IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.
PP 24-30
www.iosrjournals.org

FPGA Implementation of a Digital Atermarking System for Video Authentication

Anju Devassy,Aparna M.S,Shaly Laurence,Sona Davis Chakkoria
PG Scholar DEPT Of Electronics and communication, Sahrdaya college of Engineering, Kerala, India

Abstract: *Digital video sequences are very susceptible to manipulations and alterations using widely available editing tools. So some authentication techniques are needed in order to maintain authenticity, integrity, and security of digital video content. As a result, digital watermarking (WM), a data hiding technique has been considered as one of the key authentication methods. This paper presents a hardware implementation of a digital watermarking system that can insert invisible, semifragile watermark information into compressed video streams in real time. The watermark embedding is treated in the discrete cosine transform domain. It's a hardware-based video authentication system using this watermarking technique structures minimum video quality degradation and can withstand certain potential attacks, i.e., cover-up attacks, cropping, and segment removal on video sequences. Furthermore, the proposed hardware based watermarking system features low power consumption, low cost implementation, high processing speed, and reliability.*

Keywords: *Authentication, Digital watermarking, FPGA,hardware implememtation,Robust Techniques, Semi-fragile very largewatermarking scale integration (VLSI), video authentication.*

I. Introduction

Digital video sequences are very susceptible to manipulations and variations using commonly available editing tools. This issue turns to be more important when the video sequence is to be used as evidence. So some validation techniques are needed in order to maintain authenticity, integrity, and security of digital video content. As a result, digital watermarking (WM)[1], a data hiding technique has been considered as one of the key authentication methods. To provide copy protection and copyright protection[2][3] for digital audio and video data, twocorresponding techniques are being developed: encryption and watermarking. Encryption techniques can be used to secure digital data during the transmission from the sender to the receiver. However, after the receiver has received and decrypted the data, the data is in the clear and no longer sheltered. Watermarking techniques can complement encryption by embedding a secret imperceptible signal, a watermark, directly into the clear data. This watermark signal[4] is embedded in such a way that it cannot be detached without disturbingthe quality of the audio or video data. The watermark signal can for occurrence be used for copyright protection as it can hide information about the author in the data. The watermark can now be used to prove ownership in court. Another interesting application for which the watermark signal can be used is to trace the source of illegal copies by means of fingerprinting techniques.

In this case, the media worker embeds watermarks in the copies of the data with a serial number that is related to the customer's identity. If banned copies are found, for instance on the Internet, the intellectual property owner can easily identify customers who have broken their license agreement by supplying the data to third parties. The watermark signal can also be used to control digital recording devices as it can show whether certain data may be recorded or not. In such case the recording devices must be equipped with watermark detectors, of course. Other applications of the watermark signal include: automated monitoring systems for radio and TV broadcasting, data authentication and transmission of secret messages. Each watermarking application has its own specific requirements. Nevertheless, the most important requirements that are to be met by most watermarking techniques are that the

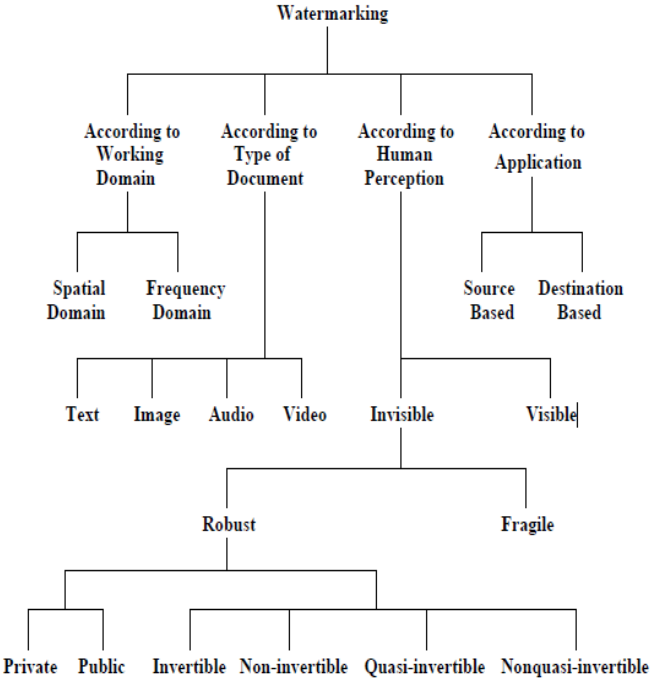
Watermark is imperceptible in the data in which it is hidden, that the watermark signal can contain a reasonable amount of information and that the watermark signal cannot be removed easily without affecting the data in which the watermark is hidden.The main objective of this paper is to describe an efficient hardware-based concept of a digital video WM system, which features low power consumption, efficient and low cost implementation[5], high processing speed, reliability and invisible, and semifragile watermarking in compressed video streams. It works in the discrete cosine transform (DCT) domain in real time. The proposed WM system can be joined with video compressor unit, and it attains performance that matches complex software algorithms within a simple efficient hardware implementation. The system also features minimum video quality degradation and can withstand certain potential attacks, i.e., cover-up attacks, cropping, segment removal on video sequences. The proposed WM system is realized using the Verilog hardware description language (HDL)

synthesized into a field programming gate array (FPGA) and then investigated using a custom versatile breadboard for performance evaluation.

II. Video Watermarking Systems

A: Types Of Digital Watermarks

Watermarks and watermarking techniques can be distributed into various categories in several ways. The watermarks can be applied in spatial domain. An alternative to spatial domain watermarking is frequency domain watermarking. It has been pointed out that the frequency domain methods are more robust than the spatial domain techniques. Different types of watermarks are shown in the figure below.



B: Robustness

A digital watermark is called "fragile" if it fails to be detectable after the slightest modification. Fragile watermarks are commonly used for tamper detection (integrity proof). Modifications to an original work that clearly are noticeable, commonly are not referred to as watermarks, but as generalized barcodes.A digital watermark is called semi-fragile if it resists benign transformations, but fails detection after malignant transformations. Semi-fragile watermarks commonly are used to detect malignant transformations.A digital watermark is called robust if it resists a designated class of transformations. Robust watermarks may be used in copy protection applications to carry copy and no access control information.The semifragile approaches are generally processed in the frequency domain.

C. The Watermarking In The Frequency Domain

Several approaches can be used in the frequency domain, for example, JPEG-based [6], spread spectrum , and content-based approaches [7]. How can we embed data into the frequency domain of a host image and it appears unperceivable? The transformation functions often-used are DCT, DWT, and DFT. Generally, we can insert data into the coeGcients of a transformed image. As shown in Figs. 1 and 2, we embed watermarkinto the coeGcients of a transformed host image. The important consideration is what locations are best to place for embedding watermarkin the frequency domain to avoid distortion . Let Hm and Wn be the subdivided images from H and W, respectively, Hm DCT be the image transformed from Hm by DCT, and Hm F be the image combined by Hm DCT and Wn in the frequency domain.
The algorithm is described as follows.
Algorithm:

1. Divide the host image into sets of 8 × 8 blocks.
H = {h(i; j); 06i; j¡N},
Hm = {hm(i; j); 06i; j¡8}, where hm(i; j)∈{0; 1;
2; : : : ; 2L − 1} and m is the total number of the 8 × 8 blocks.
2. Divide the watermarkimage into sets of 2 × 2 blocks.
W = {w(i; j); 06i; j¡M},
Wn={wn(i; j); 06i; j¡2}, where wn(i; j)∈{0; 1} and n is the total number of the 2 × 2 blocks.
3. Transform Hm to Hm DCT by DCT.
4. Insert Wm into the coeGcients of HmDCT .
Hm F={hm F (i; j) = hm DCT (i; j)⊕wm(i; j); 06i; j¡8};
hm DCT (i; j)∈{0; 1; 2; : : : ; 2L − 1}:
5. Transform the embedded host image, HmF , by Inverse DCT.
The criterion for embedding the watermarkimage into the frequency domain of a host image is that the total number
of 8×8 blocks must be larger than the total number of 2×2 blocks.

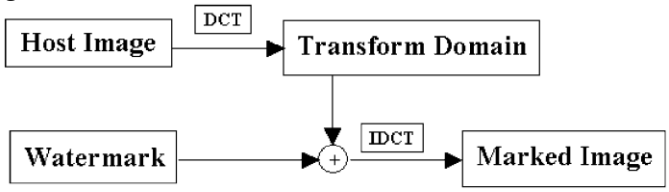


Fig. 1. The Fowchart in frequency domains.

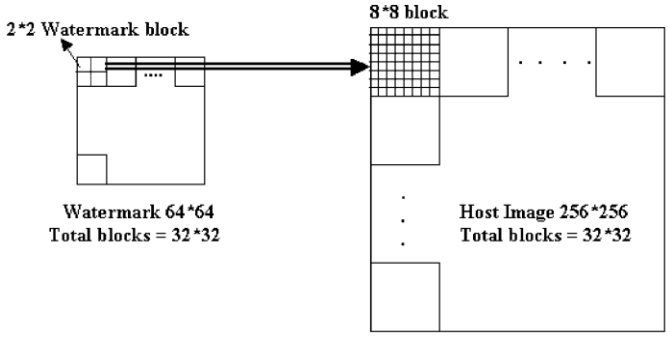


Fig. 2. The embedding skill in frequency domain.

Cannot view whom the writer is in the images. Therefore, itis embedded into the frequency domain and the rest into the spatial domain. In this way, we not only enlarge the capacity, but also secure the information that we are concernedwith.

C. Watermark Implementations

A WM system can be implemented on either software or hardware platforms, or some combinations of the two. By programming the code and making use of available software tools, it can be easy to design and implement any WM algorithm at various levels of complexity. the software approach has the advantage of flexibility, computational limitations may arise when attempting to utilize these WM methods for video signals or in portable devices. . Therefore, there is a strong incentive to apply hardware-based implementation for real-time WM of video streams. The hardware-level design offers several distinct advantages over the software implementation in terms of low power consumption, reduced area, and reliability. It enables the addition of a tiny, fast and potentially cheap watermark embedder as a part of portable consumer electronic devices. Such devices can be a digital camera, camcorder, or other multimedia devices, where the multimedia data are watermarked at the origin.

D Video Watermarking

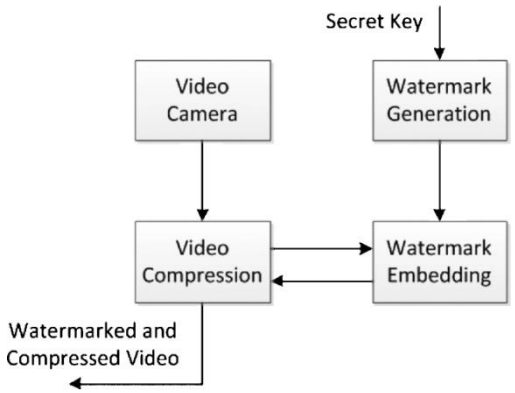


Fig. 3 . Overview of the proposed video WM system.

In general, digital WM techniques proposed so far for media authentication are usually designed to be visible or invisiblerobust or invisible-fragile watermarks according to the level of required robustness [8], [9]. Each of the schemes is equally important due to its unique applications. In this paper, however, we present the hardware implementation of the invisible semifragile watermarking system for video authentication. The motivation here is to integrate the video watermarking system with a surveillance video camera for real-time watermarking in the source end. Our work is the first semifragile watermarking scheme for video streams with hardware architecture.

Fig. 3 illustrates the general block diagram of the proposed system that is comprised of four main modules: a video camera, video compression unit, watermark generation, and watermark embedding units. Compression is divided into three elementary phases: DCT transformation, quantization, and Huffman encoding. Each of the video frames undergoes 8 × 8 block DCT and quantization. Then, they are passed to the watermark embedding module. The watermark generation unit produces a specific watermark data for each video frame, based on initial predefined secret keys. The watermark embedding module inserts the watermark data into the quantized DCT coefficients for each video frame according to the algorithm detailed below. Finally, watermarked DCT coefficients of each video frame are encoded by the video compression unit which outputs the compressed frame with embedded authentication watermark data.

III. Video Compression

Video compression enables more efficient use of transmission and storage resources.If a high bitrate transmission channel is available, then it is a more attractive proposition to send high-resolution compressed video or multiple compressed video channels than to send a single, low-resolution, uncompressed stream. Even with constant advances in storage and transmission capacity, compression is likely to be an essential component of multimedia services for many years to come. An information-carrying signal may be compressed by removing redundancy from the signal. In a lossless compression system statistical redundancy is removed so that the original signal can be perfectly reconstructed at the receiver. Unfortunately, at the present time lossless methods can only achieve a modest amount of compression of image and video signals. Most practical video compression techniques are based on lossy compression, in which greater compression is achieved with the penalty that the decoded signal is not identical to the original. The goal of a video compression algorithm is to achieve efficient compression whilst minimising the distortion introduced by the compression process. Video compression algorithms operate by removing redundancy in the temporal, spatial and/or frequency domains.

Generally, a video sequence is divided into multiple group of pictures (GOP), representing sets of video frames which are neighboring in display order. An encoded MPEG-2 video sequence is made up of two frame-encoded pictures: intraframes (I-frame) and interframes (P-frame or B-frame). P-frames are forward prediction frames and B-frames are bidirectional prediction frames. Within a typical sequence of an encoded GOP, P-frames may be 10% of the size of I-frames and Bframes are about 2% of the I-frames.

There can be two types of redundancies in video frames: temporal redundancy and spatial redundancy. MPEG-2 video compression technique reduces these redundancies to compress the images.

Within a GOP, the temporal redundancy among the video frames is reduced by applying temporal differential pulse code modulation (DPCM). The major video coding standards, such as H.261, H.263, MPEG-1,

MPEG-2, MPEG-4, and H.264, are all based on the hybrid DPCM/DCT CODEC, which incorporates motion estimation and motion compensation function, a transform stage and an entropy encoder [10], [11]. It has been illustrated in Fig. 4 that an input video frame $F_n$is compared with a reference frame (previously encoded) F'_{n-1} and a motion estimation function finds a region in F'_{n-1} that matches the current macro-block in F_n. The offset between the current macro-block position and the chosen reference region is a motion vector, d_k. Based on this d_k, a motion compensated prediction $F_n$is generated, and it is then subtracted from the current macro-block to produce a residual or prediction error, e [12]. For proper decoding this motion vector, d_k, has to be transmitted as well.

The spatial redundancy in the prediction error, e (also called the displaced frame difference) of the predicted frames, and the I-frame is reduced by the following operations: each frame is split into blocks of 8 × 8 pixels that are compressed using the DCT followed by quantization (Q) and entropy coding (run-level-coding and Huffman coding) (Fig. 4).

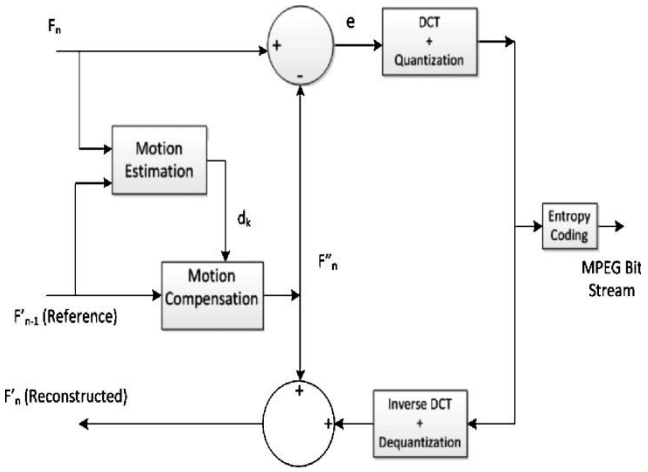


Fig .4. Block diagram of hybrid DPCM/DCT coding scheme

IV. Watermark Generation

Since simple watermark data can be easily cracked, it is essential that the primitive watermark sequence will be encoded by an encipher. This insures that the primitive watermark data are secured before being embedded into each video frame. The WM generator generates a secure watermark sequence for each video frame using a meaningful primitive watermark sequence and secret input keys. a primitive watermark pattern can be defined as a meaningful identifying sequence for each video frame. As shown in Fig. 5, the unique meaningful watermark data for each video frame contain the time, date, camera ID, and frame serial number (that is related to its creation). This will establish a unique relationship of the video stream frames with the time instant, the specific video camera, and the frame number. Any manipulation, such as frame exchange, cut, and substitution, will be detected by the specific watermark. The corresponding N-bit (64-bit) binary valued pattern, $a_{i,}$, will be used as a primitive watermark sequence. This would generate a different watermark for every frame (time-varying) because of the instantaneously changing serial number and time.

The block diagram of the proposed novel watermark generator is depicted in Fig. 6. A secure watermark pattern is generated by performing expanding, scrambling, and modulation on a primitive watermark sequence. There are two digital secret keys: Key 1 is used for scrambling and Key 2 is used for the random number generator (RNG) module that generates a pseudorandom sequence.

Initially, the primitive binary watermark sequence, a_i(of 64 bit), is expanded (a_i') and stored in a memory buffer. It is expanded by a factor c_r. For example, if we use a 64bit primitive watermark sequence then for a 256 × 256- pixels video frame, c_r will be (256 × 256/(8 × 8)) or 1024. This is done to meet the appropriate length for the video frame.

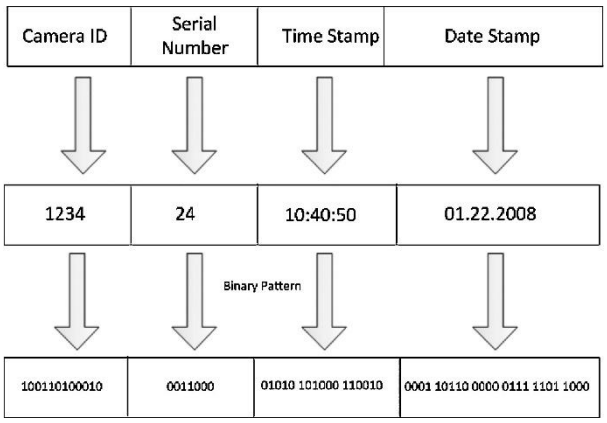


Fig. 5. Structure of the primitive watermark.

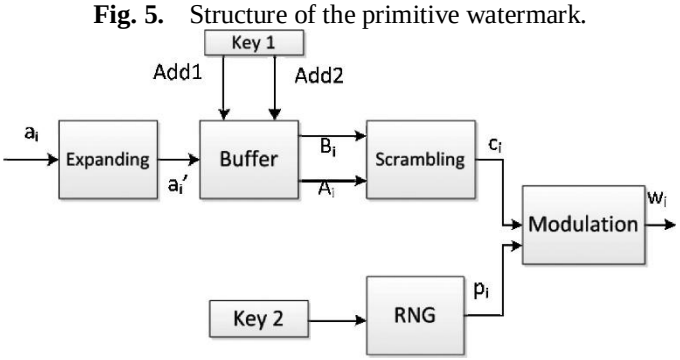


Fig. 6. Block diagram of the proposed watermark generator.

Scrambling is actually a sequence of XOR operations among the contents (bytes) of the expanded primitive WM in the buffer. Key 1 initiates the scrambling process by specifying two different addresses (Add1 and Add2) of the buffer for having the XOR operation in between them. The basic purpose of scrambling is to add complexity and encryption in the primitive watermark structure. After that, the expanded and scrambled sequence c_i is obtained. The bit size of c_i is the same as the size of the video of frame. Finally, the expanded and scrambled watermark sequence, c_i, is modulated by a binary pseudorandom sequence to generate the secured watermark sequence w_i. Due to the random nature of the pseudorandom sequence p_i, modulation makes the watermark sequence c_i a pseudorandom sequence and thus difficult to detect, locate, and manipulate.
A secure pseudorandom sequence p_i used for the modulation can be generated by an RNG structure using the Key 2.

V. Watermark Embedding

The watermarking algorithm should be hardware friendly in a way that it can be implemented in hardware with high throughput. For this purpose, one concern for the algorithm development should be that it must support pipelining architecture so that two or more macroblocks inside a single video frame or more than one frame can be watermarked simultaneously. This feature will aid in increasing the processing speed of watermarking.
The proposed WM algorithm along with MPEG-2 video encoding standard is presented as a flow chart in Fig. 7. This can be described as follows.

1) Split I frame and watermark data into 8 × 8 blocks.
2) For each 8×8 block (both watermark data and I frame), perform DCT, quantization, and zig–zag scan to generate quantized DCT coefficients.
3) Identify N watermarkable cells for each block and calculate the modification value for each selected cell.
4) Modify the identified watermarkable DCT coefficients according to the modification values.
5) Modify the identified watermarkable DCT coefficients according to the modification values.
6) Perform inverse DCT and inverse quantization for each 8 × 8 block watermarked coefficient to reconstruct the original I pixel values.
7) Buffer the reconstructed watermarked I frame.

8) Perform motion estimation for B/P frames to obtain the motion vector.
9) Using the motion vector and reconstructed watermarked I frame motion compensation is done.
10) Difference between the motion-compensated prediction frame and the watermarked reference frame I is the prediction error.
11) Perform DCT, quantization, and zig–zag scan on the prediction error.
12) Perform entropy coding for the blocks of the different frames.
13) Generate compressed and watermark embedded video steam.
14) To avoid heavy computationally demanding operations and to simplify the hardware implementation, watermarking can be done with MJPEG standard video compressing unit. Since watermark is only embedded on I frames, the steps stated above will be the same for the MJPEG video standard except for the motion estimation and motion compensation.

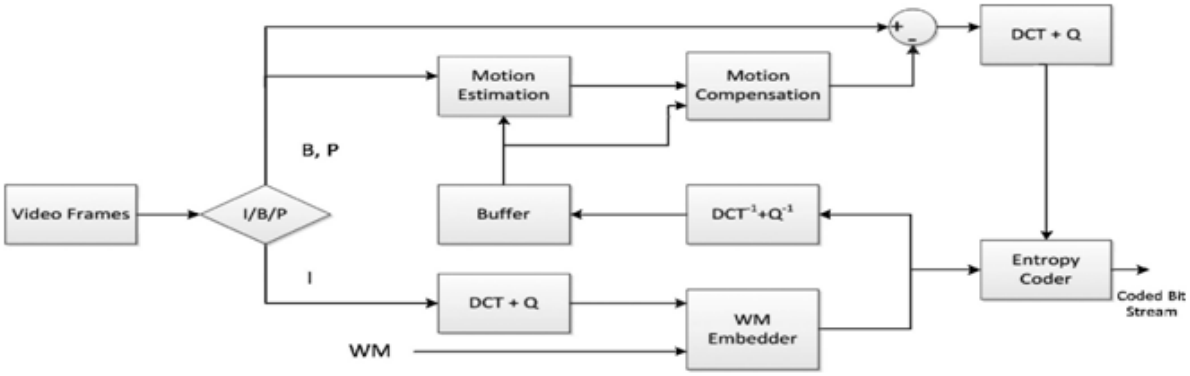


Fig. 7 Dataflow of the proposed WM algorithm.

Reference

[1]. V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in Proc. IEEE Int. Conf. Ind. Informatics, Aug. 2005, pp. 709–716.
[2]. A. D. Gwenael and J. L. Dugelay, "A guide tour of video watermarking," Signal Process. Image Commun., vol. 18, no. 4, pp. 263–282, Apr. 2003.
[3]. A. Piva, F. Bartolini, and M. Barni, "Managing copyright in open networks," IEEE Trans. Internet Comput., vol. 6, no. 3, pp. 18–26, May– Jun. 2002.
[4]. Y. Shoshan, A. Fish, X. Li, G. A. Jullien, and O. Yadid-Pecht, "VLSI watermark implementations and applications," Int. J. Information Technol. Knowl., vol. 2, no. 4 pp. 379–386, Jun. 2008.
[5]. X. Li, Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, "Hardware implementations of video watermarking," in International Book Series on Information Science and Computing, no. 5. Sofia, Bulgaria: Inst. Inform. Theories Applicat. FOI ITHEA, Jun. 2008, pp. 9–16 (supplement to the Int. J. Inform. Technol. Knowledge, vol. 2, 2008).
[6]. K.E. Zhao J., Embedding robust labels into images for copyright protection, Technical Report, Fraunhofer Institute for Computer Graphics, Darmatadt, Germany, 1994.
[7]. P. Bas, J.-M. Chassery, B. Macq, Image watermarking: an evolution to content based approaches Pattern Recognition 35 (2002) 545–561.
[8]. L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, "Implementation of a real-time digital watermarking process for broadcast monitoring on Trimedia VLIW processor," Proc. Inst. Elect. Eng. Vision, Image Signal Process., vol. 147, no. 4, pp. 371–376, Aug. 2000.
[9]. Y. Shoshan, A. Fish, X. Li, G. A. Jullien, and O. Yadid-Pecht, "VLSI watermark implementations and applications," Int. J. Information Technol. Knowl., vol. 2, no. 4 pp. 379–386, Jun. 2008.
[10]. X. Li, Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, "Hardware implementations of video watermarking," in International Book Series on Information Science and Computing, no. 5. Sofia, Bulgaria: Inst. Inform. Theories Applicat. FOI ITHEA, Jun. 2008, pp. 9–16 (supplement to the Int. J. Inform. Technol. Knowledge, vol. 2, 2008).
[11]. K. Jack, Video Demystified: A Handbook for the Digital Engineer, 2nd ed. Eagle Rock, VA: LLH Technology Publishing, 2001.
[12]. I. E. G. Richardson, H.264 and MPEG-4 Video Compression. Chichester, U.K.: Wiley, 2003.