VOL. 10, NO. 17, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

FPGA IMPLEMENTATION OF A DIGITAL WATERMARKING SYSTEM FOR VIDEO AUTHENTICATION

Anju Devassy and Kavya Anoop
Sahrdaya College of Engineering, Kodakara, India
E-Mail: anjudevassy139@gmail.com

ABSTRACT

Digital video sequences are very vulnerable 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, semi fragile 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 survive 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, semi-fragile, watermarking scale integration.

INTRODUCTION

Digital watermarking [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, two corresponding techniques are being developed: encryption and watermarking. 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 disturbing the quality of the audio or video data. The watermark signal can for existence 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.

The main objective of this paper is to describe an effective 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 semi fragile watermarking in compressed video streams. It works in the discrete cosine transform (DCT) domain in real time.

The Watermarking in the Frequency Domain

Several methods can be used in the frequency domain, for example, JPEG-based [6], spread spectrum, and content-based approaches [7]. The transformation functions often-used are DCT, DWT, and DFT. Generally, we can insert data into the coefficients of a transformed image. As shown in Figure-1 and 2, we embed watermark into the coefficients of a transformed host image. The important consideration is what locations are best to place for embedding watermark in the frequency domain to avoid distortion.

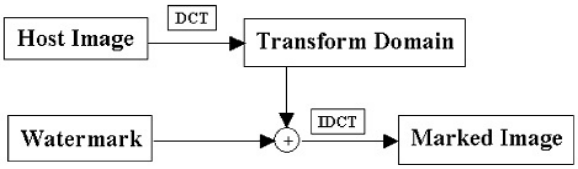

Figure-1. The Flowchart in frequency domains.

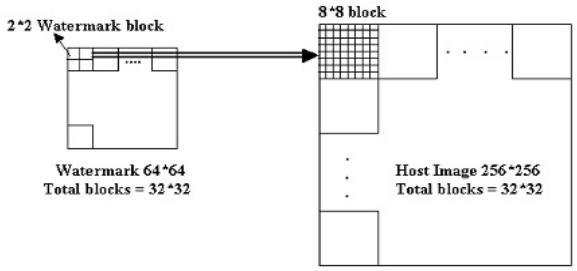

Figure-2. The embedding skill in frequency domain.

Video Watermarking

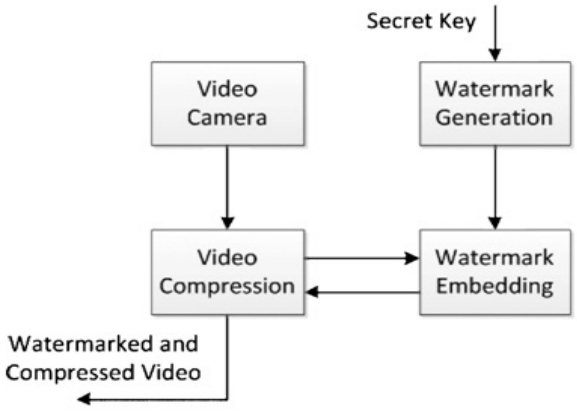

Figure-3. Overview of the proposed video WM system.

In general, digital WM techniques proposed so far for media authentication are usually designed to be visible or invisible robust or invisible-fragile watermarks according to the level of required robustness [8-9]. Figure-3 shows the general block diagram of the planned system. Compression is divided into three elementary phases: DCT transformation, quantization, and Huffman encoding. Every single 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 first 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 [10-12] of each video frame are encoded by the video compression unit which outputs the compressed frame with embedded authentication watermark data.

Video Compression

A data-carrying signal may be compressed by removing redundancy from the signal. In a lossless compression system statistical redundancy is detached so that the original signal can be perfectly reconstructed at the receiver. The goal of a video compression algorithm is to achieve efficient compression with minimum alteration introduced by the compression process. Video compression algorithms operate by removing redundancy in the temporal, spatial and/or frequency domains.

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 produces a secure watermark sequence for each video frame using a meaningful primitive watermark structure and secret input keys. a primitive watermark pattern can be defined as a meaningful identifying sequence for each video frame. As shown in Fig. 4, the single 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 watermark generator is represented in Fig. 5. 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.

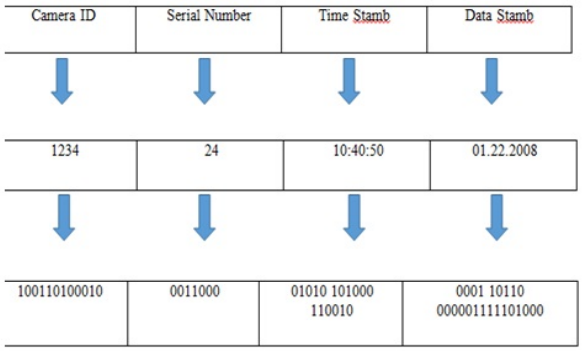

Figure-4. Structure of the primitive watermark.

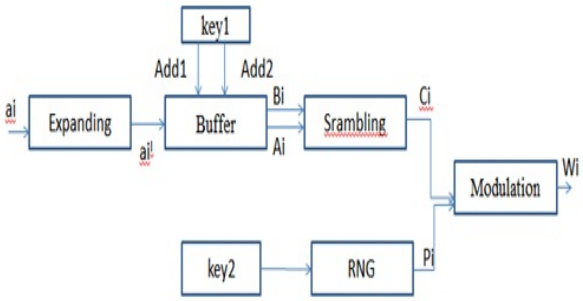

Figure-5. Block diagram of the proposed watermark generator.

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. 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 difficulty 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*.

Watermark Embedding

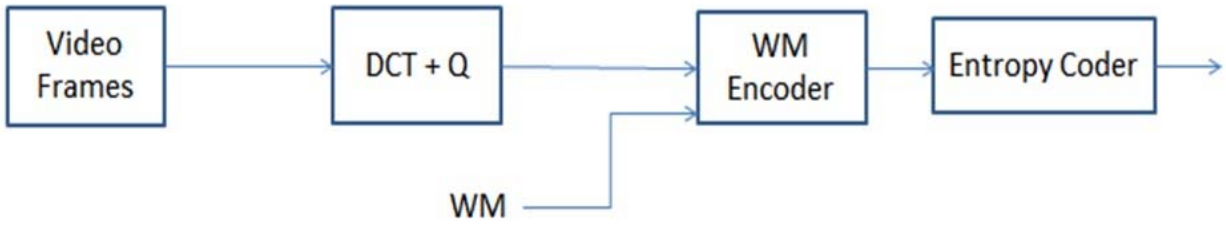

Figure-6. Data flow of the proposed WM.

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 macro blocks 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 Figure-6. This can be described as follows [14],[15].

Split I frame and watermark data into 8 × 8 blocks. For each 8×8 block (both watermark data and I frame), perform DCT, quantization, and zig–zag scan to generate quantized DCT coefficients.

Identify *N* watermarkable cells for each block and calculate the modification value for each selected cell.

Modify the identified watermarkable DCT coefficients according to the modification values.

Perform DCT, quantization, and zig–zag scan on the prediction error.

Perform entropy coding for the blocks of the different frames.

Generate compressed and watermark embedded video steam.

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

HARDWARE ARCHITECTURE

There exists a wide range of available techniques to implement the peripheral blocks of the proposed video WM system. Here, [16][17][18]the focus is on simplifying the process as much as possible, thus making it fit easily within existing video processing circuitry. At the same time, the security level and video frame quality are kept high. An overall view of the hardware implementation for the video WM system is depicted in Fig. 8. The proposed system architecture includes six modules: video camera, video compressor, watermark generator, watermark embedder, control unit, and memory. The parts implemented by the FPGA are shown in shaded blocks. The hardware implementation for the complete design is developed using Verilog HDL. As previously mentioned, all the processing in the implementation is assumed to be done on a block basis (such as 8 × 8 pixels). First, the captured video frame is temporarily stored in a memory buffer, and then each block of the frame data is continuously processed by the video compressor unit using DCT and quantization cores. The watermark embedder block inserts an identifying message, generated using the watermark generator unit, in the selected block data within the video frame and sends it to memory for storage. The control unit is responsible for driving the operations of the modules and the data flow in the whole system.

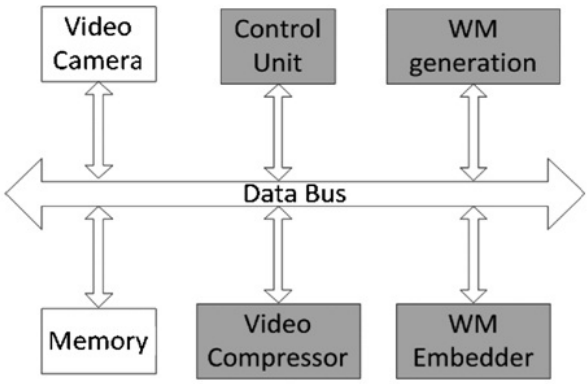

Figure-7. Block diagram of the hardware system architecture.

Watermark Embedding and Entropy coding

DCT coefficients stored in to buffer shown in Figure-8. The values taken from buffer and watermark data are combined and get 64 four bit outputs. After embedding is Entropy coding, it is a type of lossless coding to compress digital data by representing frequently occurring patterns with few bits and rarely occurring patterns with many bits. Huffman coding is a type of entropy coding. It includes three processes Zigzag scanning, run length encoding, huff man encoding. The zigzag scanning pattern for run-length coding of the quantized DCT coefficients was established in the original MPEG standard. The same pattern is used for luminance and for chrominance. A modified (alternate) pattern more suitable for coding of some interlaced picture blocks was added in the MPEG-2 standard. Run-length encoding (RLE) is a very simple form of data compression

in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. [19] Uncompressed, a character run of 15 A characters would normally require 15 bytes to store: AAAAAAAAAAAAAAA. The same string after RLE encoding would require only two bytes: 15A. In MPEG, Huffman coding in combination with Run-Level coding and zigzag scanning is applied to quantized DCT coefficients.

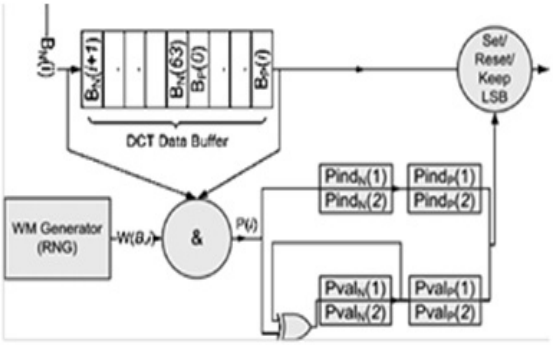

Figure-8. Hardware of watermark embedder.

"Run-Level" refers to a run-length of zeros followed by a non-zero level. Huffman coding is also applied to various types of side information.

A Huffman code is an entropy code that is optimum in the sense that it achieves the shortest average possible code word length for a source. This average code word length is >= the entropy of the source. After output is taken as bit streams.

Each module in the proposed digital video WM system, including the MJPEG compressor, watermark generator and watermark embedder has been implemented and tested individually, and then integrated together to obtain the final system architecture. The proposed architecture was first modeled in Verilog HDL and the functional simulation of the HDL design was performed using the Xilinx tool. Finally, the system design was synthesized to FPGA Spartan 3E.

EXPERIMENTAL RESULTS

Simulation Result

Performance Analysis

The performance of the overall FPGA implementation was evaluated in terms of hardware cost, power consumption, processing speed, and security issues.

1. Hardware Cost and Power Consumption: The hardware resources used by the different modules are given in Table I. The results clearly indicate that the addition of the watermark generator and embedder modules caused only 4.99 % increase in logic cells usage and 9.28% increase in memory resources consumed with relation to the hardware of the video compressor. The combined system would easily fit in the original FPGA device.

2. Processing Speed: The data of each frame are processed macro block (8×8) wise. Each macro-block first passes through the DCT module, and then through the WM embedder [20-22] and inverse DCT module, respectively. The WM generator takes 64 clock cycles to generate the WM data that are completed at the first 64 clock cycles of the frame processing period, and then WM data are stored in a WM buffer. Hence, the WM generator does not contribute to the initial latency of the frame processing period. The pipelining architecture and parallelism [23-25]of the designed system helped in achieving this high throughput after the initial latency state. Processing of P frame will require less time as they are not watermarked.

3. Security Issues: The encryption of the watermark is mainly dependent on the statistical property of the pseudorandom sequence generated by the random number generator (RNG) module, since the primitive watermark is modulated by the pseudorandom sequence.[26-27].

CONCLUSIONS

Design of the hardware architecture of a digital video watermarking system to authenticate video stream in real time was presented in this paper. FPGA-based prototyping for the hardware architecture was developed. The proposed system was suitable for implementation using an FPGA and can be used as a part of an ASIC. In the current implementation, FPGA was the simple and

available way of the proof-of concept. The implementation made integration to peripheral video (such as surveillance cameras) to achieve real-time image data protection. The aim of this paper was to achieve three objectives. First, to propose a new HW architecture of a digital watermarking system for video authentication and making it suitable for VLSI implementation. Second, to ensure that the watermarking algorithm achieves a certain level of security to withstand certain potential threats. Third, to make the watermarking system suitable for a real time video, which can be easily adapted with commonly used digital video compression standards with minor video frame degradation.

Contradictory to existing solutions, where robust WM algorithms were mainly used, a semi fragile WM system for video authentication was developed in this paper. The proposed watermark system was capable of watermarking video streams in the DCT domain in real time. It was also demonstrated that the designed system was capable of achieving the required security level with minor video frame quality degradation.

FUTURE RESEARCH

Future research should concentrate on applying the watermarking algorithm to other modern video compression standards, such as MPEG-4/H.264, so that it can be utilized in various commercial applications as well. Embedding the watermark information within high resolution video streams in real time is another challenge.

REFFERENCES

[1] V. M. Potdar, S. Han and E. Chang. 2005. “A survey of digital image watermarking techniques,” in Proc. IEEE Int. Conf. Ind. Informatics, Aug. pp. 709–716.

[2] A. D. Gwenael and J. L. Dugelay. 2003. “A guide tour of video watermarking,” Signal Process. Image Commun., Vol. 18, No. 4, pp. 263–282.

[3] A. Piva, F. Bartolini and M. Barni. 2002. “Managing copyright in open networks,” IEEE Trans. Internet Comput., Vol. 6, No. 3, pp. 18–26.

[4] Y. Shoshan, A. Fish, X. Li, G. A. Jullien and O. Yadid-Pecht. 2008. “VLSI watermark implementations and applications,” Int. J. Information Technol. Knowl., Vol. 2, No. 4 pp. 379–386.

[5] X. Li, Y. Shoshan, A. Fish, G. A. Jullien and O. Yadid-Pecht. 2008. “Hardware implementations of video watermarking,” in International Book Series on Information Science and Computing, no. 5. Sofia, Bulgaria: Inst. Inform. Theories Applicat. FOI ITHEA, pp. 9–16 (supplement to the Int. J. Inform. Technol. Knowledge, Vol. 2.

[6] K.E. Zhao J. 1994. Embedding robust labels into images for copyright protection, Technical Report, Fraunhofer Institute for Computer Graphics, Darmatadt, Germany.

[7] P. Bas, J.-M. Chassery and B. Macq. Image watermarking: an evolution to content based approaches Pattern Recognition, Vol. 35, pp. 545–561.

[8] L. D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes and G. Depovere. 2000. “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.

[9] Y. Shoshan, A. Fish, X. Li, G. A. Jullien and O. Yadid-Pecht. 2008. “VLSI watermark implementations and applications,” Int. J. Information Technol. Knowl., Vol. 2, No. 4 pp. 379–386.

[10] X. Li, Y. Shoshan, A. Fish, G. A. Jullien and O. Yadid-Pecht. 2008. “Hardware implementations of video watermarking,” in International Book Series on Information Science and Computing, no. 5. Sofia, Bulgaria: Inst. Inform. Theories Applicat. FOI ITHEA, pp. 9–16 (supplement to the Int. J. Inform. Technol. Knowledge, Vol. 2.

[11] K. Jack. 2001. Video Demystified: A Handbook for the Digital Engineer, 2nd ed. Eagle Rock, VA: LLH Technology Publishing.

[12] I. E. G. Richardson, H.264 and MPEG-4 Video Compression. Chichester, U.K.: Wiley, 2003.

[13] X. Li, Y. Shoshan, A. Fish and G. A. Jullien. 2008. “A simplified approach for designing secure random number generators in HW,” in Proc. IEEE Int. Conf. Electron. Circuits Syst., Aug. pp. 372–375.

[14] Y. Shoshan, A. Fish, G. A. Jullien and O. Yadid-Pecht. 2008. “Hardware implementation of a DCT watermark for CMOS image sensors,” in Proc. IEEE Int. Conf. Electron. Circuits Syst., Aug. pp. 368–371.

[15] D.W. Trainor J.P. Heron" and R.F. Woods,” Implementation of the 2D DCT using a XILINX XC6264 FPGA, “0-7803-3806-5/97.

[16] B. Pfitzmann. “Information Hiding Terminology”, Proc. of First Int. Workshop on Information Hiding, Cambridge, UK, May30-June1, 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.347-350.

[17] 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.

[18] N. J. Mathai, A. Sheikholesami, and D. Kundur, "VLSI implementation of a real-time video watermark embedder and detector," in Proc. Int. Symp. Circuits Syst., vol. 2. May 2003, pp. 772–775.

[19] T. H. Tsai and C. Y. Wu, "An implementation of configurable digital watermarking systems in MPEG video encoder," in Proc. Int.Conf. Consumer Electron., Jun. 2003, pp. 216–217.

[20] M. Maes, T. Kalker, J. P. Linnartz, J. Talstra, G. Depoyere, and J. Haitsma, "Digital watermarking for DVD video copy protection,"IEEE Signal Process. Mag., vol. 17, no. 5, pp. 47–57, Sep. 2000.

[21] Petitjean, J. L. Dugelay, S. Gabriele, C. Rey, and J. Nicolai,"Towards realtime video watermarking for systems-on-chip," in Proc. IEEE Int. Conf. Multimedia Expo, vol. 1. 2002, pp. 597–600.

[22] S. P. Mohanty and E. Kougianos, "Real-time perceptual watermarking architectures for video broadcasting," J. Syst. Softw., vol. 84, no. 5, pp. 724–738, May 2011.

[23] Sonjoy Deb Roy, Xin Li, Yonatan Shoshan, Alexander Fish, Member, IEEE, and Orly Yadid-Pecht, Fellow, IEEE Hardware implementation of a Digital Watermarking System for Video Authentication.

[24] Sanjana Sinha, Prajnat Bardhan, Swarnali Pramanick, Ankul Jagatramka, Dipak K. Kole, Aruna Chakraborty, "Digital Video Watermarking using Discrete Wavelet Transform and Principal Component Analysis" International Journal of Wisdom Based Computing, Vol. 1 (2), August 2011,

[25] Majid Masoumi1 , Shervin Amiri," Copyright Protection of Color Video Using Digital Watermarking", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 2, July 2012 ISSN (Online): 1694-0814.

[26] Paramjit Kaur M.Tech, Dr. Vijay Laxmi," An Upgraded Approach for Robust Video Watermarking Technique Using Stephens Algorithm", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.11, November- 2014, pg. 612-622.

[27] Nitin A. Shelke, Dr.P.N.Chatur," Blind Robust Digital Video Watermarking Scheme using Hybrid Based Approach", International Journal of Computer Science and Information Technologies, Vol. 5.