A Framework for Robust Watermarking of H.264-Encoded Video With Controllable Detection Performance

Maneli Noorkami, *Member, IEEE*, and Russell M. Mersereau, *Fellow, IEEE*

***Abstract*—As H.264 digital video becomes more prevalent, the need for copyright protection and authentication methods that are appropriate for this standard will emerge. This paper proposes a robust watermarking algorithm for H.264. We employ a human visual model adapted for a 4 × 4 discrete cosine transform block to increase the payload and robustness while limiting visual distortion. A key-dependent algorithm is used to select a subset of the coefficients that have visual watermarking capacity. Furthermore, the watermark is spread over frequencies and within blocks to avoid error pooling. This increases the payload and robustness without noticeably changing the perceptual quality. We embed the watermark in the coded residuals to avoid decompressing the video; however, we detect the watermark from the decoded video sequence in order to make the algorithm robust to intraprediction mode changes. We build a theoretical framework for watermark detection based on a likelihood ratio test. This framework is used to obtain optimal video watermark detection with controllable detection performance. Our simulation results show that we achieve the desired detection performance in Monte Carlo trials. We demonstrate the robustness of our proposed algorithm to several different attacks.**

***Index Terms*—Compressed domain, error pooling, generalized Gaussian, human visual model, H.264, likelihood ratio test, video watermark detection, video watermarking.**

I. Introduction

The advent of digital television, the appearance of digital versatile disks (DVDs), and the transfer of video files over the Internet demonstrate the importance of digital video. Despite the success of MPEG-2 in the video coding industry, a new standard, H.264, with higher compression efficiency, is poised to replace it. As H.264 digital video becomes more prevalent, the industry will need copyright protection and authentication methods that are appropriate for it. In this paper, we propose an H.264 watermarking algorithm that fulfills these needs.

Since video signals are usually stored and distributed in a compressed format, it is often impractical to first decode the video sequence, embed the watermark, and then re-encode it. An alternative to a low-complexity video watermarking algorithm is to embed the watermark in the compressed domain. Unfortunately, the large body of MPEG-2 video watermarking algorithms cannot be applied directly to H.264 because of differences in the standards. A detailed description of the H.264 standard is given in [1] and [2].

A few recently published papers have discussed embedding a watermark in the H.264 bitstream sequence. In [3], the authors propose a hybrid watermarking scheme that embeds a robust watermark in the discrete cosine transform (DCT) coefficients and a fragile watermark in the motion vectors. In [4], we proposed a low-complexity scheme that embeds one watermark bit in one of the quantized ac coefficients within each macroblock of an I-frame. In [5], the authors present a blind watermarking algorithm that embeds the watermark in H.264 I-frames. That scheme survives H.264 compression attacks with more than a 40:1 compression ratio in I-frames; however, it requires decompressing the video in order to embed the watermark. The algorithms proposed in [3] and [4] embed the watermark in the compressed video, but these algorithms are not robust against common watermarking attacks. This is because the watermark is embedded in and extracted from the I-frame residuals. Any simple processing, such as filtering followed by re-encoding by an H.264 encoder, changes the intramacroblock prediction modes and, thus, the residuals, which makes watermark recovery impossible.

The goal of this paper is to present a robust watermarking algorithm for H.264. To achieve this goal, we embed the watermark in the residuals to avoid decompressing the video and to reduce the complexity of the watermarking algorithm. However, the watermark is extracted from the decoded video sequence to make the algorithm robust to intraprediction mode changes. Since H.264's high compression performance leaves little room for an imperceptible signal to be inserted, we employ a human visual model to increase the payload and add robustness while limiting visual distortion. Watson *et al.* [7] derived a model for distortion perception in 8 × 8 DCT blocks [6], [7]. This perceptual model has been used in [8] and [9] to design watermarking algorithms for still images and MPEG-2 video. However, H.264 uses a 4 × 4 transform [10] instead of an 8 × 8 transform. The 4 × 4 transform is an integer orthogonal approximation to the DCT. Since the transform is defined by exact integer operations, inverse-transform mismatches are avoided. In this paper, we extend the human visual model for a 4 × 4 DCT block. If all of the coefficients with visual capacity for watermark embedding are used, the visual quality of the video will be degraded. We propose to embed the watermark in a selected subset of the coefficients that have visual watermarking capacity by using a

Manuscript received April 12, 2006; revised October 5, 2006. This work was supported in part by Texas Instruments Incorporated. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Inald Lagendijk.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: maneli@ece.gatech.edu; rmm@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2006.890306

key-dependent algorithm. This makes the algorithm more robust to malicious attacks. Furthermore, we have designed our algorithm so that the watermark bits are spread over frequencies and blocks. This reduces the error pooling effect described by Watson [7]. Error pooling has not been considered in previous perceptual watermarking algorithms [8], [9].

We build a theoretical framework for watermark detection using a likelihood ratio test with an assumed generalized Gaussian probability distribution for the ac coefficients of the DCT. We show that the sum of the products of watermark bits and selected DCT coefficients of the original video scaled by their quantization step size are a sufficient statistic to detect the watermark. We also show that the performance of the watermark detector depends upon the conditional mean of the sufficient statistic under the hypothesis that the watermark exists in the DCT coefficients. This mean value depends upon three parameters: the average of the squares of H.264 quantization step sizes of selected DCT coefficients, the standard deviation of selected DCT coefficients, and the number of DCT coefficients N, over which the sufficient statistic is computed. We cannot control the standard deviation of the DCT coefficients or the H.264 quantization step size, but when detecting watermarks in video, we can control the number of watermarked coefficients used to compute the sufficient statistic. This is not the case with images since, for images, there is a limited number of watermark bits that can be embedded in each image before the watermark is visible. Therefore, our video watermark detection algorithm calculates N to obtain the desired probability of a detection P_D for a given probability of a false alarm P_F. Our simulation results show that the theoretically chosen value for N does lead to the desired values of P_D and P_F in Monte Carlo trials. The simulation results also show that our proposed watermarking scheme is robust to 3 × 3 Gaussian filtering, 50% cropping, addition of white noise $\mathcal{N}(0, 0.001)$, and a trivial deliberate attack.

This paper is organized as follows. In Section II, we derive the human visual model for a 4 × 4 DCT from that of an 8 × 8 DCT. In Section III, we present the proposed watermarking scheme. Section IV discusses the watermark embedding procedure. In Section V, we build a theoretical framework for watermark detection and present our video watermark detection algorithm based on this framework. Simulation results are given in Section VI.

II. Human Visual Model for the 4 × 4 DCT

The DCT coefficients $X_{i,j}$ of an $M \times M$ block of image pixels $x(n_1, n_2)$ expand the block in terms of the DCT basis functions as follows:

$$x(n_1, n_2) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} X_{i,j} c_i c_j \cos\left(\frac{\pi(2n_1+1)i}{2M}\right) \times \cos\left(\frac{\pi(2n_2+1)j}{2M}\right) \quad (1)$$

where

$$c_i = \begin{cases} \sqrt{1/M}, & i = 0 \\ \sqrt{2/M}, & i > 0. \end{cases} \quad (2)$$

Human visual sensitivity for each DCT basis function varies as a function of its frequency. In [11], the authors measured quantization error thresholds at various DCT frequencies in an 8× 8 DCT block. Here, we extend the quantization error visibility thresholds for an 8 × 8 DCT block to those appropriate for a 4× 4 DCT block.

The basis functions of a 4 × 4 DCT are defined as

$$c_{i_4} c_{j_4} \cos\left(\frac{\pi(2n_1+1)i_4}{2 \times 4}\right) \cos\left(\frac{\pi(2n_2+1)j_4}{2 \times 4}\right) \qquad 0 \le i_4, j_4 \le 3 \quad (3)$$

and the basis functions of an 8 × 8 DCT are defined as

$$c_{i_8} c_{j_8} \cos\left(\frac{\pi(2n_1+1)i_8}{2 \times 8}\right) \cos\left(\frac{\pi(2n_2+1)j_8}{2 \times 8}\right) \qquad 0 \le i_8, j_8 \le 7. \quad (4)$$

Comparing (3) and (4) suggests that the basis function $i_4 j_4$ of a 4 × 4 DCT will have the same frequencies as the basis function $i_8 j_8$ of an 8 × 8 DCT, if

$$\begin{aligned} i_8 &= 2 \times i_4 \\ j_8 &= 2 \times j_4 \end{aligned} \quad (5)$$

hold. Since for all $0 \le i_4, j_4 \le 3$, there exists an i_8 and j_8 in the range $0 \le i_8, j_8 \le 7$, the visibility thresholds of a 4 × 4 DCT basis function can be derived from the known visibility thresholds of 8 × 8 DCT basis functions. The factors c_i and c_j cause the amplitude of the errors for a 4 × 4 DCT to be twice the size as those of an 8 × 8 DCT. Thus, to obtain invisibility, the visibility threshold of the $i_4 j_4$ basis function of a 4 × 4 DCT is obtained by dividing the visibility threshold of the $2i_4 2j_4$ basis function of an 8 × 8 DCT by 2. We used the quantization matrices given in [11] for an 8 × 8 DCT block in the YC_bC_r color space to derive the quantization matrices for a 4 × 4 DCT block. The actual visibility threshold $t_{i,j}$ is half the quantization step size $q_{i,j}$.

To obtain an image-dependent quantization matrix, two other effects from [7], luminance masking and contrast masking, are exploited. A simple solution to approximate luminance masking is with a power function as

$$t_{i,j,k} = t_{i,j} (X_{0,0,k} / \bar{X}_{0,0})^{a_T}. \quad (6)$$

$\bar{X}_{0,0}$ is the dc coefficient corresponding to the mean luminance of the display and $X_{0,0,k}$ is the dc coefficient of block k. a_T controls the degree to which this masking occurs. We choose $a_T = 0.649$ as suggested in [7]. Finally, contrast masking gives the masked threshold $m_{i,j,k}$ for a DCT coefficient $X_{i,j}$ of block k $(X_{i,j,k})$ as

$$m_{i,j,k} = \max\left[t_{i,j,k}, |X_{i,j,k}|^{w_{i,j}} t_{i,j,k}^{1-w_{i,j}}\right] \quad (7)$$

where $w_{i,j}$ is between 0 and 1. We choose $w_{i,j} = 0.7$ as the authors of [7] recommend.

The image-dependent approach ensures that each error falls below a threshold. Furthermore, Watson noted that the visibility of an error is not based solely on the visibility of the largest error, but instead reflects a pooling of errors over frequency and within a block. We spread the watermark over frequencies and blocks to reduce error pooling, which has not been done in the previous perceptual watermarking algorithms [8], [9].

III. Proposed Method

In this paper, the masked error visibility threshold $m_{i,j,k}$ is computed for each coefficient $X_{i,j}$ in block k. This threshold is divided by the H.264 quantization step size for that block Q_k to determine the capacity of that coefficient $s_{i,j,k}$ for holding watermark bits

$$s_{i,j,k} = \text{floor}(m_{i,j,k}/Q_k). \tag{8}$$

We denote the set of coefficients with visual capacity for watermark embedding (i.e., capacity greater than zero) as

$$CV = \{c_1, \ldots, c_n\} = \{X_{i,j,k} | s_{i,j,k} \neq 0, \forall i, j, k\}. \tag{9}$$

Inserting watermark bits in all of the coefficients in CV, creates visible artifacts. The algorithms in [8] and [9] only embed the watermark in those coefficients in CV that are greater than their corresponding visibility thresholds $m_{i,j,k}$. However, this significantly limits the number of watermark bits that can be embedded. If more watermark bits need to be embedded in the video frame, the parameters in the visual model that define the impact of masking need to be increased (a_T and $w_{i,j}$) [7]. The danger is that this may assume a greater benefit from masking than is actually available, resulting in noticeable visual artifacts. Furthermore, an adversary can more easily determine the locations of the watermark bits, making the algorithm less robust to attacks.

In this paper, a coefficient-selection algorithm chooses a subset of the coefficients in CV. A secret key controls the coefficient-selection process. The algorithm generates a palette that contains the actual locations of the watermark bits. The owner keeps this palette for watermark detection. This palette can be considered as an automatically generated confirmation number or password. Since the coefficient-selection algorithm is controlled by a key, the attacker does not know the actual location of watermark coefficients in CV. To be confident of eliminating the watermark, an attacker needs to modify most of the coefficients in CV, which creates visible artifacts.

We designed the coefficient-selection algorithm to spread the watermark over frequencies and blocks. For each 4 × 4 block, each coefficient is ranked by a key. However, to spread watermark bits over frequencies, we give a higher ranking for watermark embedding to those frequencies with the fewest coefficients in CV more frequently. We embed the watermark only in those coefficients whose magnitude is greater than a threshold T_{cof}. The algorithm spreads the watermark bits over blocks by limiting the number of watermark bits T_{block} that can be embedded in each block. Therefore, the algorithm embeds the watermark in the T_{block} highest ranking coefficients in CV that are greater than T_{coef}. One advantage of this strategy is that we can easily increase the watermark payload by increasing T_{block} or decreasing T_{cof}. Our experiments show that moderate relaxation of these thresholds increases the number of embedded watermark bits without impairing the visual quality, because the error pooling effect is limited. Furthermore, T_{block} and T_{cof} can be adaptive to control the number of watermark bits in a frame, depending on texture or temporal masking. This algorithm is an initial attempt to reduce error pooling while obtaining a large payload, but further research is needed to develop an optimal and more secure technique. Fig. 1 shows the structure of our proposed algorithm.

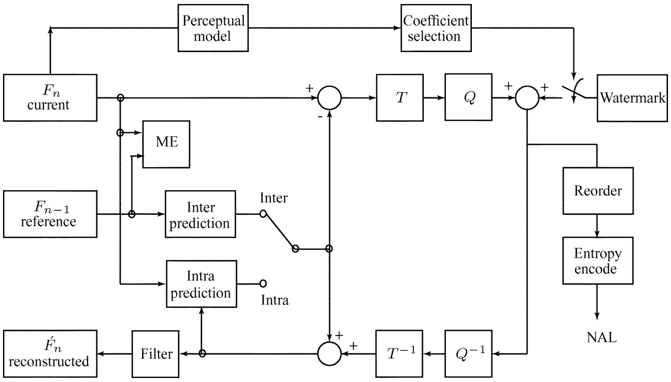


Fig. 1. Proposed watermarking system.

If we ignore the top path in this block diagram, this is the structure of an H.264 encoder. Each macroblock of the current frame is predicted in either intra or interprediction mode. The difference between the current macroblock and the prediction signal is the residual. The residuals are transformed, quantized, reordered, and entropy coded and finally written to the bitstream. There is a backward path in the encoder that reconstructs the current frame. Our watermarking algorithm embeds the watermark in the encoder. The perceptual model finds the location of coefficients with watermarking capacity CV using the original video frame. The coefficient selection algorithm selects a subset of coefficients from the coefficients with watermarking capacity and the watermark bits are added to the quantized DCT residuals at those locations.

When embedding the watermark in the compressed video, there are two different scenarios. In the first scenario, the watermark is inserted in the encoder. In this scenario, the error induced by watermarking will be corrected in future prediction and will not propagate within I-frames and to P-frames. In this scenario, more watermark bits can be embedded in the compressed video while maintaining high perceptual quality. However, there will be an increase in the bit rate of the frames that are predicted from the watermarked frames. We consider this scenario in this paper. In the second scenario, the watermark is embedded in the bitstream. In this scenario, the error will propagate within I-frames and to P-frames, and maintaining a high visual quality is a bigger problem than an increase in video bit rate. A drift compensation signal can be used to compensate for the propagated error as suggested in [12].

IV. Watermark Embedding

We propose to embed watermark information in the quantized residuals of I-frames. Thus, only entropy decoding is required to embed the watermark, and the watermark embedding algorithm has low computational complexity. We use a bipolar watermark $W \in \{-1, 1\}$ with zero mean and variance one.

After any simple attack applied to the decoded video and subsequent re-encoding, the residuals will change because the I-macroblock prediction modes will change. The linearity property of the DCT, however, guarantees that the watermark is still present in the decoded video sequence, and we can still detect it with high probability. In the following, we denote the original pixel values by i_{ijk}, the prediction by p_{ijk}, the residual by r_{ijk}, and their corresponding DCTs by I_{ijk}, P_{ijk}, and R_{ijk}. Assume s is a DCT coefficient, then $\tilde{s}$, s', and $\hat{s}$ represent the quantized, watermarked, and attacked coefficient, respectively. The addition of the prediction and residual is equal to the original pixel. This can be written as

$$i_{ijk} = p_{ijk} + r_{ijk}. \tag{10}$$

By the linearity property of the DCT

$$I_{ijk} = P_{ijk} + R_{ijk}. \tag{11}$$

The watermark is inserted onto the quantized residual as

$$\tilde{R}'_{ijk} = \tilde{R}_{ijk} + W_{ijk} \tag{12}$$

and

$$R'_{ijk} = (\tilde{R}_{ijk} + W_{ijk}) \times Q_k. \tag{13}$$

Thus

$$I'_{ijk} = I_{ijk} + W_{ijk}Q_k \tag{14}$$

and the addition of the watermark to the quantized DCT residuals is the same as the addition of the watermark times the quantization step size to the original DCT coefficients. When common signal processing operations or watermarking attacks on the video change the prediction mode of the block, the residual and prediction will change to $\hat{r}_{ijk} \neq r'_{ijk}$, and $\hat{p}_{ijk} \neq p_{ijk}$. However, if the video quality is still acceptable, then $\hat{r}_{ijk} + \hat{p}_{ijk} = \hat{i}_{ijk} \approx i'_{ijk}$, and the watermark can still be extracted from the decoded video sequence.

V. Watermark Detection

Watermark detection is a classical detection problem [13] where one hypothesis states that the watermark is present and the other states that the watermark is not present. Detecting the watermark requires choosing between the two hypotheses. The observations under the two hypotheses are as follows:

$$H_0 : y_\ell = I_\ell, \tag{15}$$

$$H_1 : y_\ell = I_\ell + W_\ell Q_\ell \tag{16}$$

where I_ℓ is the selected DCT coefficient of the video frame, Q_ℓ is the H.264 quantization step size selected by the video encoder for that coefficient, and W_ℓ is the watermark bit chosen from a bipolar watermark sequence $W \in \{-1, 1\}$ with zero mean and variance one. In this section, we have dropped the indices i, j for simplicity, and the index ℓ denotes the ℓ^{th} watermark bit or the ℓ^{th} DCT coefficient. It has been shown that the ac coefficients of the DCT are well modeled by a generalized Gaussian distribution [14], [15], which can be written as

$$p_{I_\ell}(X) = ae^{-|b(X-m)|^c} \tag{17}$$

where a and b are defined as

$$a = \frac{bc}{2\Gamma\left(\frac{1}{c}\right)}, \tag{18}$$

$$b = \frac{1}{\sigma}\sqrt{\frac{\Gamma\left(\frac{3}{c}\right)}{\Gamma\left(\frac{1}{c}\right)}} \tag{19}$$

and $\Gamma(.)$ is the gamma function. A value of $c = 0.8$ models the ac coefficients reasonably well. Note that $c = 2$ results in a normal Gaussian distribution.

The optimal detector compares the likelihood ratio to a threshold

$$\frac{p_{y|H_1}(Y|H_1)}{p_{y|H_0}(Y|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{20}$$

where η controls the tradeoff between missed detections and false alarms [13].

Assuming that the watermarked DCT coefficients are statistically independent and substituting the joint probability density into the likelihood ratio test in (20) gives

$$\frac{\prod_{\ell=1}^{N} ae^{-|b(Y_\ell - W_\ell Q_\ell)|^c}}{\prod_{\ell=1}^{N} ae^{-|b(Y_\ell)|^c}} \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{21}$$

Algebraic simplification reduces this to the equivalent test

$$Y = \sum_{\ell=1}^{N} Y_\ell W_\ell Q_\ell \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\ln \eta^{\frac{2}{c}}}{2b^2} + \frac{N\bar{Q}^2}{2} \tag{22}$$

where N is the number of selected DCT coefficients from the video and $\bar{Q}^2 = 1/N \sum_{\ell=1}^{N} Q_\ell^2$. We see that a sum of the products of watermark bits and the DCT coefficients scaled by their H.264 quantization step size is a sufficient statistic. When making decisions, knowing the value of the sufficient statistic is equivalent to knowing Y_ℓ. Therefore, we can detect the watermark by multiplying the selected DCT coefficients in the decoded frame Y_ℓ by the original watermark bits $W_\ell Q_\ell$, calculating the sum of those terms, and comparing the result with a threshold.

Looking back at (22), we see that the sufficient statistic Y is the weighted sum of generalized Gaussian random variables. Assuming that the DCT coefficients are independent, the central limit theorem implies that their sum approaches a Gaussian distribution with mean $m = m_1 + \cdots + m_n$ and variance $\sigma = \sigma_1 + \cdots + \sigma_n$. Therefore, under H_0, Y is $\mathcal{N}(0, N\bar{Q}^2\sigma^2)$ and under H_1, Y is $\mathcal{N}(N\bar{Q}^2, N\bar{Q}^2\sigma^2)$. Multiplying (22) by $1/(\sqrt{N\bar{Q}^2}\sigma)$ to normalize the Gaussian distribution and substituting for b, we have

$$\psi = \frac{1}{\sqrt{N\bar{Q}^2}\sigma} \sum_{\ell=1}^{N} Y_\ell W_\ell Q_\ell \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\sigma \ln \eta^{\frac{2}{c}} \Gamma\left(\frac{1}{c}\right)}{2\sqrt{N\bar{Q}^2}\Gamma\left(\frac{3}{c}\right)} + \frac{\sqrt{N\bar{Q}^2}}{2\sigma} \tag{23}$$

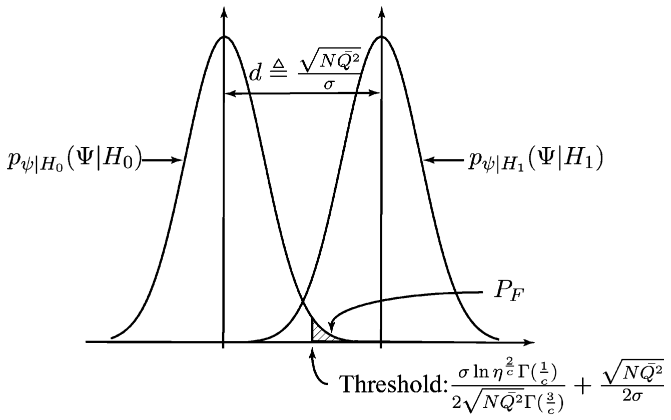

$d \triangleq \frac{\sqrt{N\bar{Q}^2}}{\sigma}$
$p_{\psi|H_0}(\Psi|H_0)$
$p_{\psi|H_1}(\Psi|H_1)$
P_F
Threshold: $\frac{\sigma \ln \eta^{\frac{2}{c}} \Gamma(\frac{1}{c})}{2\sqrt{N\bar{Q}^2}\Gamma(\frac{3}{c})} + \frac{\sqrt{N\bar{Q}^2}}{2\sigma}$

Fig. 2. Probability densities $p_{\psi|H_0}(\Psi|H_0)$ and $p_{\psi|H_1}(\Psi|H_1)$.

where the threshold T is

$$T = \frac{\sigma \ln \eta^{\frac{2}{c}} \Gamma\left(\frac{1}{c}\right)}{2\sqrt{N\bar{Q}^2}\Gamma\left(\frac{3}{c}\right)} + \frac{\sqrt{N\bar{Q}^2}}{2\sigma}. \tag{24}$$

Then, under H_0, ψ is $\mathcal{N}(0,1)$ and under H_1, ψ is $\mathcal{N}(\sqrt{N\bar{Q}^2}/\sigma, 1)$. Note that ψ is also a sufficient statistic. These probability densities are shown in Fig. 2. The distance between the means of the two densities is $d \triangleq (\sqrt{N\bar{Q}^2})/\sigma$.

To evaluate the performance of the watermark detector, we compute the probability of a detection P_D and the probability of a false alarm P_F. If the generalized Gaussian probability model is used for the DCT coefficients (but a Gaussian model for the sufficient statistic), P_D and P_F are as follows:

$$\begin{aligned} P_D &= \int_T^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-d)^2}{2}} dx = \operatorname{erfc}(T-d) \\ &= \operatorname{erfc}\left(\frac{\ln \eta^{\frac{2}{c}} \Gamma\left(\frac{1}{c}\right)}{2d\Gamma\left(\frac{3}{c}\right)} - \frac{d}{2}\right), \end{aligned} \tag{25}$$

$$\begin{aligned} P_F &= \int_T^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = \operatorname{erfc}(T) \\ &= \operatorname{erfc}\left(\frac{\ln \eta^{\frac{2}{c}} \Gamma\left(\frac{1}{c}\right)}{2d\Gamma\left(\frac{3}{c}\right)} + \frac{d}{2}\right). \end{aligned} \tag{26}$$

To achieve the specified value of P_D and P_F, the detector selects the threshold to agree with the value of P_F and then selects d to achieve the target values of P_D. Recall that d is a function of three parameters and is defined as

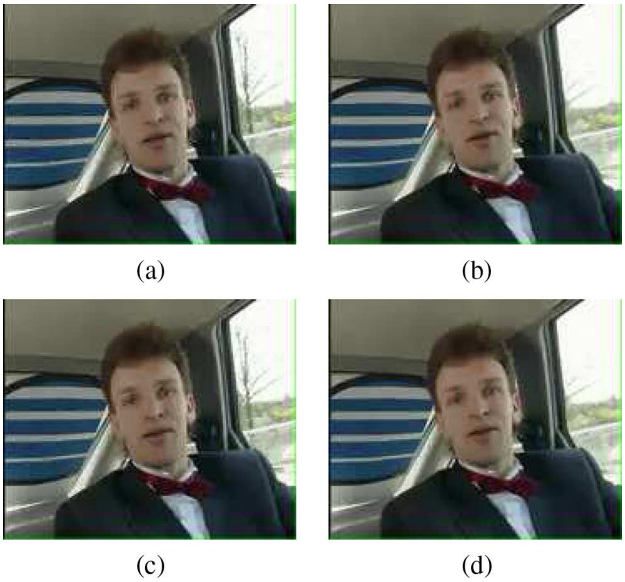
$$d \triangleq \frac{\sqrt{N\bar{Q}^2}}{\sigma} \tag{27}$$

where $\bar{Q}^2$ is the average of the squares of H.264 quantization step sizes of selected DCT coefficients and is chosen by the encoder. σ is the standard deviation of the DCT coefficients in the video and is a property of the video. Therefore, the watermark detector cannot change either of these two parameters. However, the third parameter N is the number of watermarked DCT coefficients used to compute the sufficient statistic, and it can be chosen by the detection algorithm to obtain the desired value of d. The detector finds the value of N by solving (27) for it. The detector then computes the sufficient statistic ψ over N selected DCT coefficients. The stream of N selected DCT coefficients may extend across several I-frames or may be contained within a fraction of an I-frame. Note that if $Q_\ell = Q$ is fixed for all of the selected DCT coefficients, then (27) is simplified to $d \triangleq \sqrt{N}Q/\sigma$.

Our watermark detection scheme has several advantages. First, the error rate of the detector can be maintained regardless of the video sequence given that the video is long and the detector response latency can be arbitrary in the application. The sufficient statistic ψ has the same probability distribution regardless of the video sequence. However, based upon N and the number of possibly watermarked DCT coefficients in each I-frame, the number of I-frames needed to compute ψ varies. This means that the detector will produce results more frequently for some videos than others. We believe that this is acceptable since nearly every video should have a sufficient number of possibly watermarked DCT coefficients to produce sufficient statistics and detector decisions at an acceptable rate. In the next section, we give the number of QCIF (176 × 144) I-frames used to compute one sufficient statistic for different detection performance scenarios and under several different attacks. Another advantage is that the responsibility of choosing the value of N lies completely with the detector and does not place any burden on the watermark embedding system. For example, if the watermark detector notices that the video sequence has been attacked to remove the watermark, it can increase the value of N to obtain more reliable sufficient statistics. Notice that we are taking advantage of the large amount of data in video sequences compared to images to obtain more robust watermark detection. Another advantage is that computing the sufficient statistic ψ has less computational complexity than the correlation metric proposed in [16]. Recall that ψ is defined as

$$\psi \triangleq \sum_{\ell=1}^{N} Y_\ell W_\ell Q_\ell. \tag{28}$$

The watermark sequence is a bipolar sequence of {−1,1}, and usually Q_ℓ is constant within a subset of DCT coefficients that are used to compute one sufficient statistic. Thus, computing the sufficient statistic requires only the addition or subtraction of the DCT coefficients with few multiplications. However, choosing N requires computing the standard deviation of video sequences, which has high computational complexity. In the next section, we show that the standard deviation of video sequence varies over a small range. Thus, if the computational complexity of computing σ for the video frames is undesirable, one can always assume an upper limit on the value of σ and set N accordingly.



(a) (b)

(c) (d)

Fig. 3. Comparison of the perceptual quality of our watermarking algorithm with the algorithm in [8]. (a) Our alg.'s watermarked I-frame. (b) Our alg.'s watermarked P-frame. (c) Watermarked I-frame from [8]. (d) Watermarked P-frame from [8].

TABLE I
PERCENTAGE OF WATERMARKED COEFFICIENTS FROM THE SET OF COEFFICIENTS WITH VISUAL WATERMARKING CAPACITY AND THE AVERAGE NUMBER OF WATERMARK BITS IN EACH I-FRAME

	Percentage of watermarked coefficients from *CV*		Average number of watermark bits in an I-frame	
Sequence	Our algorithm	Algorithm in [8]	Our algorithm	Algorithm in [8]
CARPHONE	19.0%	8.8%	891	609
CLAIRE	6.6%	5.0%	450	346
MOBILE	19.0%	22.4%	2291	2699
MOTHER	7.6%	3.7%	630	309
SALESMAN	20.0%	13.3%	953	626
TABLE	8.0%	8.8%	810	897

VI. SIMULATION RESULTS

We implemented our proposed watermarking algorithm in the H.264 reference software version JM10.2 [17]. To compare the perceptual quality of our proposed watermarking algorithm with the algorithm in [8], an I-frame and the following P-frame from the standard video sequence CARPHONE (QCIF, 176 × 144) are shown in Fig. 3. The frames on the top are watermarked using our algorithm, and the frames on the bottom are watermarked using the approach in [8]. Note that only the I-frame is watermarked, but adding a watermark to an I-frame will affect its dependent P-frames. This figure shows that the perceptual qualities of the watermarked frames from the two algorithms are comparable. However, there are 929 watermark bits inserted in the I-frame watermarked with our algorithm whereas the I-frame watermarked with the algorithm in [8] has only 642 watermark bits.

We used six standard QCIF video sequences (176 × 144) for our simulation. We choose $T_{\text{block}} = 2$ and $T_{\text{cof}} = 10$. Table I shows the percentage of the watermarked coefficients from the set of coefficients with visual watermarking capacity CV and the average number of watermark bits in each I-frame for each video sequence for our algorithm versus the algorithm in [8]. The results show that our algorithm increases the number of embedded watermark bits for all video sequences except the video sequences MOBILE and TABLE. These video sequences are highly textured. Thus, they have a large number of DCT coefficients with watermarking capacity. However, our algorithm limits the number of embedded watermark bits in each block to $T_{\text{block}} = 2$, which decreases the number of embedded watermark bits in those video sequences. The performance of our detection algorithm depends on the number of embedded watermark bits in each interval. Since these video sequences already have a lot of watermark bits in each frame, this does not influence the performance of our algorithm. It is more important to increase the number of embedded watermark bits in those video sequences that have a small number of DCT coefficients with watermarking capacity. On average, watermark embedding using our algorithm increases the bit rate of the video by about 5.6% versus 4.3% using the algorithm in [8]. Since these algorithms use human visual models, the peak signal-to-noise ratio (PSNR) is not an appropriate metric to compare the visual quality. However, readers might find it useful to know that the PSNR of the watermarked video decreases 0.58 dB compared to the compressed (but unwatermarked) video for our algorithm versus 0.48 dB for the algorithm in [8].

To compare the experimental results with the theoretical framework derived in the previous section, a large number of watermarked coefficients are required to compute the sufficient statistic many times. Thus, we coded and watermarked every video sequence 80 times by an H.264 encoder with an intra period of one (group of picture: I B I). Note that the more sufficient statistics we have, the more smoothly we can estimate their distribution. The H.264 encoder used a fixed quantization step size $Q = 16$ for I-frames.

The goal of our first experiment was to obtain $P_D = 0.99$ and $P_F = 0.01$. Solving (25) and (26) analytically, these probabilities can be achieved for $T = 2.325$ and $d = 4.65$. Comparing the watermarked coefficients from the decoded video with their values before the watermark was added, we notice that the difference is not exactly equal to the quantization step size, because the encoding process is lossy. On average, the difference between the watermarked coefficients and their values before watermarking is $\hat{Q}$ instead of Q, where $\hat{Q}$ is generally smaller than Q, but close to it. Therefore, we have to solve

$$d \triangleq \frac{\sqrt{N}\hat{Q}}{\sigma} \tag{29}$$

to find N. This equation suggests that the smaller the values of the quantization parameter are, the larger N must be to obtain the desired performance. Also, the larger the value of σ, the larger N must be to obtain the desired performance.

Table II shows the average value of $\hat{Q}$ and σ calculated over 80 watermarked subsequences for each video and the corresponding N to obtain $d = 4.65$. We detect the watermark by

TABLE II
EXPERIMENTAL RESULTS WHEN THE TARGET IS TO ACHIEVE $P_D = 0.99$ AND $P_F = 0.01$

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	59.17	14.40	365	4.68	0.9898	0.0108
CLAIRE	81.73	13.52	790	4.67	0.9905	0.0114
MOBILE	51.86	15.37	246	4.71	0.9917	0.0121
MOTHER	63.65	12.91	526	4.64	0.9877	0.0140
SALESMAN	54.60	14.33	314	4.67	0.9913	0.0109
TABLE	59.45	14.95	342	4.70	0.9920	0.0107

TABLE III
EXPERIMENTAL RESULTS WHEN THE TARGET IS TO ACHIEVE $P_D = 0.999$ AND $P_F = 0.001$

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	59.17	14.40	645	6.21	0.9992	0.0009
CLAIRE	81.73	13.52	1395	6.20	0.9995	0.0014
MOBILE	51.86	15.37	435	6.25	0.9990	0.0014
MOTHER	63.65	12.91	928	6.16	0.9991	0.0015
SALESMAN	54.60	14.33	554	6.20	0.9992	0.0009
TABLE	59.45	14.95	604	6.23	0.9993	0.0016

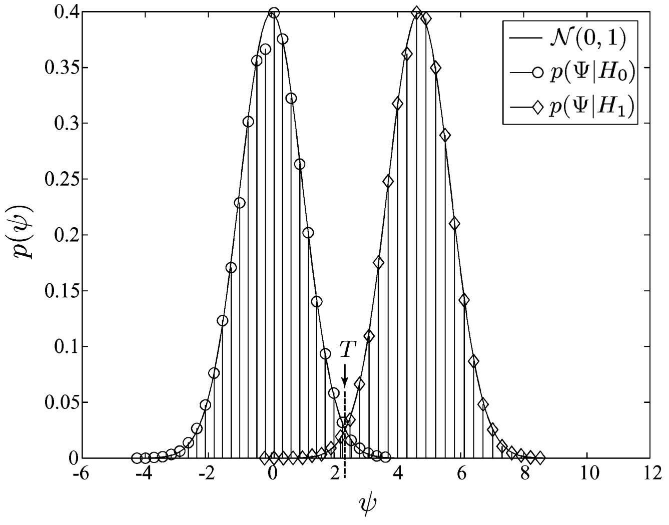


Fig. 4. Sufficient statistic probability distribution to attain $P_D = 0.99$ and $P_F = 0.01$ for CARPHONE video sequence.

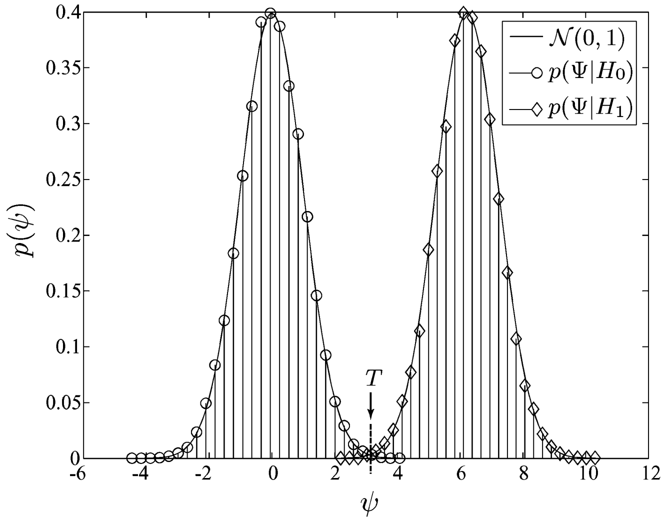


Fig. 5. Sufficient statistic probability distribution to attain $P_D = 0.999$ and $P_F = 0.001$ for CARPHONE video sequence.

computing the sufficient statistic over N-watermarked coefficients for each video. The threshold is set at $T = d/2 = 2.325$. We calculate the probability of a detection and probability of a false alarm based on this threshold. P_D and P_F and the mean value of the sufficient statistic m_ψ, obtained from our experiments, are also shown in Table II. Our results show that P_D is close to 0.99, P_F is close to 0.01, and m_ψ is close to $d = 4.65$ for all of the video sequences. In Fig. 4, we plot the probability distribution of the sufficient statistic under H_0 and H_1 for the video sequence CARPHONE. We have a total of 31978 sufficient statistics from coding this video sequence 80 times. The symbols ○ and ◇ reflect the number of sufficient statistics in the intervals centered around them. We have scaled the number of sufficient statistics in each interval so that their largest value has the same value as the peak of a Gaussian distribution with variance one. This figure shows that the experimentally determined $p(\Psi|H_0)$ and $p(\Psi|H_1)$ approximate a normal Gaussian distribution with a variance of one. This justifies our assumption that the sufficient statistic has normal Gaussian distribution.

In the next experiment, the target is to obtain $P_D = 0.999$ and $P_F = 0.001$. Solving (25) and (26) analytically, these probabilities can be achieved with $T = 3.09$ and $d = 6.18$. Table III shows the average value of $\hat{Q}$ and σ and the corresponding N to obtain $d = 6.18$. P_D and P_F and the mean of the sufficient statistic m_ψ, obtained from our experiments, are also shown in Table III. Again, our results show that P_D is close to 0.999 and P_F is close to 0.001 for all of the video sequences. In Fig. 5, we plotted the probability distribution of the sufficient statistic under H_0 and H_1 for the video sequence CARPHONE. This figure shows that $p(\Psi|H_0)$ and $p(\Psi|H_1)$ are further apart than $p(\Psi|H_0)$ and $p(\Psi|H_1)$ in Fig. 4.

In the final experiment, the target is $P_D = 0.99$ and $P_F = 0.001$. From (26), we find that $T = 3.09$ achieves $P_F = 0.001$. Then, from (25), we find that $d = 5.416$ achieves $P_D = 0.99$. Finally, we solve (29) to obtain the required N for each video sequence. Table IV gives P_D, P_F, and the mean of the sufficient statistic m_ψ for each video sequence. Fig. 6 shows $p(\Psi|H_0)$ and $p(\Psi|H_1)$, and the threshold T for the video sequence CARPHONE for this case. Since the probability of a false alarm is smaller than the probability of a missed detection, the threshold is to the right of where $p(\Psi|H_0)$ and $p(\Psi|H_1)$ intersect.

Finally, we look at the effect of different attacks on detection performance. We first consider a 3 × 3 Gaussian filtering attack. We choose N and the threshold $T = 3.09$ as in Table III to

TABLE IV
EXPERIMENTAL RESULTS WHEN THE TARGET IS TO ACHIEVE $P_D = 0.99$ AND $P_F = 0.001$

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	59.17	14.40	495	5.44	0.9904	0.0012
CLAIRE	81.73	13.52	1071	5.43	0.9895	0.0009
MOBILE	51.86	15.37	334	5.48	0.9921	0.0012
MOTHER	63.65	12.91	713	5.408	0.9882	0.0012
SALESMAN	54.60	14.33	426	5.44	0.9916	0.0009
TABLE	59.45	14.95	464	5.46	0.9924	0.0013

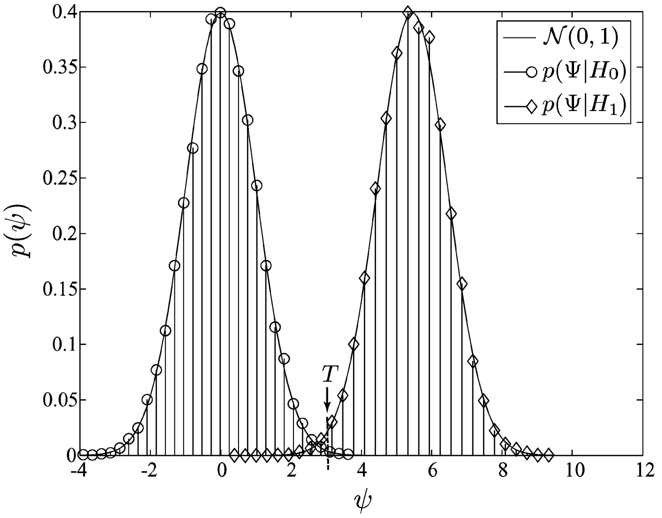


Fig. 6. Sufficient statistic probability distribution to attain $P_D = 0.99$ and $P_F = 0.001$ for CARPHONE video sequence.

TABLE V
EXPERIMENTAL RESULTS AFTER 3 × 3 GAUSSIAN FILTERING WHEN THE TARGET IS TO ACHIEVE $P_D = 0.999$ AND $P_F = 0.001$

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	60.12	10.02	645	4.25	0.9118	0.0009
CLAIRE	81.01	9.56	1395	4.42	0.9300	0.0014
MOBILE	51.92	9.43	435	3.79	0.8300	0.0014
MOTHER	65.11	9.62	928	4.49	0.9400	0.0015
SALESMAN	54.57	10.08	554	4.36	0.938	0.0009
TABLE	59.59	9.35	604	3.85	0.8500	0.0016

TABLE VI
EXPERIMENTAL RESULTS WHEN THE TARGET IS TO ACHIEVE $P_D = 0.999$ AND $P_F = 0.001$ AFTER THE 3 × 3 GAUSSIAN FILTERING ATTACK

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	59.17	9	1650	6.79	> 0.9998	0.0017
CLAIRE	81.73	9	3149	6.65	> 0.9992	0.0007
MOBILE	51.86	9	1268	6.47	> 0.9999	0.0011
MOTHER	63.65	9	1910	6.44	> 0.9997	0.0009
SALESMAN	54.60	9	1405	6.94	> 0.9999	0.0011
TABLE	59.45	9	1666	6.40	> 0.9998	0.0007

approximate $P_D = 0.999$ and $P_F = 0.001$ without any attack. Table V gives $\hat{Q}$, σ, P_D, and P_F after the Gaussian filtering attack. Comparing Table V with Table III shows that after the Gaussian filtering attack, the variance of the video sequences remains approximately the same, but $\hat{Q}$ becomes significantly smaller. Thus, if we choose N as before, $p(\Psi|H_1)$ moves toward $p(\Psi|H_0)$ because the mean value of ψ under hypothesis H_1, m_ψ, becomes smaller. Since we set the threshold T as before, we still obtain the desired $P_F = 0.001$; however, P_D is lower than 0.999. Comparing Table V to Table II shows that the m_ψ's obtained for each video sequence are similar. Thus, we can still achieve $P_D = 0.99$ and $P_F = 0.01$ by choosing T as we did for Table II.

Increasing N further, we can still achieve $P_D = 0.999$ and $P_F = 0.001$. Suppose that after the Gaussian filtering attack, $\hat{Q}$ becomes as small as 9 and suppose that the variance of the video sequences remains the same. We use these values to calculate the required N for each video sequence. Table VI shows our assumptions, the calculated value for N, and P_D and P_F after the 3 × 3 Gaussian filtering attack. Since we assumed that $\hat{Q}$ gets smaller than it actually does and we set the threshold as before, we always do better in detection than $P_D = 0.999$. We did not find any missed detections for any of the video sequences in our experiment. Therefore, the only statement that we can make is that the probability of missed detection P_M is smaller than one over the number of sufficient statistics we computed, and P_D is greater than 1 minus this value.

We also look at a cropping attack. In our experiment, we crop each video frame to 50% of its original size. We assume that the detector can determine how the video is cropped by using either the original video sequence or synchronization templates. The simulation results show that the cropping attack does not affect the detection performance; however, it does affect the number of sufficient statistics that can be extracted for each video sequence. Table VII shows N and the number of QCIF (176 × 144) I-frames F required to compute the sufficient statistic for different detection performance scenarios and after Gaussian filtering and cropping attacks. Assume that video is displayed at the rate of 30 frames/s and an I-frame is sent once per second. Then, this table suggests that we will be able to extract one sufficient statistic in every F seconds. We see that the largest value of $F = 7.41$ occurs for the CLAIRE video sequence after the Gaussian filtering attack. Note that the QCIF format (176 × 144) has one of the smallest resolutions; these results improve for higher resolution videos. Furthermore, we only embed the watermark in the luminance component of I-frames. The number of watermarked coefficients and subsequently the frequency of the sufficient statistic can be increased by embedding watermark bits in P-frames and/or chroma components.

Next, we consider the effect of additive white noise. We add white noise of mean zero and variance 0.001 to each frame of the video sequence. We choose the variance experimentally so that

TABLE VII
NUMBER OF I-FRAMES REQUIRED TO COMPUTE THE SUFFICIENT STATISTIC FOR DIFFERENT DETECTION PERFORMANCE SCENARIOS AND AFTER 3×3 GAUSSIAN FILTERING AND 50% CROPPING ATTACK

	No attack $P_D = 0.99$ $P_F = 0.01$		No attack $P_D = 0.999$ $P_F = 0.001$		Filtering $P_D = 0.999$ $P_F = 0.001$		Cropping $P_D = 0.999$ $P_F = 0.001$	
Sequence	N	F	N	F	N	F	N	F
CARPHONE	365	0.46	645	0.82	1650	2.09	645	1.37
CLAIRE	790	1.85	1395	3.28	3149	7.41	1395	5.69
MOBILE	246	0.12	435	0.21	1268	0.61	435	0.37
MOTHER	526	0.95	928	1.64	1910	2.67	928	3.47
SALESMAN	314	0.39	554	0.70	1405	1.78	554	1.01
TABLE	342	0.46	604	0.82	1666	2.26	604	1.41

TABLE VIII
EXPERIMENTAL RESULTS AFTER ADDITIVE WHITE NOISE ATTACK WHEN THE GOAL IS TO ACHIEVE $P_D = 0.999$ AND $P_F = 0.001$

Sequence	σ	$\hat{Q}$	N	m_ψ	P_D	P_F
CARPHONE	59.17	14.39	645	6.19	0.9989	0.0009
CLAIRE	81.77	13.26	1395	6.08	0.9988	0.0014
MOBILE	51.86	15.33	435	6.18	0.9991	0.0014
MOTHER	63.65	12.84	928	6.14	0.9984	0.0015
SALESMAN	54.60	14.32	554	6.19	0.9994	0.0009
TABLE	54.45	14.84	604	6.14	0.9990	0.0016

the noise is visible, but the video is not useless. We choose N as in Table III to obtain $P_D = 0.999$ and $P_F = 0.001$. Table VIII shows $\hat{Q}$, σ, P_D, and P_F after the additive white noise attack. The results show that the proposed algorithm is robust to additive white noise $\mathcal{N}(0, 0.001)$ attack without increasing N.

Finally, we examine a trivial deliberate attack to erase the watermark. The adversary does not know which coefficients from the set CV have been watermarked, nor does he or she know the value of the watermark bits inserted in each watermarked coefficient. Recall that the set CV is the set of coefficients with visual watermarking capacity. To be confident that he or she has erased the watermark, he or she has to modify all of the coefficients in CV. Since he or she does not know the value of the watermark bits, a trivial way to erase the watermark is to add the H.264 quantization step size Q to all of the coefficients in CV or randomly add $+Q$ or $-Q$ to all of the coefficients in CV. In both cases, he or she can only erase half of the watermark bits and, at the same time, he or she increases the strength of the remaining watermark bits by a factor of two. Since the sufficient statistic is a sum of the products of watermark bits and watermarked coefficients scaled by their H.264 quantization step size, the average value of the sufficient statistic remains the same. The watermark detector can detect the watermark with the same performance without increasing N. Fig. 7 shows a watermarked I-frame on the left and attacked version of that I-frame on the right. The visual quality of the attacked I-frame is not severely degraded because the adversary only modified the coefficients with visual watermarking capacity. However, there are some small artifacts around the right side of the person's face, the left side of his coat color, and more artifacts are visible in the white and blue slats on the left. Note that in the watermarked frame, 13.4% of the coefficients in CV were watermarked with a strength of Q, and the attacked frame corresponds to a frame that has 50% of its coefficients in CV watermarked with a strength of $2Q$. Therefore, we expect that the visual quality of the attacked I-frame to be worse than the watermarked I-frame. This agrees with our earlier observation that inserting watermark bits in all of the coefficients in CV can result in visible artifacts.

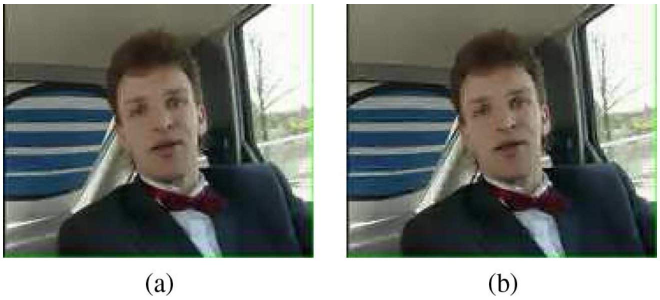


Fig. 7. Comparison of the visual quality after an adversary attempt to remove the watermark by randomly adding +Q or −Q to all of the coefficients in CV. (a) Watermarked I-frame. (b) Attacked I-frame.

We have compared our detection scheme with soft decision Viterbi decoding. Our results show that the detection performance is better when $N = 1500$ than when we construct a rate 1/3 convolutional code with $N = 500$.

VII. DISCUSSION

We showed that the problem of watermark detection for video signals is different from images because of the large watermarking capacity in videos. By appropriately choosing the number of coefficients to compute the sufficient statistic, we can achieve any probability of detection and false alarm.

In our experiments, we assume that we know the values of $\hat{Q}$ and σ. If memory is not an issue, the owner can save these values in his or her record for different video sequences. Furthermore, if computational complexity cost is not an issue, the detector can calculate σ. Otherwise, since they vary only over a small range, the detector can assume a minimum value for $\hat{Q}$, and a maximum value for σ and choose N accordingly.

We assume that the detector can synchronize after any attack that causes desynchronization. However, if the original video sequence is not available, appropriate synchronization templates [18] are required to synchronize. Furthermore, we have not considered the effect of a self-collusion attack, which is one of the most powerful attacks for video. The technique presented in [4] can be used as one solution to combat the self-collusion attack. The watermark can be embedded in different coefficients in CV by using the public and secret key proposed in [4]. In [19], we showed that a similar watermark embedding technique to the technique in this paper is fairly robust to H.264 requantization. However, because of the computational complexity of this attack, it was not implemented again in this paper. Furthermore, we only showed the robustness of our algorithm to a trivial deliberate attack. Implementing more complex adversary's attempts to erase the watermark is a subject of further study.

VIII. Conclusion

In this paper, we proposed a watermarking algorithm for H.264 that is robust to common signal processing attacks. We achieved this goal by employing a human visual model adapted for a 4 × 4 DCT block to obtain a larger payload and a greater robustness while minimizing visual distortion. We used a key-dependent algorithm to select a subset of the coefficients with visual watermarking capacity for watermark embedding to obtain robustness to malicious attacks. Furthermore, we spread the watermark over frequencies and within blocks to avoid error pooling. Our simulation results show that we increased the payload and robustness without a noticeable change in perceptual quality by reducing this effect.

We built a theoretical framework for watermark detection using a likelihood ratio test and used it to obtain optimal video watermark detection with controllable performance. Our video watermark detection scheme has several advantages. First, the error rate of the detector can be maintained regardless of the video sequence, if the video is sufficiently long and the detector latency is arbitrary. Another advantage is that control of the detector performance lies completely with the detector and does not place any burden on the watermark embedding system. Therefore, if the video has been attacked, the watermark detector can be tuned to obtain a more reliable detector response. We tested the robustness of our proposed algorithm to several common signal processing attacks such as filtering, 50% cropping, addition of white noise, $N(0, 0.001)$, and a trivial deliberate attack. Our simulation results showed that our proposed algorithm is robust against these attacks.

References

[1] I. 14496-10 and I. R. H.264, Advanced Video Coding 2003.

[2] I. E. G. Richardson*, H.264 and MPEG-4 Video Compression*. New York: Wiley, 2004.

[3] G. Qiu, P. Marziliano, A. T. Ho, D. He, and Q. Sun, "A hybrid watermarking scheme for H.264/AVC video," in *Proc. 17th Int. Conf. Pattern Recognition*, Aug. 2004, vol. 4, no. 4, pp. 865–868.

[4] M. Noorkami and R. M. Mersereau, "Compressed-domain video watermarking for H.264," in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, Sep. 2005, vol. 2, pp. 890–893.

[5] G.-Z. Wu and Y.-J. Wang, "Robust watermark embedding/detection algorithm for H.264," *J. Electron. Imag.*, vol. 14, no. 1, pp. 13 013-–13 013-9, Jan. 2005.

[6] H. A. Peterson, A. J. Ahumada, and A. Watson, "An improved detection model for DCT coefficient quantization," in *Proc. SPIE—Int. Soc. Optical Engineering*, San Jose, CA, Feb. 1993, vol. 1913, pp. 191–201.

[7] A. B. Watson, "DCT quantization matrices visually optimized for individual images," in *Proc. SPIE Int. Conf. Human Vision, Visual Processing and Digital Display*, San Jose, CA, Feb. 1993, vol. 1913, pp. 202–216.

[8] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," in *Proc. SPIE—Int. Soc. Optical Engineering*, San Jose, CA, Jan. 1999, vol. 3567, pp. 40–51.

[9] D. Simitopoulos, S. Tsaftaris, N. Boulgouris, and M. G. Strintzis, "Compressed-domain video watermarking of MPEG streams," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 2002, vol. 1, pp. 569–572.

[10] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

[11] H. A. Peterson, H. Peng, J. Morgan, and W. Pennebaker, "Quantization of color image components in the DCT domain," in *Proc. SPIE*, 1991, vol. 1453, pp. 210–222.

[12] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, no. 3, pp. 283–301, May 1998.

[13] H. L. Van Trees*, Detection, Estimation, and Modulation Theory Part I*. New York: Wiley, 1968.

[14] R. H. Jonsson, "Adaptive subband coding of video using probability distribution models," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 1994.

[15] 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.

[16] W. Zeng, "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images," *IEEE Trans. Image Process.*, vol. 8, no. 11, pp. 1534–1548, Nov. 1999.

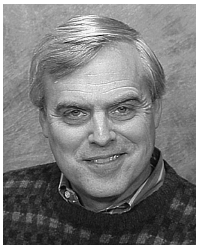
[17] H.264 Refrence Software Group. [Online]. Available: http://iphome.hhi.de/suehring/html/. 2004.

[18] A. M. Alattar, E. T. Lin, and M. U. Celik, "Digital watermarking of low bit rate advanced simple profile MPEG4 compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 787–800, Aug. 2003.

[19] M. Noorkami and R. M. Mersereau, "Towards robust compressed-domain video watermarking for H.264," in *Proc. SPIE—Security, Steganography, and Watermarking of Multimedia Contents VIII*, San Jose, CA, Jan. 2006, vol. 6072.

Maneli Noorkami (M'02) received the B.S. degree in electrical engineering from Sharif University, Tehran, Iran, in 2001 and the M.S. degree from the School of Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, in 2003, where she is currently pursuing the Ph.D. degree.

Her research interests are in image and video processing with an emphasis on watermarking techniques for multimedia. She received the Outstanding Research Award from the Center for Signal and Image Processing at the Georgia Institute of Technology in 2006. She spent two internships at Texas Instruments Incorporated, Dallas, TX, developing signal processing software for image and video applications.



Russell M. Mersereau (F'83) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 1969 and the D.Sc. degree from MIT in 1973.

He joined the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, in 1975. His current research interests are in the development of algorithms for the enhancement, modeling, and coding of computerized images, synthesis aperture radar, and computer vision. In the past, this research has been devoted to problems of distorted signals from partial information of those signals, computer image processing and coding, the effect of image coders on human perception of images, and applications of digital-signal-processing methods in speech processing, digital communications, and pattern recognition. He is the coauthor of *Multidimensional Digital Signal Processing* (Prentice-Hall, 1984) and *Circuit Analysis: A Systems Approach* (Prentice-Hall, 2006).

Dr. Mersereau has served on the Editorial Board of the Proceedings of the IEEE and as Associate Editor for signal processing of the IEEE Transactions on Acoustics, Speech, and Signal Processing and Signal Processing Letters. He was the Vice President for Awards and Membership of the Signal Processing Society and a member of its Executive Board from 1999 to 2001. He is the corecipient of the 1976 Bowder J. Thompson Memorial Prize of the IEEE for the best technical paper by an author under the age of 30, a recipient of the 1977 Research Unit Award of the Southeastern Section of the ASEE, and three teaching awards. He was awarded the 1990 Society Award of the Signal Processing Society and an IEEE Millennium Medal in 2000. He was also a Co-Founder and Vice-President of Atlanta Signal Processors, Inc., Atlanta, GA.