VLSI WATERMARK IMPLEMENTATIONS AND APPLICATIONS

Yonatan Shoshan, Alexander Fish, Xin Li, Graham Jullien, Orly Yadid-Pecht

Abstract: This paper presents an up to date review of digital watermarking (WM) from a VLSI designer point of view. The reader is introduced to basic principles and terms in the field of image watermarking. It goes through a brief survey on WM theory, laying out common classification criterions and discussing important design considerations and trade-offs. Elementary WM properties such as robustness, computational complexity and their influence on image quality are discussed. Common attacks and testing benchmarks are also briefly mentioned. It is shown that WM design must take the intended application into account. The difference between software and hardware implementations is explained through the introduction of a general scheme of a WM system and two examples from previous works. A versatile methodology to aid in a reliable and modular design process is suggested. Relating to mixed-signal VLSI design and testing, the proposed methodology allows an efficient development of a CMOS image sensor with WM capabilities.

Keywords: Watermark, CMOS sensors, image sensors, VLSI, mixed-signal circuits, fast prototyping.

ACM Classification Keywords:

1. Introduction

The field of digital imaging and its subsidiaries has been going through a continuous and rapid growth during the last decade. Research activity has been extensive in both the academic and commercial communities, and significant advances and breakthroughs are being constantly published [1]. Cost reductions and miniaturization enabled by major developments in VLSI fabrication technologies, CMOS in particular, are making high quality digital imaging products widely accessible, thus effectively taking traditional analog imaging out of the picture. Digital imaging has become a standard in almost all imaging applications, from professional photography and broadcasting to the everyday consumer digital camera. The ease of integrating CMOS imagers with supporting peripheral elements together with a significant reduction in power consumption introduced a variety of new portable products such as imagers on cell phones and narrow-band web cameras [2].

Since digital images are very susceptible to manipulations and alterations, a variety of security problems are introduced. For example, a security centre may wish to authenticate the data received from sensors spread across a facility it is supposed to protect. Another common application is resolving ownership disputes when copyrighted material is distributed illegally. Those problems and needs can be treated by embedding a secret, invisible watermark (WM) in images. A WM is an additional, identifying message, covered under the more significant image raw data, without perceptually changing it. By adding a transparent WM to the image, it can be made possible to detect alterations inflicted upon the image, such as cropping, scaling, covering, blurring and many more.

The WM can be added on either a software platform or a hardware platform, each having some benefits and some drawbacks. Although WM implementation on a hardware platform suffers from limited processing power, compared to the software implementation, it features real time capabilities and compact implementations. The advantages of hardware WM implementations are especially enhanced in CMOS imagers, where it is possible to integrate the WM embedder monolithically with the sensor array on the same die.

Many WM implementations both in software and in hardware have been proposed in the literature [3]-[11]. In 1990 the modern study of steganography and digital WM was started by Tanaka et al. [10]. They suggested hiding information in multi-level dithered images as a form of secured military communications. Following that work, digital WM arose, and the development of WM algorithms became a growing field of research. Some of the proposed algorithms were relatively simple and weak, merely substituting image least significant bits with WM data [12],[13]. Others had a similar approach but selectively chose the pixels that were to be modulated – either by a random choice to enhance security or according to image quality considerations such as the variance of luminosity. In [14], the WM was embedded in the coefficients of the discrete cosine transform (DCT) of the image to allow better robustness against JPEG lossy compression. Later algorithms modulated only middle-band DCT coefficients [15] to avoid image quality degradation while maintaining a high level of robustness. During the

second decade of digital WM research, much thought has been given to the methods in which the WM is implemented and some hardware specific algorithms have been presented [16],[17]. These implementations are usually optimized versions of the former software implementations as will be shown in later sections.

This paper aims to achieve two main objectives. First, the reader is introduced to basic principles and terms in the field of image WM. The paper presents different classification criterions and elementary WM properties such as robustness, computational complexity and influence on image quality. The second goal is to discuss a versatile methodology to design and test hardware implemented WM algorithms, integrated with an image sensor. The proposed methodology speeds up the development process while enhancing reliability.

Section 2 reviews the theory of WM algorithms. Watermark implementations in software and hardware are presented in Section 3. Hardware implementation development methodologies are discussed in Section 4. Section 5 concludes the paper.

2. Theory and implementation of watermark algorithms

2.1 Watermarks Classification

Different applications require utilization of WM with different properties, and no universal WM algorithm that can satisfy the requirements for all kinds of applications has been presented in the literature. WM can be classified into different categories according to various criterions. Figure 1 shows general classification of existing WM algorithms. First of all, all WM can be divided into two main categories: *visible* and *invisible*. The invisibility of a WM is determined by how it affects the image perceptually. Sometimes a WM is intentionally visible, in which case, the identifying image is embedded into the original one and both are visually noticeable. Figure 2 shows examples of the original image and the image with embedded visible WM. Generally, most WM algorithms aim for the WM to be as invisible as possible. Invisible WM has the considerable advantage of not degrading the host data and not reducing its commercial value. For that reason a lot of research has been carried out in this field, while visible WM has received substantially less attention.

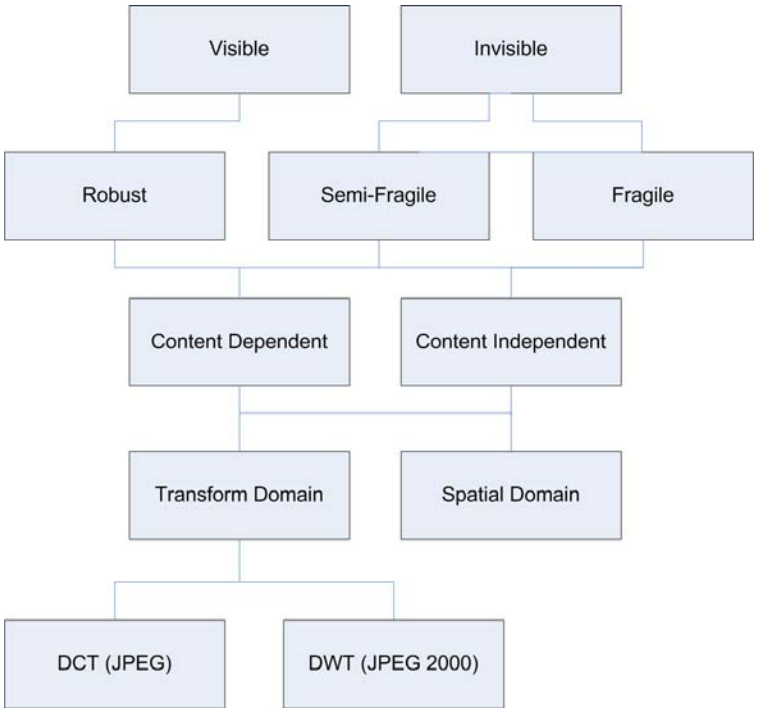

Figure 1. General classification of existing watermark algorithms

WM can also be classified according to the level of robustness to image changes and alterations. Three main categories of WM can be identified: *Fragile*, Semi-fragile and *Robust*, though no standard definition exists to explicitly determine which is which. Different applications will have different requirements, while one would need the algorithm to be robust as possible the other may be designed to detect even the slightest modification made to an image - such a WM is called fragile. A fragile WM is practical when a user wishes to directly authenticate that the image he is observing is exactly the same as it was when the WM was first embedded. This might be the case in applications where raw data is used. However, in most existing applications such modifications as lossy compression and mild geometric changes are inherently performed to the image. For those applications it is most efficient to use a semi-fragile algorithm which is designed to withstand certain legitimate modifications, but to detect malicious ones. Finally, some applications, such as copyright protection, require that the WM would be detectable even after an image goes through severe modifications and degradation, including digital-to-analog

and analog-to-digital conversions, cropping, scaling, segment removal and all sorts of attacks. A WM that answers these requirements would be called robust.

(a) (b)

Figure 2. Examples of (a) the original image and (b) the image with embedded visible watermark

Whether or not the algorithm is content dependent is another important distinction. Making the algorithm depend upon the content of the image, is good against counterfeiting attacks, however it complicates the algorithm implementation and therefore the embedding and extracting processes.

An additional classification relates to the domain in which the WM is performed. The most straight forward and simple approach is a WM implementation in the spatial domain that relates to applying the WM to the original image, for example by replacing the least significant bit (LSB) plane with an encoded one [12],[13]. Two other common representations are the discrete cosine (DCT) and the discrete wavelet transforms (DWT) [18],[19] in which the image first goes through a certain transformation, the WM is embedded in the transform domain and then it is inversely transformed to receive the watermarked image.

2.2 Watermark Design Considerations

In order to discuss WM design considerations a number of WM properties should be introduced: (1) *Capacity* (the term is adopted from the communications systems field [21]): in a watermarking system the cover image can be thought of as a channel used to deliver the identifying data (the watermark). The capacity of the system is defined as the amount of identifying data contained in the cover image, (2) *False detection ratio:* this ratio is characterized according to the probability of issuing the wrong decision. It is comprised of the probability to falsely detect an unauthentic WM (false positive), and the probability to miss a legitimate one (false negative). It is possible to manipulate the detection algorithm in order to minimize one or the other, according to the application. The value of this ratio is usually determined experimentally, (3) *Image quality degradation*: the embedding of foreign contents in the image has a degrading effect on image quality. That parameter is relatively hard to quantize and different measures such as peak signal-to-noise ratio (PSNR) or a subjective human perception measure may be applied.

These properties are elementary in every WM system and need to be carefully appreciated. The following subsections show how they are considered from different design point of views and indicate several trade-offs between them.

2.2.1 Robustness to Attacks

A good attack on fragile and semi-fragile WM will attempt to modify the perceptual content of the image, without affecting the WM data embedded in it. Knowledge of the embedding and extracting methods is assumed. There are two approaches for an attack; while the first approach requires the decryption of the encoded mark in order to produce a suitable WM on an unauthentic image, the second one aims to maintain the original mark on a modified image without knowing the mark itself. Decrypting the original WM is a cryptographic computational problem and is directly related to the capacity of the WM system. In WM however, the potential for such an attack is even greater (compared with the cryptographic case) as the attacker does not have to find the exact key, but

only one that would be close enough to pass over the detector's threshold. And still, if the capacity of the WM is large enough - using a key of several hundred bits, this attack may not be computationally tractable.

There are numerous attacks that take advantage of the existing image to create a forged one. The most intuitive one is the cover up attack [17]. This attack can be used when the mark is embedded independently to a block divided image. If the image contains homogeneous areas such as a wall, or a floor and the attacker wishes to hide a smaller object, he may do so by copying other blocks in such a way that the change would be perceptually un-noticeable, but the detector would still recognize a valid WM on the copied block. A possible counter measure for such an attack is to complicate the scheme, while increasing system complexity, and create dependencies between the marks of neighboring blocks – if a block is placed in the wrong place, detection would be false.

Attacks on copyright protection WM are designed to cause defects to the embedded WM so that it will be undetectable, while still maintaining reasonable image quality. Such attacks may include one or more of the following: (1) A geometric attack such as cropping, rotation, scaling etc, (2) A Digital-to-Analog conversion, such as printing and then Analog-to-Digital conversion by scanning (can also be done by re-sampling), (3) Lossy compression and (4) Duplicating small segments of the picture and deleting others (jitter attack) [9].

It is shown then, that several parameters must be considered for each application, in order to optimize the use of counter-measures. The goal is to maintain the required image quality desired for each application and still be robust to potential attacks. That trade-off is discussed in the next two subsections.

2.2.2 Image quality

As mentioned, an important objective of a good WM is minimizing image quality degradation. Recently we have shown [20], that for a blind content-independent algorithm, the trade-off between the security (capacity) of the mark and the negative affect on image quality is straight-forward. There, the WM is embedded by adding a pseudo-random noise to each pixel. Increasing the bit size of the mark is equivalent to increasing the variance of the noise, which is the measure for the capacity of that algorithm. It relates directly to better false detection ratio. However, it also adds significant high frequency values to the original image, affectively degrading its quality, especially in homogeneous parts of the picture.

To avoid such a significant degradation, it is possible to increase the security of the mark by making it content dependent [16],[18]. In a content dependent WM system, the embedded data is also some function of the cover image. The decoder would need the cover image data in order to extract the correct WM, making it more difficult, and sometimes even impossible to use for marking unoriginal content. This introduces higher computational complexity, but features a more secured mark without influencing the cover data severely.

2.2.3. Computational complexity

Intuitively, it is obvious that in order to apply a more complicated algorithm, more complex embedding and detecting blocks would be required. The motivation to keep the computational complexity low depends on the application and on the method of implementation. In real time applications, computations must be done in a very short time period. The speed and processing power of the computational platform at hand, limit the algorithm level of complexity that can be computed in a given time frame. When implementing in hardware, higher complexity requires additional hardware which means more area and additional costs.

In [20] we have introduced a scheme that is very easily implemented in hardware. In this implementation, computation time and hardware requirements are almost negligible, however compromising the performance achieved. As previously mentioned, in order to provide high detection rates perceptual effects are inevitable. In addition, this scheme is not able to detect local modifications. A potential attacker may take advantage of this inability, to cover parts of the picture he is trying to hide.

Depending on the intended application, more complicated schemes can be implemented to withstand expected attacks. If, for instance, localization of the changes made is important, a partition of the image into blocks may be of use. If the marked image is expected to go through lossy compression, one may consider embedding the WM in the frequency domain, as will be described in the next section. Other algorithms employ global and local mean values, temporal dependencies (in video WM) and variety of extra features to enhance their performance. However, each additional feature, added to the algorithm, increases the computational effort and hardware resources (such as memory and adders\multipliers) used. Therefore, an optimized scheme will be comprised of the minimum number of features needed to satisfy the needs of the application it is designed for.

2.3 Discussion

It has been shown that during the design of a WM system, many trade-offs are taken into account. How can one evaluate the overall quality of the final outcome? Although there is no accepted standard to uniformly asses the quality of WM [21], there are a few popular benchmarks available. A designer can use the evaluated system to embed a WM in a series of test images, and then run them through the benchmark and asses the performance by observing the quality of detection. The *StirMark* code, which is used for evaluating the robustness of WM algorithms designed for copyright protection applications, applies a series of attacks on a marked image [22]. In addition, it is possible to evaluate robustness to specific attacks by manually adding them to this benchmark. The *Checkmark* benchmark provides a framework for application-oriented evaluation of WM schemes, applicable to all sorts of WM algorithms including fragile and semi-fragile [21]. The use of such independent, third party, evaluation tools provides a good perspective on how well a WM system performs.

3. Watermark implementations – Software vs. Hardware

Figure 3 shows a scheme of a general WM system. The system consists of a WM generation, embedding and detection algorithms.

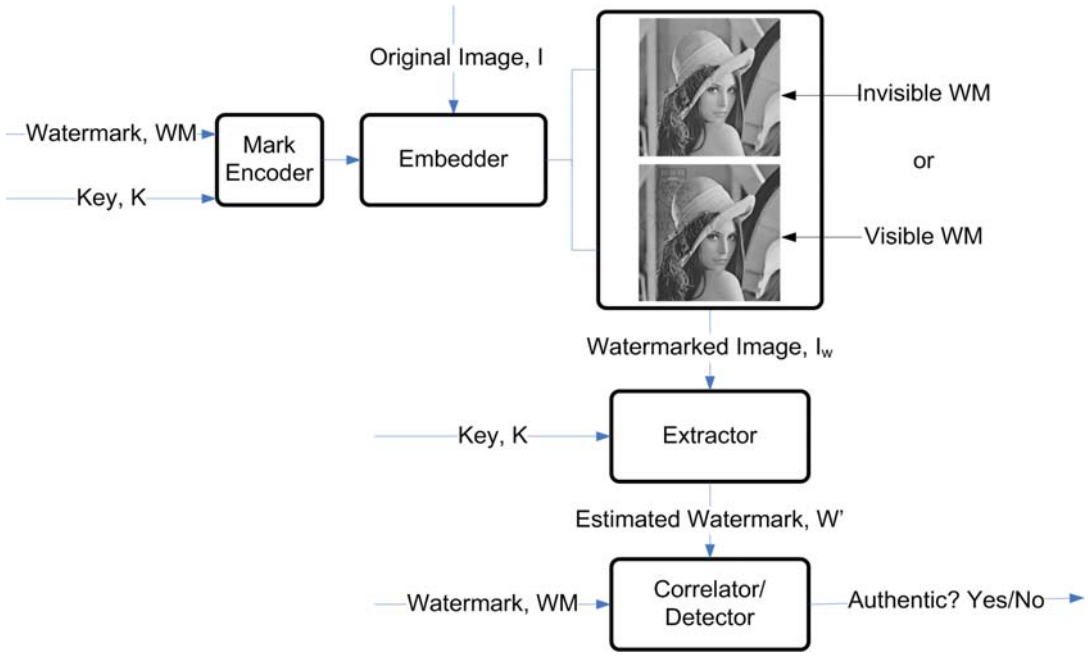

Figure 3. Scheme of general watermark system.

The identifying data (W in Figure 3) can be meaningful, like a logo, or it can just be a known stream of bits. First the identifying data is encoded using a secret key, K. Then the encoded identifying data is embedded into the original image (I in Figure 3). The result is the WM image. As previously mentioned, the WM can be visible or invisible, as shown in Figure 3. The detector part is at the receiving end. The objective is to extract the identifying data embedded in the received image, using the secret key and an inverse algorithm. Finally, a decision is made by correlating the extracted mark with the original and applying a chosen threshold.

The system can be implemented on either software or hardware platforms, or some combination of the two. A pure software WM scheme can simply be implemented in a PC environment. Such an implementation is relatively slow, as it shares computational resources and its performance is limited by the operating system. It is unsuitable for real-time applications, for it would be too slow, and it cannot be implemented on portable imaging devices that have limited processing power. On the other hand it can be easily programmed to realize any algorithm of any level of complexity, and can be used on everyday consumer PC's.

A good example of software WM solution was presented by Li [18]. In this work he proposes software implemented fragile WM, embedded in the coefficients of the block DCT. This algorithm is designed for authentication and content integrity verification of JPEG images. The algorithm embeds the WM only in a few selected DCT coefficients of every block in order to minimize the effect on the image. The author directly

addresses known issues in similar previous works, inserting additional complexity to overcome security gaps. The system utilizes the advantages of software implementation by using resources needed to store image data, transform coefficients and WM mappings. Using a combination of different security resources, including a non-deterministic mapping of the location of coefficient modulation and block dependencies, the system succeeds in facing several attacks without changing the affect on image quality – when compared to similar works. Moreover, the computations involved in the embedding process are kept relatively basic, suggesting suitability for future hardware implementation as well.

In opposite to software solutions, hardware implementations offer an optimized specific design to incorporate a small, fast and potentially cheap WM embedder. It is most suitable for real time applications, where computation time has to be deterministic (unlike software running on a windows system for example) and short. Optimizing the marking system hardware enables it to be added into various portable imaging devices. In a full imaging system that includes both the imager and WM embedder, the system security is improved as it is certain that the data entering the system is untouched by any external party. However, hardware implementations usually limit the algorithm complexity and are harder to upgrade. The algorithm must be carefully designed, minimizing any unexpected deficiencies. For example, in [16], Mohanty et al. present an implementation of both fragile and robust invisible WM algorithms in hardware. Which WM is used will be defined by the user. The WM are embedded in the spatial domain but are designed to be robust for JPEG compression. The motivation for hardware implementation is to enable the integration of a WM module within a secure digital camera system. As JPEG is the standard data format for digital cameras it is imperative that the algorithm will not be harmed by compression. The availability of two kinds of WM algorithms corresponds to applications such as image authentication for the fragile algorithm and copyright protection for the robust. The hardware employed in this implementation is comprised of image and WM RAM memories, adders/subtractors, registers and multipliers. This is a relatively large implementation and accordingly, the algorithm is rather complex.

4. WM Hardware Implementation - A Development Methodology

In this section a development methodology to a fast mixed signal hardware design is presented. This methodology can be used for the development of an image sensor with integrated WM capabilities. Figure 3 shows different elements required for such a system. The design of this complete system is a very demanding task requiring time and financial resources. It involves hardcore analog and mixed signal design as well as complex digital architecture. Although the end goal is to implement the whole system monolithically on a single chip, it is expected, as in every development process, that more than one prototype will be designed before a final version is issued. Therefore it is worthwhile, to first focus on determining the core elements of the system which are the imager, the WM digital architecture and the interface between the two. To do so without significantly compromising the quality and performance of the peripheral elements such as the A/D converter, analog voltage biasing and memory, the design is first tested on a board utilizing commercial devices.

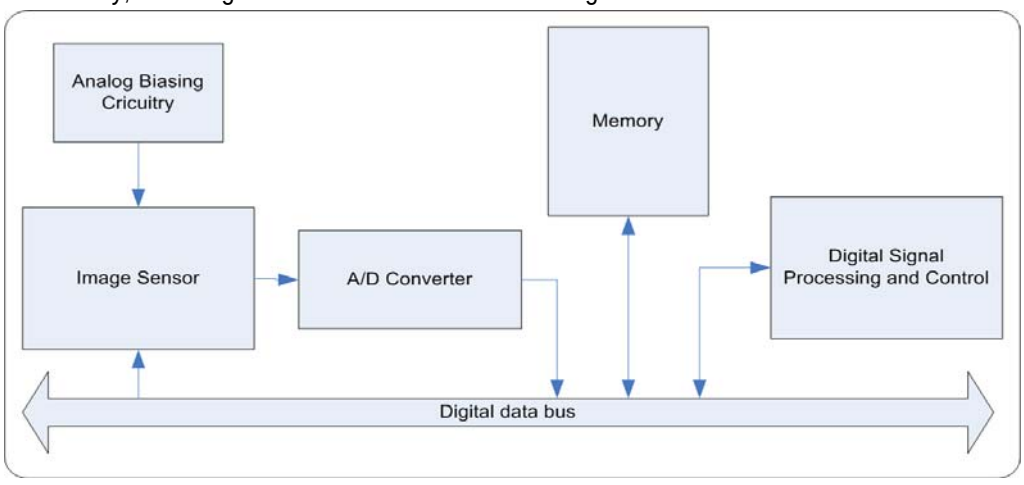

Figure 4. A general scheme of the complete image system in conjunction with digital processing.

This specifically designed board, shown in Figure 5, emulates a System-On-a-Chip (SoC) platform, allowing the incorporation of custom VLSI designs (the imager) with peripheral elements and digital logic implemented on an FPGA. It features low-noise, separated digital and analog power supplies, 12 bit analog voltage and current biasing, 12 and 18 bit A/D converters, an SRAM memory and several I/O ports including LVDS, RS-232 and direct test points for maximum testing flexibility. The designer can choose what part of the system he wants to implement in VLSI and what elements he would use of those available on board. For the discussed WM implementation a basic imager is first designed, and then the WM logic is implemented on the FPGA, together with all other required control logic, making use of the A/D converter and SRAM memory to aid the implementation of more complex algorithms. Finally the data is read out either as a WM video stream through the LVDS interface or stored in memory and read as a WM still image.

Note the presented methodology is modular and can host various kinds of SoC designs.

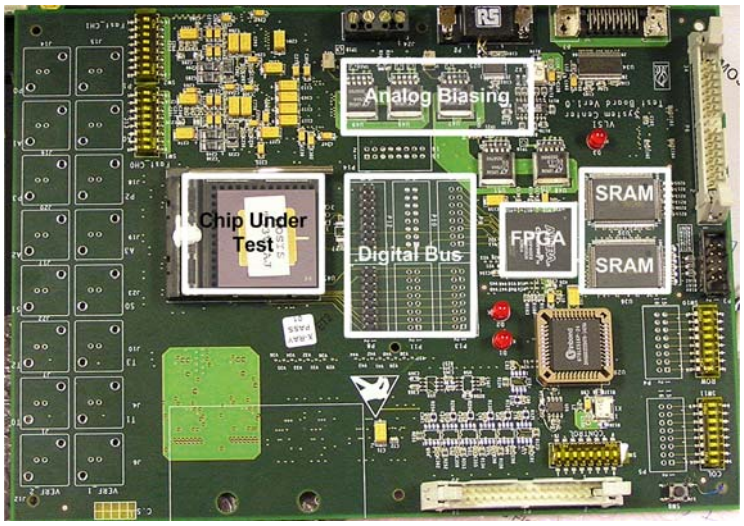

Figure 5. Mixed signal SoC fast prototyping custom development board.

5. Conclusions

Basic terms and principles in WM design and evaluation were presented. The main trade-offs and design considerations were discussed pointing out the importance of designing in light of the intended application and expected attacks. Several common attacks were also described, as well as a couple of evaluation benchmarks that facilitate the testing of different WM schemes robustness to a very large scale of attacks. The general scheme of a WM system was shown and the major benefits and shortcomings to implementations in hardware or software described. Two examples of previous works done, featuring fragile and robust WM, spatial and DCT domains, hardware and software implementations were given. In addition, a development methodology, employing custom development board for mixed signal SoC fast prototyping was shown.

Bibliography

[1] V. M. Potdar, S. Han, E. Chang, "A survey of digital image watermarking techniques", 3rd IEEE International Conference on Industrial Informatics (INDIN '05), Aug. 2005, pp. 709- 716

[2] O. Yadid-Pecht and R. Etienne-Cummings, " CMOS imagers: from phototransduction to image processing", Kluwer Academic Publishers

[3] S. P. Mohanty, "Digital Watermarking: A Tutorial Review", URL: http://www.csee.usf.edu/~smohanty/research/Reports/WMSurvey1999Mohanty.pdf

[4] F. Mintzer, G. Braudaway, and M. Yeung, “Effective and ineffective digital watermarks,” in Proc. IEEE Int. Conf. Image Process., vol. 3, 1997, pp. 9–12.

[5] C. T. Li, D. C. Lou and T. H. Chen, “Image authenticity and integrity verification via content-based watermarks and a public key cryptosystem”. Proc. IEEE Int. Conf. on Image Processing, Vancouver, Canada Sep.2000.vol. III, pp. 694-697.

[6] S. P. Mohanty, et al., "A DCT Domain Visible Watermarking Technique for Images", Proc. of the IEEE International Conference on Multimedia and Expo, July 30- August 2, 2000, Hilton New York & Towers, New York City, NY, USA.

[7] M. J. Tsai and H. Y. "Wavelet Transform Based Digital Watermarking for Image Authentication," IEEE Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science,0-7695-2296-3/05,2005.

[8] P. Meerwald S. Pereira, "Attacks applications and evaluation of known watermarking algorithms with Checkmark". SPIE Electron. Imaging. v4675

[9] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," Information Hiding: 2nd Int. Workshop, D.Aucsmith, Ed. , ser. Lecture Notes in Computer Science Berlin, Germany: Springer-Verlag, vol. 1525, pp. 218-238, 1998

[10] K. Tanaka, Y. Nakamura and K. Matsui, "Embedding Secret Information into a Dithered Multi-level Image" IEEE Military Communications Conference 1990 pp. 0216-0220

[11] F. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding - A Survey" Proc. IEEE 87(7) Jul. 1999 pp. 1062-1078

[12] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark", in Proc. IEEE Int. Conf. Image Processing, vol. 2, Austin, TX, 1994, pp. 86–90

[13] R. B. Wolfgang and E. J. Delp, "A watermark for digital images", in Proc. IEEE Int. Conf. Images Processing, Lausanne, Switzerland, Sept. 1996, pp. 219–222

[14] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia" in R. J. Anderson, Ed., "Information hiding: First international workshop", in Lecture Notes in Computer Science, vol. 1174. Berlin, Germany: Springer-Verlag, 1996, pp. 183–206

[15] C. T. Li, "Digital fragile watermarking scheme for authentication of JPEG images", IEE Proc., Vis. Image Signal Process., 2004, 151, (6), pp. 460-466

[16] S. P. Mohanty, N. Ranganathan, and R. K. Namballa, "VLSI implementation of invisible digital watermarking algorithms towards the developement of a secure JPEG encoder", in Proc. IEEE Workshop Signal Processing Systems, 2003, pp. 183-188

[17] T. H. Tsai, C. Y. Lu, "Watermark embedding and extracting method and embedding hardware structure used in image compression system", US Patent 6,993,151, 2006

[18] C. T. Li, "Digital fragile watermarking scheme for authentication of JPEG images", IEE Proc.-Vis. Image Signal Processing, Vol. 151, No. 6, December 2004, pp. 460-466

[19] M. Barni, F. Bartolini, and A. Piva, "Improved Wavelet-Based Watermarking Through Pixel-Wise Masking", IEEE Trans. Image Proc., vol. 10, no. 5, pp. 783-791, May 2001

[20] G. R. Nelson, G. A. Jullien and O. Yadid-Pecht, "CMOS image sensor with watermarking capabilities", in Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS '05), vol. 5, Kobe, Japan, May 2005, pp. 5326-5329

[21] P. Meerwald and S. Pereira, "Attacks, applications and evaluation of known watermarking algorithms with checkmark", In Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents IV, 2002

[22] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems", 11th Int. Symp. Electronic Imaging, vol. 3657, San Jose, CA: IS&T and SPIE, Jan. 25-27, 1999

[23] I. J. Cox, G. Doerr, T. Furon, "Watermarking is not cryptography", in YQ Shi, B Jeon, Eds., "Digital Watermarking : 5th International Workshop", in Lecture Notes in Computer Science, vol. 4283, Berlin, Germany: Springer-Verlag, 2006, pp. 1-15

Authors' Information

Yonatan Shoshan – ISL lab, ATIPS Lab, ECE Department, University of Calgary, Calgary AB, Canada; e-mail: shoshay@atips.ca

Alexander Fish – ISL lab, ATIPS Lab, ECE Department, University of Calgary, Calgary AB, Canada; e-mail: fish@atips.ca

Xin Li – ISL lab, ATIPS Lab, ECE Department, University of Calgary, Calgary AB, Canada; e-mail: xinli@atips.ca

Graham Jullien – ATIPS Lab, ECE Department, University of Calgary, Calgary AB, Canada; e-mail: jullien@atips.ca

Orly Yadid-Pecht – ISL lab, ATIPS Lab, ECE Department, University of Calgary, Calgary AB, Canada; The VLSI Systems Center, Ben-Gurion University, Beer-Sheva, Israel; e-mail: orly@atips.ca