An Error Propagation Free Data Hiding Algorithm in HEVC Intra-Coded Frames

Po-Chun Chang*, Kuo-Liang Chung*, Jiann-Jone Chen†, Chien-Hsiung Lin* and Tseng-Jung Lin*

*Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
E-mail: d9915014@mail.ntust.edu.tw; klchung01@gmail.com; d9409301@mail.ntust.edu.tw; tutankhamen1560bc@hotmail.com
†Department of Electrical Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
E-mail: jjchen@mail.ntust.edu.tw

***Abstract*—Efficient data hiding algorithms have been developed for video coders such as MPEG-4 and H.264/AVC, to deliver embedded information. Lin *et al.* proposed an error propagation free discrete cosine transform (DCT) based data hiding algorithm in H.264/AVC intra-coded frames. However, the state-of-the-art video codec, high efficiency video coding (HEVC), adopts both DCT and discrete sine transform (DST) such that the previous DCT based data hiding algorithms cannot afford to fully utilize available capacity for data hiding under the HEVC framework. We proposed to investigate the block DCT and DST coefficient characteristics to specify the transformed coefficients that can be perturbed without propagating errors to neighboring blocks. Experiments on four different complexity test videos justified the efficiency of the proposed algorithm in performing intra-frame error propagation free data hiding, providing higher embedding capacity in low bitrate coding, and yielding better reconstructed video quality.**

***Index Terms*—Data hiding, DCT/DST, H.264/AVC, HEVC, Intra-frame error propagation free.**

I. Introduction

With the advance of digital multimedia communication techniques, data hiding becomes important in the authentication, identification, annotation, and copyright protection of digital media items, among which digital video is most widely used. However, due to the massive information amount of digital video signals, it needs highly efficient video coding to make video communication feasible. Recently, several transform domain based data hiding algorithms [1], [2], [3], [4], [5], [6], [7], [8] have been developed in H.264/AVC [9], where the quantized discrete cosine transform coefficients (QDCTs) of the 4×4 luma blocks in intra-coded frames (IFs) are perturbed so as to carry the hidden data. Unfortunately, these algorithms have the intra-frame error propagation problem due to the improper perturbation strategies for the QDCTs, leading to severe quality degradation for the reconstructed video signals.

To solve this problem, Ma *et al.* [10] proposed an effective perturbation strategy for the QDCTs of the 4×4 luma blocks based on both the directions of intra prediction modes and the deviation in the pixel values resulted from changing the QDCTs values. Lin *et al.* [11] proposed to increase the embedding capacity for the 4×4 luma blocks achieved by Ma *et al.* and to additionally utilize the other 4×4 luma blocks to embed one-bit data. Although the intra-frame error propagation has been tackled in the algorithms [10] and [11], the inter-frame error propagation is another problem for these transform domain based data hiding algorithms and should be addressed.

As the application demands higher video signal resolution, the H.264/AVC standard can no longer achieve satisfied compression performance. A state-of-the-art video coding standard, high efficiency video coding (HEVC) [12], is developed to improve the compression efficiency for high resolution video signals. When considering data hiding in HEVC video sequences, one intuitive way is to adapt the previous data hiding algorithms used for H.264/AVC to the HEVC standard. However, since the HEVC 4×4 block does not utilize DCT for transform coding, the transform domain data hiding algorithms developed based on H.264/AVC cannot be applied to HEVC. Hence, developing a data hiding algorithm that can support the transform coding adopted by HEVC while eliminating the intra-frame error propagation and alleviating the inter-frame error propagation is required, which motivates this research.

In this paper, we proposed to categorize blocks so as to impose specific quantized coefficient perturbation patterns for intra-frame error propagation free data hiding in HEVC, denoted as EMBED. For HEVC IFs, we propose a DCT and discrete sine transform (DST) based coefficient perturbation scheme for embedding bits. The error propagation patterns corresponding to different HEVC intra prediction modes are first classified for imposing different data hiding patterns. These data hiding patterns, which correspond to specific QDCT and QDST coefficient perturbations, are used to embed bits in HEVC IFs without inducing intra-frame error propagation. In addition, one quality improvement scheme is proposed for the reconstructed IFs and meanwhile alleviates the quality degradation due to inter-frame error propagation. Experiments on four test videos confirm the capability of the proposed data hiding algorithm in providing intra-frame error propagation free, higher embedding capacity in low bitrate coding, and reconstructed video quality. To the best of our knowledge, this is the first time that such a DCT/DST based data hiding algorithm is presented for HEVC.

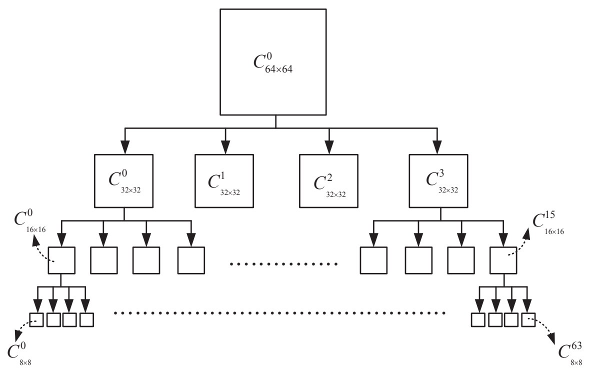


Fig. 1. The quadtree decompositions for one 64×64 coding unit.

The rest of this paper is organized as follows. In Section II, the HEVC intra coding scheme and related block transform coding operations are reviewed. The proposed DCT/DST based data hiding algorithm in HEVC and its theoretical analysis are presented in Section III. Section IV demonstrates experimental results and performance evaluation. Section V concludes this paper.

II. Intra Coding Scheme and the Related Transform Coding in HEVC

The HEVC intra coding scheme, including block transform coding operations, will be first reviewed, with which the proposed data hiding algorithm can be clearly described.

A. *Intra Coding Scheme in HEVC*

When compressing the video sequences by the HEVC encoder, coding unit (CU) is the basic unit to perform coding for each video frame. One video frame is first split into non-overlapping CUs with 64×64 pixels, each of which is further split into smaller ones with a recursive quadtree decomposition, as shown in Fig. 1, where $C^0_{N\times N}, C^1_{N\times N}, \ldots, C^{(64/N)^2-1}_{N\times N}$ denote the $(64/N)^2$ CUs of size $N \times N$ for $N = 64, 32, 16, 8$.

To exploit spatial redundancy for compression, the intra prediction is performed among adjacent blocks on one video frame in HEVC. For one $N \times N$ CU with $N \in \{64, 32, 16\}$, there exists only one prediction unit (PU) of size $N \times N$, whereas two PU sizes, 8×8 and 4×4, are available when $N = 8$. In addition to different PU sizes, the HEVC supports thirty-five intra prediction modes, as shown in Fig. 2, while only nine modes are specified in H.264/AVC.

For one $N \times N$ PU, the HEVC encoder predicts thirty-five blocks from corresponding neighboring pixels and intra prediction modes. These reference pixels for intra prediction are encoded pixels from upper-right, upper, upper-left, left and lower-left neighboring blocks, as shown in Fig. 3(a). Note that in Fig. 3(b), the reference pixels $\{s_{N+i,0}\}_{i=1,\cdots,N}$ from the lower left region are unavailable since they are not yet encoded by HEVC at the time when encoding the current PU. Under this condition, these unavailable pixels $\{s_{N+i,0}\}_{i=1,\cdots,N}$ will be set to equal to the nearest encoded reference pixel $s_{N,0}$

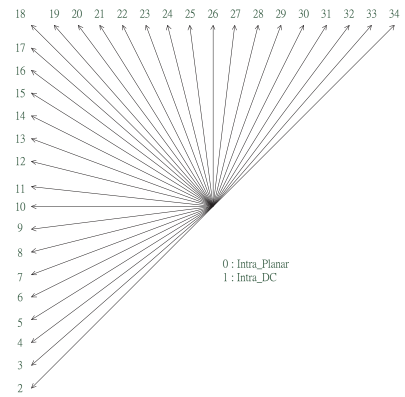
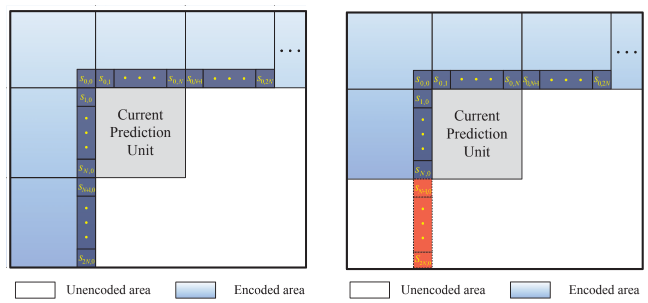


Fig. 2. The thirty-five intra prediction modes supported by HEVC.



(a) (b)

Fig. 3. Reference pixel locations of the HEVC intra prediction scheme: (a) All reference pixels are available; (b) All but part reference pixels are available.

to act as reference pixels. For each intra prediction mode, to calculate the RD cost on the corresponding residual signal, coding operations such as transform, quantization, and entropy coding have to be carried out. The intra prediction mode that yields the minimum RD cost is selected as the optimal one to predict and encode the current PU.

Once the optimal intra prediction mode for each PU is determined, the HEVC updates the RD cost associated with each CU in Fig. 1. The hierarchical structure corresponding to the CU, $C^0_{64\times64}$, in Fig. 1 will be pruned and merged according to the RD optimization procedure and hence the optimal intra coding partition corresponding to $C^0_{64\times64}$ can be obtained.

B. *Transform Coding Related to the Luma Intra Prediction*

To obtain the RD cost on the residual signal, the HEVC specifies the transform unit (TU) for transform and quantization coding of the prediction residual. In the HEVC intra coding, for one $N \times N$ PU with $N \in \{32, 16, 8, 4\}$, there exists only one TU of size $N \times N$, whereas four 32×32 TUs are used when $N = 64$. In transform coding of the luma prediction residual, the HEVC provides two kinds of transforms: the integer DST and the integer DCT. The former is used for 4×4 TUs while the latter for $N \times N$ TUs with $N = 32, 16, 8$. For simplicity, only the integer DST to 4×4 TUs and the integer DCT to 8×8 TUs are presented.

Let $R^p_{N\times N}$ denote the luma prediction residual within the $N \times N$ TU for $N = 32, 16, 8, 4$. When performing the DST on

$R^p_{4\times4}$, the QDST coefficient matrix of $R^p_{4\times4}$ can be expressed as

$$\begin{aligned} R^{QDST}_{4\times4} &= (S_f R^p_{4\times4} S_f^T) \times (1/Q) \\ &\overset{\text{def}}{=} \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}, \end{aligned} \tag{1}$$

where Q is the quantizer step size determined by a quantization parameter (QP) and

$$S_f = \begin{bmatrix} A & B & C & D \\ C & C & 0 & -C \\ D & -A & -C & B \\ B & -D & C & -A \end{bmatrix}$$

with $A = 0.22656$, $B = 0.42969$, $C = 0.57813$, and $D = 0.65625$. At the decoding stage, after performing inverse QDST (IQDST) on $R^{QDST}_{4\times4}$, the reconstructed residual can be represented as

$$\begin{aligned} R^r_{4\times4} &= IQDST(R^{QDST}_{4\times4}) \\ &= S_f^{-1}(R^{QDST}_{4\times4} \times Q)(S_f^T)^{-1}, \end{aligned} \tag{2}$$

where S_f^{-1} denotes the inverse matrix of S_f. The decoded luma PU can then be derived by adding the reconstructed residual to the corresponding prediction block.

Similarly, when performing DCT on $R^p_{8\times8}$, the corresponding QDCT coefficient matrix $R^{QDCT}_{8\times8}$ can be expressed as

$$R^{QDCT}_{8\times8} = (C_f R^p_{8\times8} C_f^T) \times (1/Q), \tag{3}$$

where C_f is the DCT matrix. At the decoder, with inverse QDCT (IQDCT) on $R^{QDCT}_{8\times8}$, the reconstructed residual can be represented as

$$\begin{aligned} R^r_{8\times8} &= IQDCT(R^{QDCT}_{8\times8}) \\ &= C_f^{-1}(R^{QDCT}_{8\times8} \times Q)(C_f^T)^{-1} \end{aligned} \tag{4}$$

and the decoded luma PU can be obtained in the same way.

III. The Proposed Data Hiding Algorithm

A. Intra-Frame Error Propagation Patterns

Since the HEVC intra prediction scheme utilizes spatial correlation among image pixels for compression, the current PU may refer to the neighboring pixels $\{s_{0,j}\}_{j=0,1,\cdots,2N}$ and $\{s_{i,0}\}_{i=1,\cdots,2N}$ collected from upper-right, upper, upper-left, left and lower-left blocks to perform intra prediction. In other words, errors on the pixels $\{R_{i,N}\}_{i=1,\cdots,N}$ and $\{R_{N,j}\}_{j=1,\cdots,N-1}$ of the current PU may be propagated, via intra prediction, to neighboring upper-right, right, lower-right, lower, and lower-left blocks, as shown in Fig. 4.

The directions of the intra-frame error propagation can be categorized into three patterns. The first pattern is defined as horizontal error propagation, which means that the errors on referred pixels $\{R_{i,N}\}_{i=1,\cdots,N}$ of the current PU would be propagated to the neighboring right and upper-right blocks. Under this condition, the intra prediction modes of the right and upper-right blocks will be within the mode numbers $\{0, 1-25\}$ and $\{0, 2-10\}$, respectively. The second one is vertical error propagation, where the errors on referred pixels $\{R_{N,j}\}_{j=1,\cdots,N}$ of the current PU would be propagated to the lower and the lower-left blocks. Under this condition, the intra prediction modes of the lower and lower-left blocks will be within the mode numbers $\{0, 1, 11-34\}$ and $\{0, 26-34\}$, respectively. The third one is diagonal error propagation, i.e., the error propagation of the referred pixel, $R_{N,N}$, of the current PU to the lower-right neighboring block, under which the prediction mode numbers will be within the set $\{0, 1, 11-25\}$.

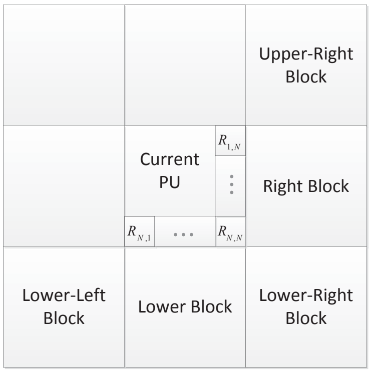


Fig. 4. The image pixels of the current PU used for intra prediction of adjacent unencoded blocks.

TABLE I
TRUTH TABLE OF ERROR PROPAGATION RELATIONSHIPS IN HEVC

Case	Error Propagation Pattern			Protected Pixel Set
	Horizontal	Vertical	Diagonal	
1	T	F	T or F	$\{R_{i,N}\}_{i=1,2,\cdots,N}$
2	F	T	T or F	$\{R_{N,j}\}_{j=1,2,\cdots,N}$
3	F	F	T	$\{R_{N,N}\}$
4	F	F	F	None
5	T	T	T or F	$\{R_{i,N}\}_{i=1,2,\cdots,N} \cup \{R_{N,j}\}_{j=1,2,\cdots,N-1}$

The occurrence of the three error propagation patterns for each $N\times N$ PU in IFs with $N = 64, 32, 16, 8, 4$ can be demonstrated by a truth table, as shown in Table I, which can be further categorized into five cases that need to protect different pixel sets for eliminating the intra-frame error propagation.

B. The DCT/DST Perturbation Pattern Analysis

Embedding hidden bits in the spatial domain is time consuming since it requires extra decoding and re-encoding operations for compressed video sequences. The proposed data hiding algorithm aims to embed bits in the transform domain, in which the QDCTs and QDSTs of intra prediction residuals are perturbed to carry the hidden bits. According to different classified cases and protected pixel sets associated with the current PU, as shown in Table I, different bit embedding patterns are presented for perturbing quantized coefficients without inducing intra-frame error propagation.

In HEVC, the residual of one $N \times N$ intra-coded PU is encoded by either the same-sized TU when $N \in \{32, 16, 8, 4\}$ or four smaller-sized TUs when $N = 64$. For simplicity, we only discuss the case that the PUs are 4×4 and the other ones, 8×8, 16×16, 32×32 and 64×64, can be carried out in the same way.

For one 4×4 PU, denote the corresponding residual block and QDST coeeficient matrix as $R^p_{4\times4}$ and $R^{QDST}_{4\times4}$, respectively. When the current 4×4 PU is categorized into $Case_1$, we select coefficients from $R^{QDST}_{4\times4}$ to yield one set of four three-coefficient-elements, i.e., $S_H = \{(X_{00}, X_{02}, X_{03}), (X_{10}, X_{12}, X_{13}), (X_{20}, X_{22}, X_{23}), (X_{30}, X_{32}, X_{33})\}$, for embedding bits without inducing error propagation to its neighboring blocks. The subscript of S_H stands for horizontal. Each element in set S_H can be used to embed one bit by perturbing the three corresponding QDSTs and then all these perturbations on QDSTs associated with the four embedded bits would keep the leftmost and rightmost column pixel values of the reconstructed block unchanged. In short, performing bit embedding on S_H will not induce intra-frame error propagation along the horizontal direction.

The following coefficient perturbing example is provided to justify the intra-frame error propagation free of the bit embedding procedure. To embed one hidden bit in the three-coefficient-element (X_{00}, X_{02}, X_{03}), the three-coefficient-element (X_{00}, X_{02}, X_{03}) is perturbed to $(X_{00} + t, X_{02} - t, X_{03} + t)$ with $t \in \mathcal{Z}$ to yield a perturbed QDST coefficient matrix $R^{QDST'}_{4\times4}$. The difference between $R^{QDST'}_{4\times4}$ and $R^{QDST}_{4\times4}$ can be represented as

$$\Delta R^{QDST}_{4\times4} = R^{QDST'}_{4\times4} - R^{QDST}_{4\times4} = \begin{bmatrix} t & 0 & -t & t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The reconstructed perturbed residual block, denoted as $R^{r'}_{4\times4}$, can be obtained by performing IQDST defined in (2) on $R^{QDST'}_{4\times4}$. The difference between $R^{r'}_{4\times4}$ and $R^{r}_{4\times4}$ can be calculated by

$$\begin{aligned} \Delta R^{r}_{4\times4} &= R^{r'}_{4\times4} - R^{r}_{4\times4} = S_f^{-1}(\Delta R^{QDST}_{4\times4} \times Q)(S_f^T)^{-1} \\ &= Q \times t \times \begin{bmatrix} 0 & 0 & 3AC & 0 \\ 0 & 0 & 3BC & 0 \\ 0 & 0 & 3C^2 & 0 \\ 0 & 0 & 3CD & 0 \end{bmatrix}. \end{aligned}$$

As shown, the rightmost column of $\Delta R^{r}_{4\times4}$ is a zero vector, which means that imposing the perturbation pattern $(+t, -t, +t)$ on (X_{00}, X_{02}, X_{03}) will not induce errors on the rightmost column vector of the reconstructed residual block $R^{r'}_{4\times4}$. It implies that the perturbed PU will not propagate errors to its neighboring blocks which refer to the $Case_1$ protected pixels $\{R_{1,4}, \cdots, R_{4,4}\}$ for intra prediction. The perturbation patterns for the remaining three three-coefficient-elements, (X_{10}, X_{12}, X_{13}), (X_{20}, X_{22}, X_{23}) and (X_{30}, X_{32}, X_{33}), to yield error free on $Case_1$ protected pixels can be carried out in the same way. In other words, we can embed four hidden bits in one 4×4 PU categorized into $Case_1$ without inducing intra-frame error propagation.

When the current PU is categorized into $Case_2$, we define a set of four three-coefficient-elements selected from $R^{QDST}_{4\times4}$ for intra-frame error propagation free along the vertical direction, i.e. $S_V = \{(X_{00}, X_{20}, X_{30}), (X_{01}, X_{21}, X_{31}), (X_{02}, X_{22}, X_{32}), (X_{03}, X_{23}, X_{33})\}$. For each element in set $S_V = \{(X_{0j}, X_{2j}, X_{3j})\}_{j=0,1,2,3}$, we can embed one hidden bit by perturbing the element (X_{0j}, X_{2j}, X_{3j}) with $(+t, -t, +t)$ such that four bits can be embedded in total for one 4×4 PU in the $Case_2$ category. Derivations, which are similar to that for $Case_1$ PUs, revealed that the matrix $\Delta R^{r}_{4\times4}$ for one $Case_2$ PU is exactly the transpose of that for $Case_1$ PUs. Since the bottom row of $\Delta R^{r}_{4\times4}$ is a zero vector, the perturbed PU block will not propagate errors to its neighboring blocks that utilize the $Case_2$ protected pixels $\{R_{4,1}, \cdots, R_{4,4}\}$ for intra prediction. For one $Case_3$ PU, since only the lower right corner pixel $R_{4,4}$ needs to be protected, it can be handled in the same way as that for either $Case_1$ or $Case_2$ PUs. For simplicity, the $Case_3$ PU is handled as that for $Case_2$ PUs and hence four bits can be embedded. For $Case_4$ PUs, since no reference pixels need to be protected, all the sixteen quantized coefficients can be used for embedding bits.

For $Case_5$ PUs, we proposed to select nine coefficients for embedding one hidden bit, i.e., $(X_{00}, X_{02}, X_{03}, X_{20}, X_{22}, X_{23}, X_{30}, X_{32}, X_{33})$. The perturbation pattern for embedding one bit can be represented by the following equation that specifies the relation between $R^{QDST'}_{4\times4}$ and $R^{QDST}_{4\times4}$, i.e.,

$$\Delta R^{QDST}_{4\times4} = R^{QDST'}_{4\times4} - R^{QDST}_{4\times4} = \begin{bmatrix} t & 0 & -t & t \\ 0 & 0 & 0 & 0 \\ -t & 0 & t & -t \\ t & 0 & -t & t \end{bmatrix}.$$

The difference between $R^{r'}_{4\times4}$ and $R^{r}_{4\times4}$ can be computed by

$$\begin{aligned} \Delta R^{r}_{4\times4} &= R^{r'}_{4\times4} - R^{r}_{4\times4} = S_f^{-1}(\Delta R^{QDST}_{4\times4} \times Q)(S_f^T)^{-1} \\ &= Q \times t \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 9C^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

As shown, the rightmost column and the lowest row of the $\Delta R^{r}_{4\times4}$ are zero vectors such that, after embedding one bit, the protected pixels of $Case_5$ would be intact, from which the neighboring blocks can perform intra-prediction without error propagation.

C. The Proposed DCT/DST Based Data Hiding Algorithm

In the proposed data hiding algorithm, the bit embedding procedure is carried out in a way such that the odd and even QDSTs/QDCTs would be made consistent with those of hidden bits, i.e., 1 or 0. The related QDSTs/QDCTs corresponding to these hidden bits are then perturbed with specific patterns described in the previous section to eliminate error propagation to neighboring intra-coded PUs. Detailed control steps of the bit embedding and extraction are described as follows.

Since the procedures of bit embedding and extraction for the quantized coefficient matrixes within different-sized TUs are basically the same, only the procedure for 4×4 TUs is presented. For the quantized coefficient matrix within one 4×4 TU which corresponds to an intra-coded PU categorized into $Case_1$, each three-coefficient-element in S_H, (X_{ij}, X_{mn}, X_{pq})s, will be checked first. If X_{ij} is nonzero, the element (X_{ij}, X_{mn}, X_{pq}) is perturbed to $(X_{ij}^{'}, X_{mn}^{'}, X_{pq}^{'})$ for carry one hidden bit h; that is,

$$(X_{ij}^{'}, X_{mn}^{'}, X_{pq}^{'}) = (X_{ij} + t, X_{mn} - t, X_{pq} + t),$$

where

$$t = \begin{cases} 1, & \text{if } (X_{ij} > 0,\ h = 1, \text{ and } X_{ij} \text{ is even}) \\ & \text{ or } (X_{ij} > 0,\ h = 0, \text{ and } X_{ij} \text{ is odd}), \\ -1, & \text{if } (X_{ij} < 0,\ h = 1, \text{ and } X_{ij} \text{ is even}) \\ & \text{ or } (X_{ij} < 0,\ h = 0, \text{ and } X_{ij} \text{ is odd}), \\ 0, & \text{otherwise.} \end{cases}$$

When the corresponding PU is categorized into $Case_2$ or $Case_3$, each three-coefficient-element in S_V can be treated as that in S_H for carrying one hidden bit. For $Case_4$, since no pixel in the current PU has to be protected, all the sixteen QDSTs, $\{X_{ij}\}_{i,j=0,1,2,3}$, can be perturbed to embed at most sixteen hidden bits. If X_{ij} is non-zero, it can be perturbed to X'_{ij} by (5) for carrying one hidden bit h.

$$X'_{ij} = X_{ij} + t, \tag{5}$$

where

$$t = \begin{cases} 1, & \text{if } (X_{ij} > 0,\ h = 1, \text{ and } X_{ij} \text{ is even}) \\ & \text{ or } (X_{ij} > 0,\ h = 0, \text{ and } X_{ij} \text{ is odd}), \\ -1, & \text{if } (X_{ij} < 0,\ h = 1, \text{ and } X_{ij} \text{ is even}) \\ & \text{ or } (X_{ij} < 0,\ h = 0, \text{ and } X_{ij} \text{ is odd}), \\ 0, & \text{otherwise.} \end{cases}$$

When the corresponding PU is categorized into $Case_5$, the nine coefficients suggested in the previous subsection, (X_{00}, X_{02}, X_{03}, X_{20}, X_{22}, X_{23}, X_{30}, X_{32}, X_{33}), are perturbed to (X'_{00}, X'_{02}, X'_{03}, X'_{20}, X'_{22}, X'_{23}, X'_{30}, X'_{32}, X'_{33}), as shown in (6), for embedding one hidden bit h if the X_{00} is non-zero. Note that embedding hidden bits into a block with all zero coefficients will lead to degraded perception quality, in additional to increase bitrate [10], [13]. That is the reason why we embed hidden bits on non-zero coefficients, X_{ij}s, only and discard blocks with all zero coefficients.

At the decoder, the hidden bit extraction procedure is carried out after the HEVC compressed video sequence is entropy decoded. Given the de-quantized coefficient matirx within one 4×4 TU, we first determine which case the corresponding PU is categorized into according to the intra-prediction modes of neighboring blocks. When the corresponding PU is categorized into $Case_1$ or $Case_2$, the first coefficient X'_{ij} of each three-coefficient-element $(X'_{ij}, X'_{mn}, X'_{pq})$ in S_H or S_V will be checked, respectively. If X'_{ij} is non-zero, the hidden bit can be extracted by

$$h = \begin{cases} 1, \text{ if } X'_{ij} \text{ is odd}, \\ 0, \text{ otherwise.} \end{cases}$$

This hidden bit extraction procedure can also be applied to $Case_3$ PUs, whose perturbation pattern is the same as that for $Case_2$. If the corresponding PU is categorized into $Case_4$, the hidden bits can be extracted from non-zero coefficients among the sixteen quantized coefficient $\{X'_{ij}\}_{i,j=0,1,2,3}$ by

$$h = \begin{cases} 1, \text{ if } X'_{ij} \text{ is odd}, \\ 0, \text{ otherwise.} \end{cases}$$

For $Case_5$ PUs, if the first coefficient X'_{00} of the nine suggested coefficients is non-zero, the hidden bit can be extracted by

$$h = \begin{cases} 1, \text{ if } X'_{00} \text{ is odd}, \\ 0, \text{ otherwise.} \end{cases}$$

D. Quality Improvement for IFs Reconstruction

In this subsection, we describe how to improve the quality of IFs in the reconstructed video sequence based on spatial domain pixel correlations at the decoder. In general, spatial inter-pixel correlation is stronger in the unperturbed block than that in the perturbed one. We proposed to exploit and utilize the inter-pixel correlation such that, after extracting the hidden bit(s), the unperturbed block can be found from all possible candidate blocks derived from the current decoded PU. By replacing with the found unperturbed block, the bit embedding errors in the current block can be eliminated and meanwhile the problem of the inter-frame error propagation is also alleviated.

The hidden bit(s) is first extracted from the current decoded PU as 1/ 0 according to the coefficient odd/even. To prevent the current decoded PU from reconstruction with bit embedding errors, the decoder is designed to select, from several candidate unperturbed blocks, the one that yields strongest inter-pixel correlation. Since this quality improvement procedure can be carried out in the same way for different-sized TUs, we present the procedure that deals with one 4×4 TU as an example for simplicity.

Given the de-quantized coefficient matrix within one 4×4 TU whose corresponding PU is categorized into $Case_1$, we first check whether the first coefficient X'_{ij} of each three-coefficient-element, $(X'_{ij}, X'_{mn}, X'_{pq})$, in S_H is zero or not. If X'_{ij} is zero, it means that no hidden bit is embedded in the three-coefficient-element, i.e. $(X'_{ij}, X'_{mn}, X'_{pq}) = (X_{ij}, X_{mn}, X_{pq})$. For a non-zero X'_{ij}, in additional to extract the hidden bit from X'_{ij}, we can infer the relation between

$$(X'_{00}, X'_{02}, X'_{03}, X'_{20}, X'_{22}, X'_{23}, X'_{30}, X'_{32}, X'_{33}) =$$
$$(X_{00}+t, X_{02}-t, X_{03}+t, X_{20}-t, X_{22}+t, X_{23}-t, X_{30}+t, X_{32}-t, X_{33}+t), \tag{6}$$

where

$$t = \begin{cases} 1, & \text{if } (X_{00} > 0,\ h = 1, \text{ and } X_{00} \text{ is even) or } (X_{00} > 0,\ h = 0, \text{ and } X_{00} \text{ is odd}), \\ -1, & \text{if } (X_{00} < 0,\ h = 1, \text{ and } X_{00} \text{ is even) or } (X_{00} < 0,\ h = 0, \text{ and } X_{00} \text{ is odd}), \\ 0, & \text{otherwise.} \end{cases}$$

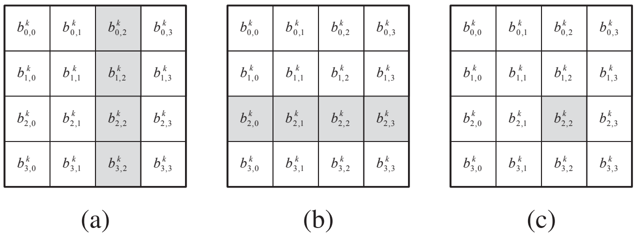


Fig. 5. The k-th 4×4 candidate reconstructed block with $k = 1, 2, \ldots, K$ and the pixel(s) suffered from the bit embedding errors corresponding to (a) $Case_1$, (b) $Case_2$ and $Case_3$, and (c) $Case_5$.

TABLE II
CONFIGURATIONS OF HM 10.0 AND JM 16.0

HM 10.0		JM 16.0	
Configuration Name	Value	Configuration Name	Value
FrameRate	30	FrameRate	30
FramesToBeEncoded	96	FramesToBeEncoded	96
IntraPeriod	1	IntraPeriod	1
QuadtreeTUMaxDepthIntra	3	ProfileIDC	77
GOPSize	1	SymbolMode	1
RDOQTS	1	LevelIDC	50
TransformSkip	1	RDOptimization	1

$(X'_{ij}, X'_{mn}, X'_{pq})$ and (X_{ij}, X_{mn}, X_{pq}) from (5) as follows:

$$(X_{ij}, X_{mn}, X_{pq}) = \begin{cases} (X'_{ij}, X'_{mn}, X'_{pq}) \text{ or } (X'_{ij}-1, X'_{mn}+1, X'_{pq}-1), \\ \text{if } X'_{ij} > 0, \\ (X'_{ij}, X'_{mn}, X'_{pq}) \text{ or } (X'_{ij}+1, X'_{mn}-1, X'_{pq}+1), \\ \text{otherwise.} \end{cases}$$

Depending on the value of X'_{ij}, there are at most two trials for each three-coefficient-element $(X'_{ij}, X'_{mn}, X'_{pq})$ to yield original one (X_{ij}, X_{mn}, X_{pq}). By combining all possible trials for the four three-coefficient-elements in S_H, there are at most K=16 candidate original blocks. For $Case_1$ PUs, only the third column pixels of the reconstructed block would suffer from bit embedding errors, as shown in Fig. 5(a), where $b^k_{i,j}$ denotes the luma value of the pixel at position (i, j) for the k-th candidate original block, $k = 1, 2, \cdots, K$. The cost function is designed to reflect the spatial correlation of the candidate original block based on inter-pixel differences between perturbed and neighboring unperturbed pixels. The candidate that yields the lowest cost function value is then considered as the most confident one and is selected as the original unperturbed block.

For $Case_2$, $Case_3$, and $Case_5$ PUs, the quality improvement procedure can be carried out in the same way under different error pattern specified in Fig. 5 for different cases. Note that $Case_4$ PUs are excluded in the proposed quality improvement procedure. For $Case_4$ PUs, since all the sixteen quantized coefficients can be used for embedding bits, all pixels in the reconstructed block may suffer from the bit embedding errors. Under this condition, since no unperturbed pixels can be referenced to estimate spatial correlation and many combinational trials from the sixteen coefficients are time-consuming, it is not feasible for the proposed quality improvement procedure to adopt $Case_4$ PUs.

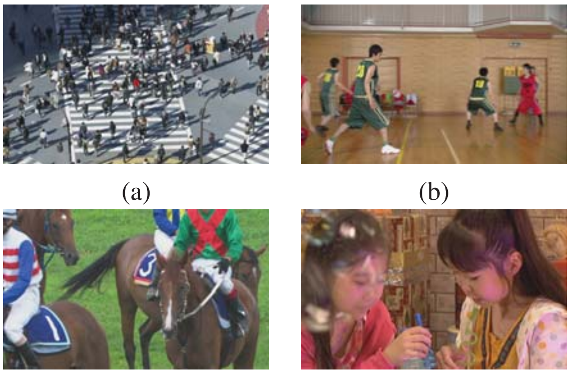
IV. Experimental Study

Although researches on performing data hiding in transform domain can be found by many, embedding hidden bits in the state-of-the-art HEVC IFs without intra-frame error propagation is challenging since more constraints are imposed on data hiding when compared with the previous ones in H.264/AVC. Moreover, since DCT is not adopted by HEVC 4×4 blocks for transform coding, the transform domain data hiding algorithms developed based on H.264/AVC, Ma *et al.* [10] and Lin *et al.* [11], cannot be adapted to the HEVC standard. Notwithstanding, experiments for these conventional data hiding algorithms are also carried out for comparison. Performance comparisons among the proposed data hiding algorithm and the other ones are made in terms of embedding capacity, PSNR, and bitrate. The standard H.264/AVC and HEVC coding algorithms are implemented to evaluate the performance of embedding capacity and quality degradation for the previous works and the proposed data hiding algorithm.

Four sample videos [14] with different resolutions, as shown in Fig. 6, are used as test ones. The GOP size is 32 and the coding structure is IPP. . .P. The values of QP are set to be 17, 22, 27, 32, 37, and 42. The video codec platforms for H.264/AVC and HEVC are JM 16.0 and HM 10.0, respectively, whose configuration parameters are listed in Table II. The data hiding algorithms and the two codec platforms are developed with Visual C++ 2012 on a 64-bit computer with Intel i7-3770 3.4GHz CPU and 8GB RAM. Detailed experimental results are available in [15].

A. Quality Improvement Scheme Analysis

To evaluate the performance of the proposed quality improvement scheme, our proposed data hiding algorithm is performed without or with the quality improvement scheme,



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

Fig. 6. Test videos: (a) *PeopleOnStreet* (2560×1600); (b) *BasketballDrive* (1920×1080); (c) *RaceHorse* (832×480); and (d) *BlowingBubbles* (416×240).

in which the former is denoted as EMBED-O and the latter EMBED-W. For analysis, we define the hit rate as

$$\text{hit rate} = \frac{N_C}{N_T} \times 100\%,$$

where N_C and N_T denote the numbers of correctly recovered blocks and total embedded blocks, respectively. The hit rate of the proposed quality improvement scheme obtained from experiments is shown in Table III. Experiments revealed that the distortion of $Case_5$ blocks is like impulse noise and can be easily detected in the quality improvement procedure that utilize inter-pixel correlations. As shown in Table III, the hit rates for $Case_5$ blocks that are transform coded by different-sized TUs are from 84% to 100%, which are the highest among all Cases.

Note that the $Case_1$, $Case_2$, and $Case_3$ blocks that are transform coded by non 4×4 TUs are not recovered in experiments since these blocks involve much high computational complexity for determining the best candidate for the original unperturbed block. As for the remaining $Case_1$, $Case_2$, and $Case_3$ blocks, their hit rates are lower than those for $Case_5$ blocks. The reason for that can be explained as follows. Since there are four bits embedded in one 4×4 DST block, the distortion due to embedding is one of the total 2^4 possible embedded distortion patterns. In addition, the embedded distortion cannot be separated into individual ones for each embedded bit. Under this condition, it is difficult to find the original block for such one perturbed block.

The PSNR gain contributed by the quality improvement procedure, or the improved PSNR by the EMBED-W over the EMBED-O, is shown in Table IV. As shown, the PSNR gain is from 0.54dB to 0.79dB, which justifies the capability of the proposed quality improvement scheme.

TABLE III
HIT RATE (%) OF THE QUALITY IMPROVEMENT SCHEME IN DIFFERENT CASES, TRANSFORM SCHEMES AND QPS

		QP17	QP22	QP27	QP32	QP37	QP42
Case 1	DST4x4	48	63	66	65	62	59
Case 2	DST4x4	51	58	63	63	59	57
Case 3	DST4x4	37	43	47	50	59	52
Case 5	DST4x4	94	97	99	100	100	100
	DCT8x8	86	94	98	99	100	100
	DCT16x16	91	97	99	100	100	100
	DCT32x32	84	98	99	100	100	100

TABLE IV
THE PSNR (DB) GAIN FROM APPLYING THE EMBED-W OVER THE EMBED-O UNDER DIFFERENT QPS

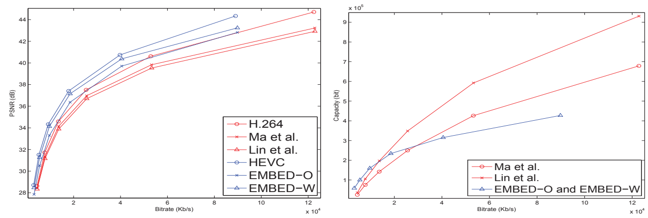
QP	PSNR Gain
17	0.56
22	0.71
27	0.74
32	0.79
37	0.69
42	0.54
Average	0.67

B. Performance Evaluations of Data Hiding Algorithms

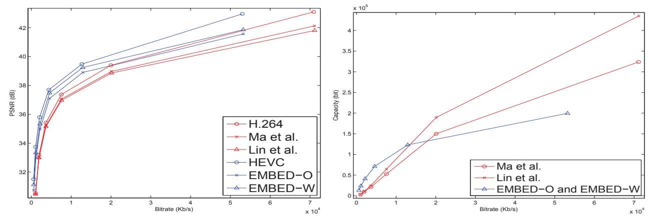
In general, allocating higher bitrates for coding video often demonstrates higher PSNRs and embedding capacity. The overall data hiding performance can be evaluated objectively by plots of bitrates vs. distortion (RD) and bitrates vs. embedding capacity (RC). The RD and RC plots of different data hiding algorithms on the four test videos are illustrated at the left and right, respectively, in Fig. 7. As shown, the proposed EMBED-W outperforms the algorithms of Ma *et al.* and Lin *et al.* in PSNR. It is expected, since more intra prediction modes and PUs/TUs sizes are available in HEVC and higher compression performance can be achieved when compared with previous codec. However, more prediction modes also imply stricter constraints on embedding hidden bits in HEVC intra-coded blocks. For example, one $N \times N$ TU with $N \in \{4, 8, 16, 32\}$ in HEVC can be embedded with N bits at most, while the same-sized residual one using 4×4 transform block(s) in H.264/AVC can be embedded with $4 \times (\frac{N}{4})^2$ bits at most.

For low bitrate video coding, the proposed EMBED-W demonstrates the best performances in both PSNRs and embedding capacity. In both HEVC and H.264/AVC, it is frequent to adopt larger transform blocks for transform coding of residual blocks for larger QPs. The algorithms of Ma *et al.* and Lin *et al.* do not embed bits in larger transform blocks, such as 16×16 ones of H.264/AVC, but the proposed algorithm can embed bits in 8×8, 16×16, and 32×32 transform blocks of HEVC. That is the reason why the latter can achieve higher embedding capacity than the former for low bitrate video coding.

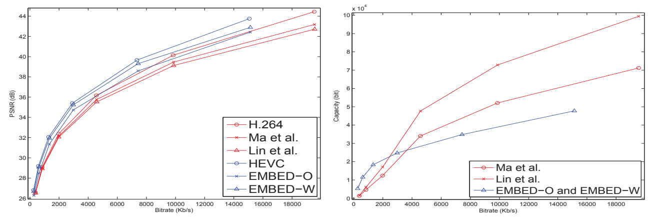
For subjective performance evaluation, the reconstructed frames with embedded bits from the videos, *BasketballDrives* and *RaceHorses*, with QP=32 are demonstrated in Figs. 8-9. As shown, the quality degradation of the embedded frames by Ma *et al.* and Lin *et al.* are severe and with poor perception quality. On the contrary, the embedded frames by the proposed EMBED-W demonstrate less visual artifacts. In summary, for



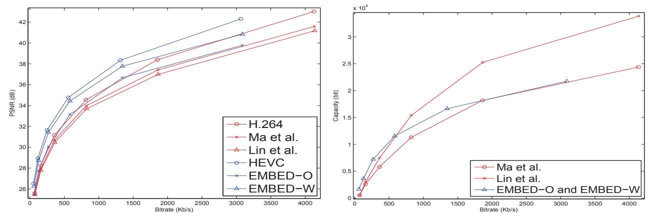
(a)*PeopleOnStreet*



(b)*BasketballDrive*



(c)*RaceHorses*



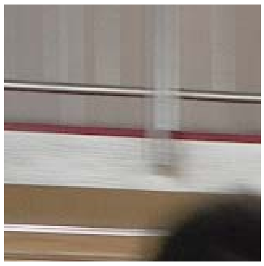
(d)*BlowingBubbles*

Fig. 7. Objective data hiding performance evaluation with PSNR vs. bitrate and capacity vs. bitrate plots for different methods on different test videos.

low bitrate coding, the proposed EMBED-W algorithm can achieve better data hiding performance, i.e., higher embedding capacity and lower objective/subjective distortions, when compared with previous works.

V. Conclusions

To embed hidden bits in HEVC IFs without propagating errors to neighboring blocks and adjacent frames, we propose to classify HEVC coding blocks according to certain intra prediction mode combinations of neighboring blocks to perform specific data hiding patterns. The signal characteristics of DCT/DST are analyzed to locate the QDCTs/QDSTs that can be perturbed without propagating errors to neighboring intra-coded blocks. In additional to the intra-frame error propagation free data hiding algorithm, one quality improvement scheme is also presented for the reconstructed IFs and meanwhile alleviates the artifact of the inter-frame error propagation. Experiments justify the efficiency of the proposed algorithm in performing intra-frame error propagation free data hiding, providing higher embedding capacity in low bitrate coding, and yielding better reconstructed video quality, when compared with previous transform domain data hiding algorithms.



(a)

(b)

(c)

(d)

Fig. 8. Subjective performance evaluation of the reconstructed IFs of the first frame of the *BasketballDrives* sequence by: (a) Original; (b) Ma *et al.*'s algorithm; (c) Lin *et al.*'s algorithm; (d) Proposed EMBED-W with QP=32.

(a)

(b)

(c)

(d)

Fig. 9. Subjective performance evaluation of the reconstructed IFs of the first frame of the *RaceHorses* sequence by: (a) Original; (b) Ma *et al.*'s algorithm; (c) Lin *et al.*'s algorithm; (d) Proposed EMBED-W with QP=32.

Acknowledgment

The work of K. L. Chung and C. H. Lin was supported by the National Science Council of R. O. C. under the contracts NSC101-2221-E-011-139-MY3 and NSC99-2221-E-011-078-MY3. The work of J. J. Chen was supported by the National Science Council of R. O. C. under the contract NSC101-2221-E-011-137.

References

[1] G. Qiu, P. Marziliano, A. T. Ho, D. He, and Q. Sun, "A Hybrid Watermarking Scheme for H.264/AVC Video," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 865-868, 2004.

[2] K. Wong, K. Tanaka, and X. Qi, "Multiple messages embedding using DCT-based mod4 steganographic method," in *Proc. LNCS Int. Workshop Multimedia Content Representation, Classification, Security*, pp. 57-65, 2006.

[3] M. Noorkami and R. Mersereau, "A framework for robust watermarking of H.264-encoded video with controllable detection performance," *IEEE Trans. Information Forensics and Security*, vol. 2, pp. 14-23, 2007.

[4] J. Zhang, A. T. S. Ho, and G. Qiu, "Robust video watermarking of H.264/AVC," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 54, pp. 205-209, 2007.

[5] X. Gong and H. M. Lu, "Towards fast and robust watermarking scheme for H.264 video," in *Proc. IEEE Int. Symp. Multimedia*, pp. 649-653, 2008.

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

[7] W. Huo, Y. Zhu, and H. Chen, "A controllable error-drift elimination scheme for watermarking algorithm in H.264/AVC stream," *IEEE Signal Processing Letters*, vol. 18, no. 9, pp. 535-538, 2011.

[8] B. Samira, A. Hassina, and H. Latifa, "H.264/AVC data hiding algorithm with a limitation of the drift distortion," in *Proc. IEEE Int. Conf. Signal Processing*, pp. 1682-1687, 2012.

[9] *Draft ITU-T recommendation and final draft international standard of joint video specification*, document ITU-T Rec. H.264/ISO/IEC 14496-10 AVC, JVT of ISO/IEC and ITU-T, 2003.

[10] X. J. Ma, Z. T. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 10, pp. 1320-1330, 2010.

[11] T. J. Lin, K. L. Chung, P. C. Chang, Y. H. Huang, H. Y. Liao, and C. Y. Fang, "An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames," *Journal of Systems and Software*, vol. 86, no. 3, pp. 604-614, 2013.

[12] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, "JCTVC-F803:WD4: Working draft 4 of high-efficiency video coding," JCTVC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 2011.

[13] M. Noorkami and R. Mersereau, "Compressed-domain video watermarking for H.264," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 11-14, 2005.

[14] F. Bossen, "Common test conditions and software reference configurations," JCTVC-L1100, 2013.

[15] [Online]. http://140.118.175.164/PCChang/APSIPA2013/HEVCDataHiding/.