Pipeline Implementation of an Image Watermarking Using Bit-Plane Congestion Adaptivity

Mohsen Hajabdollahi
Dept. of Electrical and Computer Engineering IUT Isfahan, Iran
m.hajabdollahi@ec.iut.ac.ir

Shadrokh Samavi
Dept. of Electrical and Computer Engineering IUT Isfahan, Iran
samavi96@cc.iut.ac.ir

Nader Karimi
Dept. of Electrical and Computer Engineering IUT Isfahan, Iran
nader.karimi@cc.iut.ac.ir

Abstract— **In many applications of image watermarking, its real-time implementation is required. Although Hardware implementation is regarded as a proper types of real-time implementation; most algorithms are too complex for hardware implementation. In this paper an adaptive watermarking and its improved is presented which uses bit-plane congestion to achieve transparency and robustness. A congestion analysis is introduced and performed in such a way that guarantees robustness and simplicity in hardware application point of view. Also an efficient and novel hardware implementation of the adaptive watermarking algorithm based on pipeline is presented. Low complexity and small area in hardware implementation of proposed watermarking method, makes it suitable for real time applications.**

Keywords-component; ***Information Hiding; Watermarking; Hardware Implementation; FPGA***

I. INTRODUCTION

A popular application of watermarking is to give proof of ownership of digital data by embedding copyright statements, but it has application such as image authentication and fingerprinting for traitor tracking [1]. The fast expansion of internet and its popularity have increased the concern of multimedia copyright protection and ownership [2]. The implementation of watermarking could be performed on different platforms such as software, hardware, embedded controllers, DSP processors, etc. [3].

In Recent years, watermarking algorithms have been designed and implemented on hardware by many researchers [4]. Hardware implementation has the constraints of being low power, real-time, showing high reliability, low cost, and also should have ease of integration with existing consumer electronic devices. In [5] a circuit was designed which could embed data in images and expects to produce robust and fragile watermarking. A ternary watermark system is used for invisible embedding which adds scaled gray values corresponding to the intensity of pixels of a neighborhood. In [6] a VLSI design was presented capable of generating an invisible as well as fragile watermarking in the spatial domain. They used linear feedback shift register (LFSR) for the watermark generator. The disadvantage of the watermarking algorithms implementation in [5, 6] was that the processing needed to be done pixel by pixel.

An efficient architecture for transform domain watermarking using quantization approach was proposed in [7]. Their design used pipelining and it is attempted to optimize the performance of the pipeline. The main objective of the study in [7] was to propose very-large scale integration (VLSI) architecture for the robust and blind image watermarking chip. In [8] the advantages of the both the spatial and transform domains that were less computationally complex and robustness respectively were combined for the proposed algorithm. In [9] both watermarking and encryption techniques were implemented on a Field Programmable Gate Array in DCT domain.

In [10] a VLSI watermarking design in spatial domain was developed. A simple LSB replacement scheme was offered which was operated on 8-bit pixels and the design was done at the layout level. In [11] the development of very Large Scale Integration (VLSI) architecture for a high-performance watermarking chip which could perform invisible color image watermarking using genetic algorithm was discussed. This watermarking design is done in spatial domain which also implemented genetic algorithm for image adaptivity at the cost of high hardware complexity. It is necessary to note that all works performed in the transform domain had hardware complexity. In addition, watermarking in previous implementations in the spatial domain was not influenced by different image regions.

In this paper an adaptive watermarking method and it's enhanced in spatial domain are proposed which are compatible with efficient hardware implementation. A simple congestion analysis is introduced and simulation results show desired decision quality in different image regions. Also a novel pipeline implementation of this algorithm on FPGA is performed which leads to a good implementation results. Adaptive properties of proposed method cause image to be changed minimally in term of human visual quality. The rest of the paper is organized as follows: Section 2 describes details of the proposed adaptive watermarking algorithm and its enhancement. The proposed pipelined watermark embedding circuit is explained in Section 3. Experimental results are presented in Section 4, and Section 5 is dedicated to the conclusion of the paper.

II. PROPOSED ADAPTIVE WATERMARKING SCHEME

A. *Watermarking Method*

In this implementation all regions of image are divided into two classes: smooth and congested areas. In smooth areas, where the human visual system is more sensitive, the watermarking is done with low strength and in congested areas it is performed with higher strength. Therefore the human sees fewer changes in the smooth areas and more changes in congested areas. Block diagram of general watermarking system is illustrated in Fig. 1.

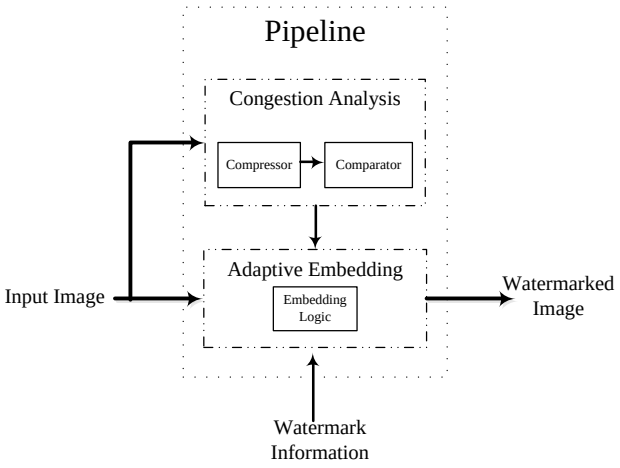


Figure. 1. General System Structure of the Proposed Method.

B. *Proposed Congestion Analysis by MSB*

To find the amount of congestion in each area, all bits in a bit-plane of pixels in an image block must be analyzed. In this paper a 3×3 mask is used to partition the image into non-overlap blocks. Congestion analysis is done based on the most significant bit-plane (MSB) of the pixels of each block. The amount of symmetry between the number of "ones" and "zeros" in a block is used to indicate the amount of disorder or congestion. According to the number of 'ones' in MSB bit-planes in each block, its congestion classifies the block as being smooth or congested as follows:

"If the number of 'ones' in MSB bit-plane in mask is a member of set $\{0,1,2,3,7,8,9\}$, then this block is of type T1."

"If the number of 'ones' in mask is a member of the set $\{4,5,6\}$, then the block is of type T2." This classification has low hardware complexity which makes the proposed method suitable for hardware implementation.

C. *Adaptive Embedding*

Suppose that P is the block under analysis and the sum of the MSBs in the 3×3 block is S. If $S \in \{0,1,2,3,7,8,9\}$, then block P is considered to be in a smooth region and if $S \in \{4,5,6\}$ then P is in a congested region. In T1 blocks, which are supposed to be smooth, watermarking is done in lower significant bit-planes. Also in T2 regions which are congested, watermarking is done to higher significant bit-planes. If the number of bit-planes varies from 1 to 8, then in smooth and congested regions, embedding is done in the 3rd and 5th bit-planes respectively. This embedding is according to the following rules: "If the block P is type T1 and the watermark bit is '0', then insert '0' in all positions of the 3rd bit-plane of the block". "If the watermark bit is '1' then insert '1' in all positions of the 3rd bit-plane of the block P". The same process is done when block is type T2, except the watermark bit is inserted in the 5th bit-plane of the block.

D. *Enhanced Adaptive Embedding*

In previous section an adaptive embedding was explained which 3rd and 5th bit-plane were selected for smooth and crowded areas respectively. Embedding process change the value of pixels and influence the visual quality of image. It is possible to reduce this effect by modifying embedding process. For this purpose a 3x3 image block with 8 bit representation is illustrated in Fig 2. Position of adaptive and enhanced embedding are illustrated by continuous arrow and dashed arrow respectively. In enhanced embedding process, in addition of main adaptive embedding, inverse bit of watermark is embedded in position depicted by dashed line. In this way the visual quality in term of PSNR and SSIM can be improved.

PSNR is used as a metric to quality measurement. In Fig. 3. all possible states of binary values in a position are showed and for these states embedding are performed. As illustrated in Fig 3, the quality of embedding is improved by enhanced adaptive watermark. To explain these results, it is necessary to note that in enhanced adaptive embedding, inserting watermark bits causes to less variation in each position. Although the sum of all variations are the same in two methods, but in enhanced embedding method, the variations are distributed more symmetric. Looking at the Mean Square Error (MSE) in formula (1) and PSNR in formula (2), represent that symmetric changes in pixel values causes less PSNR variations.

$$MSE = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}\left(P(n,m) - \tilde{P}(n,m)\right)^2 \quad (1)$$

$$PSNR = 10\ log_{10}\left(\frac{MAX(P)^2}{MSE}\right) \quad (2)$$

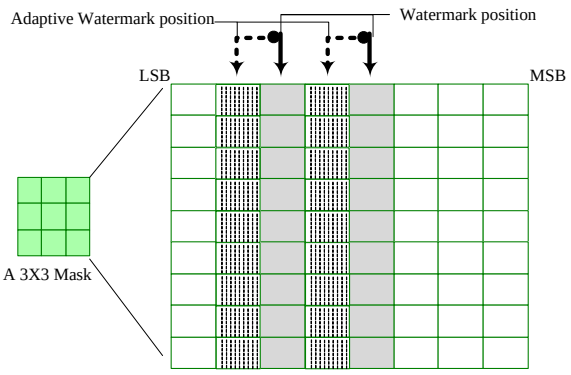


Figure. 2. Enhanced Adaptive Watermarking

All Possible States of Bit Planes	Embedding Bit	Adaptive Embedding	PSNR	Enhanced Adaptive Embedding	PSNR
Bit-Plane ith / Bit-Plane (i-1)th 0 0 0 1 1 0 1 1	'0'	0 0 0 1 0 0 0 1	45.1205	0 1 0 1 0 1 0 1	46.3699
	'1'	1 0 1 1 1 0 1 1	45.1205	1 0 1 0 1 0 1 0	46.3699

Figure. 3. All States Embedding in Enhanced Adaptive Watermarking

III. PROPOSED PIPELINED WATERMARK EMBEDDING IMPLEMENTATION

In this section the hardware implementation of watermark embedding process is presented and optimized. Efficient hardware modules for each part of embedding process is considered and discussed. Finally general pipelined of hardware architecture is presented. This pipeline has six stages as well as two main arithmetic blocks including congestion analyzer and embedding logic. These two blocks are explained as follows:

A. Congestion Analyzer

This module calculates the amount of congestion and categorizes a block into two types of smooth or congested. Two essential elements of this module are explained as follows:

1) 9-bit Adder Architecture

In Fig. 4 a 9-4 compressor is used for parallel addition of 9 bits in from 3×3 block using the design presented in [12]. In this architecture only 5 full adders and 2 half adders are used which make it proper for real-time applications.

2) Comparator

The comparator should decide whether the result of the adder is a member of the set {0, 1, 2, 3, 7, 8, 9}, which indicates a T1 block, or whether the addition result is a member of the set {4, 5, 6}, which indicates that block is type T2.

For this purpose we implemented a fast comparator based on the structure shown in Fig. 5. The output of this comparator indicates T2 type blocks. This comparator only uses 3 basic gates and is very suitable for hardware implementation. The output inverse of this comparator indicates that the block is type T1.

B. Embedding Logic

For implementation of the embedding process in hardware two multiplexers are used to select between two positions of embedding bits. The embedding implementation architecture in hardware is shown in Fig. 6. For simplicity in representation, this architecture has been shown for only one pixel. The lines for the multiplexer selectors come from the comparator which is used to determine two classes of congested and smooth. Two multiplexers put the watermark bit in the proper bit position based on the congestion type of the block.

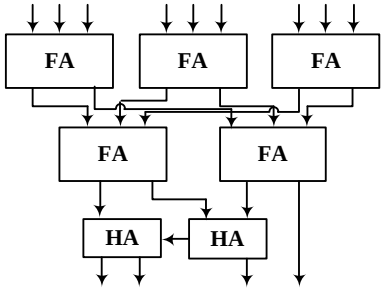


Figure. 4. Addition by a 9 to 4 compressor [12].

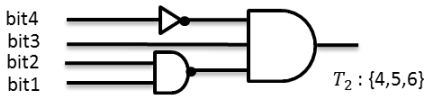


Fig. 5. Comparator for T2 detection

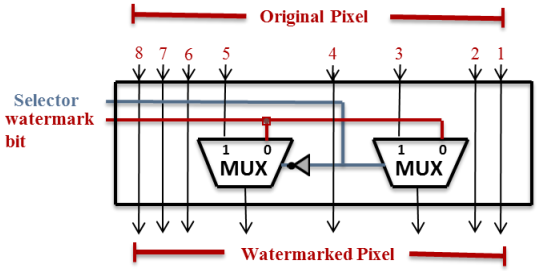


Fig. 6. Logic for adaptive in 3rd or 5th positions.

IV. EXPERIMENTAL RESULTS

A. MATLAB Implementation

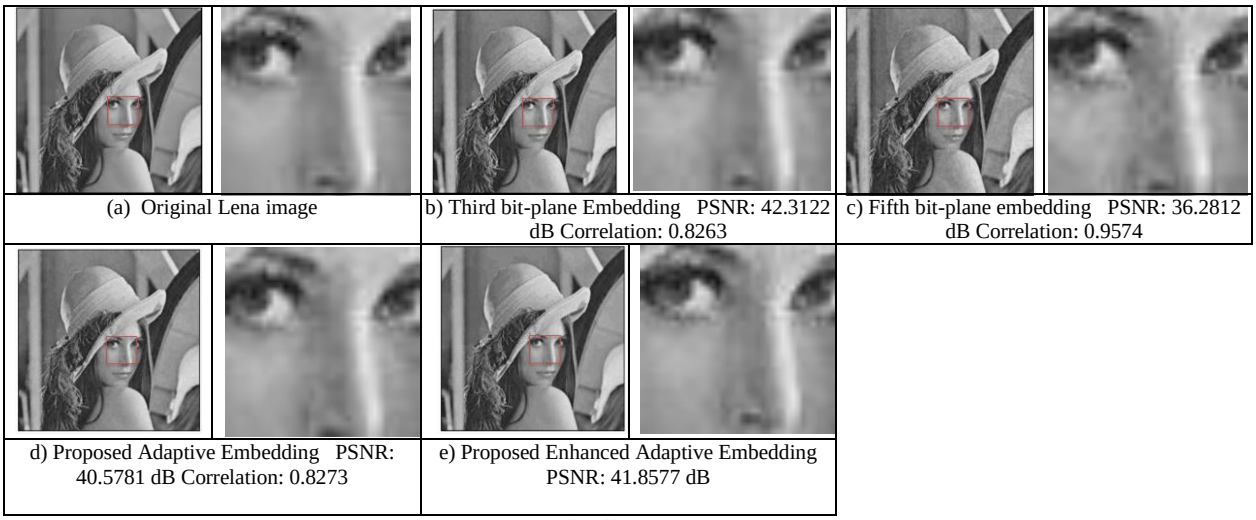
Adaptive watermarking algorithm is implemented in MATLAB 7.12 and its results are verified. In this paper JPEG attacks are tested for different images. Next the watermark extraction method is performed. Results for the correlation of the input watermark and the extracted watermark are illustrated in Table 1. In enhanced adaptive watermarking, created error is compensated such that watermarked image lead to a less variation from original image. In Fig. 7 the results for applying four methods of watermarking are shown. Watermarking in the 3rd bit-plane by the first method guaranties a good visual quality as shown in Fig. 7b, but it is much more sensitive to attacks. In other hand, watermarking in the 5th bit-plane leads to much higher robustness against attacks, but it damages the visual quality as shown in Fig. 7c. In the proposed adaptive method in the smooth and non-smooth areas watermarking is done in different bit-planes. We see that our watermark information is more robust than watermarking in the 3rd position. Also, our results are visually better than the results of the 5th bit-plane method as shown in Fig. 7d. Here it can be resulted that the desired parameters included visual quality and correlation is preserved in the proposed method. In Enhanced adaptive watermarking, by means of compensating error all results are improved in comparison with other methods.

B. Hardware Implementation

A Xilinx virtex4 family xc4vlx200 FPGA device is used for hardware implementation. Three internal RAMs for the input image, output image and the watermark message are used. In this implementation, pipelining is used to achieve high processing throughput. In each clock pulse one image line is read from the RAM. Then after three clock pulses the congestion analysis and the subsequent embedding processes are performed. The pipeline architecture is illustrated in Fig 8. For each image with the size of N×N, after initialization phase, in each clock pulse one row of image is read. Next pipeline is filled in six clock pulses and information is embedded in each clock pulse by watermarking hardware. Therefore N+6 clock pulse is needed for adaptive watermarking procedure. Fig. 9, shows ISIM simulation results where results after six clock pulses are provided and are written into the internal RAM. Implementation results for the FPGA design are reported and are summarized in Table 2.

Table 1. Robustness and transparency results for JPEG attacks

Watermark Method	Watermarking Quality Metric	Baboon	Lena	Barbara	Camera_man	Boat	Peppers
Bit-Plane 3	Correlation in JPEG Attack(Q=100)	1	1	1	1	1	1
	Correlation in JPEG Attack(Q=95)	0.5956	0.8263	0.7025	0.8895	0.7215	0.7011
	SSIM	0.9822	0.9638	0.9755	0.9559	0.9708	0.9640
	PSNR(db)	42.3344	42.3122	42.2033	42.0426	42.3486	42.4059
Bit-Plane 5	Correlation in JPEG Attack(Q=100)	1	1	1	1	1	1
	Correlation in JPEG Attack(Q=95)	0.9628	0.9574	0.9640	0.9513	0.9687	0.9554
	SSIM	0.9354	0.8881	0.9190	0.8665	0.9045	0.8869
	PSNR(db)	36.2985	36.2812	36.1883	36.1960	36.3078	36.3075
Adaptive	Correlation in JPEG Attack(Q=100)	1	1	1	1	1	1
	Correlation in JPEG Attack(Q=95)	0.7421	0.8273	0.7542	0.9002	0.7524	0.7201
	SSIM	0.9700	0.9576	0.9704	0.9484	0.9640	0.9602
	PSNR(db)	39.1798	40.5781	40.0598	40.0014	40.1453	40.8948
Enhanced Adaptive	Correlation in JPEG Attack(Q=100)	1	1	1	1	1	1
	Correlation in JPEG Attack(Q=95)	0.7904	0.8675	0.8031	0.9458	0.7823	0.7465
	SSIM	0.9752	0.9660	0.9760	0.9632	0.9711	0.9675
	PSNR(db)	40.4000	41.8577	41.2579	41.2864	41.3915	42.1254



(a) Original Lena image

b) Third bit-plane Embedding PSNR: 42.3122 dB Correlation: 0.8263

c) Fifth bit-plane embedding PSNR: 36.2812 dB Correlation: 0.9574

d) Proposed Adaptive Embedding PSNR: 40.5781 dB Correlation: 0.8273

e) Proposed Enhanced Adaptive Embedding PSNR: 41.8577 dB

Figure 7. Comparison of Different Embedding Methods.

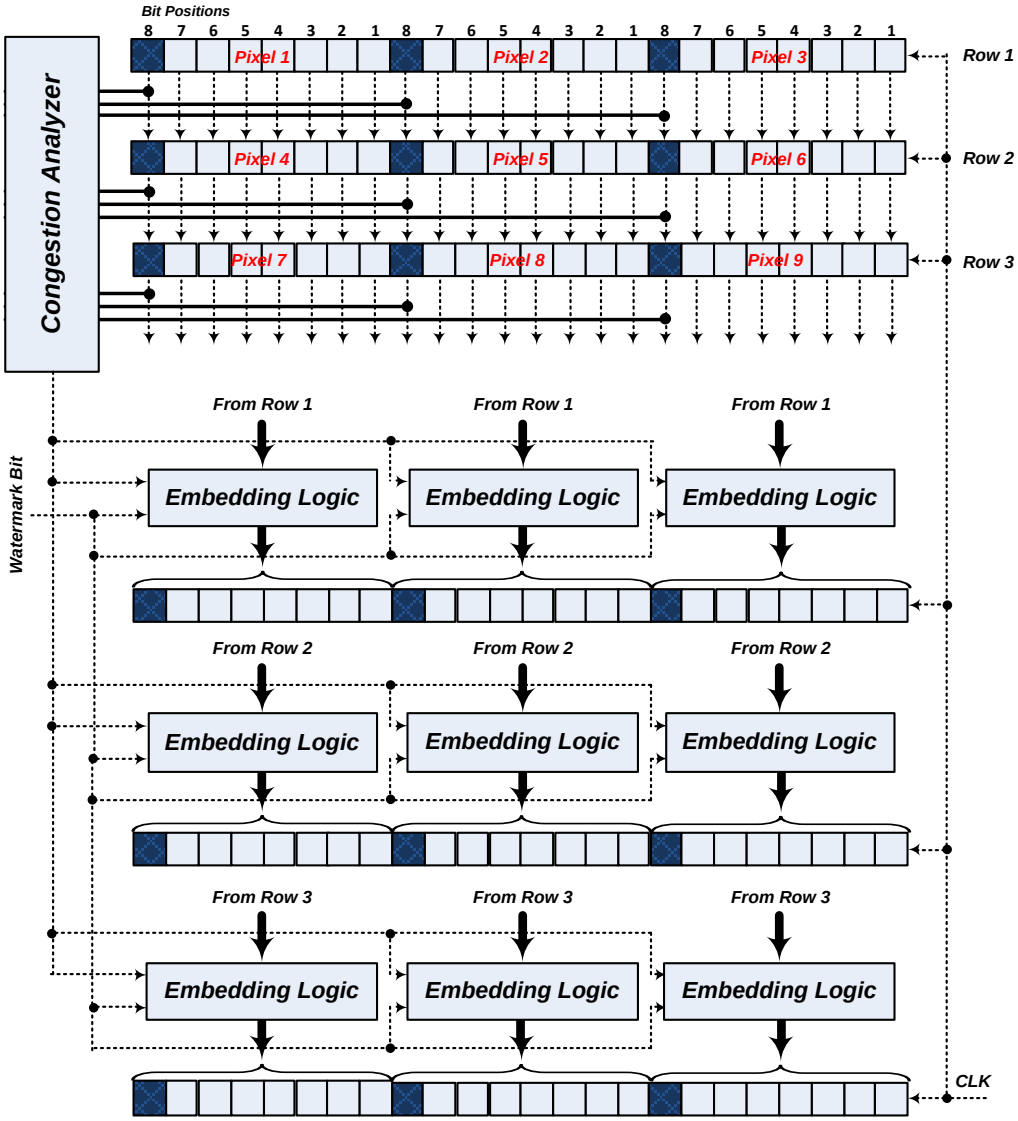


Figure. 8. Proposed Pipeline Design for Congestion analysis and Watermark Embedding

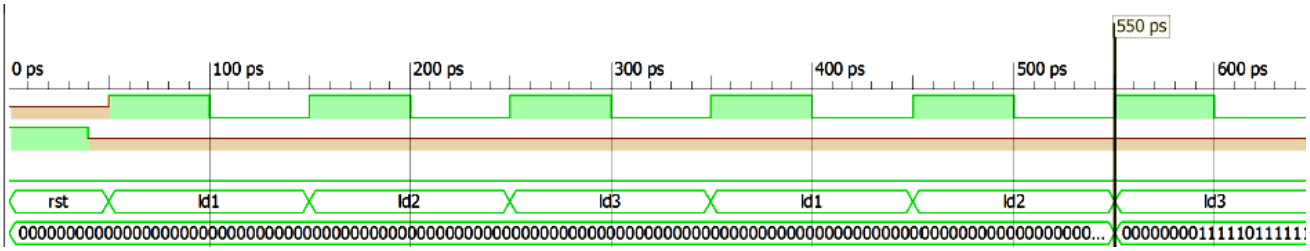


Figure. 9. Hardware Simulation Results

Table 2. Implementation Specification summary for Xilinx virtex4 Family xc4vlx200 device.

Total Area (ratio)	Maximum frequency	# of 4input LUTs	# of Slice Flip Flops	# of Slices
%8	172.873MHz (5.785ns)	11019 (%6)	12191 (%6)	7888 (%8)

V. CONCLUSION

Hardware architecture of an adaptive image watermarking system to authenticate digital images was presented in the real time space. First a simple and efficient adaptive image watermarking method, suitable for hardware implementation, was proposed. Next an enhancement design of adaptive watermarking was introduced. Efficient implementation of the proposed adaptive embedding algorithm was designed. Finally a pipeline method was used to speed up watermarking algorithm. Design reports from FPGA implementation showed, watermarking algorithm could be used in real word with a good performance. Only 8% of the total area of the FPGA was used for this efficient hardware design. Although Implementation of watermarking algorithms in the spatial domain had a low complexity in hardware but their robustness had been challenging problem. Our enhanced adaptive watermarking tried to solve this problem by detection of regions in which watermark information was preserved against attacks. Also an error compensation method to increase visual quality and robustness leads to better results. Proposed error compensation can be extended to more number of bit-planes to obtain better robustness against attacks and distortions.

REFRENCES

[1] S. Katzenbeisser and F. Petitolas, “Information hiding techniques for steganography and digital watermaking,” DNA, vol. 28, 2004.

[2] J. Cox, M. L. Miller, J. A. Bloom, and C. Honsinger, Digital watermarking vol. 53: Springer, 2002.

[3] Joshi, Amit, Vivekanand Mishra, and R. M. Patrikar. “Real time implementation of digital watermarking algorithm for image and video application.”.

[4] E. Kougianos, S. P. Mohanty, and R. N. Mahapatra, “Hardware assisted watermarking for multimedia,” Computers & Electrical Engineering, vol. 35, pp. 339-358, 2009.

[5] S. P. Mohanty, N. Ranganathan, and R. K. Namballa, “VLSI implementation of invisible digital watermarking algorithms towards the development of a secure JPEG encoder,” in Signal Processing Systems, 2003. SIPS 2003. IEEE Workshop on, 2003, pp. 183-188.

[6] S. P. Mohanty, E. Kougianos, and N. Ranganathan, “VLSI architecture, and chip for combined invisible robust and fragile watermarking,” Computers & Digital Techniques, IET, vol. 1, pp. 600-611, 2007.

[7] A. Darji, T. Lad, S. Merchant, and A. Chandorkar, “Watermarking hardware based on wavelet coefficients quantization method,” Circuits, Systems, and Signal Processing, vol. 32, pp. 2559-2579, 2013.

[8] A. Joshi, A. Darji, and V. Mishra, “Design and implementation of real-time image watermarking,” in Signal Processing, Communications and Computing (ICSPCC), IEEE International Conference on, 2011, pp.1-5.

[9] R. M. Khoshki, S. Oweis, S. Wang, G. Pappas, and S. Ganesan, “FPGA hardware based implementation of an image watermarking system.”

[10] A. Garimella, M. Satyanarayana, R. Satish Kumar, P. Murugesh, and U. Niranjan, “VLSI implementation of online digital watermarking technique with difference encoding for 8-bit gray scale images,” in Proceedings of the 16th International Conference on VLSI Design, 2003, pp. 283-288.

[11] S.Sreejith,N. Mohankumar, and M. N. Devi, “Hardware implementation of genetic algorithm based digital colour image watermarking,” Aceee International Journal on signal & Image processing, vol. 2, 2011..

[12] R. Marimuthu, D. Bansal, S. Balamurugan, and P. Mallick, “Design of 8-4 and 9-4 compressors for high speed multiplication,” American Journal of Applied Sciences, vol. 10, p. 893, 2013