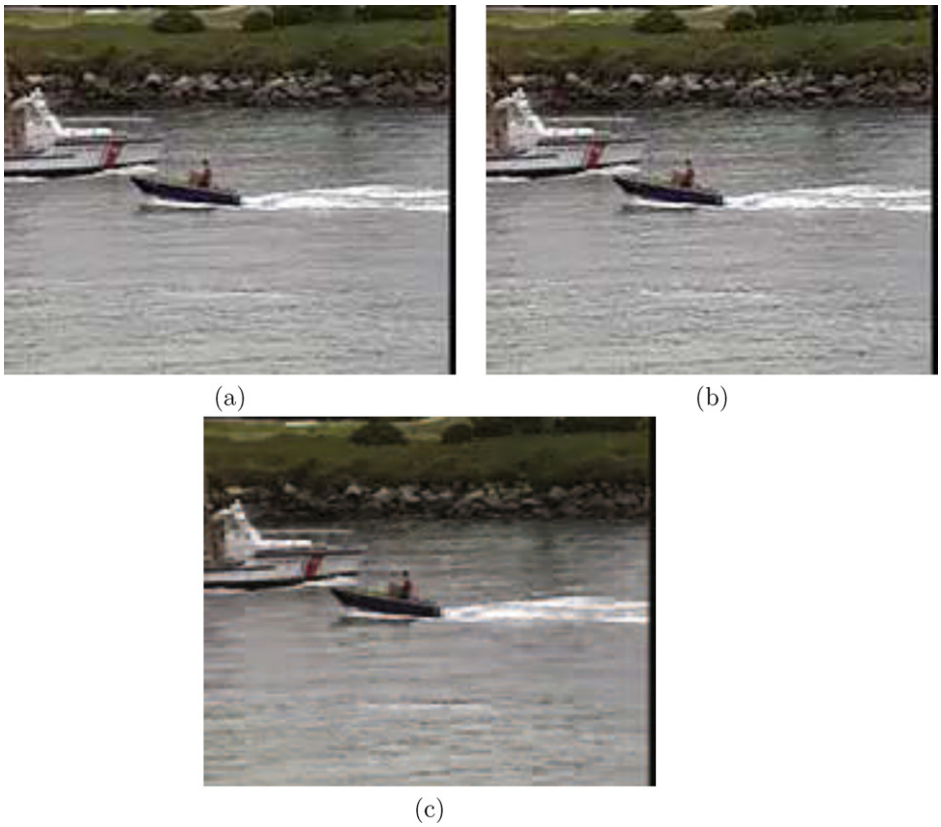

Fig. 5. Visual effect of the resultant intra frames in Coastguard sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

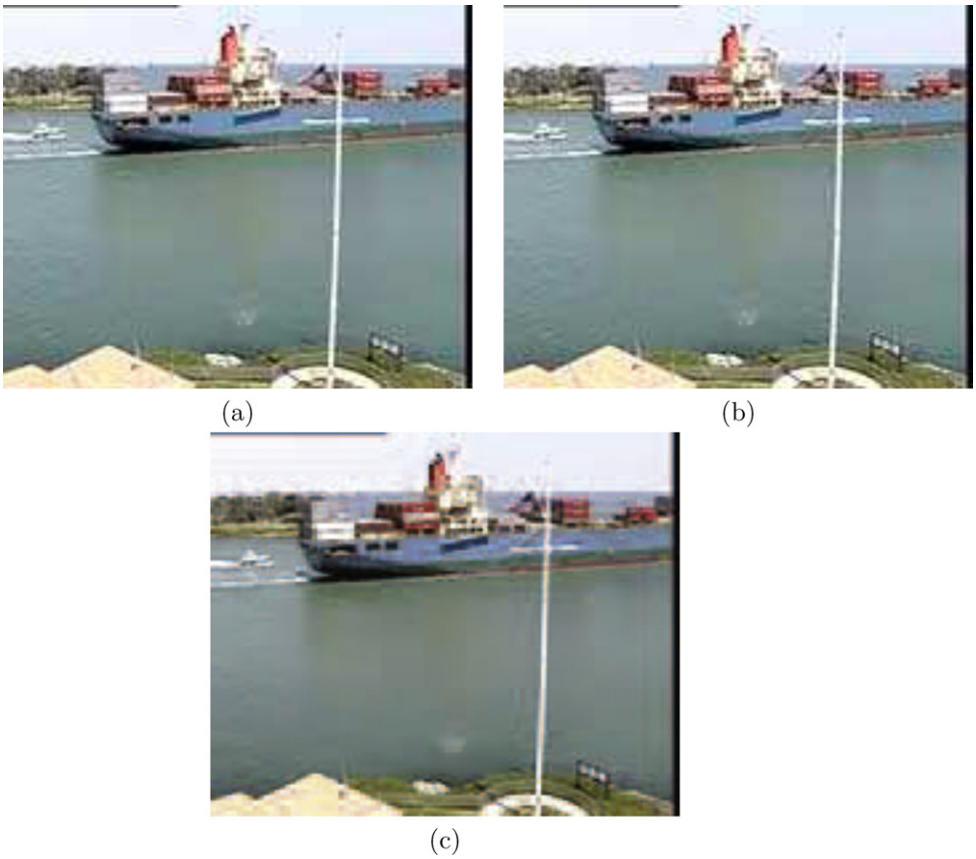

Fig. 6. Visual effect of the resultant intra frames in Container sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

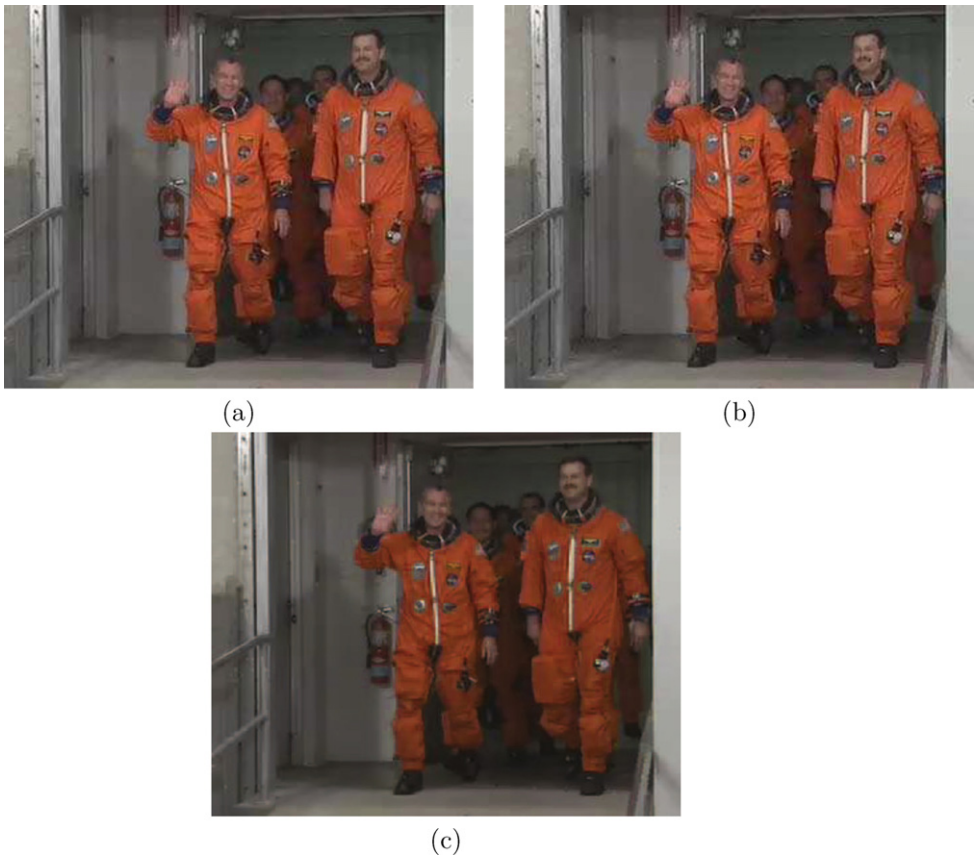

Fig. 7. Visual effect of the resultant intra frames in Crew sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

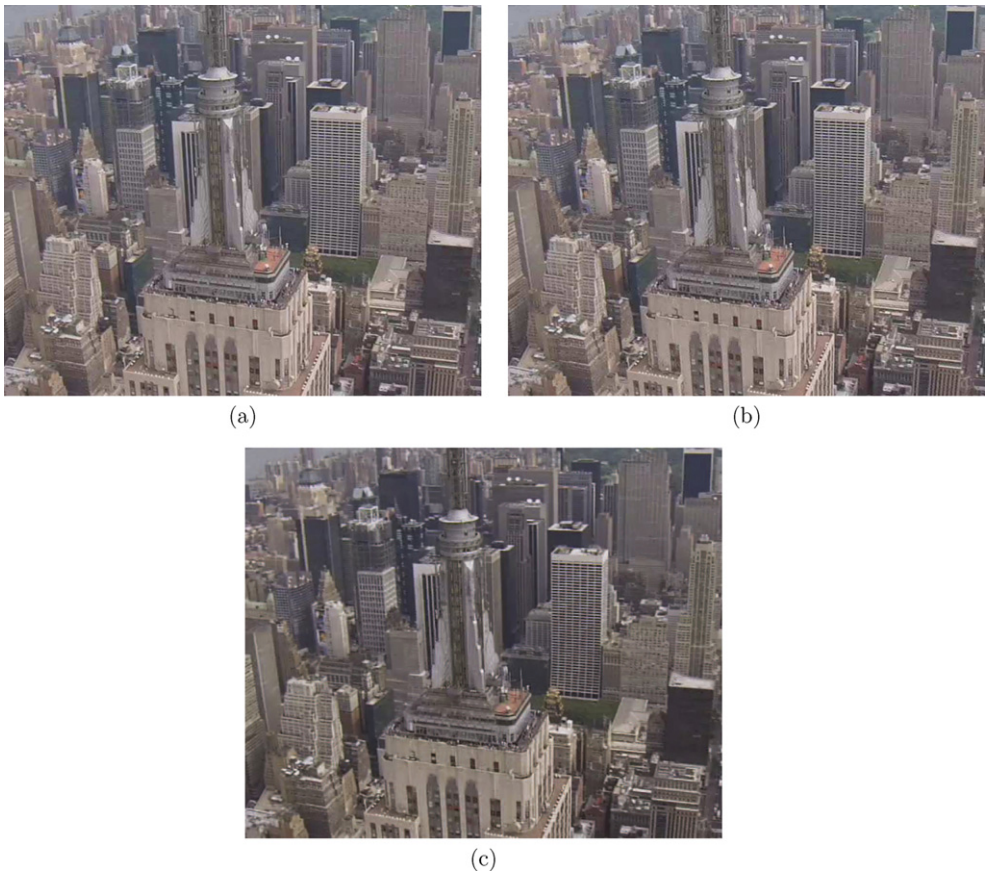

Fig. 8. Visual effect of the resultant intra frames in City sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

Fig. 9. Visual effect of the resultant intra frames in Soccer sequence by (a) H.264/AVC, (b) Ma et al.'s algorithm, and (c) the proposed algorithm.

compared to Ma et al.'s algorithm and our proposed algorithm, but the embedding capacity degradation is 41% and 93% of Ma et al.'s algorithm and the proposed algorithm, respectively. In summary, Ma et al.'s algorithm and Huo et al.'s algorithm are suitable for embedding applications with low and medium embedding capacity. However, the proposed algorithm allows the applications with higher embedding capacity; for example, embed more encoding information to assist error concealment process in H.264 video sequences and hope to keep the similar visual quality as in H.264.

Table 8
Decoding time comparison between Ma et al.'s algorithm and the proposed algorithm in terms of milliseconds per frame.

QP	Ma et al.	Proposed
18	27.814	27.978
23	21.492	21.877
28	17.962	18.331
33	15.924	16.167
38	14.886	15.028
43	14.181	14.237
Average	18.710	18.936

Table 9
Performance of Huo et al.'s algorithm.

QP	Embedding capacity	Bitrate increment rate	PSNR	SSIM	Decoding time
18	0.61	0.51%	42.84	0.994	28.256
23	0.54	0.98%	39.44	0.988	21.956
28	0.30	1.69%	36.13	0.979	18.745
33	0.14	2.18%	33.01	0.960	16.461
38	0.11	2.08%	30.26	0.925	15.478
43	0.04	1.29%	27.60	0.878	14.723
Average	0.29	1.45%	34.88	0.954	19.270

5. Conclusion

We have presented a new data hiding algorithm to resolve the capacity limitation problem in Ma et al.'s data hiding algorithm for H.264/AVC intra frames. The contribution of this work is threefold. First, unlike Ma et al.'s algorithm, the proposed scheme exploits the fifth block category, which accounts for 54% of all luma blocks. Further, for preserving the visual quality and increasing the embedding capacity, a new set of sifted 4×4 luma blocks is considered in the proposed DCT-based perturbation scheme. Based on twenty-six test video sequences, experimental results confirm the embedding capacity superiority of the proposed improved algorithm while keeping the similar human visual effect in terms of SSIM index.

References

Chen, M.J., Chen, C.S., Chi, M.C., 2005. Temporal error concealment algorithm by recursive block-matching principle. IEEE Transactions Circuits and Systems for Video Technology 15 (November (11)), 1385–1393.

Gong, X., Lu, H.M., 2008. Towards fast and robust watermarking scheme for H.264 video. In: Proc. IEEE Int. Symp. Multimedia, Los Alamitos, CA, USA, December, pp. 649–653.

Huo, W., Zhu, Y., Chen, H., 2011. A controllable error-drift elimination scheme for watermarking algorithm in H.264/AVC stream. IEEE Signal Processing Letters 18 (September (9)), 535–538.

Kapotas, S., Varsaki, E., Skodras, A., 2007. Data hiding in H.264 encoded video sequences. In: Proc. IEEE Workshop on Multimedia Signal Processing, Chania, Greece, October, pp. 373–376.

Kim, S., Hong, Y., Won, C., 2007 Data hiding on H.264/AVC compressed video. In: Proc. LNCS Int. Conf. Image Analysis and Recognition, vol. 4633, August, pp. 698–707.
Ma, X.J., Li, Z.T., Tu, H., Zhang, B., 2010. A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift. IEEE Transactions of Circuits and Systems for Video Technology 20 (October (10)), 1320–1330.
Nguyen, C.V., Tay, D., Deng, G., 2006. A fast watermarking system for H.264/AVC video. In: Proc. IEEE Asia Pacific Conf. Circuits and Systems, Singapore, December, pp. 81–84.
Noorkami, M., Mersereau, R., 2005. Compressed-domain video watermarking for H.264. In: Proc. IEEE Int. Conf. Image Processing (ICIP 2005), vol. 2, Los Alamitos, September, pp. 11–14.
Noorkami, M., Mersereau, R.M., 2007. A framework for robust watermarking of H.264-encoded video with controllable detection performance. IEEE Transactions of the Information Forensics and Security 2 (February), 14–23.
Papoulis, A., 1962. The Fourier Integral and its Applications. McGraw-Hill, New York (Chapter 2).
Qiu, G., Marziliano, P., Ho, A.T., He, D., Sun, Q., 2004. A hybrid watermarking scheme for H.264/AVC video. In: Proc. IEEE ICPR, vol. 4, August, pp. 865–868.
Richardson, I.E.G., 2003. H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. John Wiley & Sons Ltd, New Jersey.
Wong, K., Tanaka, K., Qi, X., 2006. Multiple messages embedding using DCT-based Mod4 steganographic method. In: Proc. LNCS Int. Workshop Multimedia Content Representation, Classification, vol. 4105, September, pp. 57–65.
Wong, K.S., Tanaka, K., Takagi, K., Nakajima, Y., 2009. Complete video quality-preserving data hiding. IEEE Transactions of the Circuits and Systems for Video Technology 19 (October (10)), 1499–1512.
Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13 (April (4)), 600–612.
Zhang, J., Ho, A.T.S., Qiu, G., 2007. Robust video watermarking of H.264/AVC. IEEE Transactions of the Circuits and Systems II: Express Briefs 54 (February), 205–209.

Tseng-Jung Lin received the BS degree in Computer Science and Education from National Taiwan Normal University, Taipei, Taiwan, in 1991 and the MS degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2008. He is currently pursuing the PhD degree in Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include image/video compression, image processing, image segmentation, classification, and reversible data hiding.

Kuo-Liang Chung is a Professor in the Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, since 1995. He received the BS, MS, and PhD degrees from National Taiwan University, Taipei, Taiwan, in 1982, 1984, and 1990, respectively. After two years of obligatory military services (1984–1986), he was enrolled as a research assistant in the Institute of Information Science at Academia Sinica, Taiwan from 1986 to 1987. He was a Visiting Scholar at University of Washington in summer 1999. From 2003 to 2006, he served as the chair of the Department of Computer Science and Information Engineering at NTUST. He was a Managing Editor of Journal of Chinese Institute of Engineers from Jan. 1996 to Dec. 1998. In August 2000, he served as Program Co-Chair of 2000 Conference on Computer Vision, Graphics, and Image Processing, Taiwan. Dr. Chung was the recipient of the Distinguished Research Award from National Science Council of Taiwan in 2004; the recipient of the Best Paper Award from the Image Processing and Pattern Recognition Society of Taiwan in 2007; the recipient of Distinguished Teaching Award of NTUST in 2009. Since 2009, he has been a University Chair Professor at NTUST. His current research interests include image/video compression, image/video processing, pattern recognition, 3D video processing, and shape analysis in computer vision. He now serves as an Associate Editor of Journal of Visual Communication and Image Representation.

Dr. Chung is a senior member of IEEE and a fellow of IET.

Po-Chun Chang received the BS degree in Department of Computer Science and Information Engineering, St. John's University/Taipei Campus, New Taipei City, Taiwan, in 2009. He is currently pursuing the PhD degree in Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include image/video compression, image processing, and 3D video processing.

Yong-Huai Huang received the BS degree in Information Management from Aletheia University, Danshui, Taipei, Taiwan, and the MS and PhD degrees in Computer Science and Information Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan. He is now an assistant professor in the Institute of Computer and Communication Engineering at Jinwen University of Science and Technology, Hsin-Tien Dist., New Taipei City, Taiwan. His research interests include image processing and compression, and algorithms.

Hong-Yuan Mark Liao received a BS degree in physics from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981, and an MS and PhD degree in electrical engineering from Northwestern University in 1985 and 1990, respectively. In July 1991, he joined the Institute of Information Science, Academia Sinica, Taiwan. He is a Distinguished Research Fellow now. During 2009–2011, he was the division chair of the computer science and information engineering division II, National Science Council of Taiwan. He is jointly appointed as a professor of the Computer Science and Information Engineering Department of National Chiao-Tung University and the Department of Electrical Engineering of National Cheng Kung University. During 2009–2012, he was jointly appointed as the Multimedia Information Chair Professor of National Chung Hsing University. From August 2010, he has been appointed as an Adjunct Chair Professor of Chung Yuan Christian University. His current research interests include multimedia signal processing, video-based Surveillance Systems, video forensics, and multimedia protection.

Dr. Liao was a recipient of the Young Investigators' award from Academia Sinica in 1998. He received the distinguished research award from the National Science Council of Taiwan in 2003 and 2010, and the National Invention Award of Taiwan in 2004. In 2008, he received a Distinguished Scholar Research Project Award from National Science Council of Taiwan. In 2010, he received the Academia Sinica Investigator Award. In June 2004, he served as the conference co-chair of the 5th International Conference on Multimedia and Exposition (ICME) and technical co-chair of the 8th ICME held at Beijing. In Jan. 2011, Dr. Liao served as General co-chair of the 17th International Conference on Multimedia Modeling. From 2006–2008, Dr. Liao was the president of the Image Processing and Pattern Recognition Society of Taiwan.

Dr. Liao is on the editorial boards of the IEEE Signal Processing Magazine, the IEEE Transactions on Image Processing, and the IEEE Transactions on Information Forensics and Security. He was an associate editor of IEEE Transactions on Multimedia during 1998–2001.

Chiung-Yao Fang received the BSc Degree and the MSc degree in information and computer education from National Taiwan Normal University, Taipei, Taiwan, R.O.C., in 1992 and 1994, respectively, and PhD degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2003. She is currently an Associate Professor of the Department of Computer Science and Information engineering at the National Taiwan Normal University, Taipei, Taiwan, R.O.C. Her areas of research interest include vision-based driver assistance systems, vision-based infant monitoring systems, artificial neural networks, pattern recognition, and computer vision.