World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:11, 2008

A Web Oriented Spread Spectrum Watermarking Procedure for MPEG-2 Videos

Franco Frattolillo

Abstract**—** In the last decade digital watermarking procedures have become increasingly applied to implement the copyright protection of multimedia digital contents distributed on the Internet. To this end, it is worth noting that a lot of watermarking procedures for images and videos proposed in literature are based on spread spectrum techniques. However, some scepticism about the robustness and security of such watermarking procedures has arisen because of some documented attacks which claim to render the inserted watermarks undetectable. On the other hand, web content providers wish to exploit watermarking procedures characterized by flexible and efficient implementations and which can be easily integrated in their existing web services frameworks or platforms. This paper presents how a simple spread spectrum watermarking procedure for MPEG-2 videos can be modified to be exploited in web contexts. To this end, the proposed procedure has been made secure and robust against some well-known and dangerous attacks. Furthermore, its basic scheme has been optimized by making the insertion procedure adaptive with respect to the terminals used to open the videos and the network transactions carried out to deliver them to buyers. Finally, two different implementations of the procedure have been developed: the former is a high performance parallel implementation, whereas the latter is a portable Java and XML based implementation. Thus, the paper demonstrates that a simple spread spectrum watermarking procedure, with limited and appropriate modifications to the embedding scheme, can still represent a valid alternative to many other well-known and more recent watermarking procedures proposed in literature.

Keywords**—** Copyright protection, digital watermarking, intellectual property protection.

I. Introduction

Digital watermarking [1] has gained popularity as a main technology exploited to implement copyright protection processes of MPEG videos distributed on the Internet [2]. In particular, in the most classical scenario served by watermarking, the perceptually invisible digital signature inserted in a video usually makes it possible to establish if a user is illegally in possession of it, but it does not allow for establishing the "source" of the video, that is, who has initially bought and then illegally shared it via, for example, peer-to-peer network applications [2]. Therefore, to discourage unauthorized video duplication and distribution, watermarking has to exploit specific techniques, such as fingerprinting techniques [1], [3], to trace unauthorized copies to the original owners of the videos, so as to track the authors of the infringements. In fact, such techniques enable content owners or sellers to insert a distinct watermark, also called "fingerprint", identifying the buyer within any copy of video that is distributed [3]. However, even if fingerprinting techniques are exploited together with "readable" watermarking schemes based on "blind" and not publicly available decoders [2], the dangerous "average" and "collusion" attacks are still to be considered actual threats [2], [4]. Therefore, possible countermeasures against such attacks consist both in letting the watermark depend on the host signal and of adopting "anticollusion" codes, in which case the watermarked information and the embedding strategy are chosen in such a way that averaging different watermark signals, each identifying a different colluding user, leaves certain parts of the watermark unaffected, thus permitting the recovery of some information about the colluding user pool [3], [4].

Watermarking procedures based on fingerprinting techniques are characterized by an "on buyer" behavior. In fact, since watermarks are tied to buyers, they have to be embedded into the videos to be protected upon the buyers' purchase requests. This means that watermarks have to be embedded into the required videos "on the fly", i.e. during the purchase web transactions taking place among buyers and web content providers (CPs). As a consequence, watermarking procedures have to be characterized by efficient implementations that do not compromise security and robustness. Furthermore, since a robust and secure watermarking procedure can be computationally intensive or increase the size of the protected videos, it becomes strategic to adapt it to the specific characteristics of both the terminals used to open the required videos and the transactions carried out between users and CPs. For example, a PDA or a mobile phone or a terminal with no storing capacity or limited visualization capacities could receive low-quality, "lightly watermarked" videos during transactions taking place on low performance networks. In addition, the watermarking procedures purposely developed to be exploited in web contexts should be provided with implementations that can be easily integrated into the existing web services frameworks or platforms of CPs. Finally, the effectiveness of anticollusion codes strictly depends on the length of the adopted codes [4], and this means that watermarking procedures have to enable the insertion of long fingerprinting codes without impairing the final quality of the watermarked videos. Therefore, an advanced, web oriented watermarking procedure for MPEG videos should:

- provide a good degree of robustness against the most common, nonmalevolent manipulations;
- prove to be secure against intentional attacks;
- implement a readable scheme based on a blind and not

F. Frattolillo is with the Research Centre on Software Technology, Department of Engineering, University of Sannio, Benevento, Italy (phone: +39 0824 305806; fax: +39 0824 305840; e-mail: frattolillo@unisannio.it).

publicly available decoder;

- depend on the host signal and exploit anticollusion codes;
- be provided with an efficient implementation that does not compromise security and robustness;
- directly operate on the compressed bit-stream, so as not to limit the performance of the implementation;
- be developed by employing web oriented programming technologies that promote its integration with the existing web services frameworks or platforms of CPs;
- exhibit an "adaptive" behavior, i.e. it should take into account the characteristics of both the terminals used to open the videos and the network transactions carried out to deliver them to the respective buyers.

The literature is rich of proposals concerning with watermarking procedures able to embed fingerprints in MPEG videos. Among them, the procedures based on spread-spectrum additive embedding techniques (SS) have proven robust and secure against a number of signal processing operations and attacks [5], [6], [7], [8]. Furthermore, with appropriately chosen parameters and adopting specific improvements [9], the spread-spectrum watermark can survive moderate geometric distortions without suffering from the sensitivity to amplitude scaling evidenced by other well-known watermarking techniques, such as those based on the quantization index modulation (QIM) [10], and roughly achieving the same noise robustness gain as QIM. In addition, since SS embedding techniques usually depend on a few parameters, they can be easily modified to implement adaptive and efficient behaviors able to meet the requirements reported above.

This paper describes an improved variant of a well-known watermarking procedure based on a spread spectrum method and a watermark recovery by correlation [11], [12], [13]. In particular, the proposed procedure is intended for MPEG-2 videos distributed on the Internet and directly acts on compressed video streams. The procedure is also characterized by an adaptive, on buyer behavior, according to the requirements reported above. Furthermore, it has been provided with two different implementations: the former is a parallel implementation able to run on a cluster of workstations and developed by exploiting the widely used PVM software [14]; the latter is a Java implementation exploiting XML-XSLT technologies. As a consequence, the former can be considered the high performance implementation of the proposed procedure, whereas the latter is the flexible and portable version which can be easily exploited by CPs without having to interface native code with existing web services frameworks or platforms usually based on Java technology.

The paper is organized as follows. Section II describes the proposed watermarking procedure. Section III describes the two developed implementations of the procedure. Section IV reports on some experimental results. Section V discusses related work. Finally, Section VI reports conclusion remarks.

II. The Watermarking Procedure

The proposed watermarking procedure is based on the approach described in [9], and is specialized for MPEG-2 compressed video streams. In particular, the embedding approach is an improvement of the original SS techniques [11], [12], and is based on the key idea of removing the host signal as source of interference, thus producing a dramatic improvement in the quality of the watermarking process.

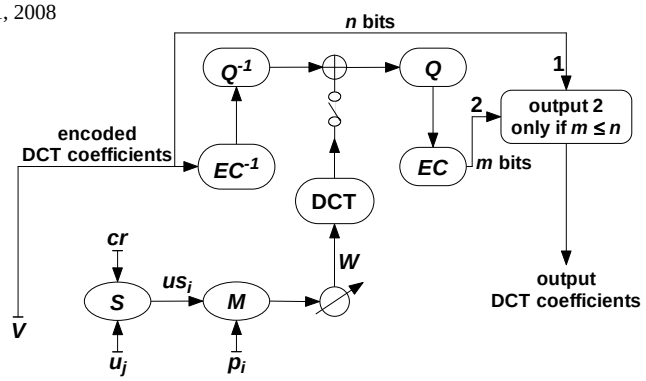

Fig. 1 The basic watermarking scheme

A. The basic scheme

In the basic scheme shown in Figure 1, the insertion of the watermark in the compressed video V is accomplished by extracting the encoded 8×8 blocks from the video and processing them together with the corresponding blocks of the watermarking signal W. In particular, the MPEG-2 bitstream is split into its main components, and only the DCT encoded signal blocks are modified.

Each encoded block is represented by a sequence of Huffman codes, each representing one (run-level)-pair and, thus, one quantized non-zero DCT coefficient of the current signal block. Therefore, to insert the watermark, each Huffman code is decoded (EC^{-1}) and inversely quantized (Q^{-1}), that is, the mapping from the quantizer index to the quantizer representative is performed. After this processing, a quantized DCT is added to the corresponding DCT coefficient from the transformed W signal, yielding a watermarked DCT coefficient. This is then quantized (Q) and Huffman encoded (EC).

It is worth noting that the basic watermarking scheme has been designed not to increase the output bit-rate. Therefore, in Figure 1, the output 2 is selected only if the number of bits used to represent the codeword for the protected video signal is less or equal than the number of bits used to represent the same codeword in the original video signal [12].

Finally, the watermarking procedure is also completed by a scheme for drift compensation, which is not shown in Figure 1 for the sake of brevity.

B. The "on buyer" behavior

The proposed procedure adds a noise-like signal to the encoded video signal processed block by block. As shown in Figure 1, the watermark signal is generated from a sequence of bits $u_j \in \{-1, 1\}$, which is used to identify a user and is spread (S) by a large factor cr, called *chip-rate*, thus obtaining the spread sequence $us_i = u_j$, with $j \cdot cr \leq i < (j+1) \cdot cr$.

Then, the noise-like signal is generated by modulating (M) the spread sequence with a binary pseudo-noise sequence $p_i \in \{-1, 1\}$, which, in the proposed solution, has to be unambiguously associated to the protected video. Thus, once a protected video has been selected, it is possible to employ the pseudo-noise sequence p_i associated to it to extract the watermark and thus obtain the user sequence $u_j \in \{-1, 1\}$. To this end, the signal of the protected video can be correlated with the p_i sequence over a cr wide correlation window, and the extracted watermark can be then analyzed to obtain the sequence of bits identifying the user who bought the video.

Finally, it is worth noting that the sequences u_j are assigned to identify users according to an anticollusion technique [4], [15] and exploit an error correction code. However, this issue is not elaborated here because this is not a main goal of the paper and for the sake of brevity.

C. The improved scheme

The scheme described in Sections II-A and II-B is based on the simple formula

$$\mathbf{s} = \mathbf{x} + u\mathbf{m} \tag{1}$$

where the vector $\mathbf{x}$ is the host signal, $\mathbf{m}$ is the chip sequence built from p_i, u represents a bit from the u_j sequence, and the vector $\mathbf{s}$ is the watermarked signal. In particular, (1) assumes that one bit of information from the u_j sequence is embedded in the vector $\mathbf{s}$ of cr values according to the common SS techniques. However, the actually implemented watermarking scheme is based on a slight modification to the SS approach, which is defined in [9] as the "linear" version of the improved SS technique (ISS). In fact, this variant assumes that the amplitude of the inserted chip sequence can vary by a linear function

$$\mathbf{s} = \mathbf{x} + (\alpha u - \lambda x)\mathbf{m} \tag{2}$$

where $x \triangleq \langle \mathbf{x}, \mathbf{m} \rangle / \langle \mathbf{m}, \mathbf{m} \rangle$ and $\langle \mathbf{x}, \mathbf{m} \rangle$ is the inner product defined as

$$\langle \mathbf{x}, \mathbf{m} \rangle \triangleq \frac{1}{cr} \sum_{i=0}^{cr-1} x_i m_i \tag{3}$$

In particular, (3) also defines the norm whenever it is used, for example, as $\langle \mathbf{x}, \mathbf{x} \rangle$.

The parameters α and λ control the distortion level and the removal of the carrier distortion on the detection statistic. In fact, if $\mathbf{y}$ is the available distorted version of $\mathbf{s}$ obtained by adding to $\mathbf{s}$ a noise $\mathbf{n}$ modeled as an uncorrelated white Gaussian random process, the sufficient statistic available at the watermark extractor r is

$$r = \frac{\langle \mathbf{y}, \mathbf{m} \rangle}{\langle \mathbf{m}, \mathbf{m} \rangle} = \alpha u + (1 - \lambda x) + n \tag{4}$$

where $n \triangleq \langle \mathbf{n}, \mathbf{m} \rangle / \langle \mathbf{m}, \mathbf{m} \rangle$. Therefore, by using the encoder knowledge about the signal, the performance of the watermarking system can be enhanced by modulating the energy of the inserted watermark to compensate for the host signal interference. In particular, the closer λ is made to 1, the more the influence of x is removed from r.

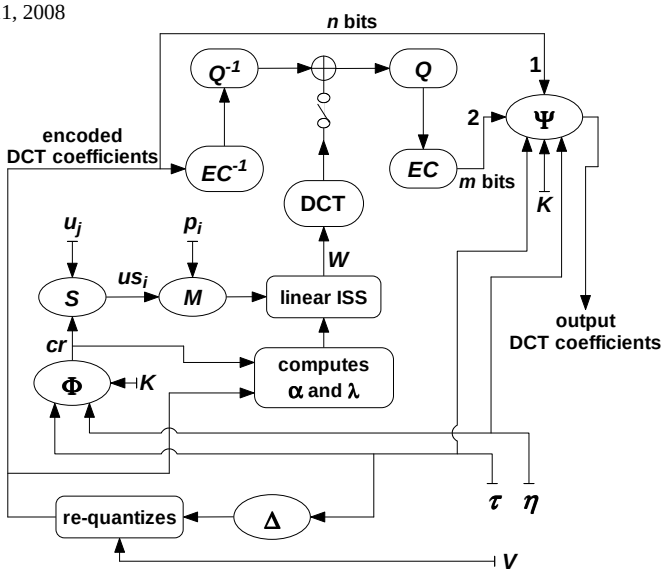

Fig. 2 The improved watermarking scheme

The detector is the same as in the SS watermarking techniques, i.e., the detected bit is $sign(r)$. Furthermore, traditional SS techniques can be obtained by setting $\alpha = 1$ and $\lambda = 0$.

The results reported in [9] make it possible to calculate the optimal values of α and λ for the watermarking system defined by (2) and under the assumptions made in Sections II-A and II-B. In particular, low values for the error probability (i.e. lower than 10^{-5}) can be achieved by setting

$$\alpha = \sqrt{\frac{cr - \lambda^2 \sigma_x^2}{cr}} \tag{5}$$

and λ close to 1 (i.e. in the range $0.9, 1$) under the assumption that cr is large enough and SNR is higher than 10 db (decibels).

D. The adaptive behavior

As reported in Section I, watermarking procedures should be characterized by an adaptive behavior. To this end, in the proposed procedure the watermark embedded in a video depends on the characteristics of both the terminal used to open the video and the quality of the network connection established between the user and the CP. This dependence, as shown in Figure 2, is controlled by two specific functions, Φ and Ψ, which determine the chip-rate cr and the video output bit-rate respectively. These functions depend on two variables, τ and η, which qualify the user terminal type and network connection respectively.

In the proposed model, τ essentially captures the terminal visualization capacities, i.e. the video resolution, whereas η synthesizes the bandwidth and latency of the user network connections. In fact, τ can be derived from what declared by

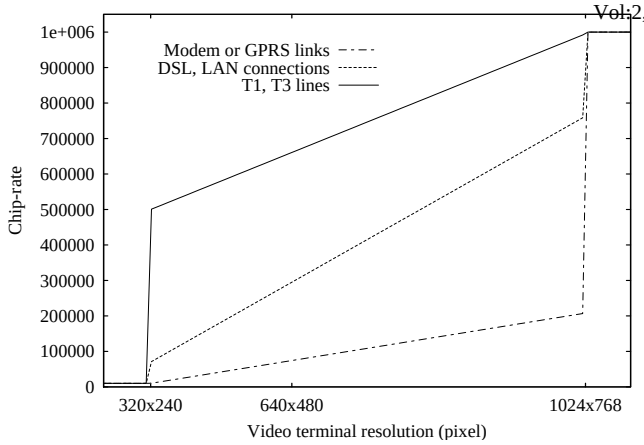

Fig. 3 The graphs of Φ

users when they interact with the web servers of CPs, whereas η can be also directly estimated by CPs during the transaction web phase with users. In particular, network connections are roughly differentiated in three main categories: modem and GPRS links, DSL and LAN connections, and T1, T3 lines.

The conducted tests have shown that cr can usefully vary in the range from $cr_{min}=10,000$ to $cr_{max}=1,000,000$. Therefore, by setting $\tau_{min}=320\times240$ and $\tau_{max}=1024\times768$, Φ can be defined as follows:

$$\Phi=\begin{cases} cr_{min} & \text{if } \tau<\tau_{min} \\ cr_{max} & \text{if } \tau>\tau_{max} \\ \eta\left(\frac{(cr_{max}-cr_{min})(\tau-\tau_{min})}{\tau_{max}-\tau_{min}}+Kcr_{min}\right) & \text{otherwise} \end{cases} \quad (6)$$

In (6) η and K may respectively assume only three different values, each corresponding to a different kind of user network connection. Table I reports the possible values for η and K derived from the conducted tests, whereas Figure 3 shows the graphs of Φ. In fact, the product $\eta\cdot K$ can be assumed as a relative weight able to characterize the network connections.

TABLE I
THE POSSIBLE VALUES OF η AND K IN (6)

	η	K
modem or GPRS links	0,2	5
DSL, LAN connections	0,7	10
T1, T3 lines	0,5	100

As reported above and shown in Figure 2, Ψ controls the video output bit-rate. It specifies the maximum increment percentage that the output bit-rate can induce in the video size. The conducted tests have shown that such increment can usefully vary in the range from $in_{min}=5\%$ to $in_{max}=50\%$ without compromising the final video quality. Therefore, Ψ can be defined as follows:

$$\Psi=\begin{cases} in_{min} & \text{if } \tau<\tau_{min} \\ in_{max} & \text{if } \tau>\tau_{max} \\ \eta\left(\frac{(in_{max}-in_{min})(\tau-\tau_{min})}{\tau_{max}-\tau_{min}}+Kin_{min}\right) & \text{otherwise} \end{cases} \quad (7)$$

As in (6), also in (7) η and K may respectively assume only three different values, each corresponding to a different kind of user network connection. Table II reports these values derived from the conducted tests, whereas Figure 4 shows the graphs of Ψ. Moreover, the product $\eta\cdot K$ can be still assumed as a relative weight able to characterize the network connections. However, the weights in Table II are less than the corresponding ones reported in Table I, and this because the possible range for the chip-rate is larger than the range specifying the increment percentage of the video size.

TABLE II
THE POSSIBLE VALUES OF η AND K IN (7)

	η	K
modem or GPRS links	0,2	5
DSL, LAN connections	0,4	8
T1, T3 lines	0,3	24,3

The behaviors of Φ and Ψ have been determined taking into account that a high value for cr increases the watermark robustness, but at the same time decreases the data rate for watermark. On the other hand, controlling the bit-rate means determining the fraction of the watermark signal that can be successfully embedded in the videos to be protected: increasing the bit-rate means to increase this fraction and thus improve the robustness of the watermarking, even though the video quality could suffer a degradation.

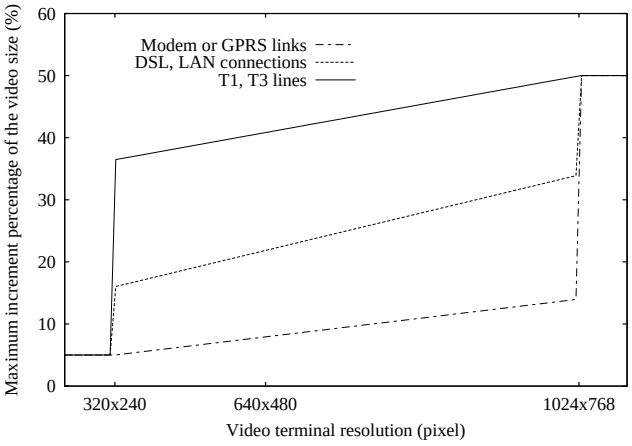

Fig. 4 The graphs of Ψ

To this end, it is worth noting that, in many well-known procedures, the watermarking is generally assumed not to increase the output bit-rate [12], [13]. On the contrary, in the proposed procedure, Ψ may increase the bit-rate, since the video size is assumed to change according to both the required protection level and the actual service conditions: the former is essentially identified by τ, whereas the latter are captured by η. Therefore, once the increment for a video size has been determined, Ψ sets a counter to the increment value. Then, Ψ updates the counter by subtracting from it the

difference between the number of bits needed to represent a codeword for the watermarked signal sent to output and the number of bits used to represent the same codeword for the original video signal: positive differences are considered "debts", whereas negative differences are considered "credits". Thus, when the counter reaches 0, further codewords for the watermarked signal are sent to output only if further credits occur that balance debts. This way, the procedure ensures that the increment of the video size remains constant.

In the proposed procedure, cr may vary to implement the adaptive behavior. As a consequence, to extract the watermark from a video, it is necessary to get the associated sequence p_i as well as the value of cr used to watermark the video. To this end, watermarking is actually performed in two phases. In the former, a cr_v value constantly associated to the video is used to embed the first n values of the sequence u_i. These values are used to identify the chip-rate cr calculated by Φ and which has to be used to watermark, in the latter phase, the remaining part of the video. Thus, given the video, the sequence p_i and the value cr_v can be identified and then applied to retrieve the first n values of the sequence u_j, which identify the cr value to be used to extract the watermark from the remaining part of the video.

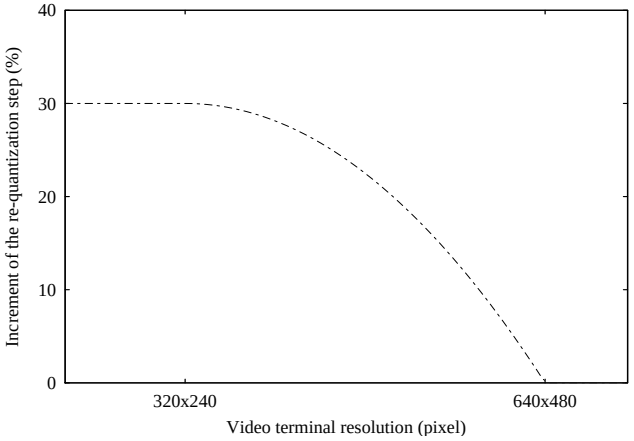

Fig. 5 The graph of Δ

The adaptive behavior of the proposed procedure is further improved by assuming that the distributed videos can be characterized by a different quality depending on the visualization capacities of user terminals. This feature is implemented by stating that the original video quality directly depends on τ. To this end, it is worth noting that the adaptive, on buyer behavior requires that a content manipulation is performed "on the fly", when the web transaction takes place, in order to adapt the video quality and the applied protection to the transaction characteristics. In particular, in order not to reduce the robustness and security levels achieved by the watermarking procedure, watermark has to be embedded after the video quality adaptation, and this means that different versions of the available videos should be handled at the CP side. In fact, two main solutions can be adopted by CPs: the former requires that different versions of each video made available by a CP are generated, stored and handled at server side, whereas the latter is based on the dynamic generation of such versions from high quality master videos. However, holding one version of a video for each possible quality level is a very heavy solution at server side, especially when the CP server has to address low to high resolution video terminals. On the contrary, the latter solution appears to be more flexible and memory saving, provided that an efficient implementation of the adaptation procedure is used.

Quality adaptation of MPEG-2 videos can be carried out by exploiting one of the two main and well-known techniques: the re-quantization of the DCT coefficients and the cut of the high frequencies, i.e. the AC coefficients [16]. The former is based on the increment of the quantization step in order to pull down ulterior DCT coefficients, whereas the latter is simply based on eliminating ulterior terms of every DCT 8×8 blocks by cutting the terms relative to the high frequencies. Therefore, both techniques reduce the dimensions of the bit-stream as well as the quality of the video, even if it is demonstrated that the former technique turns out to be more efficient than the latter in that it produces a smaller quantization error.

MPEG-2 video re-quantization is therefore controlled by the further function Δ, which determines the increment of the re-quantization step. The conducted tests have shown that such increment can usefully vary in the range from 0 to $ir_{max}=30\%$. Therefore, Δ can be defined as follows:

$$\Delta=\begin{cases} ir_{max} & \text{if } \tau \leq 320\times 240 \\ 0 & \text{if } \tau > 640\times 480 \\ ir_{max}\left(1-\left(\frac{\tau-(320\times 240)}{(640\times 480)-(320\times 240)}\right)^2\right) & \text{otherwise} \end{cases} \quad (8)$$

Obviously, a re-quantization step equals to 0 means that the original master video quality has not to be modified. Furthermore, the non linear behavior of Δ, whose graph is shown in Figure 5, allows for mostly reducing the quality of the low resolution videos, i.e., the videos that have to be lightly watermarked.

Finally, it is worth noting that re-quantization results in being strategic to implement the adaptive behavior of the proposed procedure. In fact, whenever a malicious user tries to obtain a lightly watermarked video by deceptively claiming to be provided with a low resolution video terminal and to be connected by means of a low performance link, he/she ends up obtaining only a re-quantized, low quality video which, even if unprotected, can be neither advantageously played by a high resolution video terminal nor considered interesting to Internet pirates.

III. The Implementations of the Procedure

The proposed procedure has been provided with two very different implementations in order to meet the needs of CPs. In fact, some CPs often are provided with specialized hardware resources on which they want to run high performance applications, whereas many other CPs wish to easily integrate watermarking procedures into their existing web

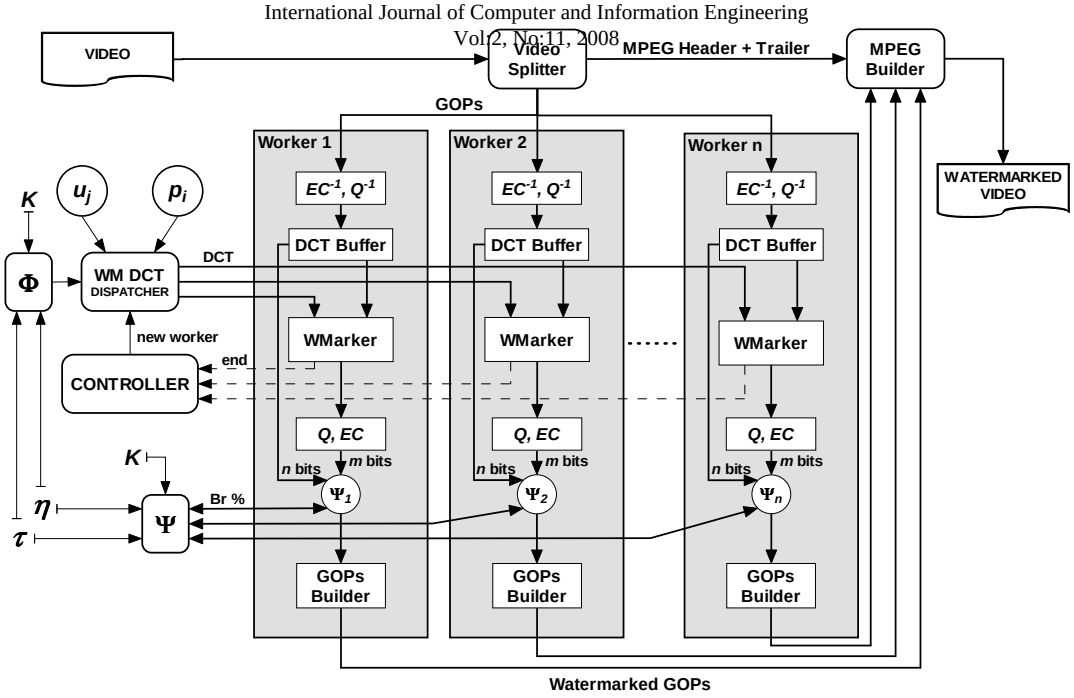

Fig. 6 The scheme of the parallel implementation

applications, which are usually implemented by exploiting the Java and XML-XSLT based technologies. As a consequence, the two developed implementations follow these two different approaches, and also demonstrate that spread spectrum watermarking procedures are simple and well suited to be implemented and exploited in web contexts.

A. The parallel implementation

Figure 6 shows the scheme of the parallel implementation of the procedure described in Section II. The implementation is based on the well-known "master-worker" paradigm, in which the video splitter behaves as the master and is responsible basically for distributing tasks among a farm of worker tasks. In particular, the splitter examines the compressed bit-stream, splits it into its main components, and assigns slices of the video, i.e. GOPs (group of pictures), to the different workers.

Each worker receives a number of GOPs, performs the first phases EC^{-1} and Q^{-1} described in Section II-A, and stores the DCT values in a specific buffer. A worker also inserts the watermarking information generated and received from the wm dispatcher task into the frames that it manages. In particular, the worker asks the wm dispatcher for the DCT watermark coefficients to be embedded, and this operation forces a synchronization among the workers in that the insertion phases performed by wmarkers are sequentialized by the behavior of the wm dispatcher, which can sequentially generate the watermark information.

Once the watermarking information has been embedded, a wmarker sends an end signal to the controller, which communicates to the wm dispatcher the new worker to be activated. Thus, while the $(i+1)$th worker starts the wmarker phase, the ith worker continues its action and performs the Q and EC phases. Then, each worker receives the increment value for the video size from the the Ψ function, which, in this scheme, solely determines the percentage of the increment value, but does not directly control the output bit-rate. This feature is implemented by the Ψ_i activities performed by the workers, which take charge of locally implementing the procedure based on "debts" and "credits" described in Section II-D. In particular, each Ψ_i initially sets its local counter to an increment value calculated on the basis of the percentage value received from Ψ and applied to the size of the video portion managed by the worker. Then, Ψ_i updates the counter by comparing the codewords of the watermarked signal and the corresponding codewords of the original video signal. If the local counter reaches 0, further codewords for the watermarked signal are sent to output only if further credits occur. It is worth noting that a Ψ_i activity can end with credits. Therefore, in order not to limit the watermarking insertion activity on the whole video, the Ψ_i has to communicate the final credit to the subsequent activity, which can exploit this value as a further increment of its local counter. Unfortunately, this induces a serialization in the activities developed by the Ψ_i functions, which forces a serialization among the workers. However, it is anyway more important to improve security rather than efficiency.

Finally, the workers complete their actions re-composing the GOPs and passing on them to the MPEG builder that merges them, thus rebuilding the whole video.

B. The Java and XML-XSLT based implementation

The proposed procedure has been also implemented by exploiting XML-based techniques of document structure trans-

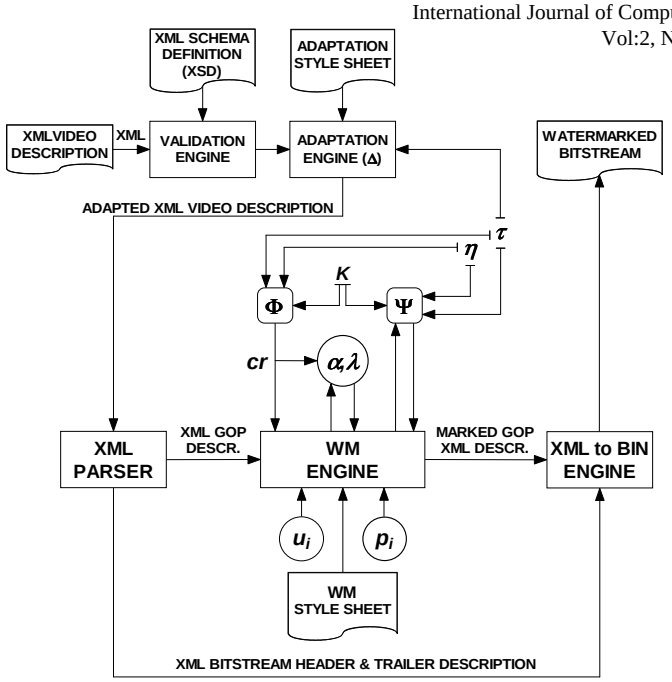

Fig. 7 The scheme of the Java and XML-XSLT based implementation

formation. In particular, this implementation, whose scheme is shown in Figure 7, assumes that the high quality master videos to be protected are all made initially available in a variant of the Bitstream Syntax Description Language (BSDL) [17], which makes it possible to describe the MPEG-2 videos by using the XML. In fact, the elaborated variant enables both the description of the whole bitstream and the addition of a further layer, similar to metadata, able to address the bitstream high-level structure, i.e. how the bitstream is organized in layers or packets of data.

Since CPs store all the master videos, they have to transform the original format of the videos contained in their web repositories, if they want to exploit this implementation. However, such a transformation appears to be an easy task, since it can be carried out by preliminary running a specific program.

Once an MPEG-2 bitstream is described in XML, all the manipulations to be performed on the video can be carried out as normal XML-to-XML editing operations. In particular, the W3C language XSLT is an efficient way to specify transformations on XML documents by means of style sheets. In fact, an XSLT style sheet contains one or several templates defining the modifications to be applied to the elements or to the attributes matching a set of conditions. Therefore, once generated and validated the BSDL description of a video (in the following referred to as BSDLd) by employing a specific XSD file, the Δ function can be calculated, and the video re-quantization can be carried out. This operation, as observed above, is an editing operation on the BSDLd performed by an XSLT transformation engine that applies a parametric adaptation style sheet. Then, the watermarking procedure computes the Φ and Ψ functions and updates the parametric style sheets that control the watermark generation. In particular, the BSDLd is parsed so as to identify only tags whose associated information has to be manipulated. To this end, a SAX parser is used, since it is characterized by an event-based behavior that, differently from a DOM parser, allows for memory saving. On the contrary, the parts of the BSDLd that have not to be manipulated can avoid the watermark XSLT engine, as shown in Figure 7.

Finally, after having generated the new XML description of the watermarked video in the variant of the BDSL, the XML-to-bitstream conversion can be performed, thus generating the protected binary version of the video.

IV. Experimental results

The proposed watermarking procedure has been assessed by performing some relevant attacks that attempt to render the embedded watermark not readable. Tables III, IV and V summarize the results obtained respectively under three different attacks: the IBM attack [6], frame dropping and frame averaging [13].

The first attack is considered an "ambiguity attack" in that it attempts to discredit the authority of the watermark by embedding one or several additional watermarks such that it is unclear which was the first, authoritative watermark. However, in this context, the IBM attack is exploited to add noise to videos so as to obscure the original watermarks.

The second attack can be considered a "simple attack" or a "detection-disabling attack", in that it attempts to impair the embedded watermark by manipulations of the whole watermarked data (host data plus watermark), without an attempt to identify and isolate the watermark.

The third attack is a "removal attack" in that it attempts to estimate the watermark, separate the watermarked data into host data and watermark, and discard only the watermark [13].

More than 80 videos have been used for the conducted tests. In particular, the tested videos are coded in MPEG-2, at 30 fps, with an original resolution of 1024×768 pixels. Their duration is in the range between 60 and 120 seconds.

For each attack, two main values have been calculated: the bit error rate (ber) affecting the watermark extraction, and the peak signal-to-noise ratio ($\overline{psnr}$). In particular, the $\overline{psnr}$ has been estimated as the mean of the $psnr$ values calculated over all the I-frames contained in the watermarked and the attacked video. To this end, each $psnr$ value has been calculated by using the following definition:

$$10 \cdot \log\left(255^2/MSE\right)$$

where MSE is for the "mean squared error" computed on an I-frame belonging to the watermarked and attacked video. Therefore, the $\overline{psnr}$ can estimate the quality of the two compared videos.

In the following tables, two pairs of values are reported under different values of the video terminal resolution (τ) and of the network connection (η). The first pair comprises the BER value, defined as the mean of the ber values calculated over the tested videos and expressed in percentage, and the corresponding standard deviation value. The second pair is

the $PSNR$ value, defined as the mean of the $\overline{psnr}$ values calculated over the tested videos and expressed in decibels, and the corresponding standard deviation value.

The user sequence u_j employed in all the conducted tests is 64 bit long, even though only 32 bits have to be considered actually used to identify a user by means of an anticollusion code. In fact, the remaining 32 bits are exploited as "check bits" needed to implement an error-correcting code. Therefore, if the ber value results in being less or equal to 9% (6 bit error) in extracting the watermark from a video, the user sequence inserted in the video can be correctly re-built. Furthermore, it is worth noting that a $\overline{psnr}$ value equal to 35 decibels is nowadays widely assumed as a lower limit for the video quality in a commercial scenario, according to the current literature [18]. Therefore, an attack can be considered valid only if the obtained ber value is greater than 9% and the $\overline{psnr}$ is greater than 35 db. Obviously, the limit of 35 db has not to be considered a hard video quality threshold, but only an estimate.

TABLE III
THE RESULTS OF THE IBM ATTACK

	320×240	640×480	1024×768
modem or GPRS	22.1%/11.2% 30,2db/5.7db	15.3%/9.1% 33,5db/4.8db	9.4%/4.4% 32,7db/4.9db
DSL, LAN	19.3%/10% 28,7db/5.9db	11.5%/6.3% 32,1db/4.9db	6.1%/4.1% 31,5db/4.7db
T1, T3	15.9%/9.8% 27,3db/4.5db	6.5%/3.9% 30,9db/5.3db	3.6%/1.8% 29,6db/5.6db

Table III shows that the proposed procedure achieves a good performance under the IBM attack. In fact, for low values of τ and η the BER is high, but the final video quality results low because the video, due to the re-quantization, is not able to contain the further information needed to make the watermark not readable. On the contrary, for high values of τ and η the procedure results in being secure, and the attack cannot impair the embedded watermark: the BER values are prevalently less than 9%. It is also worth noting that the $PSNR$ values tend to assume lower values when τ and η become high, and this because the amount of information embedded, in this hypothesis, by the watermarking procedure and by the performed attack increases, thus exceeding the video capacity.

The frame dropping attack attempts to disable the watermark extraction by removing trunk of frames. In particular, when the dropping rate of video frame is high, errors are introduced to the whole watermark, making the procedure performance poor. However, this also leads to a significant damage to the video, and the results reported in Table IV, obtained under a value of frames dropped about 20%, reflect this condition. In particular, the successful attacks performed under some values of η and τ can be contrasted by increasing the number of the check bits used to implement the error-correcting code.

TABLE IV
THE RESULTS OF THE FRAME DROPPING ATTACK

	320×240	640×480	1024×768
modem or GPRS	26.2%/12.4% 33,4db/4.8db	19.1%/11.9% 39,3db/2.1db	13.2%/6.8% 37,5db/2.8db
DSL, LAN	23.1%/12.1% 32db/4.6db	14.5%/6.9% 38,4db/2.2db	10.6%/5,3% 35,3db/2.7db
T1, T3	17.5%/8.7% 30,8db/4.9db	9.1%/3.9% 37,7db/2.6db	8.3%/3.5% 33,2db/3.7db

In the statistical averaging attack a high number of watermarked frames are collected so as the watermark can be estimated by statistical averaging. The attack has been performed by colluding about the 70% of the available frames, and the obtained results are shown in Table V. In particular, the procedure exhibits a good performance, and this is essentially due to its adaptive behavior, which can balance the final video quality with the achieved protection level.

TABLE V
THE RESULTS OF THE STATISTICAL AVERAGING ATTACK

	320×240	640×480	1024×768
modem or GPRS	13.4%/7.1% 40,3db/1.7db	12.5%/6.4% 39,6db/2.1db	9.2%/3.7% 39,2db/1.9db
DSL, LAN	10.5%/3.9% 38,5db/2.3db	9.6%/3.1% 38,2db/2.2db	7.2%/2.6% 38,4db/2db
T1, T3	9.3%/3.2% 36,3db/3.1db	5.6%/2.4% 37,6db/2.9db	3.1%/0.7db 37,7db/2.8db

Finally, it is worth noting that the $PSNR$ values obtained during the conducted tests demonstrate that the procedure can successfully protect the videos as well as reduce the final video quality to the allowed minimum values. In fact, one of the interesting aspect of the procedure, emerged from the test phase, is that, if the re-quantization phase reduces the video quality to a $\overline{psnr}$ value close to a predefined lower limit, such that of 35 db, and the subsequent watermark embedding is carried out taking care of saturating the video capacity without further reducing the final value of the $\overline{psnr}$, attacks to impair the embedded watermark end up obtaining $\overline{psnr}$ values much lower than the assumed limit, thus degrading the final video quality.

V. RELATED WORK

Even though there are some differences between images and videos which suggest specific approaches for video watermarking, most of the key ideas characterizing image watermarking techniques can be directly applicable to videos. As a consequence, while a lot of proposals has been published on image watermarking, there are fewer publications that deal with video watermarking. Therefore, in the following, the discussion is essentially focused on the mainly known

image watermarking procedures. To this end, it is worth noting that most of these procedures shares common principles. In fact, the inserted watermark is typically considered as a pseudorandom signal with low amplitude, compared to the image amplitude, and usually with spatial distribution of one information bit over many pixels. Therefore, a lot of watermarking procedures are very similar and differ only in parts or single aspects of the three topics: signal design, embedding, and recovery.

A relevant class of watermarking procedures is based on "linear" insertion schemes, such as the SS insertion scheme. These procedures embed watermark information by linearly combining the host signals represented by original images and videos with small pseudo-noise signals that are modulated by the embedded signals. Examples of such procedures are documented in [5], [11], [12]. In particular, in [5] the embedded watermark signal consists of a sequence of real numbers that are normally distributed, and it is scaled according to the strength of the frequency components of the host signal. In fact, the procedure follows a simple watermarking scheme with perceptual weighting consideration, and this has increased its robustness level with respect to the previously proposed SS procedures. Furthermore, this scheme has been also improved in [19], in which the "visual model" proposed by Watson [20] has been extended to adapt the inserted watermark to each image to be protected.

Although SS watermarking procedures have received considerable attention in the literature, they typically result in being limited by host-signal interference when the host signal is not known at the decoder. Intuitively, the host signal in an SS system is an additive interference that is often much larger, due to distortion constraints, than the pseudo-noise signal carrying the embedded watermark information. However, such a problem, as reported in Section II, has been solved by the improved scheme proposed in [9] and adopted in this paper.

Watermarking procedures based on quantization index modulation (QIM) [10] represent, on the contrary, a class of "nonlinear" methods that overcome the drawbacks due to the host-signal interference affecting the SS watermarking procedures. In fact, a QIM based watermarking procedure is based on a set of N-dimensional quantizers. The quantizers satisfy a distortion constraint and are designed such that the reconstruction values from one quantizer are "far away" from the reconstruction points of every other quantizer. The message to be transmitted is used as an index for quantizer selection. The selected quantizer is then used to embed the information by quantizing the image data in either the spatial or DCT domain. In the decoding process, a distance metric is evaluated for all quantizers and the index of the quantizer with the smallest distance identifies the embedded information. Thus, the results achieved by QIM based watermarking schemes are usually better than those ones achievable by standard SS techniques without watermark weighting.

In [21] an important study about the performance of the SS and QIM watermarking approaches for still images in the presence of lossy compression is reported. The study shows that SS and QIM based watermarking schemes have different characteristics of robustness to JPEG compression: SS watermarking is more robust to higher levels of JPEG compression, while QIM watermarking does not experience host signal interference which dominates for low compression ratios. Although the reported results concern with a scheme where watermarking occurs on still images before lossy compression, the study confirms that the idea exploited by the procedure proposed in this paper of removing the host signal as source of interference in the watermark embedding can produce a dramatic improvement in the quality of the protection process. Therefore, the proposed procedure, by adopting and adaptive scheme and exploiting the re-quantization process as a further mechanism to improve the video protection level, can enhance its performance with respect to other SS watermarking schemes without requiring complex watermark decoders or the adoption of hybrid solutions, such as the one proposed in [21].

Another interesting watermarking procedure is presented in [22]. The procedure exploits a method based on "singular value decomposition" (SVD), which is a numerical technique used to diagonalize matrices in numerical analysis and developed for a variety of applications. In fact, an image is regarded as a matrix A whose SVD can be obtained by calculating two orthogonal matrices, U and V, and one diagonal matrix S such that $A = USV$. A watermark can be inserted into an image represented by A by calculating $S + \alpha W$, where W is the matrix that represents the watermark, whereas α is a positive constant representing the scale factor which controls the strength of the watermark to be inserted. Then, the new matrices U_w, S_w and V_w have to be calculated so as to obtain $U_w S_w V_w = S + \alpha W$. Thus, the watermarked image is represented by $A_w = U_w S_w V_w$.

The discussion and the results reported in [22] show that the SVD-based watermarking procedure performs well both in resolving the problem of rightful ownership and in resisting common attacks. More precisely, the procedure exploits the SVD technique, which uses nonfixed orthogonal bases. On the contrary, other unitary transformations, such as discrete Fourier transform (DFT) or discrete cosine transform (DCT), adopt fixed orthogonal bases. As a consequence, SVD is a one-way, nonsymmetrical decomposition that leads to the good performance of the proposed procedure in both security and robustness. However, unlike the procedure proposed in this paper, an SVD-based watermarking procedure can be neither easily modified to implement an adaptive behavior, nor provided with an efficient implementation, since it cannot operate in the compressed domain.

VI. Conclusions

This paper describes a watermarking procedure for the copyright protection of MPEG-2 videos distributed on the Internet. The watermarking procedure directly acts on compressed video streams and is implemented as an on buyer variant of the improved spread spectrum scheme described in [9]. The procedure can employ long anticollusion codes to increase security against average and collusion attacks,

and is characterized by a novel adaptive behavior that is able to modulate the applied protection depending on both the terminals used to open the watermarked videos and the network transactions carried out to deliver them to buyers.

The experimental results confirm that a simple spread spectrum watermarking procedure can be made robust and secure against a variety of manipulations by performing some improvements that do not penalize efficiency. This makes the procedure suitable to be exploited in web contexts, where an "on the fly" behavior is required. Moreover, the adaptive behavior of the procedure allows for achieving a trade-off between protection needs and the final quality of the distributed videos. Thus, whenever attacks attempt to impair the embedded watermarks, the final video quality ends up being degradated, thus making the attacked videos useless in commercial web applications and not interesting to Internet pirates.

REFERENCES

[1] I. Cox, J. Bloom, and M. Miller, *Digital Watermarking: Principles & Practice*. Morgan Kaufman, 2001.

[2] M. Barni and F. Bartolini, "Data hiding for fighting piracy," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 28–39, 2004.

[3] M. Wu *et al.*, "Collusion-resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 15–27, 2004.

[4] W. Trappe, M. Wu, *et al.*, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. on Signal Processing*, vol. 41, no. 4, pp. 1069–1087, 2003.

[5] I. Cox, J. Kilian, *et al.*, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Signal Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[6] F. Hartung, J. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counterattacks," in *Electronic Imaging 1999, Security and Watermarking of Multimedia Contents*, ser. SPIE Proceedings, vol. 3657, S. Jose, CA, USA, Jan. 1999, pp. 147–158.

[7] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale and translation resilient watermarking for images," *IEEE Trans. on Image Processing*, vol. 10, no. 5, pp. 767–782, 2001.

[8] J. Lubin, J. Bloom, and H. Cheng, "Robust, content-dependent, high-fidelity watermark for tracking in digital cinema," in *Electronic Imaging 2003, Security and Watermarking of Multimedia Contents*, ser. SPIE Proceedings, P. W. Wong and E. J. Delp, Eds., vol. 5020, S. Jose, CA, USA, Jan. 2003, pp. 536–545.

[9] H. S. Malvar and D. A. F. Florêncio, "Improved spread spectrum: A new modulation technique for robust watermarking," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 898–905, 2003.

[10] B. Chen and G. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Inforormation Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[11] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, Berlin, Germany, October 1996.

[12] ——, "Digital watermarking of MPEG-2 coded video in the bitstream domain," in *Procs of the Int'l Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Munich, Germany, 1997, pp. 2621–2624.

[13] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Procs of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[14] A. Geist, A. Beguelin, J. Dongarra, and others., *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing*. The MIT Press, 1994.

[15] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. on Information Theory*, vol. 44, no. 9, pp. 1897–1905, 1998.

[16] Z. Lei and N. D. Georganas, "Rate adaptation transcoding for precoded video streams," in *Procs of the 10th ACM Int'l Conference on Multimedia*, Juan-les-Pins, France, 2002, pp. 127–136.

[17] M. Amielh and S. Devillers, "Bitstream syntax description language: Application of xml-schema to multimedia content adaptation," in *Procs of the 11th Int'l World Wide Web Conference*, Honolulu, Hawaii, USA, 2002.

[18] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.

[19] C. I. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 525–539, 1998.

[20] A. B. Watson, "DCT quantization matrices visually optimized for individual images," in *Human Vision, Visual Processing and Digital Display IV*, ser. SPIE Procs, J. P. Allebach and B. E. Rogowitz, Eds., vol. 1913, S. Jose, CA, USA, Feb. 1993, pp. 202–216.

[21] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of watermarking algorithms for improved resistance to compression," *IEEE Trans. on Image Processing*, vol. 13, no. 2, pp. 126–144, 2004.

[22] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. on Multimedia*, vol. 4, no. 1, pp. 121–128, 2002.