A Fragile Watermark Error Detection Scheme for Wireless Video Communications

Minghua Chen, Yun He, *Senior Member, IEEE*, and Reginald L. Lagendijk, *Senior Member, IEEE*

***Abstract*—In video communications over error-prone channels, compressed video streams are extremely sensitive to bit errors. Often random and burst bit errors impede correct decoding of parts of a received bitstream. Video decoders normally utilize error concealment techniques to repair a damaged decoded frame, but the effectiveness of these error concealment schemes relies heavily on correctly locating errors in the bitstream. In this paper, we propose a fragile watermark-based error detection and localization scheme called "force even watermarking (FEW)". A fragile watermark is forced onto quantized DCT coefficients at the encoder. If at the decoder side the watermark is no longer intact, errors exist in the bitstream associated with a particular macro-block (MB). Thanks to the watermark, bitstream errors can accurately be located at MB level, which facilitates proper error concealment. This paper describes the algorithm, model and analysis of the watermarking procedure. Our simulation results show that compared to the syntax-based error detection schemes, the proposed FEW scheme significantly improves the error detection capabilities of the video decoder, while the peak signal-to-noise ratio loss and additional computational costs due to watermark embedding and extraction are small.**

***Index Terms*—Error detection, error resilience, forced even watermark (FEW), fragile watermark, video coding, video communications, wireless communications.**

I. Introduction

In recent years, applications of video communications over wireless channels have emerged rapidly. Sophisticated video compression algorithms such as the H.263 ITU [1], [2] and MPEG standards [3]–[5] are employed in order to meet the bit rates provided by band-limited wireless channels. These video processing algorithms achieve highly efficient compression by using motion-compensated, DCT-based inter-frame video compression.

The transmission of compressed video streams over wireless communication channels, however, presents several challenging problems that remain to be resolved. One of the most important problems is that on one hand, compressed video streams are extremely sensitive to bit errors, while on the other hand, the imperfect wireless channel introduces vast amounts of random and burst bit errors due to fading, signal attenuations, and co-channel interference [6], impeding the correct decoding of the received video bitstreams. Particularly, variable-length coded (VLC) codes are highly susceptible to errors in the bitstream. As a result, the decoder may lose synchronization with the encoder, making it impossible to correctly decode a sequence of VLC code words until the next resynchronization code word is met. Predictive coding techniques aggravate the situations. Because decoding errors in a video frame may propagate to subsequent video frames, bit errors degrade the quality of not merely individual frames but of the entire video sequence.

There are several approaches to make video stream more resilient to the wireless channel's degradations. The approaches include: 1) error correction and data interleaving [7]; 2) error detection and localization by channel coding [7], [8]; 3) resynchronization and data partitioning [7], [9], [10]; 4) error detection on the video decoder side; and 5) error concealment [11]. forward error correction (FEC) codes are widely employed by video encoders/decoders to correct transmission errors. However, FEC techniques are generally ignorant of applications' differentiated needs for error correction, and come therefore an unjustified overhead in terms of the overall bitstream size or bit rate. Typically, FEC is applied to provide protection of the compressed video stream up to a certain level. Uncorrectable but detected errors cause an automatic repeat request (ARQ) to be issued in delay-insensitive video communication applications such as video downloading and buffered play out. In delay-sensitive video streaming scenarios, error detection is used to inform the video decoder that proper concealment actions need to be taken.

Resynchronization is a technique that enables the decoder to fall back into lock step with the encoder after bit errors have been detected. Techniques such as data partitioning can be used to avoid propagation of errors between different data portions. For instance, the data partitioning scheme in MPEG-4 [5] partitions motion and texture data, thus improving the error resilience of the compressed video stream.

In this paper, we propose a novel error *detection* scheme. More specifically, the proposed technique belongs to the class of application-oriented error detection schemes, which aims at detecting and locating any errors remaining in the received compressed video stream after FEC decoding. Previous application-oriented error detection schemes essentially detect remaining

Manuscript received March 14, 2002; revised July 17, 2003. This work was supported by the National Science Foundation of China under Grants 69972021 and 60333020, by the Advanced Research Project of Tsinghua University, and by the Ubiquitous Communications Research Program at Delft University of Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Edward W. Knightly.

M. Chen was with the State Key Laboratory of Microwave and Digital Communications, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. He is now with Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: minghua@eecs.berkeley.edu).

Y. He is with the State Key Laboratory of Microwave and Digital Communications, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: hey@video.tsinghua.edu.cn).

R. L. Lagendijk is with Information and Communication Theory Group, Delft University of Technology, The Netherlands (e-mail: R.L.Lagendijk@Ewi.TUDelft.nl).

Digital Object Identifier 10.1109/TMM.2005.843367

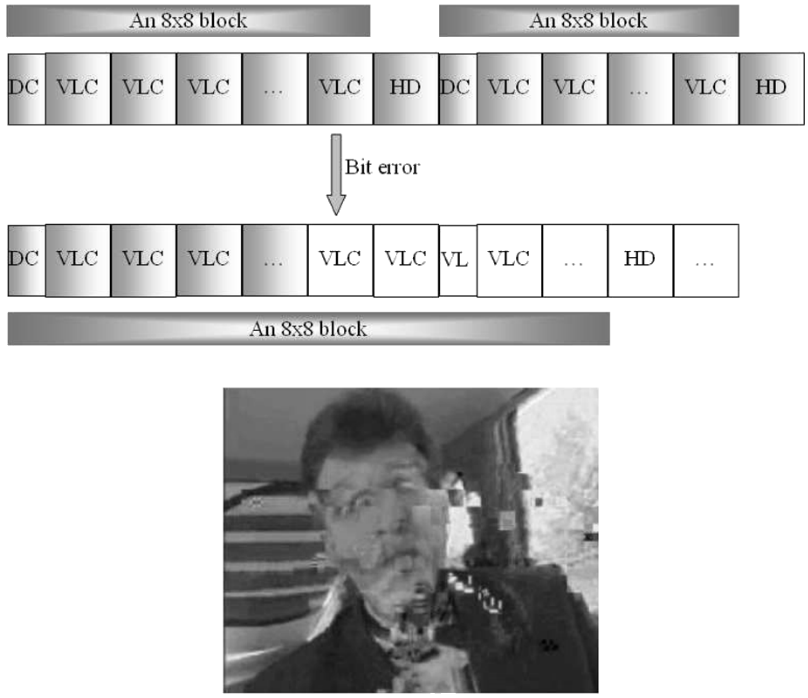


Fig. 1. Illustration of the MB shifting phenomenon. 8 × 8 blocks are inserted due to incorrectly decoded VLC codes after bit errors.

bit errors by validating the stream syntax and the interpretation given to code words found in the video stream [12]. In general, the success of error concealment techniques depends on the correct detection of erroneous macro-blocks (MB)—and not bitstream errors—since error-concealment techniques are applied at the MB level. For that reason, we focus on detecting and localizing erroneous MBs.

The rest of this paper is organized as following. In Section II, the error types and conventional syntax-based schemes will be reviewed briefly. In Section III, the basic idea of error detection and localization employing fragile watermarking will be explained. A particular scheme called "force even watermark" (FEW) will be proposed. As watermarks introduce additional degradations into the compressed video data, we will analyze the severity of this degradation in Section IV. Section V provides simulation results and performance comparison of proposed scheme to syntax-based error detection. Section VI concludes this paper with a brief discussion.

II. Error Types and Conventional Schemes

A. Error Types

Bit errors have different decoding consequences depending on the bitstream components being affected, such as header errors, motion vector errors and quantized DCT coefficient errors. An erroneous header causes subsequent components, including motion vector and quantized DCT coefficient, to be incorrectly decoded. Erroneous motion vectors cause motion-compensated information to occur at incorrect spatial positions. Erroneous DCT coefficients cause the luminance and chrominance values of current and following 8 × 8 DCT blocks to be reconstructed erroneously. Often, bit errors also destroy the proper separation between MBs, virtually removing or inserting additional 8 × 8 DCT blocks. This phenomenon results in a (horizontal) shift of decoded MB information. As an illustration of this phenomenon, Fig. 1 shows a slice of shifted macro blocks due to bit error near the right eye of the person shown in the video sequence.

Although bitstream errors that effect headers and motion vectors have more severe consequences for the decoding process than errors in quantized DCT coefficients, they happen less frequent since the largest part of the video streams consists of quantized DCT coefficients. For that reason we believe that casting additional error detection and localization mechanisms onto quantized DCT coefficients is effective to improve the correct operation of subsequent error concealment. Furthermore, providing headers and motion information with additional error detection information is relatively useless, as the consequences of errors in headers and motion information can generally not be concealed by any spatial processing of decoded video frames.

B. Conventional Error-Detection Scheme

In motion-compensated DCT-based compression system, the following *syntax-based* error detection techniques can be applied to detect bitstreams errors [12]:

- motion vectors are out of range;
- invalid VLC table entry is found;
- DCT coefficient is out of range;
- number of DCT coefficients in an 8 × 8 DCT-block exceeds 64;
- parameters (e.g. the quantizer scale factor) are out of range.

If a video decoder encounters any of these syntax errors during the decoding process, it marks the MB being decoded as erroneous and invokes an error handling procedure that usually carries out bitstream resynchronization and spatial-temporal error concealment.

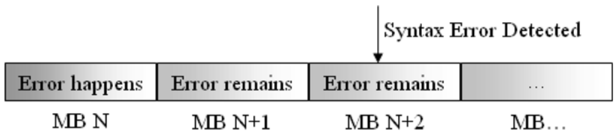


Fig. 2. Illustration of error detection lag, yielding error detection at an incorrect (macro-) block.

Unfortunately, syntax-based error detection in the decoder has two disadvantages.

1) The error detection probability is relatively low. In this paper, we define the error detection probability as the ratio of the number of detected erroneous groups of MBs to the number of all erroneous group of MBs. Experimental evaluation shows typical error detection probabilities that syntax-based scheme can achieve varies from 0.1 to 0.4.
2) The probability of errors being correctly located is very low. Our experiments show it is between 0.05 and 0.15. The reason for this poor performance is that the error detection mechanism lags the actual occurrence of errors in the bitstream. Consequently, several preceding erroneous macro blocks can not be detected and concealed. Fig. 2 graphically illustrates what we mean by "lagging detection".

In a recent proposal [13], it was shown that casting robust watermarks onto compressed video stream increases the error detection capabilities of the video decoder. The proposed scheme hide parity checking of the DCT information in the video stream, which can lead to an improvement of the probability of correctly locating errors on MB level by 10% $\sim$ 30%. On the other hand, approximately 2-dB worst-case PSNR losses were observed for the luminance component after adding the watermark. In next section, we propose an alternative to the technique of [13] and cast a fragile watermark onto the compressed bitstream that yields a smaller loss in PSNR and has a better error detection performance.

III. Proposed Technique

A. General Approach

Two categories of watermarking techniques exist, namely *robust* watermarking and *fragile* watermarking. Robust watermarking has been applied to embed author and copyright identification in multimedia data [14]–[16]. The watermark must be retained in the signal even under intentional signal distortion and intentional attacks to remove it. In contrast, fragile watermarking refers to the process of watermarking a signal such that even the slightest modification of the data causes the extracted watermark to be different from the original. Fragile watermarking can be used to detect tampering of multimedia data [17]–[19]. Although these schemes were designed for still-image authentication, the same idea can be used in the detection of "tampering" of video data by errors in the wireless channel.

The idea of the proposed technique is the following. The video encoder puts a *fragile* watermark onto the quantized DCT coefficients before these are passed to the variable length encoder. The watermark embedding is carried out within the motion-compensation loop as so to avoid degradations due to drift. The structure of the video encoder using the proposed fragile watermarking technique is shown in Fig. 3.

The fragile watermark can be embedded into the data at MB level, e.g. forcing a relation between blocks inside a MB; or at 8×8 DCT block level, e.g., forcing the relation between different quantized coefficients in a single DCT block. On the decoder side, the watermark is detected directly from the quantized DCT coefficients. Since the watermark is fragile, ideally, any uncorrected channel error will corrupt the watermark. The decoder then is able to perform the precise detection and localization of the erroneous MBs. In reality, undesired miss detection may happen, and one design goal could be to minimize the miss detection probability.

In the proposed technique, we put the fragile watermark onto 8×8 DCT blocks. In this way, we can vary the watermark parameters depending on the block properties, such as intra-coded or predicted frames, intra- or inter-coded DCT blocks, luminance or chrominance blocks, and the statistical property of the DCT block. The proposed technique is backward compatible with existing decoders. This implies that a decoder which does not have the fragile watermark detector will not be bothered by the presence of the watermark on the DCT coefficients. Obviously, since such decoder does not have any knowledge about the fragile watermark, it will not make use of the embedded error detection information. Therefore, our proposed approach does need a decoder with more functionality than the standard-compliant ones in order to locate channel bit errors more accurately.

The watermarking technique here embeds information by modifying quantized DCT coefficients to facilitate the error detection process at the decoder side. To avoid the accumulation over time of differences in DCT coefficients caused by embedding a watermark, the watermark module has to be within the inter-frame coding loop. Differences introduced by the watermarking of one video frame will show up in the (motion-compensated) frame differences calculated for the next frame. Therefore, casting a watermark on the DCT coefficients will influence the quality of the encoded video.

Irrespective of the watermarking technique used, we therefore need to trade off two performance aspects, namely:

- the error detection capability;
- the quality loss due to casting watermark onto the quantized DCT coefficient.

We address the second performance aspect theoretically in Section IV. The improvement in error detection capability is much harder to predict theoretically, we therefore confine ourselves to experimental evaluation of the error detection capability in Section V.

B. FEW Scheme

In this subsection, we will propose a fragile watermarking scheme that can greatly improve the error detection capability. The scheme is simple to implement and easy to analyze, but we do not claim this scheme to be optimal in trading off error detection capabilities and loss in (PSNR) quality. The proposed

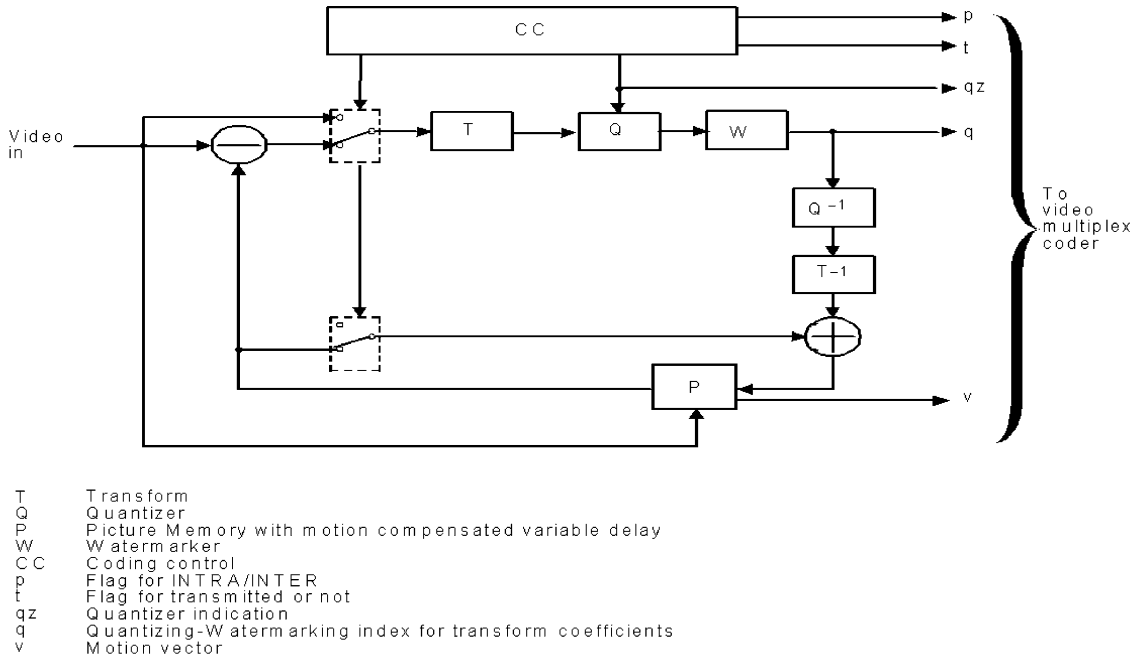


Fig. 3. Structure of the motion-compensated video encoder employing the proposed fragile watermarking technique. The watermarking module is indicated by W.

scheme illustrates the efficiency of error detection by watermarking and we hope it inspires future research into this area and into developing optimal fragile watermarking schemes for channel error detection.

The fragile watermarking scheme that we propose here is FEW to indicate that certain DCT coefficients are forced to quantized even values. The fragile watermark used is given by (1). All quantized DCT coefficients in an 8×8 DCT block after a cutoff zigzag scan position are modified to nearby smaller even numbers

$$\begin{aligned} &\text{for } i = pos\ to\ 63 \\ &\quad \mathrm{AC}_i^W = \begin{cases} \mathrm{AC}_i, & |\mathrm{AC}_i| \text{ is even} \\ \mathrm{AC}_i - sign(\mathrm{AC}_i), & |\mathrm{AC}_i| \text{ is odd} \end{cases} \\ &end \\ &sign(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0. \end{cases} \end{aligned} \tag{1}$$

Here, AC_i are the AC DCT coefficients and AC_i^W represent the watermarked version of it. $pos(0 \leq pos \leq 63)$ is the cutoff zigzag scan position.

The watermarking procedure is applied when encoding the video data, immediately after quantization process. In the decoding, if a watermarked coefficient is detected as odd, apparently, it has been corrupted and therefore the DCT coefficient must have been damaged by channel errors. Hence, an erroneous 8×8 DCT block can be detected and marked for concealment. Here, a MB is an erroneous one if any of the six 8×8 DCT blocks in it is reported as erroneous.

An obvious alternative scheme is to force quantized DCT coefficients to odd values. However, forcing to odd values has an important disadvantage. Forcing to even value can be applied to every value a DCT coefficient can take, whereas forcing to an odd value should only be applied to DCT coefficients whose absolute value is larger than two. Namely, forcing zero-valued DCT coefficients to the nearest odd values "1" or "-1" will disrupt the runlength coding of zero-valued DCT coefficient significantly, which will result in a loss of compression efficiency. Therefore, forcing quantized DCT coefficients to even values is to be preferred over odd values.

IV. Quality Loss After Watermarking the Quantized DCT Coefficients

Changing the parity of quantized DCT coefficients obviously degrades the reconstructed video sequence somewhat. The cutoff zigzag scan position pos controls the amount of additional distortion introduced. For a larger value of pos, the degradation due to watermarking is smaller, but the probability of detecting an erroneous DCT block is also smaller. Since the watermarking process is carried out on the quantized DCT coefficients, the quality loss due to watermarking depends on not only the watermarking process but also the coarseness of quantization. We will now analyze the quality loss due to watermarking in PSNR sense, as a function of the cutoff zigzag scan position pos, the quantizer coarseness (QP) and the coding type (intra-coded versus inter-coded DCT blocks). In this analysis, we express the PSNR as follows:

$$\mathrm{PSNR} = 10log_{10}\left[\frac{255^2}{E(\mathrm{MSE}_{\mathrm{DCT}})}\right] \tag{2}$$

where $E(\mathrm{MSE}_{\mathrm{DCT}})$ is the expectation of mean square error of the DCT coefficients due to watermarking ***only***.

To calculate $E(\mathrm{MSE}_{\mathrm{DCT}})$ we need a model for the probability density function of the DCT AC coefficients, i.e., OAC_i, $i = 1, 2 \ldots 63$. We use the commonly accepted Laplacian model for DCT coefficients of both intra- and inter-coded blocks [20]–[22]

$$p(x) = \frac{\lambda}{2}e^{-\lambda|x|}, \quad \lambda = \frac{1}{E(|x|)}. \tag{3}$$

TABLE I
PERFORMANCE OF FEW SCHEME COMPARED TO SYNTAX-BASED ERROR DETECTION: "AKIYO" SEQUENCE

Error detection scheme	Coded bit rate (kbps)	Coding PSNR (dB)	△ PSNR (dB)	Error detection rate (%)	Error correctly located rate (%)	Avg. decoded PSNR improvement (dB)
Syntax-based (BER=$5\cdot10^{-3}$)	92.56	36.45	–	37.1	2.1	–
Syntax-based (BER=10^{-3})	92.56	36.45	–	34.8	4.9	–
Syntax-based (BER=$5\cdot10^{-4}$)	92.56	36.45	–	29.1	11.3	–
Syntax-based (BER=10^{-4})	92.56	36.45	–	33.3	5.6	–
FEW (BER=$5\cdot10^{-3}$)	85.91	35.79	0.24	63.2	22.4	4.0
FEW (BER=10^{-3})	85.91	35.79	0.24	55.6	43.4	3.9
FEW (BER=$5\cdot10^{-4}$)	85.91	35.79	0.24	62.7	49.4	3.7
FEW (BER=10^{-4})	85.91	35.79	0.24	56.3	40.7	3.5

TABLE II
PERFORMANCE OF FEW SCHEME COMPARED TO SYNTAX-BASED ERROR DETECTION: "MOTHER AND DAUGHTER" SEQUENCE

Error detection scheme	Coded bit rate (kbps)	Coding PSNR (dB)	△ PSNR (dB)	Error detection rate (%)	Error correctly located rate (%)	Avg. decoded PSNR improvement (dB)
Syntax-based (BER=$5\cdot10^{-3}$)	158.14	34.59	–	37.2	2.3	–
Syntax-based (BER=10^{-3})	158.14	34.59	–	28.1	7.6	–
Syntax-based (BER=$5\cdot10^{-4}$)	158.14	34.59	–	31.1	6.0	–
Syntax-based (BER=10^{-4})	158.14	34.59	–	28.1	6.3	–
FEW (BER=$5\cdot10^{-3}$)	151.31	34.49	0.28	69.7	26.2	3.4
FEW (BER=10^{-3})	151.31	34.49	0.28	63.7	40.9	4.1
FEW (BER=$5\cdot10^{-4}$)	151.31	34.49	0.28	62.3	48.5	2.8
FEW (BER=10^{-4})	151.31	34.49	0.28	60.0	47.3	2.1

Let us first consider the decrease in PSNR for intra-coded blocks. The typical quantization and de-quantization function for AC intra-coefficients using in H.263 [2] are

$$\begin{aligned} &Quant. && \mathrm{AC}_i = \left\lfloor \frac{\mathrm{OAC}_i}{2\mathrm{QP}} \right\rfloor \\ &Dequant. && \widehat{\mathrm{OAC}}_i = \begin{cases} (2AC_i + sign(\mathrm{AC}_i)), & \mathrm{AC}_i \neq 0 \\ \mathrm{QP} - sign(\mathrm{AC}_i)\mathrm{QP}_0, & \\ 0, & \mathrm{AC}_i = 0 \end{cases} \\ &where && \mathrm{QP}_0 = \begin{cases} 1, & \mathrm{QP}\ \text{is even} \\ 0, & \mathrm{QP}\ \text{is odd.} \end{cases} \end{aligned} \quad (4)$$

Here, $\mathrm{OAC}_i (1 \leq i \leq 63)$ are the original AC DCT coefficient prior to quantization, AC_i are the quantized DCT coefficients, and $\widehat{\mathrm{OAC}}_i$ is the reconstructed (dequantized) version of OAC_i. QP is the coarseness of the quantizer.

As shown in (1), the watermarked AC DCT coefficients AC_i^W are given by

$$\mathrm{AC}_i^W = \begin{cases} \mathrm{AC}_i, & |\mathrm{AC}_i|\ \text{is even} \\ \mathrm{AC}_i - sign(\mathrm{AC}_i), & |\mathrm{AC}_i|\ \text{is odd.} \end{cases} \quad (5)$$

Taking the properties of the dequantization into account, we then obtain the difference between the nonwatermarked reconstructed AC DCT coefficients $\widehat{\mathrm{OAC}}_i$ and the watermarked reconstructed AC DCT coefficients $\widehat{\mathrm{OAC}}_i^W$ when AC_i is not zero

$$\widehat{\mathrm{OAC}}_i - \widehat{\mathrm{OAC}}_i^W = \begin{cases} 2\mathrm{QP}\left(\mathrm{AC}_i - \mathrm{AC}_i^W\right), & |\mathrm{AC}_i| \neq 1 \\ \pm 3\mathrm{QP} \mp \mathrm{QP}_0, & |\mathrm{AC}_i| = 1. \end{cases} \quad (6)$$

Observe that we need to consider the case $|\mathrm{AC}_i| \neq 1$ and $|\mathrm{AC}_i| = 1$ separately due to the deviant dequantization process around zero.

Since only an odd DCT coefficient will be changed to be even, it is easy to see that only a DCT coefficient with value between $2(2k-1)\mathrm{QP}$ and $4k\mathrm{QP}(k = 1, 2, \ldots)$ will be modified. Hence, using (6) we obtain

$$\begin{aligned} E&\left(\left|\widehat{\mathrm{OAC}}_i - \widehat{\mathrm{OAC}}_i^W\right|^2\right) \\ &= E\left(|3\mathrm{QP} - \mathrm{QP}_0|^2\right) \cdot P\{|\mathrm{AC}_i| = 1\} \\ &\quad + E\left(\left|2\mathrm{QP}\left(\mathrm{AC}_i - \mathrm{AC}_i^W\right)\right|^2\right) \cdot P\{|\mathrm{AC}_i| \neq 1\} \\ &= (3\mathrm{QP} - \mathrm{QP}_0)^2 (e^{-2\mathrm{QP}\lambda_i} - e^{-4\mathrm{QP}\lambda_i}) \\ &\quad + 4\mathrm{QP}^2 \frac{e^{-6\lambda_i \mathrm{QP}}}{1 + e^{-2\lambda_i \mathrm{QP}}} \end{aligned} \quad (7)$$

where $\lambda_i (1 \leq i \leq 63)$ are the probability density function (pdf) shape parameters of Laplacian distribution of the AC DCT coefficient OAC_i. Taking into account that only DCT coefficients

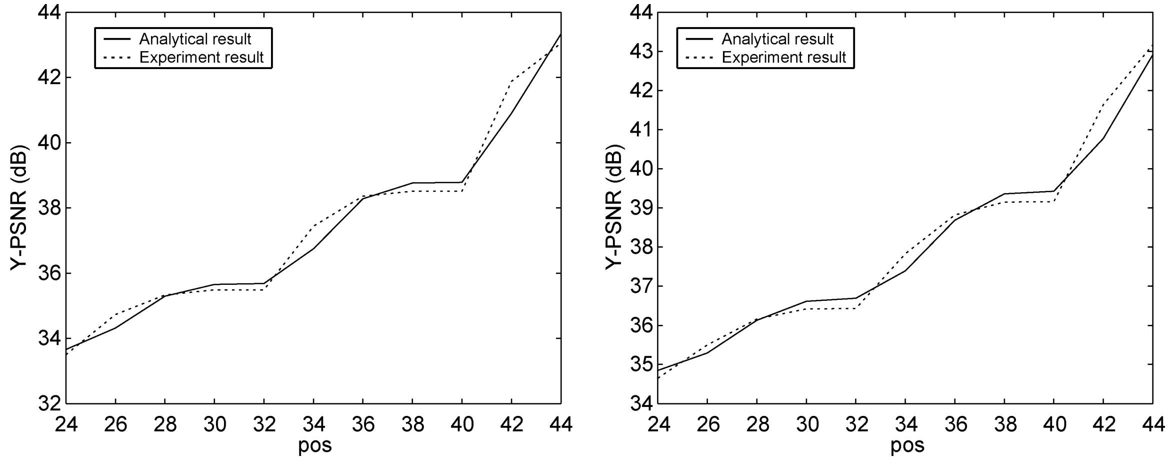


Fig. 4. Validation of (8) using the "Coast Guard" sequence. The loss in PSNR due to watermarking is shown versus the value of pos for $\mathrm{QP} = 10$ (left) and $\mathrm{QP} = 7$ (right).

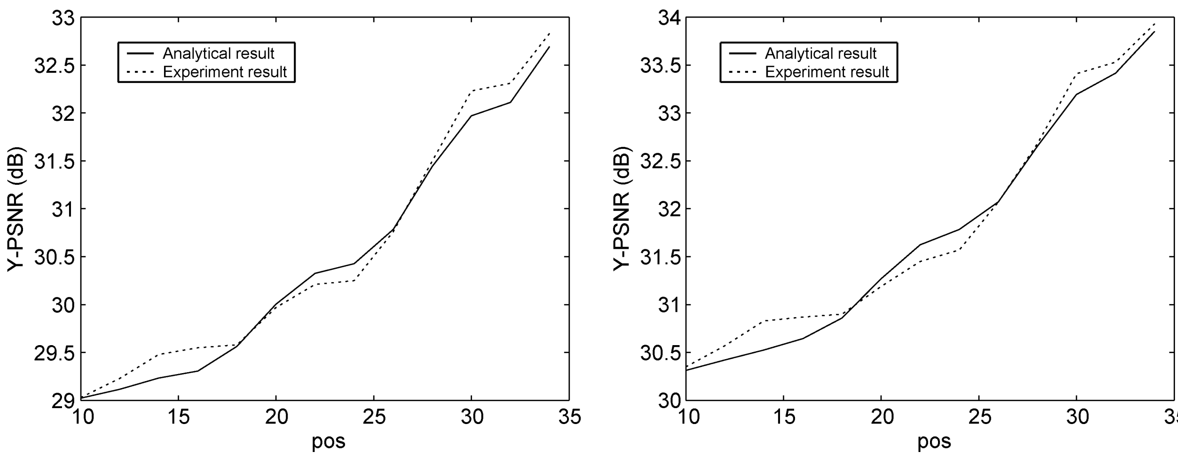


Fig. 5. Validation of (10) using the "Coast Guard" sequence. The loss in PSNR due to watermarking is shown versus the value of pos for $\mathrm{QP} = 10$ (left) and $\mathrm{QP} = 7$ (right).

with an index larger or equal to pos will be subject to watermarking, $E(\mathrm{MSE_{DCT}})$ for an intra block can be computed as follows:

$$E(\mathrm{MSE_{DCT}}) = \frac{1}{64}\sum_{i=pos}^{63} E\left(\left|\widehat{\mathrm{OAC}}_i - \widehat{\mathrm{OAC}}_i^W\right|^2\right)$$
$$= \frac{(3\mathrm{QP}-\mathrm{QP}_0)^2}{64}\sum_{i=pos}^{63}\left(e^{-2\mathrm{QP}\lambda_i} - e^{-4\mathrm{QP}\lambda_i}\right)$$
$$+ \frac{\mathrm{QP}^2}{16}\sum_{i=pos}^{63}\frac{e^{-6\lambda_i \mathrm{QP}}}{1+e^{-2\lambda_i \mathrm{QP}}} \quad (8)$$

We can do a similar analysis for the inter-coded blocks. In this case, as shown in [2], the quantization and dequantization process is given by:

$$\textit{Quant.} \quad \mathrm{AC}_i = \left\lfloor \frac{\mathrm{OAC}_i}{2\mathrm{QP}} + 0.5 \right\rfloor$$
$$\textit{Dequant.} \quad \widehat{\mathrm{OAC}}_i = \begin{cases} (2AC_i + sign(\mathrm{AC}_i)), & \mathrm{AC}_i \neq 0 \\ \mathrm{QP} - sign(\mathrm{AC}_i)\mathrm{QP}_0, & \\ 0, & \mathrm{AC}_i = 0 \end{cases}$$
$$\text{where} \quad \mathrm{QP}_0 = \begin{cases} 1, & \mathrm{QP\ is\ even} \\ 0, & \mathrm{QP\ is\ odd.} \end{cases} \quad (9)$$

The $E(\mathrm{MSE_{DCT}})$ for inter block can then be found to be

$$E(\mathrm{MSE_{DCT}}) = \frac{(3\mathrm{QP}-\mathrm{QP}_0)^2}{64}\sum_{i=pos}^{63}\left(e^{-\mathrm{QP}\lambda_i} - e^{-3\mathrm{QP}\lambda_i}\right) + \frac{\mathrm{QP}^2}{16}\sum_{i=pos}^{63}\frac{e^{-5\lambda_i \mathrm{QP}}}{1+e^{-2\lambda_i \mathrm{QP}}}. \quad (10)$$

We emphasize that the MSE of DCT coefficients calculated in (8) and (10) are due to watermarking only. The total MSE of DCT coefficients is the superposition of the MSE due to watermarking and the MSE due to quantization. The MSE we calculated here is useful when one wants to quantify the quality loss due to watermarking. Furthermore, (8) and (10) can play a role for controlling the value of the cutoff zigzag scan position pos.

To experimentally verify (8) and (10), we encoded the *Cost Guard* CIF sequence using the H.263 TMN8 codec [23]. The values of the pdf shape parameter $\lambda_i (1 \leq i \leq 63)$ were estimated from the original data. The PSNR between the reconstructed frames and watermarked reconstructed frames are shown in Figs. 4 and 5. The figures show the theoretical curve based on (8) and (10), as well as the empiric results. The close match between theoretical and empirical results shows the usefulness of our derivations.

TABLE III
PERFORMANCE OF FEW SCHEME COMPARED TO SYNTAX-BASED ERROR DETECTION: "CAR PHONE" SEQUENCE

Error detection scheme	Coded bit rate (kbps)	Coding PSNR (dB)	△ PSNR (dB)	Error detection rate (%)	Error correctly located rate (%)	Avg. decoded PSNR improvement (dB)
Syntax-based (BER=$5 \cdot 10^{-3}$)	333.76	35.36	–	49.1	3.4	–
Syntax-based (BER=10^{-3})	333.76	35.36	–	37.0	4.9	–
Syntax-based (BER=$5 \cdot 10^{-4}$)	333.76	35.36	–	35.9	3.4	–
Syntax-based (BER=10^{-4})	333.76	35.36	–	30.3	5.4	–
FEW (BER=$5 \cdot 10^{-3}$)	306.18	34.53	0.44	76.1	30.2	3.6
FEW (BER=10^{-3})	306.18	34.53	0.44	62.6	45.4	3.8
FEW (BER=$5 \cdot 10^{-4}$)	306.18	34.53	0.44	62.2	49.6	3.5
FEW (BER=10^{-4})	306.18	34.53	0.44	61.4	50.7	2.3

V. SIMULATION RESULTS AND COMPARISONS

A. *Basis of Comparison and Simulation Environment*

This section describes simulations in which we numerically evaluate the performance of the proposed error detection. Artifacts introduced by watermarking are quantified by PSNR computed as below:

$$\mathrm{PSNR}(f,z) = 10log_{10}\left[\frac{255^2}{\frac{1}{N_f}\sum_{\forall(m,n)}\left(z(m,n)-f(m,n)\right)^2}\right] \tag{11}$$

where N_f is the total number of pixels over which the PSNR is calculated, $f(m,n)$ is the original video frame and $z(m,n)$ is the reconstructed and watermarked frame. $\Delta\mathrm{PSNR}$ indicates the PSNR difference between the watermarked and nonwatermarked reconstructed video frames evaluated at the *same* encoded bit rate for fair comparison. For example, the 0.44 dB $\Delta\mathrm{PSNR}$ shown in Table III is computed as following. First, get the PSNR when encode without applying the watermarking scheme at the coded bit rate 306.18 kbps, which is the same coded bit rate FEW scheme results. Then we compare the PSNR we get in first step (34.97 dB approximately) with the PSNR that FEW scheme result, which is 34.53 dB, the difference between these two PSNRs is the $\Delta PSNR$ we have in Table III.

We used a modified TMN8 [23] as video compression and decompression simulation platform. Our watermark embedding and detection modules, as well as a "syntax checking module", are added.

In order to evaluate the robustness of the FEW scheme for different degrees of motion, three different standard video test sequences were applied in simulations. These sequences were (240 frames of) *Akiyo*, *MotherandDaughter* and *CarPhone*. All are in CIF format, and encoded at a frame rate of 30 frames/s. In order to focus on the detection capability for erroneous quantized DCT coefficient, we only cast bit errors on those coded bits that represent quantized DCT coefficients. Therefore, our simulations preclude influences of erroneous motion vectors and headers. The GOB header option in H.263 is enabled and the GOB headers act like a synchronization words at every beginning of a MB slice, thus the error propagation of VLC is stopped on the boundary of MB slices. The intra-mode update interval parameter for MB is set to default value, i.e., 0, in the H.263 encoder; for the experiments with frequent intra refresh but for JVT [24] platform, please refer to [25].

We used the binary symmetric channel (BSC) as a model for the wireless communication channel, because under sufficient error correction and interleaving, any real channel is equivalent to transmitting the unprotected data through a BSC channel. For sufficiently strong FEC and enough interleaving, the random bit error of the equivalent BSC channel is in the order of $10^{-4} \sim 10^{-3}$. Therefore we concentrated our simulations on bit error rate (BER) in this range. More specifically, the BER of the BSC channel was set to 10^{-4}, $5 \cdot 10^{-4}$, 10^{-3} and $5 \cdot 10^{-3}$.

From the analysis in Section IV, we can see that the selection of a value for pos depends on the statistics of DCT coefficients of the blocks to be watermarked. Since Y, U, and V DCT blocks as well as intra- and inter-coded DCT blocks have different statistics, different pos values would in principal be needed. In the simulation, we detect the erroneous MBs using the proposed scheme with (fixed) values for $pos = 37, 22, 15$ for intra-Y/inter-Y/both intra-/inter- U&V block. These pos values are chosen according to empirical experience. To avoid the influence of different bit rate control scheme, we set the quantization parameter QP to a constant value of 10 for both intra- and inter-coded blocks.

One point we would like to make clear in simulation is that we do not select different sets of pos value optimally for each test sequence respectively. The reason is that choosing **different** sets of pos value for each sequence will result in an overhead for synchronizing encoder and decoder about the pos values used. Also, sometime it is not possible to get the statistics of DCT blocks ahead of time (e.g. real time encoding). Instead, we follow a more pragmatic approach to select just **one** set of pos value that will have good performance for any sequence. This will make it easy and clear to deploy and understand, while one can still analyze the PSNR loss using (8) and (10) when these statistics is given. Since the test sequences we used in simulation represent different level of motion complexity well, we do believe that the results presented are representative for the performance of real system, and the set of pos value we selected is suitable for most sequences.

TABLE IV
COMPARISON OF FEW SCHEME FOR TWO VALUES OF pos AND TWO BIT ERROR RATES USING THE "CAR PHONE" SEQUENCE

Error detection scheme	Coded bit rate (kbps)	Coding PSNR (dB)	$\triangle$ PSNR (dB)	Error detection rate (%)	Error correctly located rate (%)	Avg. decoded PSNR improvement (dB)
FEW ($pos = 22$, BER=10^{-3})	306.62	34.51	0.46	59.3	36.4	3.3
FEW ($pos = 22$, BER=$5 \cdot 10^{-4}$)	306.62	34.51	0.46	63.1	40.4	3.5
FEW ($pos = 37$, BER=10^{-3})	313.71	34.78	0.28	44.5	23.1	2.3
FEW ($pos = 37$, BER=$5 \cdot 10^{-4}$)	313.71	34.78	0.28	47.5	28.3	2.4

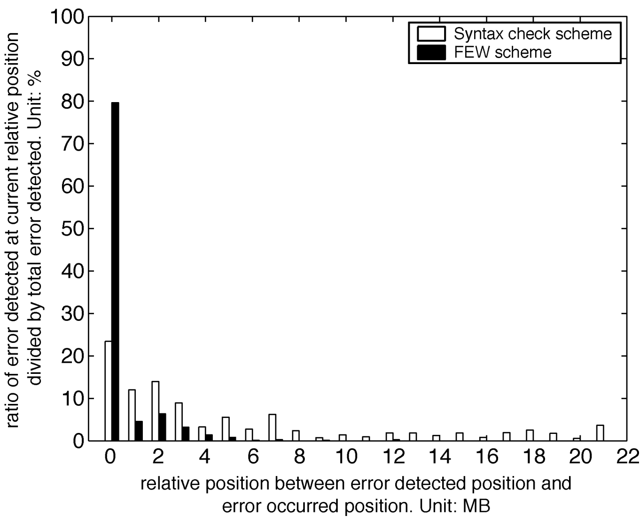


Fig. 6. Capability to locate erroroneous MBs at the correct position. Horizontal axis: the difference in position between the MB affected by a bit error and the MB where the error is first detected. Vertical axis: the percentage of cases. The solid bars indicate the results for the FEW scheme, and the open bars show the results for the syntax-based detection scheme.

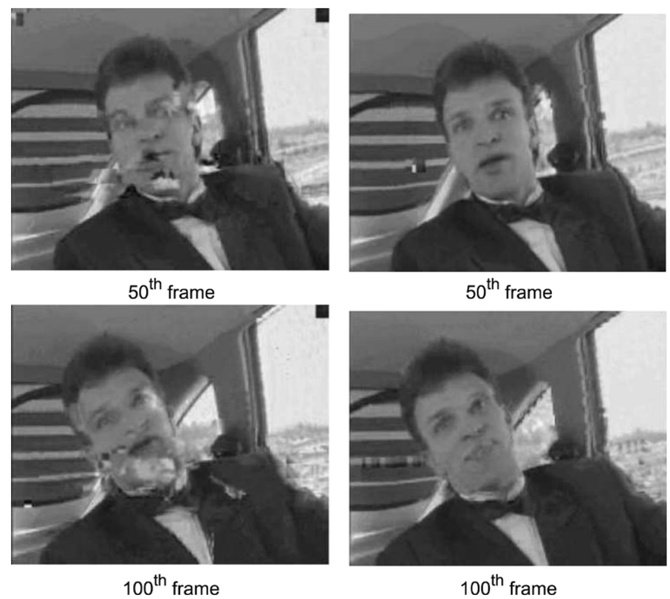


Fig. 7. Two reconstructed frames using different error detection scheme. The results on the left are obtained by using the syntax-based error detection scheme; the results on the right are obtained using the FEW error detection scheme.

B. Results

Tables I–III compare the error detection performance and ΔPSNR for the *syntax-based* error detection scheme [12] and our proposed FEW scheme. The tables show that the proposed FEW scheme:

- doubles the error detection probability;
- improves the probability of correctly locating an error by $3 \sim 13$ times;
- yields a ΔPSNR of less than 0.44 dB.

The effect of the much higher probability of correctly locating an erroneous MB can be seen from Fig. 6. The FEW scheme detects the error at the correct position (indicated by $\text{position} = 0$) with a significantly higher probability than the syntax-based error detection scheme. Therefore, the method that we propose has superior performance in avoiding lagging effects. Also notice in Tables I–III there are some difference between coded rates of FEW and syntax based scheme. This is because FEW forces some coefficients to be zero thus reducing the number of coefficients that need to code.

In Fig. 7, reconstructed frames under $\text{BER} = 5 \cdot 10^{-4}$ situation are shown to compare syntax-based scheme and the FEW scheme. In both schemes, we used simple copying-from-the-last-frame as the error-concealment technique, which involves motion compensation when locating the blocks to copy. From the results, we see that—under the same error concealment technique—the reconstructed frames from the decoder that applies the FEW scheme has better subjective quality due to the significantly stronger error detection ability.

Fig. 8 shows the encoding/reconstructed PSNR comparison graphs for the *Car Phone* sequence. Although after applying the FEW scheme the PSNR is about 0.5 dB less on the encoder side, the PSNR of the reconstructed video frames is much better than the one resulting from syntax-based detection. For instance, Fig. 8(b) shows a PSNR difference of $1 \sim 5$ dB.

Table IV and Fig. 9 show the performance of the FEW scheme for different values of the watermarking control parameter pos. These results clearly illustrate the tradeoff between the error detection property and quality loss. In Table IV, for example, changing the watermark parameter pos from 37 to 22 increases the error detection probability from 44.5% to 59.3%, sacrificing an additional 0.2 dB PSNR loss in the encoder. On the other hand, the pay off of the change can clearly been seen in the right hand side curve in Fig. 9. On average, an increase of 3.5 dB is observed due to the improved error detection capability at the decoder side when decreasing pos from 37 to 22.

It is also interesting to see the difference in error detection and error location capabilities of the FEW scheme and the syntax-based scheme. We carry out the comparison for different QP

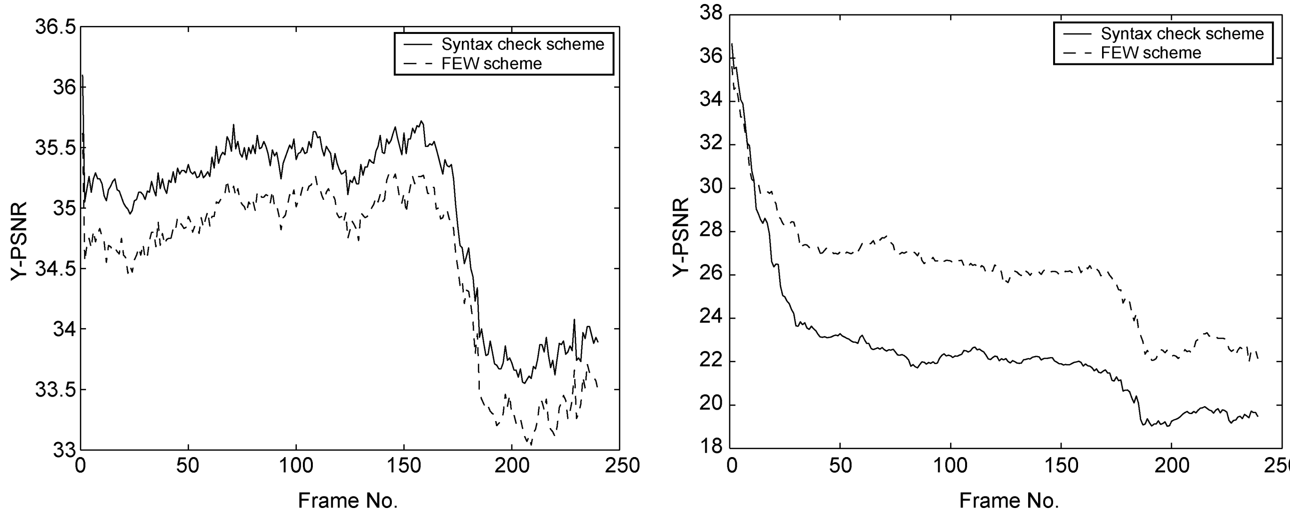


Fig. 8. PSNR of the compressed video at the encoder (left hand side) and reconstructed video at the decoder after channel errors and error detection (right hand side). The solid curve shows the result for the syntax-based error detection scheme, and the dashed curve shows the result for the FEW error detection scheme. Results are computed using the "Car Phone" sequence; channel $\mathrm{BER} = 5 \cdot 10^{-4}$.

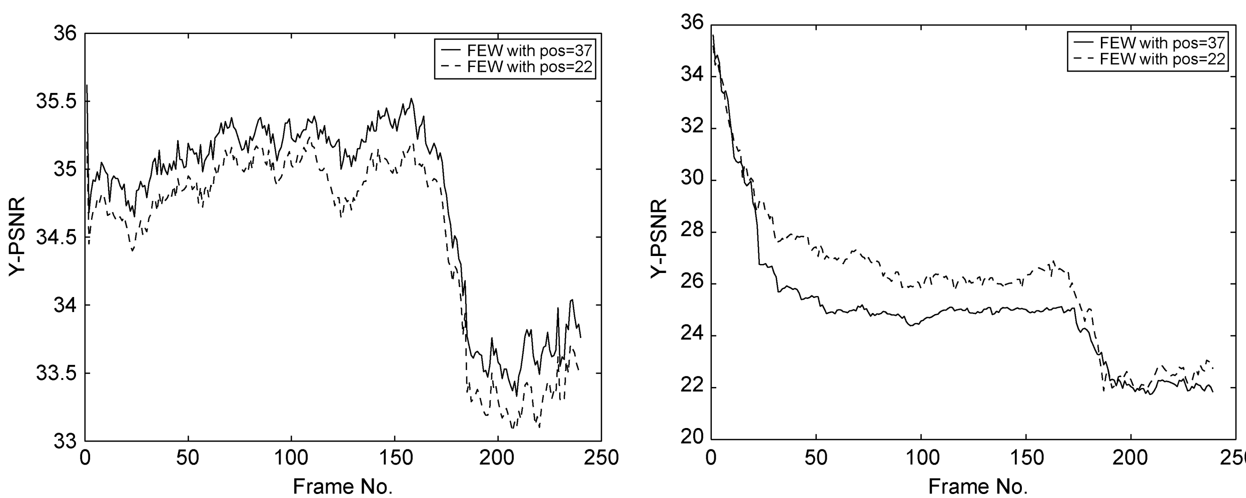


Fig. 9. PSNR of the compressed video at the encoder (left-hand side) and reconstructed video at the decoder after channel errors and error detection (right-hand side). Both curves show results for the FEW error detection scheme. The solid line is for $pos = 37$, the dashed line for $pos = 22$. Results are computed using the "Car Phone" sequence; channel $\mathrm{BER} = 5 \cdot 10^{-4}$.

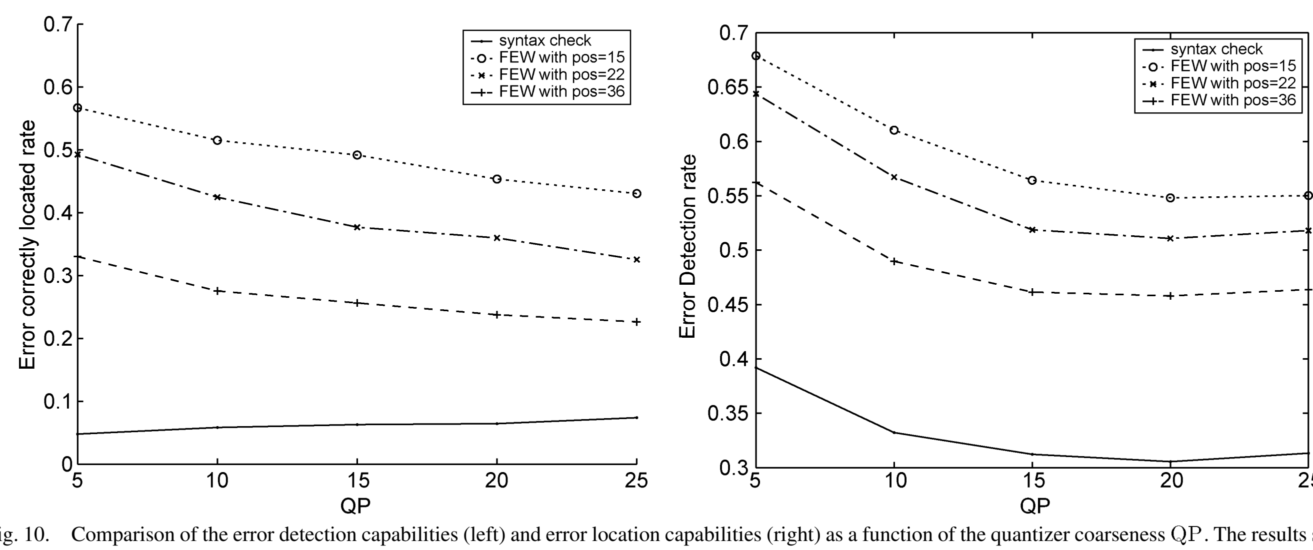


Fig. 10. Comparison of the error detection capabilities (left) and error location capabilities (right) as a function of the quantizer coarseness QP. The results show the performance for the syntax-based error detection (solid curve) and the FEW error detection scheme with $pos = 15$, 22, and 36. Results are computed using the "Car Phone" sequence; channel $\mathrm{BER} = 5 \cdot 10^{-4}$.

values and different pos values. The *Car phone* sequence is used in this experiment. The results are shown in Fig. 10. The first observation is that the FEW scheme always performances better than the syntax-based error detection scheme for all QP

settings evaluated. This is reasonable since we embed information to support the error detection at the decoder. Secondly, for a fixed pos, the FEW error detection capability decreases as the quantization step QP increases. This is because as QP increase, more coefficients whose indexes are larger than pos are *quantized* to zero. These quantized coefficients no longer contribute to the fragile watermark. Consequently, when a MB is corrupted, there is a smaller probability for the watermark pattern to be corrupted, effectively decreasing the error detection capability. An extreme case occurs when QP is very large. All DCT coefficients whose indexes is larger than pos are then quantized to zero, and no watermark is embedded at all. In this case, the performance of the proposed FEW scheme degenerates to performance of the syntax-based error detection scheme.

VI. Conclusions

A watermark-based error detection technique has been proposed in this paper. By embedding a fragile watermark into quantized DCT coefficients during the encoding, the error detection capability of the decoder can be greatly improved compared to *syntax-based* error detection schemes. The PSNR loss at approximately the same encoded bit rate is less than 0.5 dB. However, it was shown that this loss in PSNR is by far compensated if channel bit errors occur. More erroneous DCT blocks can be correctly detected, located and thus concealed. The proposed technique is backward compatible to any existing compression standard, as it does not change the syntax of the bitstream for fixed watermarks.

We have pointed out three aspects of our approach that deserve more research. In the first place, the fragile watermarking method itself can be optimized for optimally trading-off the loss in PSNR and improvement in error detection abilities. Secondly, we have derived an analytical expression for the loss in PSNR as a function of the cutoff zigzag scan position pos, but it would be desirable to also have an analytical expression for the error detection probability as a function of this parameter. Finally, our experiments are based on a fixed value for the cutoff zigzag scan position pos. The optimization of the value of pos on a frame-by-frame, stripe-by-stripe, or even DCT block-by-block basis may further improve the tradeoff between quality (PSNR) loss and error detection ability.

Acknowledgment

The authors would like to thank J. Lan and L. Xie for their discussion. The authors would also like to thank the anonymous reviewers for their helpful comments.

References

[1] *Video Codec for Audio-Visual Services at px64kbit/s*, ITU-T Recommendation H.261, 1993.
[2] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263, Mar. 1996.
[3] *Coding of Moving Pictures and Associated Audio for Storage Media Up to 1.5 Mb/s: Video*, ISO/IEC DIS 11172-2, 1991.
[4] *Information Technology—Generic Coding of Moving Pictures and Associated Audio: Video*, ISO/IEC 13818-2, 1994.
[5] ISO/IEC JTC1/SC29/WG11 14496-2, "Amd X, Coding of Moving Pictures and Audio," International Standard, Maui, HI, Dec. 1999.
[6] B. Sklar, "Raleigh fading channels in mobile digital communication systems, part I: Characterization," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 90–100, Sep. 1997.
[7] Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.
[8] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bit-rate video coding systems," *IEEE J. Select. Areas Commun.*, vol. 15, no. 9, pp. 1764–1774, Dec. 1997.
[9] R. Talluri, I. Moccagatta, Y. Nag, and G. Cheung, "Error concealment by data partitioning," *Signal Process.: Image Commun.*, vol. 14, pp. 505–518, 1999.
[10] L. Ducla-Soares and F. Pereira, "Error resilience and concealment performance for MPEG-4 frame-based video coding," *Signal Process.: Image Commun.*, vol. 14, pp. 447–472, 1999.
[11] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Process.: Image Commun.*, vol. 14, pp. 473–492, 1999.
[12] M. Barni, F. Bartolini, and P. Bianco, "On the performance of syntax-based error detection in H.263 video coding: a quantitative analysis," in *Electronic Imaging 2000—SPIE Conf. Image and Video Communications*, San Jose, CA, Jan. 2000.
[13] T. S. Wang, P.-C. Chang, C.-W. Tang, H.-M. Hang, and T. Chiang, An Error Detection Scheme Using Data Embedding for H.263 Compatible Video Coding, Jul. 2000.
[14] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Real-time labeling for MPEG compressed video," *J. Vis. Commun. Image Repres.*, Dec. 1998.
[15] F. Hartung and B. Girod, "Digital watermarking of MPEG-2 coded video in bitstream domain," in *Proc. IEEE ICASSP'97*, 1997, pp. 2621–2624.
[16] Q. Cheng and T. S. Huang, "Blind digital watermarking for images and videos and performance analysis," in *Proc. IEEE Multimedia and Expo 2000 (ICME 2000)*, vol. 1, 2000, pp. 389–392.
[17] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 1996, pp. 219–222.
[18] J. Fridrich, "A hybrid watermark for tamper detection in digital images," in *Proc. 5th Int. Symp. Signal Processing and Its Applications (ISSPA '99)*, Brisbane, Australia, 1999, pp. 301–304.
[19] D. Kundur and D. Hatzinakos, "Digital watermark for telltale tamper proofing and authentication," *Proc. IEEE*, vol. 87, no. 7, pp. 1167–1180, Jul. 1999.
[20] F. Muller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electron. Lett.*, vol. 15, no. 20, pp. 664–665, Sept. 1979.
[21] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistics analysis of the 2D-DCT coefficients of the differential signal of images," *Signal Processing: Image Commun.*, vol. 4, pp. 477–488, 1992.
[22] K. A. Birney and T. R. Fischer, "On the modeling of DCT and subband image data for compression," *IEEE Trans. Image Process.*, vol. 4, no. 2, pp. 186–193, Feb. 1995.
[23] ITU-T Study Group 16, TMN8 Video Codec Test Model Near-Term Version 8[R], Q15-B-46. Sunriver, 1997.
[24] Joint Committee Draft (CD), Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, May 2002.
[25] P. Zhou and Y. He, "A fragile watermark error detection scheme for jvt," in *Proc. 2003 Int. Symp. Circuits and Systems (ISCAS '03)*, vol. 2, May 2003, pp. 956–958.

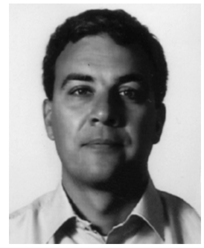
Minghua Chen received the B.Eng. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1999 and 2001, respectively. He is currently pursuing the Ph.D degree in the Department of Electrical Engineering and Computer Science, University of California at Berkeley.

He received a Pao Family fellowship from the University of California at Berkeley in 2001. He is co-author of the book IPv6 Principle and Practice (Beijing, China: People's Posts and Telecommunication Press, 2000). His research interests include signal processing, networking, and wireless communication, with current emphasis on video processing and transmission.

Yun He (M'97–SM'01) received the B.S. degree in signal processing from Harbin Shipbuilding Institute, Harbin, China, in 1982, the M.S. degree in ultrasonic signal processing from Shanghai Jiaotong University, Shanghai, China in 1984, and the Ph.D. degree in image processing from Liege University, Liege, Belgium, in 1989.

She is currently a Full Professor at Tsinghua University, Beijing, China. Her research interests include picture-coding theory and methodology, picture-coding algorithm software and hardware complexity analysis, and multiview and three-dimensional picture coding.

Dr. He is a member of the IEEE Circuits and Systems Society and the IEEE Signal Processing Society.



Reginald L. Lagendijk (S'87–M'90–SM'97) received the M.Sc. and Ph.D. degrees in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1985 and 1990, respectively.

He became an Assistant Professor and Associate Professor at Delft University of Technology in 1987 and 1993, respectively. He was a Visiting Scientist in the Electronic Image Processing Laboratories of Eastman Kodak Research, Rochester, NY, in 1991, and a Visiting Professor at Microsoft Research and Tsinghua University, Beijing, China, in 2000 and 2003. Since 1999, he has been a Full Professor in the Information and Communication Theory Group of the Technical University of Delft. He has been involved in the European Research projects DART, SMASH, STORit, DISTIMA, and CERTIMARK. He is currently leading a number of projects in the field of intelligent information processing for wireless and *ad-hoc* communications. He is author of *Iterative Identification and Restoration of Images* (Norwell, MA: Kluwer, 1991), and co-author of *Motion Analysis and Image Sequence Processing* (Norwell, MA: Kluwer, 1993) and *Image and Video Databases: Restoration, Watermarking, and Retrieval* (New York: Elsevier, 2000). At present, his research interests include signal processing and communication theory, with emphasis on visual communications, compression, analysis, searching, and watermarking of image sequences.

Dr. Lagendijk has served as Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and is currently Area Editor of EURASIP's journal *Signal Processing: Image Communication*. He was a member of the IEEE Signal Processing Society's Technical Committee on Image and Multidimensional Signal Processing.