DBpia

Watermarking of Compressed Video in the Bitstream Domain: An Efficient Algorithm and its Implementation

Inna G. Drobouchevitvh*, Sungjun Lim**, Byungwan Han***
Hangbae Chang**, Kyungkyu Kim** ***Regular Members***

ABSTRACT

Digital watermarking of multimedia data is a very active research area that has enjoyed a considerable amount of attention in recent years. In this paper, we propose an algorithm for embedding/detecting a fragile watermark in MPEG-4 compressed video domain (Simple and Advance Simple Profiles). The watermark bits are put directly into Huffman VLC-codespace of quantized DCT domain. The advantage of watermark embedding into the compressed domain is the significant savings for a real-time implementation as it does not require a full decoding operation. The watermark embedding does not change the video file size. The algorithm demonstrates high watermarking capacity, thereby providing reliable foolproof authentication. The results of experimental testing demonstrate that watermark embedding preserves the video quality. Watermark detection is performed without using the original video.

Key Words : Video Watremarking, Fragile Watermark, MPEG-4, Compressed Domain, VLC

Ⅰ. Introduction : Watermarking for Video Authentication

Our work focuses on the design of a watermarking algorithm for MPEG-4 video sequences, coded in compliance with Simple and Advance Simple Profiles of MPEG-4 standard. The proposed watermarking scheme is intended for digital video authentication.

Digital watermarking of multimedia data is a very active research area that has received a considerable amount of attention in recent years. With MPEG-4 format becoming increasingly popular in multimedia applications dealing with video creation and processing, the watermarking of MPEG-4 compressed video data is a relatively unexplored area. We also remark that, while there has been a great number of studies focused on authenticating digital images, comparatively little work has been done for authenticating video. The majority of published research on video watermarking focuses on non-authentication watermarking methods (mostly, robust watermarking for copyright protection). A good watermark for hard-authenticating of streaming video is still a research problem[30].

Evidently, the design and features of watermarking software are governed by its targeted application. In the digital video authentication, the purpose of the watermark is to prove the video ownership as well as to establish that the video has not been tampered with. In the case of surveillance applications, the purpose of a watermark is to validate the integrity of digital data recorded by the system, i.e., to prove that the video stream is intact as originally recorded. The watermark

* Hitron Systems ** Yonsei Graduate School of Information *** Tongwon College
논문번호 : KICS2005-12-510, 접수일자 : 2005년 12월 27일, 최종논문접수일자 : 2006년 3월 27일

www.dbpia.co.kr

must provide reliable authentication evidence (the situation when digital video is used as evidence before court can be an example of application), and thus, ought to be fragile to any manipulation or alteration of the content. The latter is commonly referred to as “hard authentication” and is done by means of fragile and semi-fragile watermarking.

The application scenario imposes certain restrictions and application-driven criteria on the design of watermarking software. Below we outline general requirements and key elements for hard-authentication watermarking that we pursue in our algorithm development :

Foolproof authentication : the embedded watermark should detect any tampering/modification of the original data.

Perceptual transparency : the embedded watermark should not be perceptible under normal observation or interfere with the visual content and functionality of the multimedia.

Blind detection : it is not desirable for video watermarking software to use the original video during watermark detection simply because the video files are usually of considerable size, and it is not convenient to store them twice.

Security : it should be hard for an unauthorized party who has full knowledge of the algorithm to fake a valid authentication, to derive the authentication secret from the public data and knowledge, or to undertake a malicious manipulation without detection.

Low Error Probability : even in the absence of attacks or signal distortions, the probability of failing to detect the watermark when it is present or detecting a watermark when, in fact, one does not exist, must be very small.

Bit-rate control : the procedure of watermark embedding should not significantly increase the data rate of the watermarked video stream.

Computational efficiency : while allowing for a useful amount of information to be embedded into a visual image, the watermarking procedure should be of reasonable computational complexity. Thus, every effort has to be made to make its implementation as fast as possible, which is necessary for real-time applications, such as, e.g., video servers in Video on Demand applications. Implementation speed is also important when a large number of video sequences has to be watermarked.

Note that some of the requirements above are in conflict amongst themselves (e.g., foolproof authentication (high embedding capability) and perceptual transparency (low image degradation)). Thus, a good watermarking algorithm should seek the optimal balance between the conflicting elements.

In this paper, we propose a scheme for fragile watermarking of MPEG-4 compressed bitstreams. The underlying method is the Least Significant Bit (LSB) modification in the variable length codeword (VLC) domain, which is a computationally efficient technique with a high embedding capacity and small-degree embedding distortion. In this work, we propose a watermarking procedure designed specifically to deal with MPEG-4 coded video. The MPEG-4 VLC codespace is explored for its watermarking capacity. Moreover, the proposed watermarking procedure exploits the AC/DC prediction mechanism which is an essential coding tool in MPEG-4 standard, so as to ensure the perceptual transparency of the watermarking signals. Extensive experimental testing has been conducted to verify the performance of the proposed algorithm. Furthermore, the practical implementation of the method is investigated. As the result, the computational efficiency and the use of the memory are essentially reduced to the feasible minimum to satisfy the requirements of real-time embedding/detection.

The remainder of the paper is organized as follows. In the next Section, we give an outline of the basic mechanisms used by MPEG-4 standard in video compression and briefly overview the relevant results in the area of the digital video watermarking. In Section 3, we present our watermarking scheme. We describe the main features of the proposed algorithm along with its practical implementation and discuss the experimental results. Concluding remarks are given in Section 4.

www.dbpia.co.kr

Ⅱ. MPEG-4 Compression and Watermarking

We start by briefly outlining the basic mechanisms used by MPEG-4 standard in video-compression.

MPEG-4 Coding scheme employs the following guidelines:

1) An RGB-to-YCbCr conversion is performed for color domain, followed by sub-sampling of Y and Cb/Cr components.
2) A DCT transformation into frequency domain is applied to each video block (8x8 pixels) of each component (Y, Cb and Cr), the 8x8 blocks are grouped to form 16x16 macroblocks.
3) Quantization is applied (with different quantization scaling for different frequencies), DC/AC prediction is done, and Run-Length Coding(RLC) is performed with the scan order being defined by DC-prediction mechanism (zig-zag order, vertical scan or horizontal scan).
4) Motion compensation exploits temporal redundancy by attempting to predict a frame from a previous reference frame.

MPEG-4 provides two basic coding modes: Intra-Mode and Inter-Mode. In Intra-Mode (I- frames), both the spatial redundancy and irrelevancy are exploited with block based DCT coding, quantization, run length and Huffman coding. Only information from the picture itself is used and so, every frame can be decoded independently. In Inter-Mode (P-/B-frames), the temporal redundancy between the pictures in a video sequence is taken into account. A macroblock-based motion estimation between two successive images is done allowing a motion compensated prediction of the current picture. The residual (the difference between the predicted image and the reference image) is calculated, DCT coded, quantized and VLC coded. The motion vectors describing the motion of the blocks are also encoded with VLC.

As far as the bitstream structure is concerned, an MPEG-4 video sequence is divided into a series of GOVs(Group of Video Object Planes), where each GOV contains a series of frames, or VOPs(Video Object Plane): a single I(intra)-frame followed by a sequence of P(predicted) -frames and B(bidirectional)-frames (the latter appear only in Advanced Simple Profile). Under SP/ASP, a VOP corresponds to a complete video frame of rectangular shape.

The additional coding tools used in Simple Profile are Short Header, AC/DC Prediction, 4 MV/ Unrestricted MV, Video Packet Resynchronisation, Data Partitioning and Reversible VLC. The tools of Advanced Simple Profiles comprise those used in Simple Profile plus H.263/MPEG-2 Quantization, Global Motion Compensation, Quarter-Pel Motion Compensation and Interlace.

For more details on MPEG-4 video standard, we refer an interested reader to MPEG-4 documentation (ISO/IEC 14496-2[1] and ISO/IEC JTC1/ SC29/WG11[2]), "The MPEG-4 Book" by T. Ebrahimi and F. Pereira and "H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia" by I.E.G. Richardson[4].

The advantage of watermark embedding into the compressed domain is the significant savings for real-time implementation as it does not require a full decoding operation. It is very often impractical or, indeed, simply unfeasible to decompress and then recompress the entire video data, due to both high storage capacity requirements and the processing time escalation. A general watermarking scheme for video compressed domain operates as follows: the compressed video stream is parsed by the watermark embedder, video is partially decoded to identify and expose the syntactic elements of the compressed video data for watermarking, and the watermark is inserted. Thus, recompression is also not necessary after the watermark has been embedded.

It should be stressed that digital video is not merely a sequence of images displayed at regular time intervals. One of the issues of concern is high spatial correlation between successive frames. In most cases, successive frames of video are not

www.dbpia.co.kr

independent and have a high degree of similarity. Watermarking in the compressed stream can be seen as a form of video editing in the compressed domain. Needless to say, such editing is far from trivial to be practically realized. In addition to the basic, well-known problems for image watermarking (such as, e.g., adjusting the local strength of the watermark according to the properties of the human visual system without accessing the fully decompressed video), new, video-specific, issues emerge. In particular, we face two major problems here. Firstly, watermark embedding must be tied tightly with a compression method. This coupling not only restricts the portability of the watermarking algorithm, but also imposes limitations caused by the bitstream syntax and coding algorithm. It also should be ensured that the watermarking embedding process does not increase the output bit-rate beyond the allowed threshold. The second issue is to prevent the distortion introduced by the watermark to propagate from one frame to another one. The MPEG standard relies on motion prediction and any distortion is likely to be propagated to neighboring frames. Namely, drift occurs when (spatial or temporal) prediction is used and a predictor is modified without adjusting the residual to compensate for the new predictor. The accumulation of such propagating signals may result in a poor quality video, and motion compensation can be seen as a constraint. A watermarking technique must be designed to prevent the distortion from spreading and accumulating, which is currently a prominent challenge for compressed-domain watermark embedding. For additional information on challenges of video watermarking, we refer the interested reader to comprehensive surveys on multimedia/video watermarking: Doerr and Dugelay[20], Langelaar et al.[28], Li and Yuan [29], Lin et al.[30], Nicholson et al.[36], Swanson et al. [40], Zhu et al.[41]-[42].

While there has been a good deal of studies devoted to video watermarking, the watermarking for MPEG-4 video is a relatively new area with many open issues and challenges. The watermarking in dequantized DCT-domain has received much attention due to its potential for robust watermarking (especially, as applicable for copyright protection): Bas et al.[10]-[11]], Qiao and Nahrstedt [37]-[38], Celik et al.[13], Hartung and Girod [22]-[24], Alattar et al.[5], Cox et al.[7], Arena et al.[16], etc. There has being a number of research studies on watermarking in quantized DCT domain of MPEG-compressed video: Langelaar and Lagendijk[26], Barni et al.[9], Doerr and Dugelay[18]-[20], Simitopoulos et al.[39].

In this study, we pursue the problem of compressed-domain video watermarking in VLC domain. The watermark signals are inserted directly into variable length codewords used for entropy encoding in the quantized DCT domain. Our work has been inspired by the ideas for watermarking in compressed bitstream that previously appeared in the literature on digital multimedia watermarking. The data hiding in VLC domain is a well-established watermarking technique. Undoubtedly, it owns its popularity to its superior computational efficiency, which is a primal concern in the real-time applications. In recent years, there has been a number of research studies devoted to the design and discussion of the video watermarking schemes operating in the VLC domain.

Berger II and Mobasseri[12],[34] employ the “VLC mapping” approach that relies on a certain redundancy of the VLC space. In particular, the VLC codewords used in a bitstream are mapped to the outside of the used codespace but within the original VLC table and the watermark is embedded into a bitstream as forced bitstream errors. The algorithm is tested on JPEG-images. Mobasseri and Marcinak[35] extend the latter approach to MPEG-2 video. Because of the very limited unused capacity of the MPEG-2 VLC codespace, the approach of the one-to-one VLC codeword mapping turns out to be unsuitable. Instead, the authors come with the idea of mapping a pair of the original VLCs used in the same 8x8 block to an unused pair. The significance of looking at pairs of VLCs is that there is a considerable number of code pairs that do not appear together in one block. As in the case with JPEG-water-

www.dbpia.co.kr

marking, the watermarking scheme is lossless. It is also noted that for the "VLC mapping" approach to work, the watermark decoder must possess the knowledge of the original, not watermarked VLCs. Furthermore, the forced bitstream errors introduced by watermarking may obviously affect the visual quality of an image. To lessen such an effect, the mapping procedure has to be designed with a special care. The computational complexity of both used/unused VLC codespace generation and VLC mapping are two important factors that should be taken into consideration, especially if large data streams are expected to be watermarked in real-time.

In our work, we follow the approach when watermark data is embedded into the VLC codewords by forcing the LSB of their quantized level to the value of a watermark bit. A VLC is considered suitable for watermarking if it can be paired with another valid VLC that has the same last/run-values and the level difference of one. This approach has been suggested and implemented by Langelaar et al.[27] for MPEG-2 bitstreams. In recent years, there has been a number of studies confirming the successful realization of this technique in digital multimedia watermarking, in particular for JPEG images and MPEG-1/2 video (see e.g., the works by Fridrich et al.[21], Cinalli et al.[15], Liu and Chang [31]). Cross and Mobasseri[17] implemented a more sophisticated realization of LSB modification in VLC domain technique for MPEG-1/2 bitstreams. In particular, they have suggested an approach to derive the watermark data from the video itself so that the watermark data cannot be pirated. The proposed algorithm uses an I-VOP from the previous GOP to watermark intra-coded blocks in the current GOP. The video tampering or frame removal can be pinpointed unless two frames are removed. From the computational point of view, the algorithm demands a double bitstream-decoding pass through a complete GOP: first, to establish its watermarking capacity (defined by the number of watermarkable VLCs), and then-to actually embed a watermark. Lu et al.[33] have devised a frame-dependant watermarking scheme that operates in the VLC domain and uses a special filtering mechanism for choosing quantized DCT coefficients to watermark. The algorithm operates on a macroblock level and the filter is based on the mean level value of a macroblock. The algorithm is basically an LSB modification (apart from the cases when the watermarked VLC falls out of VLC codespace, and has to be substituted with the closest valid VLC). The length of watermarked VLCs doesn't need to match the original one, but the mechanisms to control bit-rate are introduced. Hwang et al.[25] proposed an error detection method for real-time video communication by the means of watermarking VLC codes. In particular, a special pattern is embedded in LSBs of selected quantized coefficients, and both DC- and AC-coefficients are considered eligible for embedding. The method has been tested on H.263- coded data.

Our primal goal has been the design of a fragile compressed-domain watermarking scheme compliant with MPEG-4 coding standard (Simple and Advance Simple Profiles), as well as the investigation of the watermarking capacity of the VLC space. The proposed algorithm heavily relies on the MPEG-4 video compression method, and, to the best of our knowledge, pioneers the applicability of the VLC domain watermarking technique with file size preservation to MPEG-4 coded bitstreams. The implemented method demonstrates the good watermarking capacity, the perceptual transparency of watermarking signals and high computationally efficiency. Furthermore, we exploit the fact that the watermarking procedure does not change the bitstream size. Namely, in using both the video frame size and the time of the recording as the parameters in the watermark data generation we find a useful tool to fight watermark pirating.

Ⅲ. MPEG-4 Algorithm for MPEG-4 Video Watermarking

3.1 Structure and Main Features of Algorithm

Our watermarking scheme is a data-hiding algo-

www.dbpia.co.kr

rithm that works in the bitstream domain of MPEG-4 encoded video, i.e. a watermark is inserted directly into the compressed bitstream. More specifically, watermark bits are introduced into the MPEG-4 bitstream via the Least Significant Bit (LSB) modification of quantized DCT coefficients (the method is also known in the literature as watermarking by Parity Bit modification, see, e.g. Langelaar et al.[28]). Our algorithm operates on the bitstream level of MPEG-4 coded video - the lowest level domain, where we deal with 8x8 blocks of quantized DCT-coefficients. During the embedding process, the data to be watermarked are extracted from the stream, analyzed, watermarked and placed back into the stream.

The main features of the proposed watermarking algorithm are as follows:

1) Embedding and detection are performed without fully decompressing the video stream.
2) We chose to use I-VOPs only in our watermarking scheme. As the lack of the reference I-VOPs makes P- and B- encoded frame information unusable, this is well-justified, especially when GOVs are small. There has been a good deal of studies and argumentation to support this approach[32]-[38].
3) Watermark is inserted into quantized frequency domain of luminance components.
4) The watermarking procedure preserves the bitstream length.

At the system level, our watermarking software mimics MPEG-4 decoder: the video bitstream is parsed through the watermark embedder/detector; the headers are fully decoded; I-VOPs are identified and submitted for watermark embedding/ detection. According to MPEG standard, I-VOP is coded as the sequence of 16x16 macroblocks (MB); each MB comprises six 8x8 elementary blocks: four Y-blocks(luminance) followed by Cb-/Cr-blocks(chrominance). In our algorithm, the Y (luminance) channel is chosen to host watermark data. The watermarking signals are embedded into the LSB-plane of the quantized AC-coefficients of luminance blocks. For each 8x8 luminance block, after decoding quantized DC coefficient, we perform a procedure, as prescribed by DC/AC-prediction mechanism, to establish the scan order for AC coefficients (if necessary). Next, quantized AC coefficients are scanned one-by-one till the end of block is reached. Note, that in MPEG-4 there is no compulsory EndOfBlock indicator. Thereby, in the case when resync markers are absent, the only way to reach a block is to completely decode all preceding blocks, on a bitstream level.

The watermark data is a sequence of 0/1-$bits. The watermarking algorithm treats an I-VOP as a collection of 8x8 luminance blocks, that are processed sequentially. The algorithm's objective is to incorporate one watermark bit per each block. Given a block, the algorithm operates on a set of quantized AC-coefficients that are declared suitable for watermarking for this block.

According to MPEG standard, the quantized AC coefficients are coded by the means of Run-Length Coding (RLC) mechanism. The entropy encoding method can be either Huffman or Arithmetic encoding. In our work, we deal exclusively with Huffman encoding. MPEG-4 standard allows both Variable Length Codes (VLC) and Fixed Length Codes (FLC). The latter are used in case of Short Video Header, as well as in a situation when Escape mode 3 is used (Table B-18 in [1]). The Huffman codeword is the string of 0/1-bits. The codeword bitstrings are unique in the sense that no codeword may be a prefix of a bitstring for another valid codeword. The Huffman codewords translate into RLC-triples of the *(Last, Run, Level)*-form, where *Level* is the non-zero value of the (currently coded) coefficient, *Run* is the number of zero-value coefficients that immediately precede the current coefficient in the block, and *Last* $\in$ {0, 1} indicates whether the current coefficient is the last non-zero coefficient of the block.

Letxdenote the family of all feasible 3D RLC events (*Last, Run, Leve*l) to code an intra AC-coefficient in compliance with MPEG-4 standard.

www.dbpia.co.kr

Table 1. Examples of lcRLC events

RLC	RLC length	(Last, Run)	Level	LSB of Level
010101s 010011s	7	(0,0)	6 7	0 1
0000011 0 01101s 0000011 0 01100s	14	(0,0)	31 32	1 0

Let H denote the family of all valid Huffman codewords (i.e., RLC space) used by MPEG-4 for intra AC-coefficients encoding (Tables B-16, B-18, B-19 and B-21 in [1]). We use $l(x)$ to denote the length of Huffman RLC codeword for$x \in X$. By the virtue of feasibility, $l(x) > 0$, for any$x \in X$.

Definition 1. *An RLC event*$x = (F, R, L), x \in X$, *is called label--carrying RLC event (lcRLC event) if there exists an RLC event* $x' = (F, R, L')$, $x' \in x$, *such that*

(i) $L' \in \{L-1, L+1\}$;

(ii) $l(x) = l(x')$.

An event x' *is then called a co-RLC event for event* x.

Table 1 gives two examples of lcRLC events. Namely, two pairs of label-carrying (Last, Run, Level)-events are given. The Huffman codeword for the corresponding event is presented in the first two columns, where the column 1 gives the codeword itself (the symbol "s" represents the sign bit) and column 2 indicates the codeword's length. Columns 3 and 4 present the event, and the last column specifies the value of the Level's LSB. Note that the Huffman coding is in accordance with MPEG-4 standard: the first pair of events is coded by Table B-16 in [1] whereas the events of the second pair fall under Escape mode 1 (Escape '0' coding type) and are coded by Tables B-16 and B-19 in [1].

The proposed watermarking method operates as follows. Within a block, the algorithm goes through suitable for watermarking AC-coefficients targeting for an lcRLC event. When an lcRLC event is found, the algorithm watermarks it by the following mechanism: if the LSB of a coefficient (i.e., of the level of an RLC event) is the same as the watermark bit to be embedded, then there are no changes made; otherwise the original lcRLC is switched with its co-RLC event. In our watermarking scheme, the set of suitable candidates for watermarking is formed based on the number of consecutive non-watermarked blocks preceding the current block. In particular, at the outset of I-VOP watermarking, we set the value of Skipped Blocks Tolerance (SBT).

We are now ready to present a detailed description of our watermarking algorithm. The proposed watermarking software imitates an MPEG-4 decoder: the video bitstream is parsed; the headers are decoded; video elementary stream packets with coded I-VOP data are detected. The I-VOP is watermarked by sequential processing of macroblocks. Only luminance blocks within the macroblock are watermarked, the chrominance blocks stay unchanged. Specifically, for each I-VOP, the algorithm performs as follows:

Algorithm wmIVOP
Input: MPEG-4 coded I-VOP
Output: Watermarked I-VOP

Step 1. Subject to a given I-VOP, generate the sequence $W = (W[i])_{i=\overline{1,X}}$, $W[i] \in \{0, 1\}$ of watermark bits.

Step 2. Parse I-VOP bitstream, decoding macroblocks headers and identifying luminance blocks within each macroblock. For each luminance block B_i do the following:

Step 2.1 Decode quantized DC-coefficient.

Step 2.2 Identify block scan order by performing the reverse of DC-prediction mechanism.

Step 2.3 Form the set of AC-coefficients eligible for watermarking for this block.

Step 2.4 Set block status as "open".

Step 2.5 Till block is completely decoded:

(i) Perform Huffman decoding of the next non-

www.dbpia.co.kr

zero AC-coefficient.

(ii) If the block is open and the decoded coefficient is eligible for watermarking, go to Step (iii); otherwise go to Step (i).

(iii) If the coefficient produces an lcRLC event, force the LSB of the levelto the value of a watermark bit, $W[i \bmod X]$ (if the LSB coincides with a watermark bit there are no changes; otherwise the current lcRLC event is switched with its co-RLC event), and declare the block "closed".

Below, we briefly discuss the performance of Algorithm wmIVOP.

3.1.1 Generation of watermark data (Step 1)

The watermark data is a sequence of 0/1-bits to be embedded into I-VOP, at most one bit per block. The length of the sequence (X) should be sufficiently large. Watermark data is generated subject to a given I-VOP by the means of a set of hash functions and cryptographic procedures with a secret key. The secret key is generated on the basis of I-VOP size, the user-defined password and I-VOP recording timestamp data.

We also note that the security of watermark embedding procedure can be further strengthened by employing simple procedures to make the watermark data more dependant on the VOP content, such as the dependence of a watermark bit for the current block on the previously (non)watermarked blocks; the dependence of the watermark location on the block content (especially when there is a choice of an lcRLC event to watermark), etc.. Those easy to realize procedures would add extra precaution against watermark pirating in an attempt to produce forged video.

3.1.2 Determining the block scan order(Step 2.2)

The scan order for AC-coefficients is established based on the value of ac_pred_flag as decoded out off the macroblock header and (if applicable) the mechanism of DC-prediction. The latter can be efficiently executed as demonstrated by the following Property:

Property 1. *Reversing DC-prediction requires to keep in memory at most* $(\omega + 2)$ *DC-values from previously decoded blocks, where* ω *denote the VOP's width in 8x8 blocks.*

Proof. The procedure of Step 2.2 performed for a given block requires the knowledge of the DC-values of three adjacent blocks. Let $A_{i,j}$ denote a macroblock, such that $0 \leqq i < W, 0 \leqq j < H,$ where W and H denote the frame width and height, respectively, in macroblocks. Furthermore, let $A_{i,j}[x, y], x, y \in \{0, 1\}$, denote 8x8 luminance blocks of macroblock $A_{i,j}$

$A_{i,j}[0, 0]$ $A_{i,j}[0, 1]$

$A_{i,j}[1, 0]$ $A_{i,j}[1, 1]$

Then at the outset of decoding a macroblock A_{i^*,j^*}, it suffices to have in storage the values of DC-coefficients as used for DC-prediction for the following blocks:

$$\{\ A_{i,j^*}[1,x] \mid x \in \{0,\ 1\},\ 0 \leq i < i^*\ \} \cup$$
$$\{\ A_{i,j^*-1}[1,x] \mid x \in \{0,\ 1\},\ i^* \leq i < W\ \} \cup$$
$$\{\ A_{i^*-1,j^*}[0,\ 1]\ \} \cup \{\ A_{i^*-1,j^*-1}[1,\ 1]\ \}$$

After the decoding of each block its DC-value goes in storage by replacing the DC-value of its top-left diagonal block:

current block	$A_{i,j}[0, 0]$	$A_{i,j}[0, 1]$	$A_{i,j}[1, 0]$	$A_{i,j}[1, 1]$
top-left diagonal block	$A_{i-1,j-1}[1, 1]$	$A_{i,j-1}[1, 0]$	$A_{i-1,j}[0, 1]$	$A_{i,j}[0, 0]$

Thus, the DC of block $A_{i,j}[x, y], x, y \in \{0, 1\}$, replaces the DC of block $A_{i-\bar{y},j-\bar{x}}[\bar{x}, \bar{y}]$, where $\bar{0} = 1$ and $\bar{1} = 0$. We leave it to the reader to verify the correctness of the above argument. By the above procedure, the Property follows. ■

3.1.3 Identifying AC-coefficients eligible for watermarking (Step 2.3)

As mentioned earlier, at the outset of I-VOP watermarking, we set the value of Skipped Blocks Tolerance (SBT).

www.dbpia.co.kr

Let z be the number of unwatermarked blocks (i.e., the blocks that Algorithm wmIVOP failed to watermark) immediately preceding the current block, and let SBT denote the preset value of Skipped Blocks Tolerance. Set Define the matrix $U=\{u[i,j]\}_{i,j\in\{0,\ldots,7\}}$as follows: if $z<SBT$, then set $U=U'$, otherwise set $U=U''$. An AC-coefficient $c[i,j],0\le i,j\le 7$, is declared eligible for watermarking if $u[i,j]=1$.

Obviously, the main idea behind the above approach is to avoid changing the AC-coefficients of the block's first row/column, as those (and only those) coefficients could be used for AC-prediction for the blocks yet to be decoded. In effect, this is our measure to avoid the spatial propagation of watermark throughout the remainder of I-VOP as well as the temporal drift throughout the GOV. By extensive testing, we have chosen to set the value of SBT within $[7,15]$ interval.

$$U'=\begin{pmatrix}0&0&0&0&0&0&0&0\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\end{pmatrix}$$

$$U''=\begin{pmatrix}0&0&1&1&1&1&1&1\\0&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{pmatrix}$$

3.1.4 Watermark Detection

The watermarking scheme of Algorithm wmIVOP enables the watermark detection/extraction without the use of an original (non-watermarked) video sequence, i.e., the video authentication is done by "blind detection". Indeed, the watermark data is generated based on the parameters that remain unchanged. Therefore, the detection procedure can be viewed as the embedding algorithm that goes through the luminance blocks of I-VOPs and seeks to insert a watermark. If there are no changes to be made (all watermark bits are already in place), than the detection procedure reports successful output.

3.2 Experimental Results

An extensive series of experiments was conducted to examine the performance of the proposed watermarking scheme on real-life video sequences. The tests were aimed to investigate: 1) the watermark capacity, i.e., the watermarking information hidden in the picture should be sufficient for reliable video authentication; 2) the watermark fidelity, i.e., the watermark should not be noticeable to the viewer nor should the watermark degrade the quality of the content. We tested the algorithm on MPEG-4 bitstreams supported under SP/ASP. The tests were conducted for various video image sizes (in pixels): 640x480, 320x240 and 160x112. The video sequences contained smooth areas, textured areas and sharp edges.

For our algorithm, the watermark capacity is evidently determined by the population of lcRLC events in bitstream. Our experiments support the known observation[1] that the lcRLC events are fairly uniformly distributed over the DCT-spectrum, and thus, each non-zero AC coefficient has an equal probability of being watermarked. Note that, when an AC coefficient is coded in FLC mode[1], the operation of LSB substitution with preserving the codeword length is trivially executed (under the obvious proviso that Escape mode 3, Table 18a/b in [1], is prohibited for combinations listed in Table 16 in [1]). Thus, our prime concern is VLC codespace. According to MPEG-4 standard (tables B-16, B-19, B-21 in [1]), there exist plenty of lcRLC events. To be exact, 38% (64 out of 169) of VLC-events with Last=0 are lcRLC events, and 6% (6 out of 93) of VLC-events with Last=1 are lcRLC events (both Escape mode 1 and Escape mode 2 are taken into account). This fact justifies our expectations for the high capacity of the proposed watermark-embedding scheme. The quantitative results of the empirical evaluation demonstrate that, indeed, the algorithm embeds an ample amount of

www.dbpia.co.kr

watermark data to provide reliable hard authentication.

Evidently, a block with more non-zero AC-coefficients has more chances of being watermarked. We will refer to an 8x8 Elementary Block (EB) of quantized DCT coefficients as *non-empty* if it contains at least one none-zero AC-coefficient, and *empty*, otherwise. Furthermore, we use a term of "*lc-EB*" for an Elementary Block that contains at least one lcRLC event. Our tests show that the watermarking capacity of VOP appears to be strongly dependent on two factors: (i) the amount of non-empty EBs in I-VOP and (ii) the population of lcRLC events. These two factors are highly correlated, though some abnormal exceptions can take place, as we observed. Obviously, one of two major constraints on the presence of non-empty EBs in I-VOP is a bit-rate. The other constraint is the VOP content itself. In particular, smooth areas are coded with only one or a few DCT-coefficients, and thus have little chance to be affected by watermarking. Hence, the watermark-carrying capacity strongly depends on the presence of edges and textured areas in the video content. This, in fact, explains why the watermark capacity may vary greatly from GOV to GOV in the same bitstream.

The video sequences chosen for our experimental tests came from two sources: the series of sample MPEG-4 coded benchmark clips and the video recorded in real-life conditions by a DVR system. The sample benchmark clips were used to test the behaviour of the algorithm for its capacity and fidelity under different conditions and for various video data parameters. Table 2 portraits the numerical results of our testing. The first two columns give the image size in pixels and the recording bit-rate. The third column shows the percentage of non-empty EBs in I-VOP; the fourth column gives the percentage of lcRLC events (only AC values are taken into account, DC values are ignored); the fifth column reports on the percentage of lc-blocks (i.e., the blocks that contain at least one lcRLC event and, thus, can be watermarked if the corresponding AC-coefficient is declared eligible for watermarking); and the last column displays the outcome of watermarking done by Algorithm wmIVOP, with the value of SBT=10.

As the data from Table 2 shows, the VOP's watermarking capacity may vary dramatically.

Our main subject of investigation has been watermarking of real-life video bitstreams produced by a DVR surveillance system. In particular, we conducted the exhaustive testing of video data coded with the following characteristics: (i) image size (in pixels): 320x240 and 160x112, bit-rate in the range of 200-500Kbit/s, and (ii) image size: 640x480, bit-rate in the range of 0.9-2.0 Mbit/s. All sequences were coded at 30 frames/sec, with GOV size being set to 30/60 VOPs. The experi-

Table 2. The numerical results of test

size (pixels)	bit-rate (Kbits/s)	$\frac{empty\ EBs}{all\ EBs}$ (%)	$\frac{lcRLC\,events}{all\ RLC\,events}$ (%)	$\frac{lc-EBs}{non-empty\ EBs}$ (%)	$\frac{W\,Marked\ EBs}{non-empty\ EBs}$ (%)
160x112	200~500	0~5	25~30	78~85	50~65
320x240	800~900	0~5	20~25	65~75	25~35
320x240	800~900	6~15	13~20	20~30	7~11
320x240	300~500	10~15	23~25	43~53	15~20
320x240	250~350	13~20	27~31	60~70	32~35
320x240	900~1000	15~20	15~18	20~45	6~15
640x480	1200~2000	15~30	22~25	45~52	16~19
320x240	600~800	20~23	25~28	75~80	27~30
640x480	800~1000	30~35	20~25	45~55	11~20
640x480	900~1200	45~50	10~11	15~20	3~8
320x240	800~900	50~55	14~15	30~32	5~6
320x240	800~900	50~75	6~11	13~23	5~10

www.dbpia.co.kr

ments demonstrated that Algorithm wmIVOP generally produced 15-30% watermarked blocks per intra-coded VOP, which has been accepted as very satisfactory.

The advantage of LSB embedding is that it allows high perceptual transparency as the watermark embedding yields perceptually invisible degradation in the image quality. Furthermore, by the choice of eligible-for-watermarking AC coefficients, the proposed algorithm takes a simple precaution to avoid the temporal/spatial propagation of watermarking artefacts. Accordingly, the practical experiments with the various MPEG-4 streams demonstrate that watermark embedding preserves the video quality. The informal subjective tests show that the perceptual impact of the watermarking is negligible as watermarking does not introduce any perceptible artefacts, and the embedded data is invisible under normal viewing conditions. Furthermore, we have conducted PSNR(Peak-Signal-to-Noise Ratio) analysis to numerically measure the distortion effect of watermarking on video quality. The PSNR was calculated on the uncompressed luminance channel for I-VOPs of the original(compressed but not watermarked) video sequences and the compressed-and-watermarked sequences. The PSNR [dB] is defined as follows:

$$PSNR = 20log_{10}\frac{255}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x}_i)^2}}$$

where x_i and $\overline{x}_i$ denote the luminance values in original and watermarked frames, respectively, and N stands for the total number of pixels. As an objective quality measure, the average PSNR above 40 is generally considered a solid guarantee against perceptible errors in realistic scenarios. For all video sequences watermarked by Algorithm wmIVOP with $SBT = 10$, the computed PSNR values were above 40 (all observed PSNR fell in the range [44,62]).

Ⅳ. Conclusion

A scheme for fragile watermarking of MPEG-4 compressed bitstreams was developed and implemented. The underlying method is LSB modification - a computationally efficient embedding technique with a high embedding capacity and small-degree embedding distortion. The watermarking procedure works in strict adherence to MPEG-4 coding standard and preserves the integrity of the MPEG-4 stream. Moreover, the bit-length of the stream remains unchanged. The embedded watermark can be used for video authentication and protection against tampering.

The proposed algorithm demonstrates a good level of security, as both preparation of the watermark payload and watermark insertion are guarded by the means of cryptography and original routines for watermark generation/embedding. Moreover, the watermark signals are made frame-dependent in a continuous manner(and thus, defeating frame-averaging attacks).

Embedding and detection are performed without fully de-multiplexing the video stream. The algorithm requires bitstream parsing and only minimal decoding as only inverse entropy encoding must be executed. Furthermore, the use of memory is essentially reduced to the feasible minimum. As the result, the algorithm demonstrates high computational efficiency which is necessary for real-time practical applications and also when a large number of video sequences must be watermarked. Since there are no lossy operations after watermark embedding, the watermark will exist intact in the quantized coefficients when the detection process is carried out. The watermark will remain intact across different user platforms and interfaces, as long as no recompression is performed.

The embedded watermark does not appear to be perceptible under normal observation, as well as it does not introduce noticeable artefacts into video content and functionality. The watermark extraction/detection does not require an original file, i.e., the video authentication is done by "blind detection".

Obtaining theoretical bounds on the watermark capacity/fidelity for the LSB modification in quantized DCT domain as well as the further explora-

www.dbpia.co.kr

tion of RLC codeword space of MPEG-4 compressed bitstreams is an interesting topic that deserves further study. Another issue for future work is an extension of the developed technique for watermarking inter-coded data of MPEG-4 video.

REFERENCE

[1] ISO/IEC 14496-2: "*Information technology - coding of audio-visual objects: Part 2 -- Visual*", 2001.

[2] ISO/IEC JTC1/SC29/WG11: "*Coding of moving pictures and audio: MPEG4 Video Verification Model version 18.0*", 2001.

[3] T. Ebrahimi, F. Pereira "*The MPEG-4 Book*", Prentice Hall PTR, 2002.

[4] I.E.G. Richardson "*H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*", John Wiley & Sons, 2003.

[5] A.M. Alattar, E.T. Lin and M.U. Celik "*Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video*", IEEE Transactions CirSysVideo (13-8), 2003.

[6] A.M. Alattar, E.T. Lin and M.U. Celik "*Watermarking low bit-rate Advanced Simple Profile MPEG-4 Bitstreams*", ICASSP Proceedings, 2003.

[7] A.M. Alattar, M.U. Celik and E.T. Lin "*Evaluation of watermarking low bit-rate MPEG-4 bit streams*", SPIE (5020) 2003.

[8] S. Arena, M. Caramma, R. Lancini "*Data hiding in the bit stream domain for Mpeg-2 coded video sequences exploiting space and frequency masking*", ICASSP Proceedings, 2000.

[9] M. Barni, F. Bartolini, V. Capellini and N. Checcacci "*Object watermarking for mpeg-4 video streams copyright protection*", SPIE Proceedings (3971) 2000.

[10] P. Bas, N. V. Boulgouris, F. D. Kovaros, J-M Chassery, M. G. Strintzis and B. Macq "*Robust Watermarking of Video Object for MPEG-4 Applications*", SPIE 2001.

[11] P. Bas and B. Macq "*A New Video-object Watermarking Scheme Robust to Object Manipulation*", ICIP 2001.

[12] R.J. Berger II, B.G. Mobasseri, "*Watermarking in JPEG Bitstream*", Proc. SPIE Conference on Security, Steganography and Watermarking of Multimedia Contents VII, San Jose, January 2005.

[13] M.U. Celik, G. Sharma, A.M. Tekalp and E. Saber "*Lossless generalized LSB data embedding*", IEEE Trans. on Image Processing, 2004.

[14] L. Chun-Shien, J. Chen, H. Liao, and K. Fan, "*Real-Time MPEG-2 Video Watermarking in the VLC Domain*", International Conference on Pattern Recognition (Vol. 2), 2002.

[15] D. Cinalli, B. Mobasseri, and C. O'Connor "*MetaData Embedding in Compressed UAV Video*", American Society of Naval Engineers, Intelligent Ship Symposium V, Philadelphia, May 2003.

[16] I.J. Cox, J. Kilian, F.T. Leighton and T. Shamoon, "*Secure Spread Spectrum Watermarking for Multimedia*", Image Processing 6(12), 1997.

[17] D. Cross and B.G. Mobasseri "*Watermarking for self-authentication of compressed video*", Proc. IEEE International Conference on Image Processing," Rochester, NY, September 2002.

[18] G. Doerr and J.L. Dugelay "*A guide tour of video watermarking*", Signal Processing: Image Communications 18(4), 2003.

[19] G. Doerr, and J.L. Dugelay "*Secure Video Watermarking via Embedding Strength Modulation*", Proceedings of Digital Watermarking Workshop, 2003.

[20] G. Doerr and J.-L. Dugelay "*Video Watermarking: Overview and Challenges*" chapter 42 in Handbook of Video Databases: Design and Applications, CRC Press, 2003.

[21] J. Fridrich, M. Goljan, Q. Chen, and V. Pathak "*Lossless Data Embedding with File Size Preservation*" Proc. Of EI SPIE, Security and Watermarking of Multimedia Contents VI, vol. 5306, San Jose, 2004.

[22] F. Hartung and B. Girod "*Watermarking of MPEG-2 encoded video without decoding and re-encoding*", Multimedia Computing and Networking (MMCN) Proceedings, 1997.

[23] F. Hartung and B. Girod "*Digital Watermarking*

www.dbpia.co.kr

of MPEG-2 coded video in the bitstream domain", ICASSP Proceedings (Vol. 4) 1997.

[24] F. Hartung and B. Girod "*Watermarking of Uncompressed and Compressed Video*", Signal Processing 66(3), 1998.

[25] Y. Hwang, B. Jeon, and T.M. Chung "*Improved Error Detection Method for Real-time Video Communication using Fragile Watermarking*", in Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing, Lecture Notes on Computer Science (vol. 2352) 2002.

[26] G.C. Langelaar and R.L. Lagendijk "*Optimal Differential Energy Watermarking of DCT Encoded Images and Video*", IEEE Transactions on Image Processing, 10(1), 2001.

[27] G.C. Langelaar, R.L. Lagendijk, and J. Biemond, "*Real-time labeling of MPEG-2 compressed video*", J. Visual Commun. Image Representation, 9(4), 1998.

[28] G.C. Langelaar, I. Setyawan, and R.L. Lagendijk "*Watermarking Digital Image and Video Data*" IEEE Signal Processing Magazine, September 2000.

[29] C.T. Li and Y. Yuan "*Digital Watermarking Schemes for Multimedia Authentication*", in Digital Watermarking for Digital Media, Idea Group Publishing, 2005.

[30] E. T. Lin, C. I. Podilchuk, T. Kalker and E.J. Delp "*Streaming video and rate scalable compression: what are the challenges for watermarking?*", Journal of Electronic Imaging 13(1), 2004.

[31] H.-H. Liu, L.-W. Chang "*Real Time Digital Video Watermarking for Digital Rights Management via Modification of VLCs*", 11th International Conference on Parallel and Distributed Systems (ICPADS'05), 2005.

[32] C.S. Lu, J.R. Chen, H.Y.M. Liao, K.C. Fan "*Real-time frame dependent watermarking in MPEG-2 video*", TR-IIS-04-004, Institute of Information Science, Academia Sinica, Taiwan, ROC, 2004.

[33] C.S. Lu, J.R. Chen and K.C. Fan "*Real-time frame-dependent video watermarking in VLC domain*", Signal Processing: Image Communication 20, 2005.

[34] B. G. Mobasseri, R. J. Berger II, "*A Foundation for Watermarking in Compressed Domain*", IEEE Signal Processing Letters, 12(5), 2005.

[35] B.G. Mobasseri, M.P. Marcinak "*Watermarking of MPEG-2 Video in Compressed Domain Using VLC Mapping*", ACM Multimedia and Security Workshop, 2005.

[36] D. Nicholson, P. Kudumakis and J.-F. Delaigle "*Watermarking in the MPEG-4 context*", Lecture Notes In Computer Science, Proceedings of ECMAST, 1999.

[37] L. Qiao and K. Nahrstedt "*Watermarking schemes and protocols for protecting rightful ownership and customer's rights*", Journal of Visual Communication and Image Representation (9) 1998.

[38] L. Qiao and K. Nahrstedt "*Watermarking methods for MPEG encoded video: towards resolving rightful ownership*", Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS), 1998.

[39] D. Simitopoulos, S.A.Tsaftaris, N.V. Boulgouris and M.G. Strintzis "*Fast MPEG watermarking for copyright protection*", IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2002.

[40] M.D. Swanson, M. Kobayashi and A.H. Tewfik, "*Multimedia Data Embedding and Watermarking Technologies*", Proc. IEEE 86(6) 1998.

[41] B.B. Zhu and M.D. Swanson "*Multimedia Authentication and Watermarking*", in Multimedia Information Retrieval and Management, Springer-Verlag, 2003.

[42] B.B. Zhu, M.D. Swanson and A.H. Tewfik "*When Seeing Isn't Believing*", IEEE Signal Processing Magazine, March 2004.

Inna G. Drobouchevitch 정회원
하이트론씨스템즈 연구원
<관심분야> 보안, 멀티미디어, 신호처리

www.dbpia.co.kr

임 성 준 (Sungjun Lim) 정회원
현재 연세대학교 정보대학원 박사과정, 동원대학 모바일컨텐츠과 조교수
<관심분야> 멀티미디어, 네트워크, 신호처리

한 병 완 (Byungwan Han) 정회원
현재 동원대학 애니메이션과 조교수
<관심분야> 멀티미디어, 네트워크, 임베디드 시스템

장 항 배 (Hangbae Chang) 정회원
현재 연세대학교 박사
<관심분야> 정보보호, 유비쿼터스시스템

김 경 규 (KyungKyu Kim) 정회원
현재 연세대학교 정보대학원 교수
<관심분야> 유비쿼터스 시스템

www.dbpia.co.kr