

Implementation of a Visible Watermarking in a Secure Still Digital Camera Using VLSI Design

M.Mohamed Ismai Majeed¹, S.C.Ramesh² and R.Anuja³⁺

^{1 2 3} Department of Electrical & Electronics Engg.,
PSN College of Engineering and Technology, Tirunelveli, Tamilnadu, India

Abstract. Watermarking is the process that embeds data called a watermark, a tag, or a label into a multimedia object, such as images, video, or text, for their copyright protection. According to human perception, the digital watermarks can either be visible or invisible. A visible watermark is a secondary translucent image overlaid into the primary image and appears visible to a viewer on a careful inspection. The invisible watermark is embedded in such a way that the modifications made to the pixel value are perceptually not noticed, and it can be recovered only with an appropriate decoding mechanism. This paper presents new very large scale integration (VLSI) architecture for implementing two visible digital image-watermarking schemes. The proposed architecture is designed to aim at easy integration into any existing digital camera framework.

Keywords: Digital watermarking, JPEG encoder, MATLAB, spatial domain watermarking, still digital camera, visible and invisible watermarking

1. Introduction

Watermarking is the process that embeds data called a watermark, a tag, or label into a multimedia object such that the watermark can be detected or extracted later to make an assertion about the object. The object may be an image, audio, video, or text. Whether the host data is in spatial domain, discrete cosine-transformed, or wavelet-transformed, watermarks of varying degree of visibility are added to present media as a guarantee of authenticity, ownership, source, and copyright protection. In general, any watermarking scheme (algorithm) consists of three parts, such as Watermark, Encoder and Decoder and Comparator.

Watermarks and watermarking techniques can be divided into various categories. The watermarks can be applied either in spatial domain or in frequency domain. It has been pointed out that the frequency-domain methods are more robust than the spatial-domain techniques. On the other hand, the spatial domain watermarking schemes have less computational overhead compared with frequency-domain schemes. According to human perception, the digital watermarks can be divided into four categories: (1) Visible, (2) Invisible-robust, (3) Invisible-fragile, (4) Dual

Whether each owner has a unique watermark or an owner wants to use different watermarks in different objects, the marking algorithm incorporates the watermark into the object. The verification algorithm authenticates the object determining both the owner and the integrity of the object.

A visible watermark is a secondary translucent image overlaid into the primary image and appears visible to a casual viewer on careful inspection. The invisible-robust watermark is embedded in such a way that modifications made to the pixel value are perceptually not noticed, and it can be recovered only with appropriate decoding mechanism. The invisible-fragile watermark is embedded in such a way that any

⁺ Corresponding author.

E-mail address: (mmimajeed@gmail.com,ramesh.psnct@gmail.com,anu_lalli@yahoo.co.in).

manipulation or modification of the image would alter or destroy the watermark. A dual watermark is a combination of a visible and an invisible watermark. In this type of watermark, an invisible watermark is used as a back up for the visible watermark.

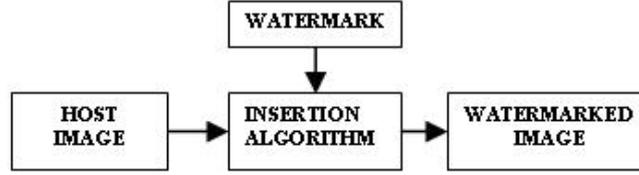


Fig 1 Watermark Insertion Procedure

The proposed watermarking chip can be easily incorporated as a module in any existing JPEG encoder, and a secured JPEG encoder can be developed. An outline of such a secure JPEG encoder is provided in Fig.2. The secure JPEG codec can be a part of a scanner or a digital camera so that the digitized images are watermarked right at the origin. The proposed watermarking chip can also be directly integrated with any existing digital still camera. We provide the schematic view of a still camera that includes a watermarking module in Fig.3. The SDC (secure digital still camera) is conceptually similar to the “trustworthy digital camera” proposed by Friedman [8], in which cryptography is used for image authentication.

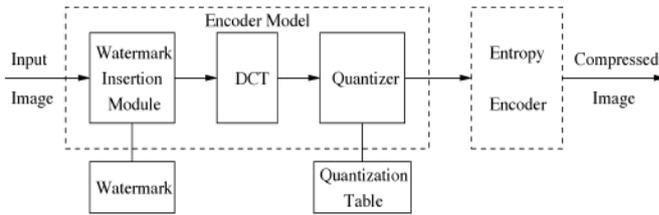


Fig. 2 Block-level view of a secure JPEG encoder

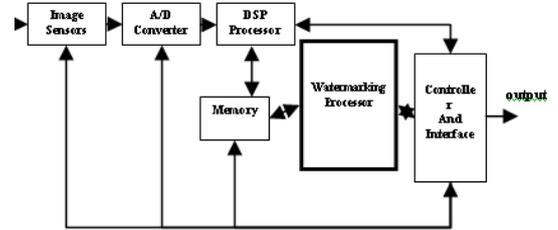


Fig. 3 System architecture of a secure digital still camera

2. Watermarking Algorithm

In this section, we discuss the image watermarking algorithm in brief and then discuss the modifications necessary to facilitate hardware implementation. The modifications are aimed at reducing silicon area through module sharing.

- Visible Watermarking Algorithm

In general, visible watermarking has three goals such as (i) Visible watermark should identify the ownership (ii) Visual quality of the host image should be preserved (iii) Watermark should be difficult to remove from the host image. To satisfy these three conflicting criteria, schemes have been proposed for adding a watermark with the original image. The watermarked image is obtained by adding a gray-scaled value of the watermark image to the host image. The amount of scaling is done in such a way that the alteration of each original image pixel occurs to a perceptual equal degree. The original formulas have been simplified as shown subsequently, where the scaling factor determines the strength of watermark.

$$I_w(m,n) = \begin{cases} I(m,n) + W(m,n) \left(\frac{I_{white}}{38.667} \right) \left(\frac{I(m,n)}{I_{white}} \right)^{2/3} \alpha_1, & \text{for } \frac{I(m,n)}{I_{white}} > 0.008856. \\ I(m,n) + W(m,n) \left(\frac{I(m,n)}{903.3} \right) \alpha_1, & \text{for } \frac{I(m,n)}{I_{white}} \leq 0.008856. \end{cases} \quad \text{---(1)}$$

The above equation can be simplified to make it amenable for hardware implementation. At the same time, it is ensured that the computation in hardware yields results that are as accurate as the software implementation. We assume and $I_{white} = 255$ simplify the above equations to the following:

$$I_w(m,n) = \begin{cases} I(m,n) + \left(\frac{\alpha_1}{6.0976} \right) W(m,n) (I(m,n))^{2/3}, & \text{for } I(m,n) > 2.2583. \\ I(m,n) + \left(\frac{\alpha_1}{903.3} \right) W(m,n) I(m,n), & \text{for } I(m,n) \leq 2.2583. \end{cases} \quad \text{---(2)}$$

The above expression involves cubic root calculation, which is complex to implement in hardware. Therefore, we further simplify the above expressions and remove the cubic root function with a piecewise

linear model. We divide the gray values range to four ranges, and we fit four linear regression coefficients that best approximate the cubic root in each of these ranges. Moreover, we round up the fraction involved in the comparison operation, and the final expression that is implemented using hardware is as follows:

$$I_w(m,n) = \begin{cases} I(m,n) + \left(\frac{\alpha_I}{6.0976}\right)W(m,n)I(m,n), & \text{for } I(m,n) \leq 2. \\ I(m,n) + \left(\frac{\alpha_I C_1}{6.0976}\right)W(m,n)I(m,n), & \text{for } 2 < I(m,n) \leq 64 \\ I(m,n) + \left(\frac{\alpha_I C_2}{6.0976}\right)W(m,n)I(m,n), & \text{for } 64 < I(m,n) \leq 128 \\ I(m,n) + \left(\frac{\alpha_I C_3}{6.0976}\right)W(m,n)I(m,n), & \text{for } 128 < I(m,n) \leq 192 \\ I(m,n) + \left(\frac{\alpha_I C_4}{6.0976}\right)W(m,n)I(m,n), & \text{for } 192 < I(m,n) \leq 256. \end{cases} \quad \text{---(3)}$$

We performed extensive software simulations for various test images and found that the pixel values of the watermarked images obtained using the above set of equations match with that obtained using original equation with the help of MATLAB.

3. Vlsi Architecture and Implementation of the Chip

In this section, we discuss the VLSI architectures for the watermarking algorithm. We assume that both the original image and the watermark image are stored in the memory within the digital camera framework and are available for processing. The images may be in either a compressed format or as raw ASCII data. We need to have a corresponding decoder to decode the image and get the uncompressed data in case it is in compressed format.

The insertion operation for the watermarking algorithm is described in [1]. This insertion function is simplified to using a piecewise linear model such that we have a compact and efficient hardware design. Fig. 4 shows the architecture proposed for the algorithm. The watermarking in this scheme is performed pixel by pixel as evident from the insertion function. A register file is used to store the constants needed to scale the image–watermark product and we store the constants.

The comparator is used to determine the range in which a particular pixel gray value lies such that an appropriate constant can be picked up from the register file. The left-hand-side multiplier calculates appropriate constant times the host image pixel gray values, and the right-hand-side multiplier is used to find times the watermark image pixel gray value.

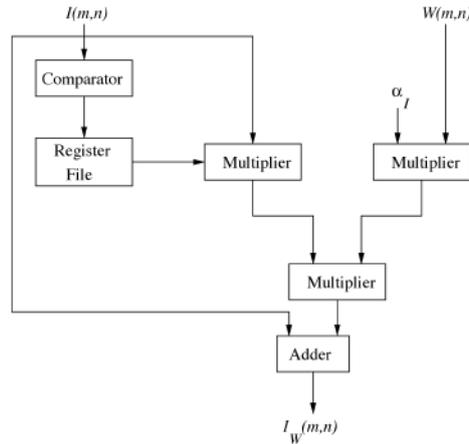


Fig. 4 Data path architectures for the visible watermarking algorithm

The results of the above two multipliers are fed to the third multiplier, which effectively calculates the product of constants, the host image pixel gray value, and the watermark image pixel gray value. The above product is added to the host image pixel gray values using the adder to obtain the watermarked image pixel gray values. This process has to be carried out for all the pixels in order to obtain the watermarked image

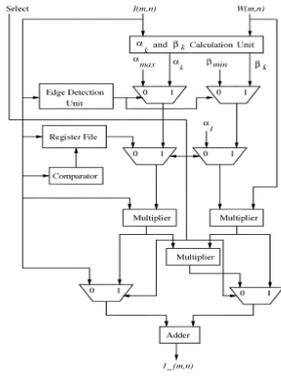


Fig 5 (a) Merged data path Architecture for the proposed watermarking processor

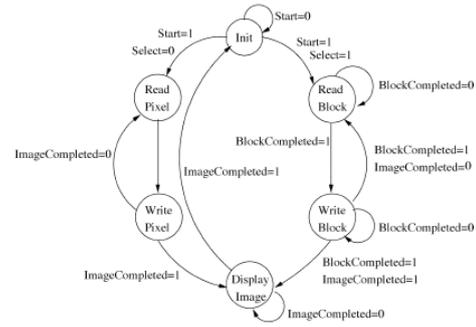


Fig. 5 (b) Architecture of the controller for the merged data path for the proposed watermarking processor

- Architecture for the Watermarking Processor

The data paths for the algorithm shown in Fig. 4 stitched together using multipliers and the combined data path is shown in Fig. 5(a). The controller that drives the data path is shown in Fig. 5(b). The controller has six states, such as Init, Read Block, Write-Block, Read Pixel, Write Pixel, and Display Image. When the Start signal is “1”, the watermarking process is initiated.

The data path construction involves the implementation of the proposed architecture in the previous section. The fundamental functional units are 8-bit adders, 8-bit multipliers, and 8-bit adder–subtractor. Each adder is constructed using 1-bit adders in a ripple-carry manner. The adder–subtractor unit is obtained from the adder using XOR gates. The carry inputs to the adder–subtractor and one of the inputs to the XOR gate are set to high whenever the select signal for this unit is “2” so that a subtraction is carried out. The output of the adder–subtractor module gives the absolute value of the difference of two numbers when the difference is positive. When the difference is less than 0 (which is indicated by the carry bit taking a value 0), the absolute value is obtained by taking the 2’s complement of the output of the adder–subtractor module.

The comparator was designed to compare the values of two 8-bit numbers for greater than, equal to, or less than relations. First, a single-bit comparator was designed to compare the values of two single-bit numbers, and later, instances of this module were cascaded to compare two 8-bit numbers, starting from the most significant bit position and proceeding toward the least significant bit position

Area	3.34 X 2.89 mm ²
Number of gates	28469
Clock frequency	292.27 MHz
Number of I/O ports	72
Power	6.9286 mW

Table 1 overall statistics of the watermarking chip

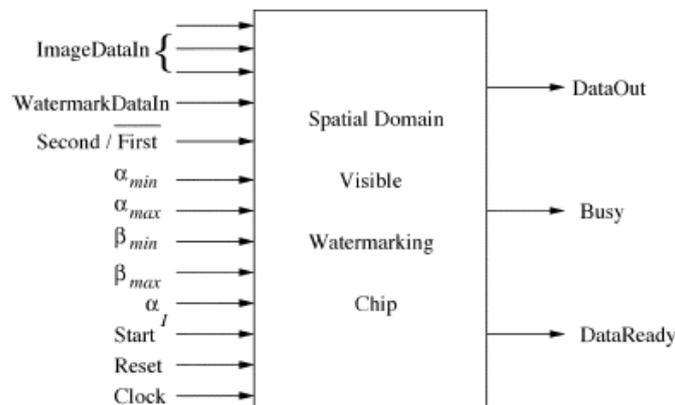


Fig. 6 Pin diagram for the proposed watermarking chip

4. Experimental Results

Each of the functional units is simulated individually before being integrated together to develop the whole chip. Performing watermarking on various test images does the functional verification of the whole chip. Fig. 7 shows test image and Fig. 8 shows the watermark image used. The test image as well as the watermark image is of 256 X 256 dimensions. The watermarked image obtained using the algorithm is shown in Fig. 9.

To verify whether the proposed chip produces results as effective as the software implementations, we have conducted several tests. The algorithms we have chosen for our implementation are well-accepted algorithms and are proven to be satisfying the vis-à-vis goals of the watermarking scheme. Thus, as long as the pixel values of a watermarked image from the hardware implementation matches with the pixel values of the same watermarked image obtained using software implements, we prove that hardware implementation do match with software implementations in satisfying the goals.



Fig. 7 Original host image



Fig. 8 Watermark image



Fig 9 Watermarked image

5. Conclusion

In this paper, we presented a watermarking chip that can be integrated within a digital camera framework for watermarking images. The watermarking chip can also be integrated in any existing JPEG encoder. The chip has two different types of watermarking capabilities, in spatial domain. This algorithm does pixel-by-pixel processing, and comparable in terms of signal-to-noise ratio (SNR) values. The design can be improved by a data path organization in which the blocks can be pipelined to obtain better throughput.

6. References

- [1] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA: Artech House, 2000.
- [2] S. P. Mohanty, "Watermarking of digital images," M.S. thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India, 1999.
- [3] N. Memon and P. W. Wong, "Protecting digital media content," *Commun. ACM*, vol. 41, no. 7, pp. 34–43, Jul. 1998.
- [4] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.
- [5] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A dual watermarking technique for images," in *Proc. 7th ACM Int. Multimedia Conf.*, vol. 2, 1999, pp. 49–51.
- [6] S. P. Mohanty, N. Ranganathan, and R. K. Namballa, "VLSI implementation of in digital watermarking algorithms toward the development of a secure JPEG encoder," in *Proc. IEEE Workshop Signal Processing Systems*, 2003, pp. 183–188.
- [7] G.W. Braudaway, K. A. Magerlein, and F. Mintzer, "Protecting publicly available images with a visible image watermark," in *Proc. SPIE Conf. Optical Security Counterfeit Deterrence Technique (Vol. SPIE-2659)*, 1996, pp. 126–132.
- [8] G. L. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Trans. Consumer Electronics.*, vol. 39, no. 4, pp. 905–910, Nov. 1993.