WATERMARKING – VOLUME 2

Edited by **Mithun Das Gupta**

INTECHOPEN.COM

Watermarking – Volume 2
Edited by Mithun Das Gupta

Published by InTech
Janeza Trdine 9, 51000 Rijeka, Croatia

Notice
Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

Publishing Process Manager Sasa Leporic
Technical Editor Teodora Smiljanic
Cover Designer InTech Design Team

First published May, 2012
Printed in Croatia

A free online edition of this book is available at www.intechopen.com
Additional hard copies can be obtained from orders@intechopen.com

Watermarking – Volume 2, Edited by Mithun Das Gupta
p. cm.
ISBN 978-953-51-0619-7

Contents

Preface

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

Mithun Das Gupta
Bio Signals and Analysis lab at GE Global Research Bangalore
India

1

Recent Advances in Watermarking for Scalable Video Coding

Dan Grois and Ofer Hadar
Ben-Gurion University of the Negev, Beer-Sheva
Israel

1. Introduction

The H.264/AVC (ISO/IEC MPEG-4 Part 10) video coding standard (Wiegand & Sullivan, 2003), which was officially issued in 2003, has become a challenge for real-time video applications. Compared to the MPEG-2 standard, it gains about 50% in bit rate, while providing the same visual quality. In addition to having all the advantages of MPEG-2 (ITU-T & ISO/IEC JTC 1, 1994), H.263 (ITU-T, 2000), and MPEG-4 (ISO/IEC JTC 1, 2004), the H.264 video coding standard possesses a number of improvements, such as the content-adaptive-based arithmetic codec (CABAC), enhanced transform and quantization, prediction of "Intra" macroblocks, and others. H.264 is designed for both constant bit rate (CBR) and variable bit rate (VBR) video coding, useful for transmitting video sequences over statistically multiplexed networks, the Ethernet, or other Internet networks). This video coding standard can also be used at any bit rate range for various applications, varying from wireless video phones to high definition television (HDTV) and digital video broadcasting (DVB). In addition, H.264 provides significantly improved coding efficiency and greater functionality, such as rate scalability, “Intra” prediction and error resilience in comparison with its predecessors, MPEG-2 and H.263. However, H.264/AVC is much more complex in comparison to other coding standards and to achieve maximum quality encoding, high computational resources are required (Grois et al., 2010a; Kaminsky et al., 2008).

Due to the recent technological achievements and trends, the high-definition, highly interactive networked media applications pose challenges to network operators. The variety of end-user devices with different capabilities, ranging from cell phones with small screens and restricted processing power to high-end PCs with high-definition displays, have stimulated significant interest in effective technologies for video adaptation for spatial formats, consuming power and bit rate. As a result, much of the attention in the field of video adaptation is currently directed to the Scalable Video Coding (abbreviated as "SVC" or "H.264/SVC"), which was standardized in 2007 as an extension of H.264/AVC (Schwarz et al., 2007), since the bit-stream scalability for video is currently a very desirable feature for many multimedia applications (Grois et al., 2010b; Grois et al., 2010c).

Scalable video coding has been an active research and standardization area for at least 20 years (Schwarz et al., 2007). The prior international video coding standards MPEG-2 (ITU-T & ISO/IEC JTC 1, 1994), H.263 (ITU-T, 2000), and MPEG-4 (ISO/IEC JTC 1, 2004) already include several tools by which the most important scalability modes can be supported. However, the scalable profiles of those standards have rarely been used. Reasons for that include the characteristics of traditional video transmission systems as well as the fact that the spatial and quality scalability features came along with a significant loss in coding efficiency as well as a large increase in decoder complexity as compared to the corresponding non-scalable profiles (Schwarz et al., 2007; Wiegand & Sullivan, 2003).

To fulfill these requirements, it would be beneficial to simultaneously transmit or store video in variety of spatial/temporal resolutions and qualities, leading to the video bitstream scalability. Major requirements for the Scalable Video Coding are to enable encoding of a high-quality video bitstream that contains one or more subset bitstreams, each of which can be transmitted and decoded to provide video services with lower temporal or spatial resolutions, or to provide reduced reliability, while retaining reconstruction quality that is highly relative to the rate of the subset bitstreams. Therefore, the Scalable Video Coding provides important functionalities, such as the spatial, temporal and SNR (quality) scalability, thereby enabling the power adaptation. In turn, these functionalities lead to enhancements of video transmission and storage applications (Grois et al., 2010b; Grois et al., 2010c; Grois & Hadar, 2011).

Scalable Video Coding bitsream contains a Base-Layer (*Layer 0*) and one or more Enhancement Layers (*Layers 1, 2, etc.*), while the Base-Layer provides the lowest bitsream resolution with regard to the spatial, temporal and SNR/Quality scalability, as schematically presented in *Figure 1* (Schierl et al., 2007).

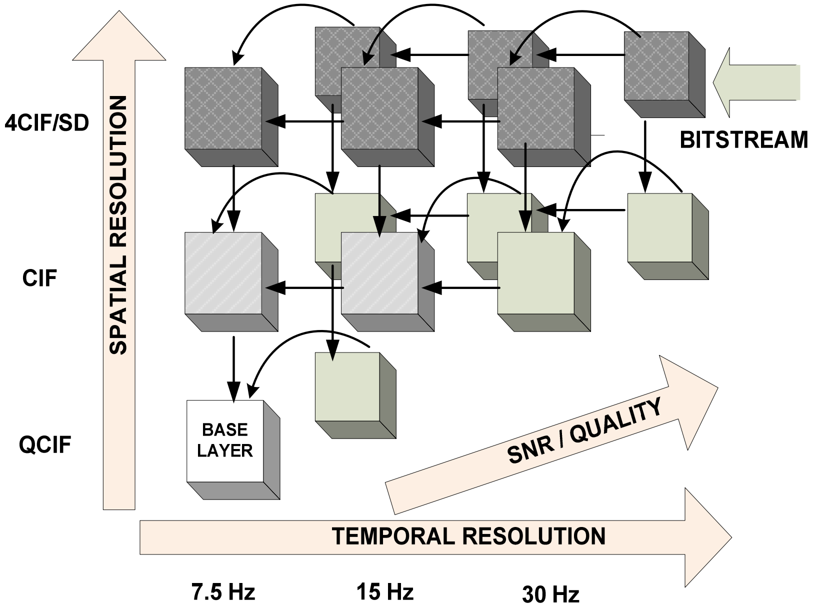


Fig. 1. Schematic representation of the SVC bitsream: the resolution is increased with the increase of the layer index, while the Base-Layer (*Layer 0*) has the lowest bitsream resolution (Schierl et al., 2007).

The term "scalability" refers to the removal of parts of the video bit stream in order to adapt it to the various needs or preferences of end users as well as to varying terminal capabilities or network conditions. According to (Schwarz et al., 2007), the objective of the SVC standardization has been to enable the encoding of a high-quality video bit stream that contains one or more subset bit streams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the existing H.264/AVC design with the same quantity of data as in the subset bit stream. *Figure 2* below presents a block-diagram of a SVC encoder, which has for simplicity two spatial layers: *Layer 0*, which is the Base Layer, and *Layer 1*, which is the first Enhancement Layer. It should be noted that in order to improve the coding efficiency of the Scalable Video Coding in comparison to simulcasting of different spatial resolutions, additional "inter-layer prediction mechanisms" are incorporated (Schwarz et al., 2007).

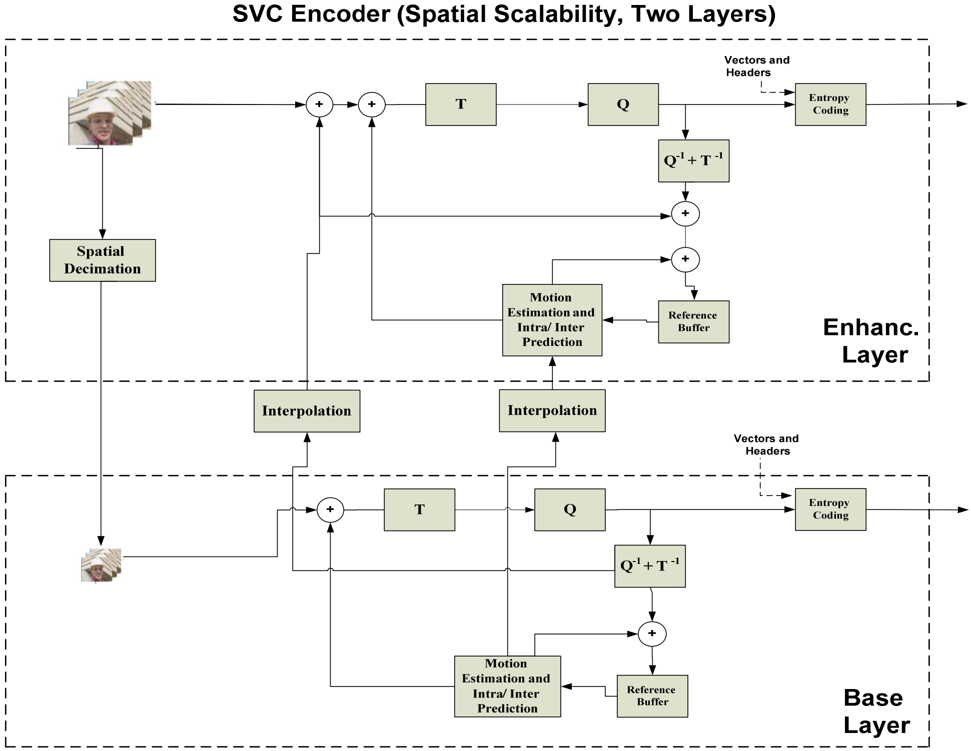


Fig. 2. Block-diagram of the spatial SVC encoding scheme (for simplicity, only two layers are presented: *Layer 0*, which is the Base Layer, and *Layer 1*, which is the first Enhancement Layer).

The Scalable Video Coding has achieved significant improvements in coding efficiency comparing to the scalable profiles of prior video coding standards. As a result, the Scalable Video Coding is currently a highly attractive solution to the problems posed by the characteristics of modern video transmission systems (Schwarz et al., 2007).

Scalable Video Coding poses new challenges for watermarking that need to be addressed to achieve full protection of the scalable content (Meerwald, 2011; Lin et al., 2004), while maintaining low bit-rate overhead due to watermarking. Challenges that complicate watermark detection include the very different statistics of the transform domain coefficients of scalable base- and enhancement layers, the combination of multi-channel detection results for incremental detection performance (Piper et al., 2005), as well as the prediction of data between scalability layers which complicates the modeling of the embedding domain. Despite intense research in the area of image and video watermarking (Meerwald, 2011; Lin et al., 2004), the peculiarities of watermarked scalable multimedia content have received limited attention and a number of challenges remain.

One of the main challenges for watermarking the rate-scalable compressed video is that not all receivers will have access to the entire (watermarked) video stream (Lin et al., 2001). The embedded watermark must be detectable when only the base layer is decoded (for layered and hybrid layered/embedded methods) or for a low rate version of the video stream (for embedded methods.) However, the enhancement information adds value to the video stream and should not be left unprotected by a watermark. Ideally, there should be a uniform improvement in the detectability of an embedded watermark as the decoded rate increases.

According to one method for watermarking the rate-scalable video streams, a watermark is embedded in the base layer and a separate watermark is embedded in the enhancement layer(s) (Lin et al., 2001). For temporal scalability, this is an effective method for watermarking as the enhancement information does not alter the frames encoded in the base layer. However, for other forms of scalability, care must be taken so that the multiple watermarks do not interfere with each other once the decoder merges the base and enhancement information. The watermarks could interfere in visibility, where the distortions introduced by adding all watermarks is unacceptable, or detectability, where the presence of all the watermarks impair the ability to detect each watermark individually. The ability to detect each embedded watermark individually (before the enhancement and base information are merged) is not sufficient for a robust watermark, as such a system would be vulnerable to a collusion attack between the non-enhanced and enhanced versions of the video.

For embedded scalability modes, one could design a watermark analogous to an embedded coding scheme, where the most significant structures of the watermark are placed near the beginning of the video stream, followed by structures of lesser significance (Lin et al., 2001).

With this regard, *Figure 3* below presents different watermarking embedding schemes by using the SVC spatial scalability (Meerwald, P. & Uhl, A., 2010a).

Watermarking systems are oftern characterized by a set of common features and the importence of each feature depends on the application requrements. As known, the watermarks are generally devided to three main groups (Piper, 2010):

a. *Robust:* Robust watermarks are designed to be resistant to manipulations of the content. Therefore, a robust watermark can be still detected after the content has undergone processing, such as resampling, cropping, lossy compression, and the like.
b. *Fragile:* fragile watermarks are very sensitive to any manipulations to the content. This does not make the fragile watermark inferior to the robust watermark, since different applications demand different amounts of robustness or fragility.

c. *Semi-Fragile:* semi-fragile wateermarks are designed to be fragile with respect to some changes but to tolerate other changes. For example, they may be robust to compression but will be able to detect malicious tampering. This can be achived by carefully designing the watermark to be robust for particular image/video manipulations.

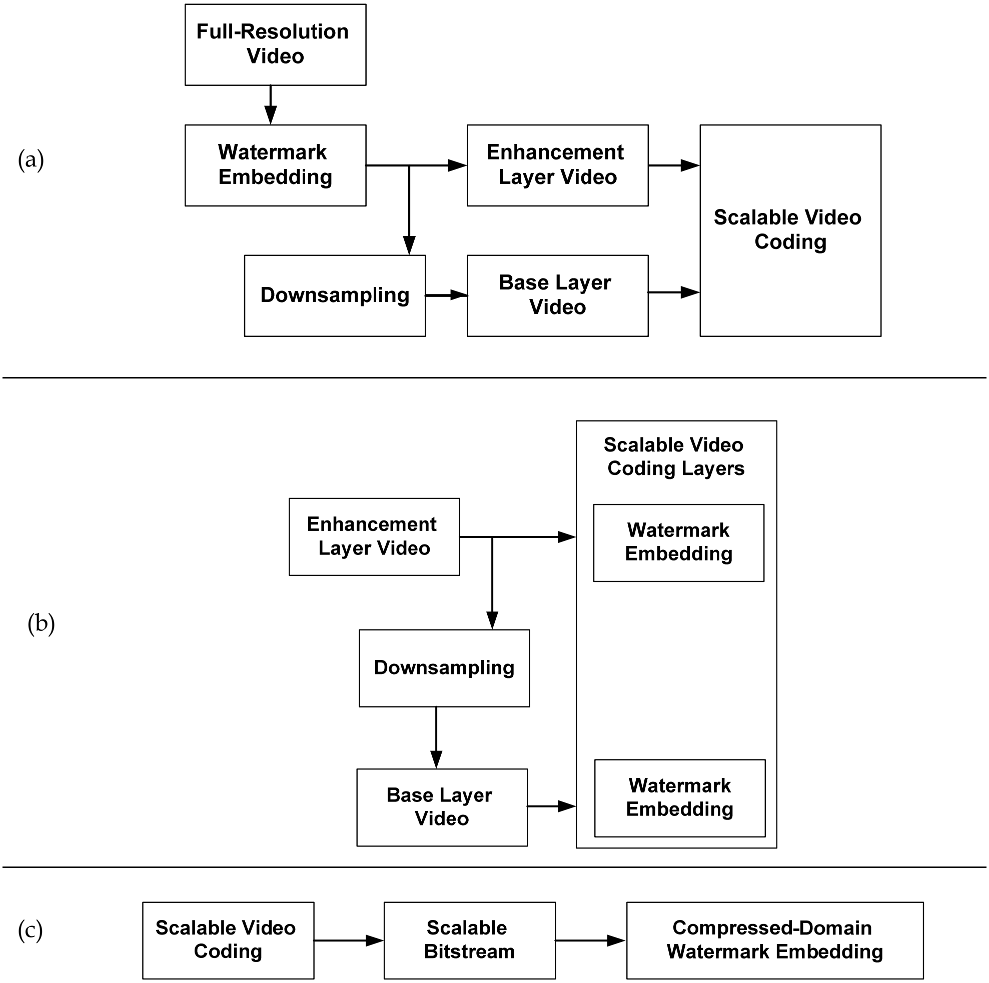


Fig. 3. Three different watermarking embedding schemes by using spatial Scalable Video Coding (Meerwald, P. & Uhl, A., 2010a): a) Watermark embedding prior to the video encoding; b) Integrated watermark embedding and coding; and c) Compressed-domain embedding after encoding.

Further, *Table 1* below presents common watermarling applications, which are used with regard to different watermark features (Bhowmik, 2010):

Application Name	Description
Broadband Monitoring	Passive monitoring by the automatic watermark detection of the broadcasted watermarked media.
Copyright Identification	Resolving copyright issues of digital media by using the watermark information as the copyright data.
Content Authentication	Authentication of original art work, performance and protection against digital forgery.
Access Control	Access control applications, such as, Pay-TV.
Copy Control	Disabling copy of CD/DVD by the watermarked permission.
Packaging and Tracking	Transaction tracking and protection against forged consumable items (including pharmaceutical products, and the like) by embedding a watermark on packaging.
Medical Record Authentication	Authentication of digitally preserved patient's medical record, including a blood sample, X-ray, etc.
Insurance / BankingDocument Authentication	Digital authentication of an insurance claim, banking, financial, mortgage and corporate documents.
Media Piracy Control	Tracking of the source of the media piracy.
Ownership Identification	Supporting a legitimate claim, such as, royalty by the the media owner.
Transaction Tracking	Tracking of the media ownership in a buyer-seller scenario.
Meta-data Hiding	Hiding meta-data within the media instead of a big header.
Video Summary Creation	Instant retrieval of video summary by embedding the summary within the host video.
Video Hosting Authentication	Piracy control by video authentication at video hosting servers, including Youtube™, etc.

Table 1. Common watermarling applications (Bhowmik, 2010).

Since, the robust watermarking algorithms, which are designed specifically for robustness, are preferred in a majority of watermarking applications, we mainly fosus this chapter on this type of watermarking. Also, we make a special emphazis on the combined schemes of watermarking and encryption by using the H.264/SVC due to the increasing interest with regard to this issue.

This chapter is organized as follows: in *Section 2*, we present recent advances in robust watermarking by using the Scalable Video Coding, in *Section 3,* we discuss recent advances in the scalable fragile watermarking, then in *Section 4*, we present recent compressed-domain watermarking techniques by using the Scalable Video Coding, and after that in *Section 5*, we talk about combined schemes of watermarking and encryption by using the Scalable Video Coding. The future research directions are outlined in *Section 6*, and this chapter is concluded in *Section 7.*

2. Robust watermarking by using scalable video coding

In general, digital watermarking has been proposed as a solution to the problem of copyright protection of multimedia data in the complicated network environment (Shi et al., 2010). Especially, in today’s society, with the progress of 3G/4G wireless networks and the plurality of heterogeneous mobile devices, the multimedia resources must be accessed by many different terminals, which require the source single multimedia stream to meet the varying terminal capabilities. Thus, the Scalable Video Coding can be efficiently employed to achieve these goals. However, due to the SVC scalability, the source video stream can be decoded into a plurality of streams, each having a different resolution, frame rate and video presentation quality, according to each end-user terminal. Therefore, there are many challenges for watermarking by using the Scalable Video Coding approach (Shi et al., 2010).

It should be noted that using the prior knowledge of the Scalable Video Coding system and the transmission channel are beneficial for the watermarking system (Meerwald & Uhl, 2008), thereby enabling to use a number of supported spatial and temporal layers, denosing and deblocking filters, and the like (as schematically shown in *Figure 4*). As it is known, by exploiting the host video as the side-information at the encoder, in message coding and watermark embedding, the negative impact of host signal noise on the watermark decoder performance can be cancelled (Cox et al., 2002).

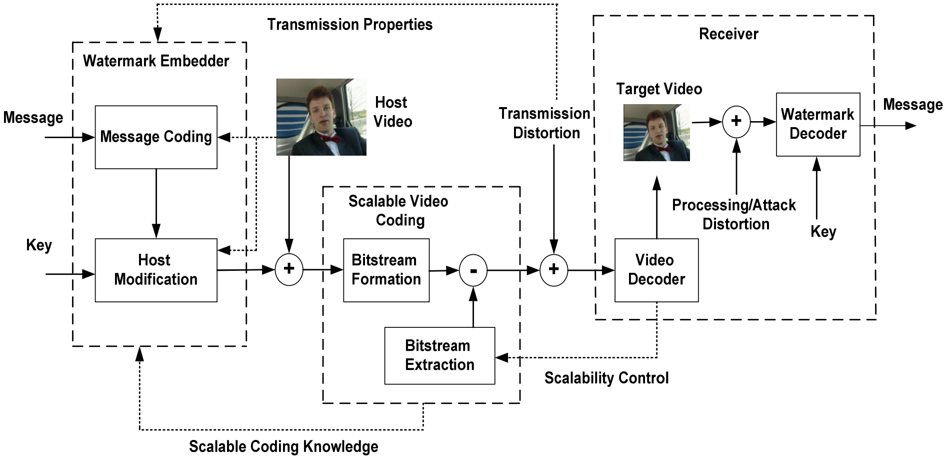


Fig. 4. Schematic diagram of the watermark communication channel by using Scalable Video Coding for blind watermarking (Meerwald & Uhl, 2008).

With regard to this issue, (Meerwald & Uhl, 2008) present a frame-by-frame scalable watermarking scheme that is robust for spatial, temporal and quality scalabilities, in which the luminance component of each frame is decomposed using a two-level wavelet transform with a 7/9 bi-orthogonal filter. Separate watermarks are embedded in the approximation and each detail subband layer. According to (Meerwald & Uhl, 2008), an additive spread-spectrum watermark $w_l(n,m)$ is added to the detail subband coefficients $d_{l,o}(n,m)$,

$$d^{'}_{l,o}(n,m) = d_{l,o}(n,m) + \alpha \cdot s_{l,o}(n,m) \cdot w_l(n,m), \tag{1}$$

where α is a global strength factor and $s_{l,o}(n,m)$ is a perceptual shaping mask derived from a combined local noise and frequency sencitivity model. l and o indicate a hierarchical level and orientation of the subband. Blind watermark detection can be performed independently for each hierarchical layer by using the normalized correlation coefficient detection. By applying a high-pass 3X3 Gaussian filter to the detail subbands prior to the correlation, some of the host interference is suppresses, which improves the detection statistics. Also, a different key is used for each frame to generate the watermark pattern (Meerwald & Uhl, 2008).

Further, (Meerwald, P. & Uhl, A., 2010b) focus on the watermark embedding in the intra-coded macroblocks of an H.264-coded base layer. Each macroblock of the input frame is coded by using either intra- or inter-frame prediction, and the difference between input pixels and the prediction signal is the residual. The watermarked SVC base layer representation is used for predicting the SVC enhancement layer, as seen from *Figure 5* below (Meerwald, P. & Uhl, A., 2010b).

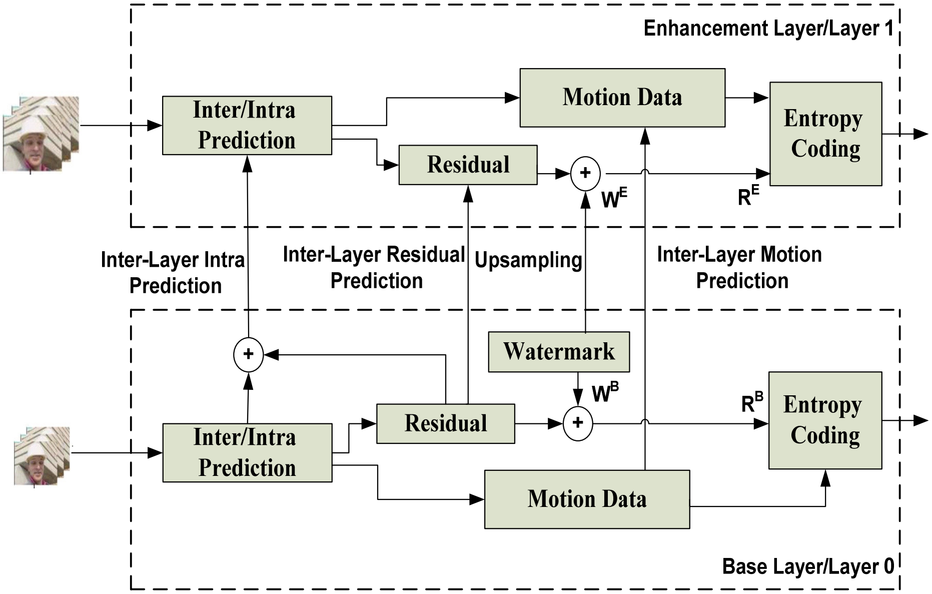


Fig. 5. Sample encoding watermarking structure of two spatial SVC layers (Meerwald, P. & Uhl, A., 2010b).

As already mentioned, for the scalable watermark system, the key scalable property is that the detection process is scalable (Shi et al., 2010). In other words, the system should be able to detect a watermark in all different scalable bits-streams. As the quality of multimedia decreases, the correlation between the watermark and watermarked signal may be decrease as well. So, it will not work effectively if the same threshold is used for each SVC layer. However, if different detective thresholds are used for different layers, the watermark system is required to transmit some extra side information. One potential measure is that the detective threshold can be adjusted adaptively according to the multimedia content.

With this regard, (Shi et al., 2010) propose a scalable and credible watermarking algorithm towards Scalable Video Coding (SVC), which aims to build Copyright Protection System

(CPS). The authors first investigate where to embed the watermark to ensure it can be detected in the SVC Base Layer as well as in the Enhancement Layers, and then the authors propose a model that combines the frequency masking, contrast masking, luminance adaption and temporal masking. Finally, whether watermark exists or not is judged by the adaptive detection, which guarantees the proposed method has a good legal credibility, since its False Alarm Rate (FAR) is close to zero.

In the *Section 3*, we discuss recent advances in scalable fragile watermarking.

3. Recent advances in scalable fragile watermarking

The good authentication watermarking can detect and localize any change to the video, including changes in frame rate, video size or related video object (Wang et al., 2006). If the watermarked video is attacked by frame removing, and then the watermark extracting procedure is applied on the attacked video, the procedure returns a false alarm to indicate that the video content becomes incomplete. Also, if one change the size of watermarked video and then one applies the watermark extraction procedure on this resized video, the procedure returns an output that resembles random noise, meaning a false alarm. Similarly, if one modifies certain related video object, then the procedure will output a false alarm (Wang et al., 2006).

With this regard, (Wang et al., 2006) propose to embed the watermark information into the Enhancement Layer of MPEG-4 Fine Granularity Scalability (FGS), as schematically shown in *Figure 6*, to detect the integrality of video stream. According to (Wang et al., 2006), it is supposed that w_i denotes the *i*-th watermark bit, and T_j denotes the total number of "1" bits in the *j*-th 8x8 bit-plane. The watermark w_i should be embedded into the *k*-th specified bit B_k in *j*-th bit-plane, and the detail of embedding watermarking can be described as follows. First, the specified bit (*k*-th bit) in the *j*-th bit-plane is selected by a run-length-selection algorithm for embedding *i*-th watermark bit. The run-length-selection algorithm can determine a specified bit for embedding watermark in 8x8 residue bit-plane and obtaining an optimal coding efficiency in run-length coding. If w_i is "1", then T_j will be enforced to be as an odd value. Similarly, if w_i is "0", then T_j will be enforced to be as an even value. That is, the specified bit B_k can be modified as $B_k^{'}$ by the following expression:

$$B_k^{'} = \begin{cases} 0 \ , & if \ w_i \oplus E(T_j) = 0 \\ 1 \ , & if \ w_i \oplus E(T_j) = 1 \end{cases} \tag{2}$$

where $E(T_j) = (T_j + 1) \bmod 2$, and "$\oplus$" denotes the exclusive "OR" operation.

Since fragile watermarking has extremely low resistance for various attacks, the extracted watermark signal fairy easy lose its completeness when multimedia content is modified or changed by a pirate or hacker. Thus, the multimedia can be determined where it has been changed or modified illegally according to the completeness of extracted watermark. (Wang et al., 2006) propose a BCW (Bitplane-Coding Watermarking) algorithm to add watermark information to the residual bit-planes of the Enhancement Layer. In embedding procedure, the watermark information is embedded into every 8×8 block of residual bit-planes in the Enhancement Layer, while encoding to MPEG-4 FGS video stream. The watermark bit is modulated by modifying a specified bit that is selected from each 8×8 bit-plane such that the even/odd value of the total number of "1" bits can meet the corresponding watermark information. The main reasons for hiding watermark into enhancement layers is that minimal degradation of the host data can be imperceptible as the watermark signal is inserted into the enhancement layer.

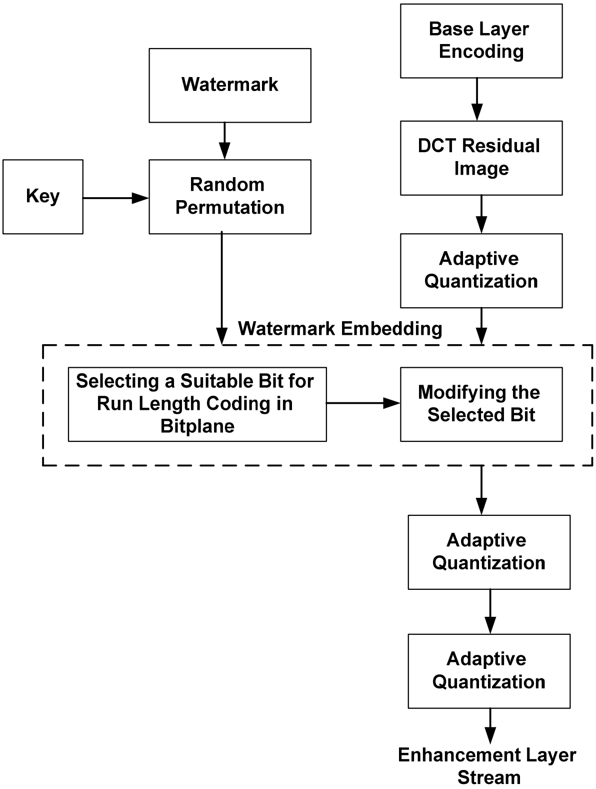


Fig. 6. Embedding a watermark in an Enhancement Layer of the MPEG-4 FGS video stream (Wang et al., 2006).

In turn, in *Figure 7* is presented a block diagram for the watermark extraction from the Enhancement Layer of MPEG-4 FGS video stream (Wang et al., 2006). If $E(T_j)$ is "1", the extracted watermarking data is equal to "1". Otherwise, if $E(T_j)$ is "0", the extracted watermarking data is also "0". The equation for extracting watermark can be expressed as follows:

$$w_i^{'} = \begin{cases} 0 \ , & if \ E(T_j) = 0 \\ 1 \ , & if \ E(T_j) = 1 \end{cases} \tag{3}$$

where $w_i \, (i = 0,1,2,3,4,...)$ is the *i*-th data of watermark. Also, in the watermark extracting of (Wang et al., 2006), the received Enhancement Layer (EL) stream with the watermarking data can be decoded to bit-planes through the Variable-Length Decoding (VLD) at the receiver end.

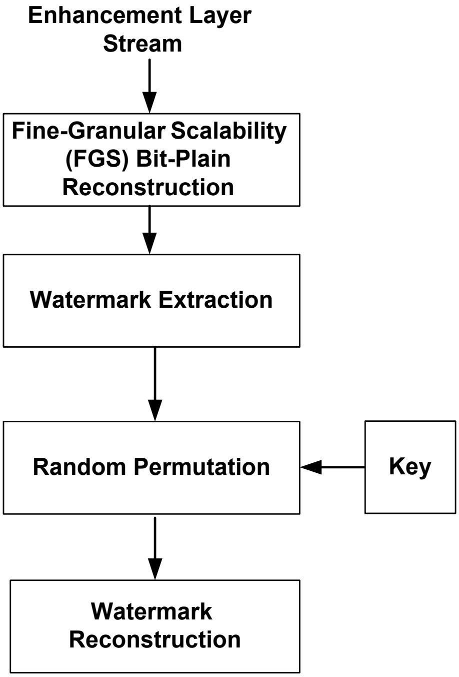


Fig. 7. Extracting a watermark from an Enhancement Layer of the MPEG-4 Fine Granularity Scalability (FGS) video stream (Wang et al., 2006).

In the following *Section 4*, we discuss compressed-domain watermarking by using Scalable Video Coding techniques.

4. Compressed-domain watermarking by using scalable video coding

The concept of scalable watermarking is composed of the expansion of progressive coding and the watermark system (Seo & Park, 2005). Progressive watermarking techniques enables to transmit images with a built-in watermark progressively, and then to extract the watermark from the decoded images. The scalable digital watermarking is mostly related to the scalable video coding techniques. Therefore, the scalable digital watermarking enables to protect contents regardless of the transmission of a specific domain, and enables to extract watermark from any domain of the scalable contents. Also, the increase of the scalable domain can also reduce an error of the watermark extraction (Piper et al., 2004). In *Figure 8*, the compression is performed on the original image after the wavelet transform, and the selected coefficients and watermark key are combined, followed by the spectrum quantization and encoding (Seo & Park, 2005). Therefore, by progressively transmitting the image from the low frequency band to the high frequency band, the receiver can extract the watermark from the corresponding image portion, which that contains the built-in watermark; the bit error rate is decreased, as the transmitted data of images, with the built-in watermark, is increased (Seo & Park, 2005).

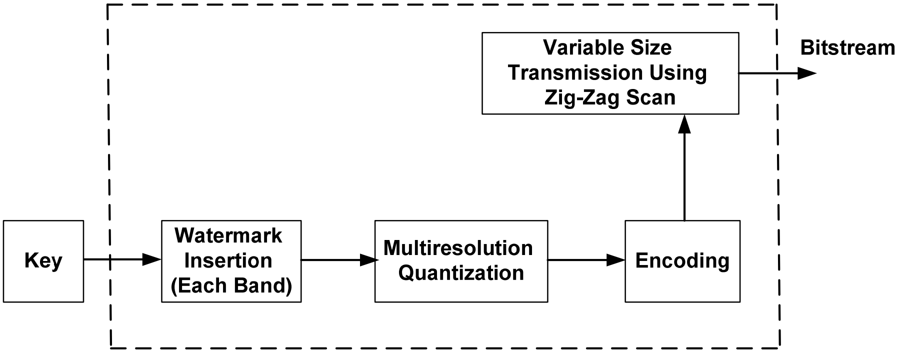

Variable Size Transmission Using Zig-Zag Scan
Bitstream
Key
Watermark Insertion (Each Band)
Multiresolution Quantization
Encoding

Fig. 8. Scalable watermarking in the compressed domain (Seo & Park, 2005).

In the following *Section 5*, we discuss combined schemes of watermarking and encryption by using the H.264/SVC.

5. Combined schemes of watermarking and encryption by using Scalable Video Coding

Intellectual Property (IP) protection is a critical element in a multimedia transmission system (Chang et al., 2004; Chang et al., 2005). Conventional IP protection schemes can be categorized into two major branches: *encryption* and *watermarking*. The content protection can be increased when combining the encryption and the robust watermarking, as proposed and implemented by (Chang et al., 2004; Chang et al., 2005). By taking advantage of the nature of cryptographic schemes and digital watermarking, the copyright of multimedia contents can be well protected.

In general, the Scalable Video Coding encryption can be defined as follows (Stutz & Uhl, 2011):

- Encryption before compression: There are no dedicated encryption proposals that take SVC-specifics into account (Stutz & Uhl, 2011).
- Compression/Integrated encryption: The base layer is encoded similar to AVC, thus all encryption schemes for AVC can be basically employed in the base layer. The enhancement layers can employ inter-layer prediction, but not necessarily have to, e.g., if inter-layer prediction does not result in better compression. The compression integrated encryption approaches for AVC can be applied as well for SVC, e.g., the approaches targeting the coefficient data can also be applied for SVC.
- Bitstream/ Oriented encryption: The approach of (Stutz & Uhl, 2008) takes advantage of SVC to implement transparent encryption after compression. The following approaches have been proposed for SVC encryption (Arachchi et al., 2009; Hellwagner et al., 2009; Nithin et al., 2009) which all preserve the NALU structure and encrypt

almost the entire NALU payload. As the NALU structure is preserved, scalability is preserved in the encrypted domain.

The scalable transmission method over the broadcasting environment for layered content protection is adopted by (Chang et al., 2004; Chang et al., 2005). As a result, the embedded watermark can be extracted with the high confidence and the next-layer keys/secrets can be perfectly decrypted and reconstructed. The watermarking is added to order to aid the encryption process, since the watermarked data content can withstand different types of attacks, such as distortions, image/video processing, and the like.

Further, (Park & Shin, 2008) presents a combined scheme of encryption and watermarking to provide the access right and the authentification of the video simultaneously, as schematically presented in *Figure 9*. The proposed scheme enables to protect the data content in a more secure way since the encrypted content is decrypted when the watermark is exactly detected. The encryption is performed for the access right, and the watermarking is implemented for the authentication. Particulalry, the encryption is preformed by encrypting the intra-prediction modes of the 4x4 luma block , the sign bits of texture, and the sign bits of MV difference values in the intra frames and the inter frames. In turn, a reversible watermarking scheme is implemented by using intra-prediction modes. The watermarking scheme proposed by (Park & Shin, 2008) has a small bit-overhead; however, no degradation of the visual quality occurs.

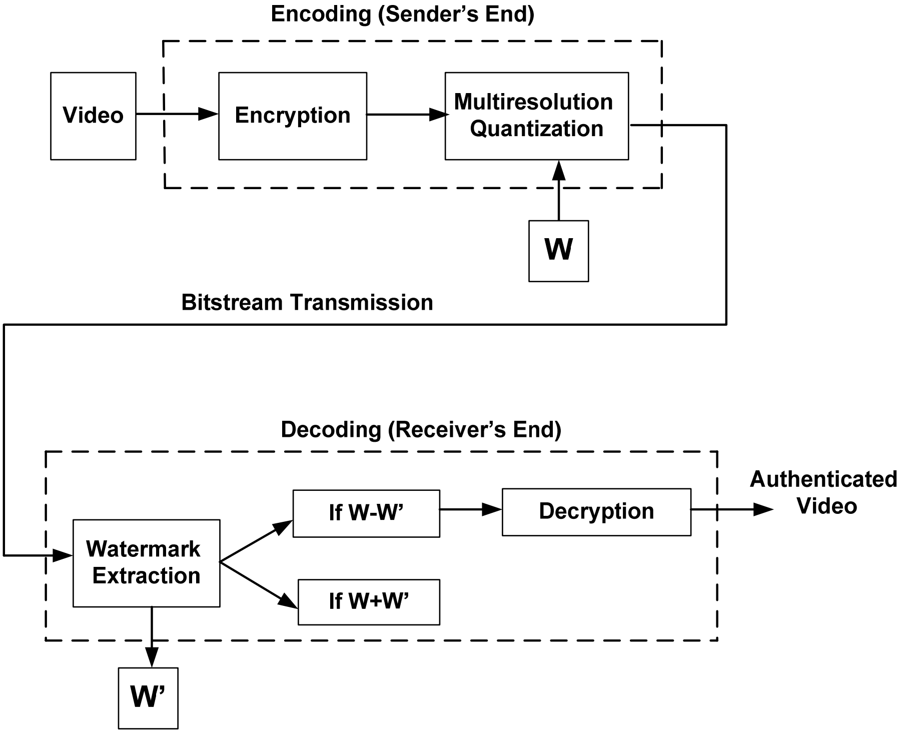


Fig. 9. Combined scheme of encryption and watermarking (Park & Shin, 2008).

The method of (Park & Shin, 2008) is applied in the Scalable Video Coding on the macroblock (MB) level in the Base Layer. The encryption and watermarking are implemented in the encoding process almost simultaneously. In turn, in the decoding process, the receiver's device extracts the watermark from the received bitstream. The extracted watermark is compared to the original one. If they match, then the received video s trusted and the encrypted bitsream is decrypted. In other words, according to (Park & Shin, 2008), only authenticated contents can be decoded in the decoding process.

In the following *Section 6*, we present possible future research directions for optimizing the existing watermarking techniques for use with the Scalable Video Coding.

6. Future research directions

The existing watermarking techniques for the Scalable Video Coding have still many issues to be solved in order to provide a complete solution, and possible future research directions can be outlined as follows (Bhowmik, 2010):

- Developing watermarking techniques for the Region-of-Interest (ROI) video coding by using the H.264/SVC;
- Modeling the transmission channel error and its influence on the watermark robustness for SVC applications;
- Developing real-time watermarking authentication schemes by using bitstream-domain watermarking for the H.264/SVC;
- Developing comprehensive compressed-domain SVC watermarking schemes, which enable scalability in the media distribution, while resolving digital rights management (DRM) issues.

7. Conclusions

In this chapter we have presented a comprehensive overview of recent developments in the area of watermarking by using the Scalable Video Coding. As discussed, the Scalable Video Coding poses new challenges for watermarking, which have to be addressed to achieve full protection of the scalable content, while maintaining low bit-rate overhead due to watermarking. Particularly, we presented recent advances in robust watermarking and discussed recent advances in the scalable fragile watermarking; also, we presented recent compressed-domain watermarking techniques by using the Scalable Video Coding, and presented combined schemes of the SVC watermarking and encryption.

As clearly seen from this overview, there are still many challenges to be solved, and therefore further research in this field should be carried out.

8. References

Arachchi, H. K., Perramon, X., Dogan, S. & Kondoz, A.M. (2009). Adaptation-aware encryption of scalable H.264/AVC video for content security, *Scalable Coded Media beyond Compression, Signal Processing: Image Communication*, iss. 24, vol. 6, pp. 468–483, 2009.

Bhowmik, D. (2010). Robust Watermarking Techniques for Scalable Coded Image and Video, Ph.D. Thesis, Department of Electronic and Electrical Engineering, University of Sheffield, 2010.

Chang, F.-C., Huang, H.-C. & Hang, H.-M. (2004). Combined encryption and watermarking approaches for scalable multimedia coding, *Pacific-Rim Conf. on Multimedia (PCM2004)*, pp. 356–363, Dec. 2004.

Chang, F.-C., Huang, H.-C. & Hang, H.-M. (2005). Layered access control schemes on watermarked scalable media, *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on* , pp. 4983- 4986, vol. 5, 23-26 May 2005.

Cox, I.J., Miller, M.L. & Bloom, J.A. (2002), Digital Watermarking, Morgan Kaufmann, 2002.

Grois, D.; Kaminsky, E. & Hadar, O. (2010). Optimization Methods for H.264/AVC Video Coding, The Handbook of MPEG Applications: Standards in Practice, (eds M. C. Angelides and H. Agius), John Wiley & Sons, Ltd, Chichester, UK, 2010.

Grois, D.; Kaminsky, E. & Hadar, O., (2010). ROI adaptive scalable video coding for limited bandwidth wireless networks, *Wireless Days (WD), 2010 IFIP*, pp.1-5, 20-22 Oct. 2010.

Grois, D.; Kaminsky, E. & Hadar, O. (2010). Adaptive bit-rate control for Region-of-Interest Scalable Video Coding, *Electrical and Electronics Engineers in Israel (IEEEI), 2010 IEEE 26th Convention of* , pp.761-765, 17-20 Nov. 2010.

Grois, D. & Hadar, O. (2011). Complexity-aware adaptive bit-rate control with dynamic ROI pre-processing for scalable video coding, *Multimedia and Expo (ICME), 2011 IEEE International Conference on* , pp.1-4, 11-15 Jul. 2011.

Hellwagner, H., Kuschnig, R., Stutz, T. & Uhl, A. (2009). Efficient in-network adaptation of encrypted H.264/SVC content, *Journal on Signal Processing: Image Communication*, iss. 24, vol. 9, pp. 740 – 758, Jul. 2009.

ITU-T and ISO/IEC JTC 1 (1994). Generic coding of moving pictures and associated audio information – Part 2: Video, ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), Nov. 1994.

ITU-T (2000). Video coding for low bit rate communication, ITU-T Recommendation H.263, version 1: Nov. 1995, version 2: Jan. 1998, version 3: Nov. 2000.

ISO/IEC JTC 1 (2004). Coding of audio-visual objects – Part 2: Visual, ISO/IEC 14492-2 (MPEG-4 Visual), version 1: Apr. 1999, version 2: Feb. 2000, version 3: May 2004.

Kaminsky, E.; Grois, D. & Hadar, O. (2008). Dynamic Computational Complexity and Bit Allocation for Optimizing H.264/AVC Video Compression, *J. Vis. Commun. Image R.*, Elsevier, vol. 19, iss. 1, pp. 56-74, Jan. 2008.

Lin, E., Podilchuk, C. & Kalker, T. (2001). Streaming video and rate scalable compression: what are the challenges for watermarking? *In Proceedings of SPIE 4314, Security and Watermarking of Multimedia Content III*, pp. 116–127, 2001.

Meerwald, P. & Uhl, A. (2008). Toward robust watermarking of scalable video, *In Proceedings of SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, pp. 68190J ff., San Jose, CA, USA, 6819, Jan. 27 - 31, 2008

Meerwald, P. & Uhl, A. (2010). Robust watermarking of H.264/SVC-encoded video: quality and resolution scalability, In H.-J. Kim, Y. Shi, M. Barni, editors, *In Proceedings of the 9th International Workshop on Digital Watermarking, IWDW '10*, pp. 159-169, Seoul, Korea, Lecture Notes in Computer Science, 6526, Springer, October 1 - 3, 2010.

Meerwald, P. & Uhl, A. (2010). Robust watermarking of H.264-encoded video: Extension to SVC, *In Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP '10, pp. 82-85, Darmstadt, Germany, Oct. 15 - 17, 2010.

Meerwald, P. (2011). Digital Watermark Detection in Visual Multimedia Content, Ph.D. Thesis, University of Salzburg, Austria, Feb. 2011.

Nithin, T., Bull, D. & Redmill, D. (2009). A novel H.264 SVC encryption scheme for secure bit-rate transcoding. *In Proceedings of the Picture Coding Symposium, PCS'09*, Chicago, IL, USA, May 2009.

Park, S. & Shin, S. (2008). Combined scheme of encryption and watermarking in H.264/Scalable Video Coding (SVC). In New Directions in Intelligent Interactive Multimedia, Springer, Studies in Computational Intelligence, vol. 142, pp. 351—361, Sep. 2008.

Piper, A., Safavi-Naini, R. & Mertins, A. (2004). Coefficient selection methods for scalable spread spectrum watermarking, *IWDW 2003*, pp. 235-246, 2004.

Piper, A., Safavi-Naini, R. & Mertins, A. (2005). Resolution and quality scalable spread spectrum image watermarking, *In Proceedings of the 7th Workshop on Multimedia and Security, MMSEC '05*, pp. 79–90, New York, NY, USA, Aug. 2005.

Piper, A. (2010). Scalable Watermarking for Images, Ph.D. Thesis, School of Computer Science and Software Engineering, University of Wollongong, 2010.

Schierl, T., Hellge, C., Mirta, S., Gruneberg, K. & Wiegand, T. (2007). Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks, *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 3455-3458, 27-30 May 2007.

Schwarz, H.; Marpe, D. & Wiegand, T. (2007). Overview of the scalable video coding extension of the H.264/AVC standard, *IEEE Trans. Circ. Syst. for Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.

Seo, J. & Park, H. (2005). Data protection of multimedia contents using scalable digital watermarking, *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on* , pp. 376- 380, 2005.

Shi, F., Liu, S., Yao, H., Liu, Y. & Zhang, S. (2010). Scalable and Credible Video Watermarking towards Scalable Video Coding, *Advances in Multimedia Information Processing, PCM 2010*, Lecture Notes in Computer Science, vol. 6297/2010, pp. 697-708, 2010.

Stutz, T. & Uhl, A. (2008). Format-compliant encryption of H.264/AVC and SVC, *In Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'08)*, Berkeley, CA, USA, Dec. 2008.

Stutz, T. & Uhl, A. (2011), Survey of H.264 AVC/SVC Encryption, *Circuits and Systems for Video Technology, IEEE Transactions on*, vol.PP, no.99, pp. 1-15, 2011.

Wang, C., Lin, Y., Yi, S. & Chen, P. (2006). Digital authentication and verification in MPEG-4 fine-granular scalability video using bit-plane watermarking, *Proc. of Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV'06)*, pp. 16–21, Las Vegas, NV, Jun. 2006.

Wiegand, T. & Sullivan, G. (2003). Final draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 ISO/IEC 14 496-10 AVC), in Joint Video Team (JVT) of ITU-T SG16/Q15 (VCEG) and ISO/IEC JTC1/SC29/WG1, Annex C, Pattaya, Thailand, Mar. 2003, Doc. JVT-G050.

Wiegand, T.; Schwarz, H.; Joch, A.; Kossentini, F. & Sullivan, G. J. (2003). Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. Circuit Syst. Video Technol.*, vol. 13, iss. 7, pp. 688- 703, Jul. 2003.

Wiegand, T.; Sullivan, G.; Reichel, J.; Schwarz, H. & Wien, M. (2006). Joint draft 8 of SVC amendment, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 9 (JVT-U201), 21st Meeting, Hangzhou, China, Oct. 2006.

2

Perceptual Image Hashing

Azhar Hadmi[1], William Puech[1], Brahim Ait Es Said[2]
and Abdellah Ait Ouahman[2]
[1]*University of Montpellier II, CNRS UMR 5506-LIRMM*
[2]*University of Cadi Ayyad, ETRI Team*
[1]*France*
[2]*Morocco*

1. Introduction

With the fast advancement of computer, multimedia and network technologies, the amount of multimedia information that is conveyed, broadcast or browsed via digital devices has grown exponentially. Simultaneously, digital forgery and unauthorized use have reached a significant level that makes multimedia authentication and security very challenging and demanding. The ability to detect changes in multimedia data has been very important for many applications, especially for journalistic photography, medical or artwork image databases. This has spurred interest in developing more robust algorithms and techniques to allow to check safety of exchanged multimedia data confidentiality, authenticity and integrity. Confidentiality means that the exchange between encrypted multimedia data entities, which without decryption key, is unintelligible. Confidentiality is achieved mainly through encryption schemes, either secret key or public key. Authentication is an another crucial issue of multimedia data protection, it makes possible to trace the author of the multimedia data and allow to determine if an original multimedia data content was altered in any way from the time of its recording. Integrity allows degradation detection of multimedia and helps make sure that the received multimedia data has not been modified by a third party for malicious reasons. Many attempts have been noted to secure multimedia data from illegal use by different techniques fields such as encryption field, watermarking field and perceptual image hashing field. The field of encryption is becoming very important in the present era in which information security is of the utmost concern to provide end-to-end security. Multimedia data encryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication, etc. Although we may use the traditional cryptosystems to encrypt multimedia data directly, it is not a good idea for two reasons. The first reason is that the multimedia data size is almost always much great. Therefore, the traditional cryptosystems need much more time to directly encrypt the multimedia data. The other problem is that the decrypted multimedia data must be equal to the original multimedia data. However, this requirement is not necessary for image/video data. Due to the characteristic of human perception, a decrypted multimedia containing small distortion is usually acceptable. Deciding upon what level of security is needed is harder than it looks. To identify an optimal security level, the cost of the multimedia information to be protected and the cost of the protection itself are to be compared carefully. At present, many available image encryption algorithms have been proposed (Ozturk & Ibrahim, 2005;

Puech et al., 2007; Rodrigues et al., 2006). In some algorithms, the secret-key and algorithm cannot be separated effectively. This does not satisfy the requirements of the modern cryptographic mechanism and are prone to various attacks. In recent years, the image encryption has been developed to overcome above disadvantages as discussed in (Furht et al., 2004; Stinson, 2002). The other field to secure multimedia data is the watermarking field. Watermarking schemes have been developed for protecting intellectual property rights, which embed imperceptible signal, called watermark, carrying copyright information into a multimedia data *i.e.* image to form the watermarked image. The embedded watermark should be robust against malicious attacks so that it can be correctly extracted to show the ownership of the host multimedia data whenever necessary (Bender et al., 1996; Memon & Wong, 1998). A fragile or semi-fragile watermark detects changes of the host multimedia data such that it can provide some form of guarantee that the multimedia data has not been tampered with and is originated from the right source. In addition, a fragile watermarking scheme should be able to identify which portions of the watermarked multimedia data are authentic and which are corrupted; if unauthenticated portions are detected, it should be able to restore it (Cox et al., 2002). Watermarking has been widely adopted in many applications that require copyright protection, copy control, image authentication and broadcast monitoring (Cox et al., 2000). Watermarking can be used in copyright check or content authentication for individual images, but is not suitable when a large scale search is required. Furthermore, data embedding inevitably cause slight distortion to the host multimedia data (Wang & Zhang, 2007) and change its content. Recently, researchers in the field of security/authentication of multimedia data have introduced a technique inspired from the cryptographic hash functions to authenticate multimedia data called the *Perceptual hash functions* or *Perceptual image hashing* in case of image applications. It should be noted that the objective of a cryptographic hash function and a perceptual image hash function are not exactly the same. For example, there is no robustness or tamper localization requirement in case of a cryptographic hash function (Ahmed & Siyal, 2006). Traditionally, data integrity issues are addressed by cryptographic hashes or message authentication functions, such as MD5 (Rivest, 1992) and SHA series (NIST, 2008), which are sensitive to every bits of the input message. As a result, the message integrity can be validated when every bit of the message are unchanged (Menezes et al., 1996). This sensitivity to every bit is not suitable for multimedia data, since the information it carries is mostly retained even when the multimedia has undergone various content preserving operations. Therefore, bit-by-bit verification is no longer a suitable method for multimedia data authentication. A rough classification of content-preserving and content-changing manipulations is given in Table 1 (Han & Chu, 2010). Robust perceptual image hashing methods have recently been proposed as primitives to overcome the above problems and have constituted the core of a challenging developing research area to academia as well as the multimedia industry. Perceptual Image hashing functions extract certain features from image and calculate a hash value based on these features. Such functions have been proposed to establish the "perceptual equality" of image content. Image authentication is performed by comparing the hash values of the original image and the image to be authenticated. Perceptual hashes are expected to be able to survive on acceptable content-preserving manipulations and reject malicious manipulations. In recent years, there has been a growing body of research on perceptual image hashing that is increasingly receiving attention in the literature. Perceptual image hashing system generally consists of four pipeline stages: the *Transformation* stage, the *Feature extraction* stage, the *Quantization* stage and the *Compression and Encryption* stage as shown in Figure 1. The *Quantization* stage in a perceptual image hashing system is very important to enhance robustness properties and increase randomness to minimize collision probabilities in a perceptual image hashing system. This step is very difficult especially if it is followed by the *Compression and Encryption* stage because we do not know the behavior of the extracted continuous features after content-preserving/content-changing manipulations (manipulations examples are given in Table 1). For this reason, in most proposed perceptual image hashing schemes, the *Compression and Encryption* stage is ignored.

Content-preserving manipulations	Content-changing manipulations
- Transmission errors	- Removing image objects
- Noise addition	- Moving of image elements or changing their positions
- Compression and quantization	- Adding new objects
- Resolution reduction	- Changes of image characteristics: color, textures, structure, etc.
- Scaling	- Changes of the image background: day time or location
- Rotation	- Changes of light conditions: shadow manipulations etc.
- Cropping	
- γ Distortion	
- Changes of brightness hue and saturation	
- Contrast adjustment	

Table 1. Content-preserving and content-changing manipulations.

In this chapter we analyze the importance of the *Quantization* stage problem in a perceptual image hashing pipeline. This chapter is arranged as follows. In Section 2, a classification of perceptual image hashing methods is presented followed by an overview of the unifying framework for perceptual image hashing. Then, the basic metrics and important requirements of a perceptual image hashing function wherein a formulation of the perceptual image hashing problem is given. Then, perceptual hash verification measures are presented followed by an overview of recent published schemes proposed in the literature. In Section 3, we present the quantization problem in perceptual image hashing systems, then we discuss the different quantization techniques used for more robustness of a perceptual image hashing scheme where we show their advantages and their limitations. In Section 4, a new approach of analysis of the quantization stage is presented based on the theoretical study presented in Section 3 and it is followed by a presentation and discussion of some obtained experimental results. Finally, Section 5 offers a discussion on the issues addressed and identifies future research directions. The objective of the latter section is to present prospects and challenges in the context of perceptual image hashing.

2. Perceptual image hashing

In this Section, we give a classification of different perceptual image hashing techniques followed by the presentation of perceptual image hashing framework and basic requirements related to perceptual image hashing are discussed. Furthermore, related work is reviewed and the challenging problems that are not yet resolved are identified.

2.1 Perceptual image hashing methods classification

Most of the existing image hashing studies mainly focus on the feature extraction stage and use them during authentication, which can roughly be classified into the four following categories (Zhu et al., 2010), (Han & Chu, 2010):

- *Statistic-based schemes* (Khelifi & Jiang, 2010; Schneider & Chang, 1996; Venkatesan et al., 2000): This group of schemes extracts hash features by calculating the images statistics in the spacial domain, such as mean, variance, higher moments of image blocks and histogram.
- *Relation-based schemes* (Lin & Chang, 2001; Lu & Liao, 2003): This category of approaches extracts hash features by making use of some invariant relationships of the coefficients of discrete cosine transform (DCT) or wavelet transform (DWT).
- *Coarse-representation-based schemes* (Fridrich & Goljan, 2000; Kozat et al., 2004; Mihçak & R.Venkatesan, 2001; Swaminathan et al., 2006): In this category of methods, the perceptual hashes are calculated by making use of coarse information of the whole image, such as the spatial distribution of significant wavelet coefficients, the low-frequency coefficients of Fourier transform, and so on.
- *Low level feature-based schemes* (Bhattacharjee & Kutter, 1998; Monga & Evans, 2006): The hashes are extracted by detecting the salient image feature points. These methods first perform the DCT or DWT transform on the original image, and then directly make use of the coefficients to generate final hash values. However, these hash values are very sensitive to global as well as local distortions that do not cause perceptually significant changes to the images.

2.2 Perceptual image hashing framework

A perceptual image hashing system, as shown in Fig. 1, generally consists of four pipeline stages: the *Transformation* stage, the *Feature extraction* stage, the *Quantization* stage and the *Compression and Encryption* stage.

In the *Transformation* stage, the input image undergoes spacial and/or frequency transformation to make all extracted features depend the the values of image pixels or the image frequency coefficients. In the *Feature Extraction* stage, the perceptual image hashing system extracts the image features from the input image to generate the continuous hash vector. Then, the continuous perceptual hash vector is quantized into the discrete hash vector in the *Quantization* stage. The third stage converts the discrete hash vector into the binary perceptual hash string. Finally, the binary perceptual hash string is compressed and encrypted into a short and a final perceptual hash in the *Compression and Encryption* stage (Figure 1).

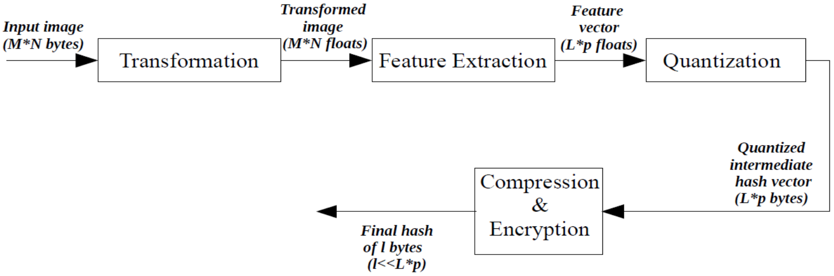


Fig. 1. Four pipeline stages of a perceptual image hashing system.

2.2.1 Transformation stage

In the *Transformation* stage, the input image of size $M \times N$ *bytes* undergoes spatial transformations such as color transformation, smoothing, affine transformations, etc. or frequency transformations such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), etc. When the DWT transformation is applied, most perceptual image hashing schemes take into account just the LL subband because it is a coarse version of the original image and contains all the perceptually information. The principal aim of those transformations is to make all extracted features, in the *Feature Extraction* stage, depend upon the values of image pixels or its frequency coefficients in the frequency space.

2.2.2 Feature Extraction stage

In the *Feature Extraction* stage, the image hashing system extracts the image features from the transformed image to generate the feature vector of L features where $L << M \times N$. Note that each feature can contain p elements of type $float$ which means that we get $L \times p$ $floats$ at this stage. It is still an open question, however, which mappings (if any) from DCT/DWT coefficients preserve the essential information about an image for hashing and/or mark embedding applications. We can at this stage add another features selection as shown in Fig. 2, so only the most pertinent features are selected which are statistically more resistant against a specific allowed manipulation like addition of noise and image rotation, etc. The selected features can be presented as an intermediate hash vector of $K \times p$ $floats$, where $K < L$.

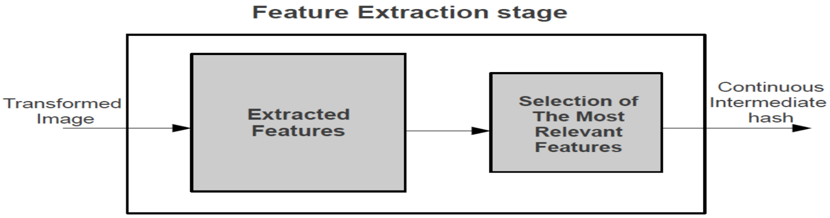


Fig. 2. Selection of the most relevant features in the Feature Extraction stage.

2.2.3 Quantization stage

In the next stage, the *Quantization* stage, we get a quantized intermediate perceptual hash vector which contains $L \times p$ elements of type *byte*. Uniform quantization can be applied to quantize each component of the continuous perceptual hash vector. Adaptive quantization (Mihçak & R.Venkatesan, 2001) is another quantization type which is is the most famous quantization scheme in the field of image hashing. The difference between the two quantization schemes is that the partition of uniform quantization is based on the interval length of the hash values, whereas the partition of adaptive quantization is based on the probability density function (pdf) of the hash values. This kind of quantization is detailed in Section 3.

2.2.4 Compression and Encryption stage

Compression and Encryption stage is the final step of a perceptual image hashing system, the binary intermediate perceptual hash string is compressed and encrypted into a short perceptual hash of fixed size of *l bytes*, where $l << L \times p$, which presents the final perceptual hash that allows image verification and authentication at the receiver. This stage can be ensured by cryptographic hash functions i.e. SHA series which generate the final hash of fixed size (hash of 160 bits in case SHA-1).

In the next section, we give the most important requirements that a perceptual image hashing must achieve and show how they conflict with each other.

2.3 Metrics and important requirements of a perceptual image hashing

Perceptual hash functions can be categorized into two categories: unkeyed perceptual hash functions and keyed perceptual hash functions. An unkeyed perceptual hash function $H(x)$ generates a hash value h from an arbitrary input x (that is $h = H(x)$). A keyed perceptual hash function generates a hash value h from an arbitrary input x and a secret key k (that is $h = H(x;k)$). The design of efficient robust perceptual image hashing techniques is a very challenging problem that should address the compromise between various conflicting requirements. Let P denote probability. Let $H()$ denote a perceptual hash function which takes one image as input and produces a binary string of length l. Let I denote a particular image and I_{ident} denote a modified version of this image which is "perceptually similar" to I. Let I_{diff} denote an image that is "perceptually different" from I. Let h_1 and h_2 denote hash values of the original image I and the perceptually different image I_{diff} from I. $\{0/1\}^l$ represents binary strings of length l. Then the four desirable properties of a perceptual image hashing function are identified as follows:

- Equal distribution (unpredictability) of hash values:

$$P(H(I) = h_1) \approx \frac{1}{2^l}, \forall h_1 \in \{0,1\}^l \tag{1}$$

- Pairwise independence for perceptually different images I and I_{diff}:

$$P(H(I) = h_1 | H(I_{diff}) = h_2) \approx P(H(I_{ident}) = h_1), \qquad \forall h_1, h_2 \in \{0,1\}^l \tag{2}$$

- Invariance for perceptually similar images I and I_{ident}:

$$P(H(I) = H(I_{ident})) \geq 1 - \theta_1, \qquad for\ a\ given\ \theta_1 \approx 0 \tag{3}$$

- Distinction of perceptually different images I and I_{diff}:

$$P(H(I) \neq H(I_{diff})) \geq 1 - \theta_2, \qquad for\ a\ given\ \theta_2 \approx 0 \tag{4}$$

To meet property in equation (3), most perceptual hash functions try to extract features of images which are invariant under insignificant global modifications such as compression or enhancement. Equation (4) means that, given an image I, it should be nearly impossible for an adversary to construct a perceptually different image I_{diff} such that $H(I) = H(I_{diff})$. This property can be hard to achieve because the features used by published perceptual hash functions are publicly known (Kerckhoffs, 1883; Mihçak & R.Venkatesan, 2001). Also, it makes property in equation (3) be neglected in favor of property in equation (4). Likewise for perfect unpredictability, an equal distribution (equation (1)) of the hash values is needed. This would deter achieving the property in equation (3) (Monga, 2005). Depending on the application, perceptual hash functions have to achieve these conflicting properties to some extent and/or facilitate trade-offs. From a practical point of view, both robustness and security are important. Lack of robustness (equation (3)) renders an image hash useless as explained above, while security (equations (1),(4)) means that it is extremely difficult for an adversary to modify the essential content of an image yet keep the hash value unchanged. Thus, trade-offs must be sought, and this usually forms the central issue of perceptual image hashing research.

2.4 Perceptual hash verification

Perceptual image hashing system calculates hashes for similar images that must be equal. Referring to the image space as shown in Figure 3, let I denote an image, and X denote the set of images I_{ident} that are modified from I by means of content-preserving manipulations and are defined to be perceptually similar to I. Let Y contains all other images I_{diff} that are irrelevant to I and its perceptually similar versions. I_{diff} are the results of content-changing manipulations. Consequently, $\{I\} \cup X \cup Y$ forms an entire image space. Let h, h_{ident} and h_{diff} denote hash values of the original image I, the perceptually similar image I_{ident} from I and the perceptually different image I_{diff} from I respectively. In robust and secure perceptual image, the following properties are required when Encryption and Compression stage is applied in a perceptual image hashing system: $\boldsymbol{h} = \boldsymbol{h}_{ident}$ **for all identical images** $\boldsymbol{I}_{ident} \in \boldsymbol{X}$ and $\boldsymbol{h} \neq \boldsymbol{h}_{ident}$ **for all different images** $\boldsymbol{I}_{diff} \in \boldsymbol{Y}$ (Figure 3). Since the requirement of bit-by-bit hashes equality is usually hard to achieve, most of the proposed schemes compute distances and similarities between perceptual hashes. The most often used are the Bit Error Rate (BER), the Hamming distance and the Peak of Cross Correlation (PCC). The first two measure the distance between two hash values, whereas the latter measures the similarity between two hash values. Using theses measures, the sender determines the threshold τ. The proper selection of τ is very important as it defines the boundary between content-preserving and content-changing manipulations.

Let $d(.,.)$ indicates the used measure *i.e.* a normalized Hamming distance function. Let h, h_{ident} and h_{diff} denote hash values of the original image I, the perceptually similar image I_{ident} from I and the perceptually different image I_{diff} from I respectively. The error-resilience of multimedia data hashing is defined as follows. I_{ident} is successfully identified to be perceptually similar to I if $d(h, h_{ident}) \leqslant \tau$ holds. In other words if two images are perceptually similar, their corresponding hashes need to be highly correlated. If $d(h, h_{diff}) \gg \tau$, then I_{diff} is identified as modified from I by means of content-changing manipulations. Overall, the main theme of perceptual image hashing is to develop a robust perceptual image hash function that can identify perceptually similar multimedia contents and reject content-changing manipulations.

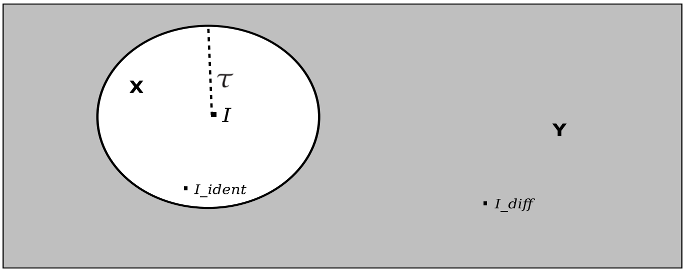


Fig. 3. The image space $\{I\} \cup X \cup Y$ formed by an image $\{I\}$, its perceptually similar versions set X and its modified version set Y.

2.5 Review of some related work on perceptual image hashing techniques

In recent years, there has been a growing body of research on perceptual image hashing that is increasingly receiving attention in the literature. Most of these existing papers focus on studies of the feature extraction stage because they believe that extracting a set of robust features that resist, and to stay relatively constant, content-preserving manipulations and at the same time should detect content-changing manipulations is the most important objective in perceptual image hashing system. Few papers address perceptual image hashing system security. In (Fridrich, 2000), the extraction of the hash is based on the projection of image coefficients onto filtered pseudo-random patterns. The final perceptual hash is used for generating a pseudo-random watermark sequences, that depend sensitively on a secret key yet continuously on the image, for authentication and integrity verification of still images. In (Venkatesan et al., 2000), a perceptual image hashing technique based on statistics computed from randomized rectangles in the discrete wavelet domain (DWT) is presented. Averages or variances of the rectangles are then calculated and quantized with randomized rounding to obtain the hash in the form of a binary string. The quantized statistics are then sent to an error-correcting decoder to generate the final hash value. Statistical properties of wavelet subbands are generally robust against attacks, but they are only loosely related to the image contents therefore rather insensitive to tampering. This method has been shown to be robust against common image manipulations and geometric attacks. The proposed method in (Schneider & Chang, 1996) is using the intensity histogram to sign the image. Since the global histogram does not contain any spatial information, the authors divide the image into blocks, which can have variable sizes, and compute the intensity histogram for each block separately. This allows some spatial information to be incorporated into the signature. The method in (Fridrich & Goljan, 2000) is based on the observation of the low frequency DCT coefficient. If a low frequency DCT coefficient of an image is small in absolute value, it cannot be made large without causing visible changes to the image. Similarly, if the absolute value of a low frequency coefficient is large, it cannot change it to a small value without influencing the image significantly. To make the procedure dependent on a key, the DCT modes are replaced with DC-free random smooth patterns generated from a secret key. Other researchers have used others techniques to perform image perceptual hashing. Authors in (Swaminathan et al., 2006) used Fourier-Mellin transform for perceptual image hashing applications. Using Fourier-Mellin transform's scale invariant property, the magnitudes of the Fourier transform coefficients were randomly weighted and summed. However, since Fourier transform did not offer localized frequency information, this method was not able to detect malicious local modifications. In a more recent development, a perceptual image hashing scheme based Radon Transform is proposed in (Lei et al., 2011) where the authors perform Radon Transform on the image and calculate the moment features which are invariant to translation and scaling in the projection space. Then Discrete Fourier Transform (DFT) is applied on the moment features to resist rotation. Finally, the magnitude of the significant DFT coefficients is normalized and quantized as the final perceptual image hash. The proposed method can tolerate almost all the typical image processing manipulations, including JPEG compression, geometric distortion, blur, addition of noise and enhancement. The Radon transform was first used in (Lefebvre et al., 2002), and further expanded in (Seo et al., 2004). Authors in (Guo & Hatzinakos, 2007) propose a perceptual image hashing scheme based on the combination of discrete wavelet transform (DWT) and the Radon Transform. Taking the advantages of the frequency localization property of DWT and shift/rotation invariant property of the Radon transform, the algorithm can effectively detect malicious local changes, and at the same time, be robust against content-preserving modifications. Obtained features derived from the Radon Transform are then quantized by the probabilistic quantization (Mihçak & Venkatesan, 2001) to form the final perceptual hash.

In this Section, we have presented some reviews of different schemes proposed in the field of perceptual image hashing. In Section 3, we develop the quantization problem in perceptual image hashing and we present some approaches to address this problem which surely have limitations in practice.

3. Quantization problem in perceptual image hashing

3.1 Problem statement

The goal of the quantization stage, in the perceptual image hashing system, is to discretize the continuous intermediate hash vector (continuous features) into a discrete intermediate hash vector (discrete features). This step is very important to enhance robustness properties and increase randomness to minimize collision probabilities of a perceptual image hashing system. Quantization is the conventional way to achieve this goal. The quantization step is difficult because we do not know how the values in the continuous intermediate hash drop after content-preserving (non-malicious) manipulations in each quantization interval Q. This difficulty of an efficient quantization increases more when it is followed by an encryption and compression stage *i.e.* SHA-1, because the discrete intermediate hash vectors must be quantized in a correct way for all perceptual similar images. For this reason this stage is ignored in most schemes presented in the literature. To understand the quantization problem statement, let us suppose that the incidental distortion introduced by content-preserving manipulations can be modeled as noise whose maximum absolute magnitude is denoted as B, which means that the maximum range of additive noise is B. Suppose that the original scalar value $x_l \in \mathbb{R}$ for $l \in \{1, ..., L\}$ of the continuous intermediate hash is bounded to a finite interval $[-A, A]$. Furthermore, suppose that we wish to obtain a quantized message $q(x_l)$ of x_l in P quantization points given by the set $\tau = \{\tau_1, ..., \tau_P\}$. The points are uniformly spaced such that $Q = \tau_j - \tau_{j-1} = 2A/(P-1)$ for $j \in \{1, ..., P\}$. Now suppose $x_l \in [\tau_j, \tau_{j+1})$, then it will be quantized as τ_j. However, when this value is corrupted after noise addition, the distorted value could drop in the previous quantization interval $[\tau_{j-1}, \tau_j)$ or in the next interval $[\tau_{j+1}, \tau_{j+2})$ and it will be quantized as τ_{j-1} or τ_{j+1}, respectively, and the quantized x_l value will not remain unchanged as τ_j before and after noise addition. Thus, the noise corruption will cause a different quantization result and automatically cause different perceptual hashes (Hadmi et al., 2010). Figure 4 shows the distribution of the original DWT

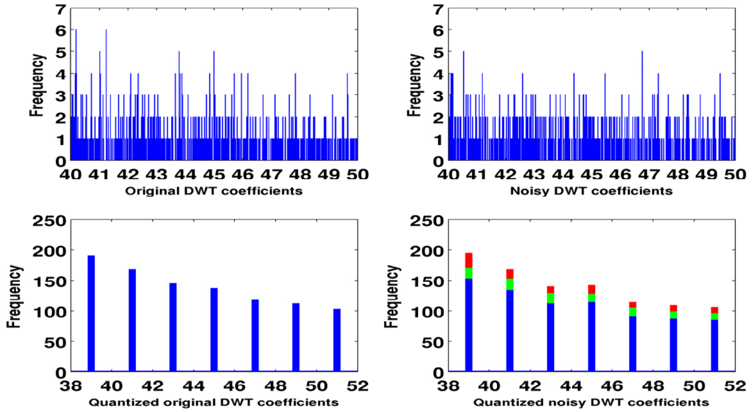


Fig. 4. The influence of additive Gaussian noise on the quantization ($Q = 2$) of the original DWT LL-subband coefficients and their noisy version in the interval $[40, 50]$. In green: DWT LL-subband quantized coefficients that dropped from the right neighboring quantization interval. In red: DWT LL-subband quantized coefficients that dropped from the left neighboring quantization interval.

LL-subband (level 3) coefficients, of Lena image sized 1024×1024, in the interval $[40, 50]$ and their noisy version, in the same interval $[40, 50]$, by an additive Gaussian noise of standard deviation equals $\sigma = 1$. When applying a Gaussian noise with $\sigma = 1$, the noisy image remains visually the same than the original image however it causes changes on extracted features distribution as we can see in Figure 4. This causes errors in the quantization step because the quantized features do not remain unchanged after noise addition as shown in Figure 4. To avoid such cases, many quantization schemes have been proposed in the literature. Authors in (Sun & Chang, 2005) proposes an error correction coding (ECC) to correct errors of extracted features caused by corruption from additive noise to get the same quantization result before and after additive noise. In their work, they assume that the quantization step $Q > 4B$, which is not always true at the practical point of view, and they push the points away from the quantization decision boundaries and create a margin of at least $Q/4$ so that original x_l value when later contaminated will not exceed the quantization decision boundaries. The illustration of the concept of error correction is illustrated in Figure 5. The original feature P is quantized in nQ before adding noise, but after adding noise there is also a possibility that the noisy feature value could drop at the range $[(n-1)Q, (n-0.5)Q)[$ and will quantized as $(n-1)Q$. As a solution to this, Authors propose to add or subtract $0.25Q$ to remain the features at the range $[(n-0.5)Q, (n+0.5)Q)]$ and then remain the quantized value the same as the original quantized value nQ even after adding noise.

Other similar work based on this approach has recently been proposed (Ahmed et al., 2010) where the authors calculate and record a vector of 4-bits called “Perturbation information”. This additional transmitted information has the same dimension of the extracted features. It is used at the receiver’s end to adjust the intermediate hash during the image verification stage before performing quantization. Therefore, the information carried in the “Perturbation information” helps to make a decision to positively authenticate an image or not. Their theoretical analysis is more general than in (Sun & Chang, 2005) from a practical point of view. One main disadvantage of such schemes is that vectors used to correct errors of extracted

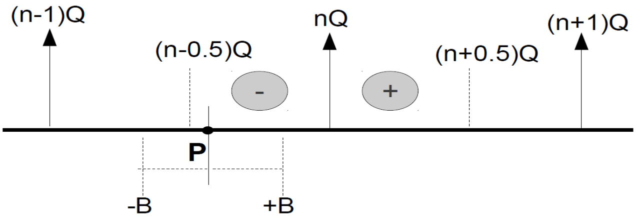


Fig. 5. Illustration on the concept of error correction in Sun's scheme (Sun & Chang, 2005).

features need to be transmitted or stored beside the image and the final hash as shown in Figures 6 and 7.

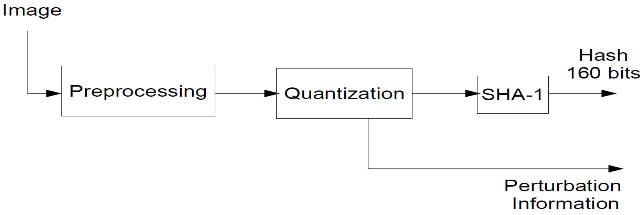


Fig. 6. Hash generation module with quantization in Fawad's scheme (Ahmed et al., 2010).

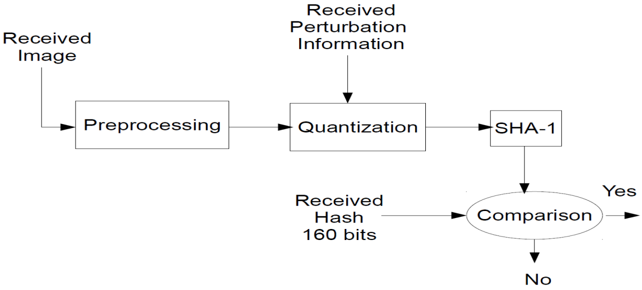


Fig. 7. Image verification module with quantization in Fawad's scheme (Ahmed et al., 2010).

Another quantization scheme which is widely applied in perceptual image hashing (Swaminathan et al., 2006), (Zhu et al., 2010) proposed by (Mihçak & Venkatesan, 2001) called *Adaptive Quantization* or *Probabilistic Quantization* in (Monga, 2005). Its property is that it takes into account to the distribution of the input data. The quantization intervals $Q = \tau_j - \tau_{j-1}$ for $j \in \{1, ..., P\}$ are designed so that $\int_{\tau_{j-1}}^{\tau_j} p_X(x)\,dx = 1/P$, where P is the number of quantization levels and $p_X(.)$ is the pdf of the input data X. The central points $\{C_j\}$ are defined so as to make $\int_{\tau_{j-1}}^{C_j} p_X(x)\,dx = \int_{C_j}^{\tau_j} p_X(x)\,dx = 1/(2P)$. Around each τ_j, a randomization interval $[A_j, B_j]$ is introduced such that $\int_{A_j}^{\tau_j} p_X(x)\,dx = \int_{\tau_j}^{B_j} p_X(x)\,dx = r/P$, where $r \leq 1/2$. The randomization interval is symmetric around τ_j for all j in terms of distribution p_X. The natural constraint must be respected $C_j \leq A_j$ and $B_j \leq C_{j+1}$. The overall quantization rule is then given by:

$$q(x_l) = \begin{cases} j-1 & \text{w.p.} \quad 1 & \text{if } C_j \leq x_l < A_j, \\ j-1 & \text{w.p.} \quad \left(\frac{P}{2r}\int_{x_l}^{B_j} p_X(t)\,dt\right) & \text{if } A_j \leq x_l < B_j, \\ j & \text{w.p.} \quad \left(\frac{P}{2r}\int_{A_j}^{x_l} p_X(t)\,dt\right) & \text{if } A_j \leq x_l < B_j, \\ j & \text{w.p.} \quad 1 & \text{if } B_j \leq x_l < C_{j+1}. \end{cases} \tag{5}$$

where w.p. stands for “with probability”.

The discrete scheme of *Adaptive Quantization* has recently been developed by (Zhu et al., 2010) to make it applicable in practice.

3.2 Theoretical analysis

In this section, we analyze statically the behavior of the extracted features under additive uniform noise, Section 3.2.1 and Gaussian noise, Section 3.2.2, as well as the probability of a false quantization for these selected features. The main goal of this analysis is to give a theoretical behavior of the extracted image features to be hashed against content-preserving /content-changing manipulations, that are simulated by an additive noise, that may undergo an image (Hadmi et al., 2011).

3.2.1 Case of an additive uniform noise

To analyze the influence of an additive noise on perceptual image hashing robustness, we have decided to lead a statical analysis of the quantization problem. The idea is to compute the length of the quantization interval Q for a noise whose maximum absolute magnitude is B, which represents the content-preserving manipulations, and a previously fixed probability that a value in this interval drops out, that is denoted as P_{drop}.

To address this problem, we have started by developing the convolution product between two distributions defined as follows:

- Let $P_\rho(x)$ denote the extracted feature distribution limited to an interval $[a,b]$ of length $\rho = b - a$. $P_\rho(x)$ is given by:

$$P_\rho(x) = \begin{cases} \frac{1}{\rho} & \text{for } x \in [a,b], \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

- Let $P_B(x)$ denote the probability density function of the continuous uniform noise, which presents content-preserving manipulations, in the interval $B = [-\frac{B}{2}, \frac{B}{2}]$, with $B < \rho$. $P_B(x)$ is expressed as:

$$P_B(x) = \begin{cases} \frac{1}{B} & \text{for } x \in [-\frac{B}{2}, \frac{B}{2}], \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The convolution product $h(x)$ of $P_\rho(x)$ by $P_B(x)$ is:

$$h(x) = \int_{-\infty}^{+\infty} P_\rho(y) P_B(x-y)\, dy = \int_a^b \frac{1}{\rho} P_B(x-y)\, dy \tag{8}$$

Finally, we get the convolution product $h(x)$ (equation (9)) expressed as:

$$h(x) = \begin{cases} 0 & \text{for } x \leq a - \frac{B}{2}, \\ \frac{1}{\rho B}\left(x + \frac{B}{2} - a\right) & \text{for } x \in \left]a - \frac{B}{2}, a + \frac{B}{2}\right], \\ \frac{1}{\rho} & \text{for } x \in \left]a + \frac{B}{2}, b - \frac{B}{2}\right], \\ \frac{1}{\rho B}\left(-x + \frac{B}{2} + b\right) & \text{for } x \in \left]b - \frac{B}{2}, b + \frac{B}{2}\right], \\ 0 & \text{for } x > b + \frac{B}{2}. \end{cases} \tag{9}$$

An example of $h(x)$ is presented in Figure 8, with $B < \frac{\rho}{2}$.

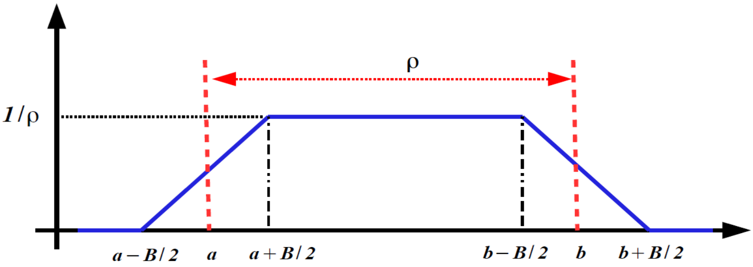


Fig. 8. Convolution product of $P_\rho(x)$ by $P_B(x)$.

Suppose that y presents an extracted feature which is in the interval $[a, b]$ and let P_{drop} be the probability that y drops out from $[a, b]$ because of the adding noise B. Thus, $P_{drop}(y)$ is calculated and expressed as follows (Equation 10):

$$\begin{aligned} P_{drop}(y) &= P(y \notin [a,b]) \\ &= \int_{a-\frac{B}{2}}^{a} h(x)\, dx + \int_{b}^{\pm\frac{B}{2}} h(x)\, dx \\ &= \frac{B}{4\rho} \end{aligned} \tag{10}$$

Equation (10) allows us to get an information of the extracted features behavior after adding noise. For example, for a uniform noise of length $B = 4.10^{-2}$, if we want to have $P_{drop} = 10^{-3}$, then the length of the quantization interval ρ that must be chosen is: $\rho = 10$.

To make a comparison between the theoretical probability that extracted features drop out from the quantization interval given by Equation 10 and the experimental probability, we

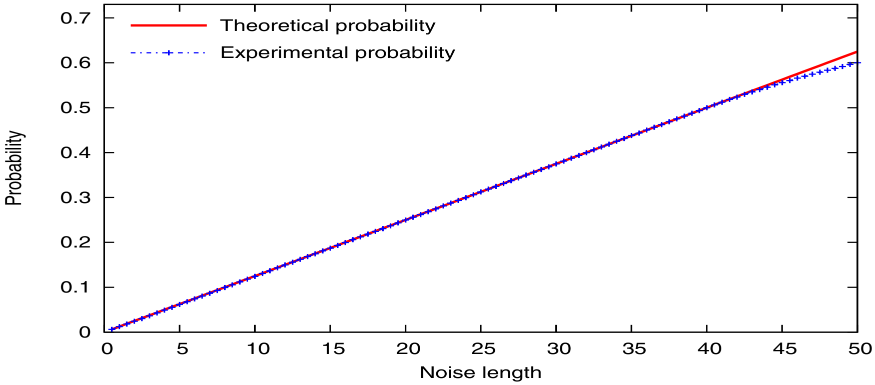


Fig. 9. Comparison between the theoretical and the experimental probabilities that extracted features drop out from the quantization interval for various noise lengths.

applied continuous uniform noise of different lengths from $B = 0$ to $B = 50$ on the same $N = 10000$ samples in the interval $\Delta = [-10, 10]$, and then we calculated the probability P_{drop} for each noise length. We note that the experimental results presented in Figure 9 coincide with the theoretical results calculated from Equation 10 for all noise lengths until $B = 44$. Some divergences are observed after this noise length which can be considered as content-changing (malicious) manipulations.

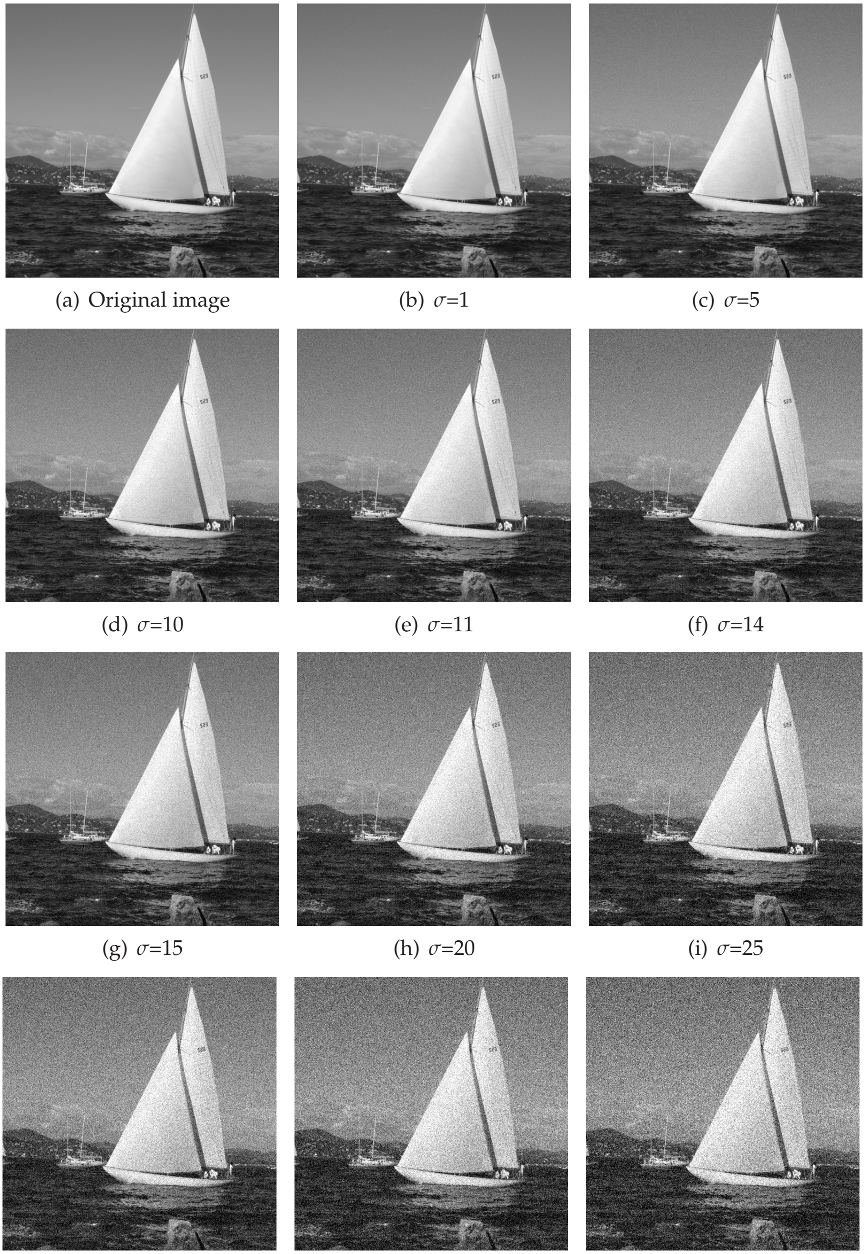
The same analysis can be performed for other noise distributions such as Gaussian distribution or triangular distribution. Thus, by just modeling the content-preserving manipulations by the aforementioned distributions, we can precisely obtain the probability from which the extracted features will drop from a fixed quantization interval to its neighboring intervals. Alternately, we can beforehand fix the maximum range of additive noise that we judge to be a content-preserving manipulation and the probability that extracted features change of quantization interval. This will allow us to fix the length of the appropriate quantization interval which respects to this probability.

3.2.2 Case of an additive Gaussian noise

Figure 10 shows an example of an original image of size 512 × 512 and their noisy versions with many levels of additive Gaussian noise controlled by its standard deviation σ. Note that the applied additive Gaussian noise is 0-mean, and changing its standard deviation σ allows us to increase or decrease its level.

To evaluate the perceptual similarity between the original and their modified versions, we can based on the perceptual aspect provided by the Human Visual System (HVS), on the method of the Structural SIMilarity (SSIM)[1] (Wang et al., 2004), or on the method of Peak Signal to Noise Ratio (PSNR). Table 2 gives the SSIM and PSNR values for noisy images obtained by applying different standard deviation values σ of the additive Gaussian noise. The quality of the Gaussian noisy images is compared to the original image and they are classified into four

[1] SSIM is a classical measure well correlated to the Human Visual System.The SSIM values are real positive numbers lower or equal to 1. Stronger is the degradation and lower is the SSIM measure. A SSIM value of 1 means that the image is not degraded.



(a) Original image (b) σ=1 (c) σ=5

(d) σ=10 (e) σ=11 (f) σ=14

(g) σ=15 (h) σ=20 (i) σ=25

(j) σ=30 (k) σ=35 (l) σ=40

Fig. 10. Original image and their noisy versions with different additive Gaussian noise parametrized with different standard deviations σ.

categories: very similar, similar, different and very different. The changed images qualified very similar and similar (Figures 10(b), 10(c), 10(d), 10(e)) must have the same perceptual hash of the original image noted by I_{ident}. Other cases of images *i.e.* images qualified as different or very different Figures (10(f), 10(g), 10(h), 10(i), 10(j), 10(k), 10(l)) from the original image must have a different perceptual hash noted by I_{diff} as presented in Table 2.

Standard deviation σ	SSIM	PSNR (dB)	Image quality	Perceptual hash
1	0.997	47.79	very Similar	I_{ident}
5	0.946	34.15	Similar	I_{ident}
10	0.828	28.16	Similar	I_{ident}
11	0.802	27.32	Similar	I_{ident}
14	0.728	25.25	Different	I_{diff}
15	0.704	24.70	Different	I_{diff}
20	0.600	22.24	Different	I_{diff}
25	0.517	20.36	Different	I_{diff}
30	0.450	18.86	very Different	I_{diff}
35	0.397	17.59	very Different	I_{diff}
40	0.354	16.50	very Different	I_{diff}

Table 2. SSIM and PSNR values for noisy images obtained by applying different standard deviation values σ of the additive Gaussian noise.

In the case of σ=1, the noisy image remains visually the same as the original image and it has high values of SSIM ($SSIM = 0.997$) and PSNR ($PSNR = 47.79$). For σ=5, σ=10 and σ=11, the changes in the noisy images are very small and we can consider that the noisy images are still similar to the original image. In the case of $\sigma = 5, 10, 11$, the SSIM values remain smaller than 80% and the PSNR values remain larger than $27db$. When the level of the additive Gaussian noise increases, the noisy images are perceptually different from the original image as it is shown in Figure 10 for σ=14,. . . ,40 and both the SSIM and PSNR values degrade. We can fix the threshold of the additive Gaussian noise that holds a good content in the sense of human perception fixed at σ=11 as it is justified in term of the SSIM and PSNR values. We fixed the degradation to a SSIM value of 80% and the PSNR value at $27db$ to consider a noisy image similar to the original image. The threshold of the SSIM and PSNR values is justified in terms of the subjective measure based on the HVS for many tests that we have done for a large database of grayscale images as we can see in Figure 10.

To address theoretically the influence of an additive Gaussian noise whose 0-mean and standard deviation σ on an uniform distribution of features limited in an interval $[a, b]$, we compute the convolution product between the distribution of the extracted features and the distribution of the additive Gaussian noise defined as follows:

- Let $P_\rho(x)$ denote the extracted feature distribution limited to an interval $[a, b]$ of length $\rho = b$ - a. $P_\rho(x)$ is given by:

$$P_\rho(x) = \begin{cases} \frac{1}{\rho} & \text{for } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

- Let $P_\sigma(x)$ denote the probability density function of the Gaussian noise whose 0-mean and standard deviation σ, which presents content-preserving manipulations. $P_\sigma(x)$ is expressed as:

$$P_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{12}$$

The convolution product $h(x)$ of $P_\rho(x)$ by $P_\sigma(x)$ is:

$$\begin{aligned}
h(x) &= \int_{-\infty}^{+\infty} P_\rho(y) P_\sigma(x-y)\, dy \\
&= \frac{1}{\rho}\Big(\int_{-\infty}^{x-a} P_\sigma(y)\, dy - \int_{-\infty}^{x-b} P_\sigma(y)\, dy\Big) \\
&= \frac{1}{\rho}\Big(\int_{-\infty}^{x-a} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}\, dy - \int_{-\infty}^{x-b} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}\, dy\Big) \\
&= \frac{1}{\rho}\Big(\int_{-\infty}^{\frac{x-a}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy - \int_{-\infty}^{\frac{x-b}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy\Big) \\
&= \frac{1}{2\rho}\Big[erf\Big(\frac{x-a}{\sqrt{2}\sigma}\Big) - erf\Big(\frac{x-b}{\sqrt{2}\sigma}\Big)\Big]
\end{aligned} \tag{13}$$

with $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$.

The convolution product $h(x)$ models the behavior of the original features after adding the Gaussian noise in each quantization interval. Figure 11 shows a normalized uniform distribution of 10000 features belonging in the interval $[10, 20]$ before and after the quantization stage where the quantization step Q=10. All these features are quantized to the value 15 as shown in Figure 11. Figure 12 presents the normalized distribution of the noisy features after adding a Gaussian noise with 0-mean and standard deviation σ=2. This distribution coincides exactly with the theoretical results given by Equation 13. As shown in Figure 12, the noisy features are quantized and spread in 3 quantization intervals and are quantized to three values: 5, 15 and 25. The 5 quantized value presents the quantized value to the left neighbor quantization interval and the 25 presents the quantized value to the right neighbor quantization interval. Statistically, for the same experiment settings we have 8% of features drop to the left neighbor quantization interval and 8% of features drop to the right neighbor quantization interval. For the other experiments settings, we always have a symmetric percentage of features drop in the left and right neighbor quantization interval.

4. Experimental results

4.1 Experimental analysis protocol

In this section, we describe the quantization analysis protocol for perceptual image hashing based on statistical invariance of extracted block mean features. The aim is to find agreement between the density of the additive Gaussian noise, the size of the image block and the quantization step size that must be taken to ensure a good level of image hashing robustness. As shown in Figure 13, the original input image I of size $N \times M$ pixels is split to non

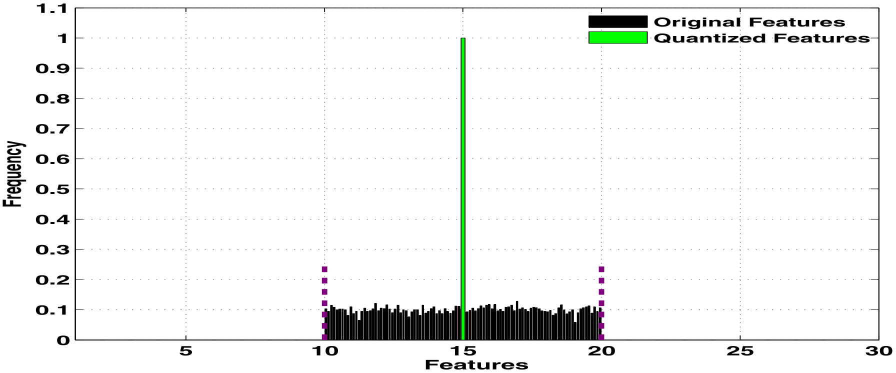


Fig. 11. 10000 original features uniformly distributed in one quantization interval $[10, 20]$ before quantization (black) and after uniform quantization (green) where the quantization step Q=10.

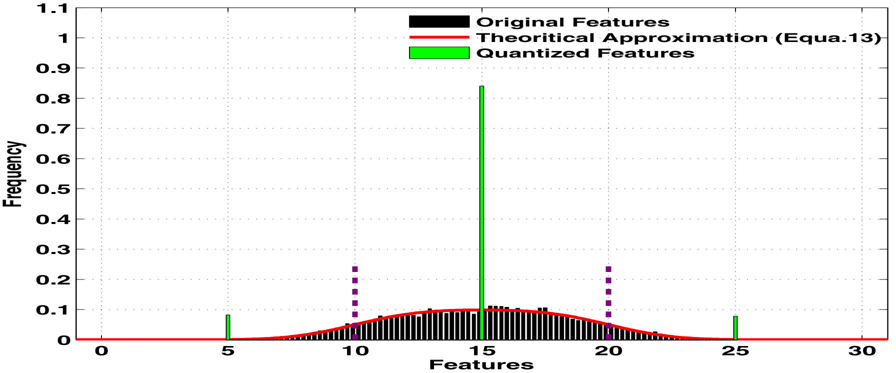


Fig. 12. 10000 noisy features after adding Gaussian noise whose 0-mean and standard deviation $\sigma = 2$ before quantization (black) and after uniform quantization (green) where the quantization step Q=10.

overlapping blocks of size $q \times p$ pixels that we note by $B_{i,j}$, where $i \in \{1, 2, \ldots, \frac{N}{q}\}$ and $j \in \{1, 2, \ldots, \frac{M}{p}\}$. The float mean value $m_{i,j}$ of each block $B_{i,j}$ is computed and stored in a one dimensional vector that we note by $\boldsymbol{V_m(k)}$, where $k \in \{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$. Quantization step is the conventional way to descretize the continuous vector V_m. For a given quantization size step Q, the quantized vector $\boldsymbol{V'_m(k)}$ of $\boldsymbol{V_m(k)}$ is given by the floor operation:

$$V'_m(k) = \lfloor \frac{V_m(k)}{Q} \rfloor \times Q + \frac{Q}{2} \tag{14}$$

where k=$\{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$.

The distribution $Dist_I$ of the quantized vector $\boldsymbol{V'_m}$ is then calculated and stored as a reference enabling us to make a comparison with distributions of other candidate images for verification of their integrity with the original image.

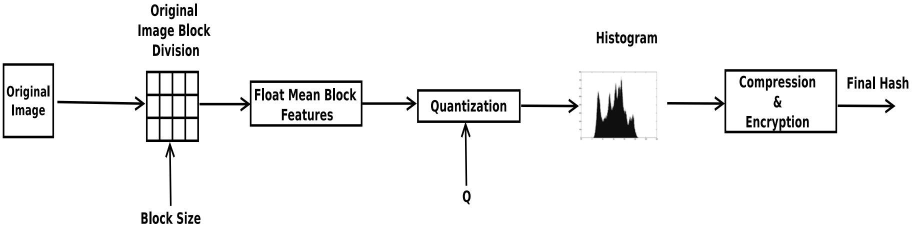


Fig. 13. Proposed quantization analysis protocol for perceptual image hashing based image block mean.

The image hashing system assumes that the original image I may be sent over a network consisting of possibly untrusted nodes. During the untrusted communication the original image could be manipulated for malicious purposes. Therefor, the received image $\bar{I}$ may undergo non-malicious operations like JPEG compression, etc. or malicious tampering. The final perceptual hash of I should be used to authenticate its received version $\bar{I}$. In the case of non-malicious operations, the original feature vector and the received one should differ by a small Euclidean distance which makes quantization control easier, and by a large Euclidean distance in the case of content-changing manipulations. This allows to have different results after the quantization step. Note, that even if the feature vector undergo small changes under small additive noise may cause false authentication of the received image $\bar{I}$ where it has to be considered similar to I. The received image $\bar{I}$, that we simulate like the original image plus a Gaussian noise with 0-mean and a standard deviation σ, will undergo the same steps than the original image (Fig.13) which allows to get the distribution $Dist_{\bar{I}}$ of $\boldsymbol{\bar{V}'_m(k)}$. Let $\boldsymbol{V_m(k)}$ be the mean of an original image block of size $q \times p$ pixels noted by $p_{i,j}$. By the same way, we note by $\boldsymbol{\bar{V}'_m(k)}$ the mean of noisy image block noted by $p'_{i,j}$. $\boldsymbol{\bar{V}'_m(k)}$ can be expressed as function of $\boldsymbol{V_m(k)}$ as follow:

$$\begin{aligned} \bar{V}'_m(k) &= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} p'_{i,j} \\ &= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} (p_{i,j} + n_{i,j}) \\ &= \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} p_{i,j} + \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} n_{i,j} \\ &= V_m(k) + \frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} n_{i,j} \end{aligned} \tag{15}$$

where $n_{i,j}$ is a Gaussian noise belongs to $\mathcal{N}_{0,\sigma}$ and $k \in \{1, 2, \ldots, \frac{N}{q} \times \frac{M}{q}\}$.

The term "$\frac{1}{p \times q} \sum_{i=1}^{p} \sum_{j=1}^{q} n_{i,j}$" in Equation 15 belongs to Gaussian distribution with 0-mean and standard deviation $\frac{\sigma}{\sqrt{p \times q}}$.

$\bar{\boldsymbol{V}}_{\boldsymbol{m}}'$ is the discrete vector which contains the quantized values of the computed means of the received image blocks. The comparison between $Dist_I$ and $Dist_{\bar{I}}$ allows us to get the information about the percentage of stable features that stayed fix after the additive Gaussian noise, the percentage of the features that moved to the left neighbor quantization interval and the percentage of the features that moved to the right neighbor quantization interval. This information of the features behavior is very useful, it allows us to take into account the percentage of the stable features that resist to non-malicious operations, simulated by an additive Gaussian noise. Also, it allows us to control the parameters of blocks size division and quantization step size to achieve an aimed level of the image hashing system robustness against a given level of additive noise. Selected features will then be hashed in the step of "Compression and Encryption" as shown in Figure 1. The "Compression and Encryption" stage is achieved by the cryptographic hash function SHA-1 generating a final hash of 160-bits with height level of security.

4.2 Experimental analysis of the quantization problem in a perceptual image hashing system

In the experiments of the proposed scheme, the features are the means of different image block sizes. The computed image block are sized: 4 × 4, 8 × 8 and 16 × 16. Then after, they are quantized by different quantization step sizes: Q=1, Q=4 and Q=16. In other words, for each given quantization step size, we tested different image block sizes against different levels of the additive Gaussian noise. The experiments are tested for a large database of grayscale images of size 512 × 512. Figure 3 shows the variation of mean distribution for different image block sizes and different levels of additive Gaussian noise in the case of quantization step size $Q = 4$ applied for the image Figure 10(a). In the case of the quantization step size Q=4 and standard deviation σ=1 (Figure 10(b)) (Table 3), we observe that unstable mean block features decrease when we increase the block size. We note also that the percent of stable mean block features is significant even in the case of block size equals to 4 × 4 (Table 4). When the standard deviation in the additive Gaussian noise increase (case of σ=5 shown in Table 3) while keeping the visual contents of the noisy image the same as the original image 10(a), the percentage of the stable mean block features decrease compared to the case of σ=1. When the visual contents of the noisy/attacked image changes Figure 10(l) than the original one (case of σ=40), we observe that a little of mean block features remain stable for all the block size that we tested as shown in Table 3.

The obtained numerical results in Table 4 present the percentage of features that have not moved and remain stable under different additive Gaussian noise and also those that drop from the left neighbor quantization interval or from the right neighbor quantization interval for each size of image block. As we can observe in Table 4, the percentage of stable features that remain fixed after adding Gaussian noise decreases when the level of the noise increases. For the same level of noise, the percentage of stable features increase when the the image block size increase. Thus, if we set the quantization step size to $Q = 1$, we can take into account the percentage of stable features that resist against tolerable level of the additive Gaussian noise. For example, if we fix the quantization step size Q equals to value 1 and we

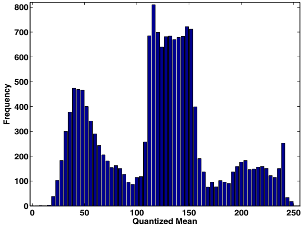
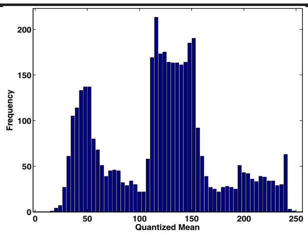
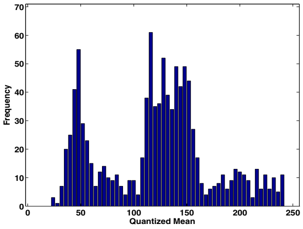
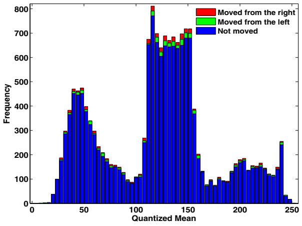
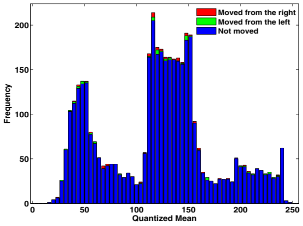
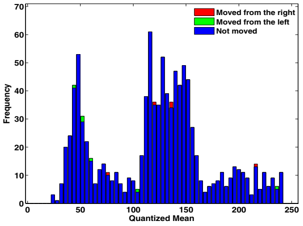
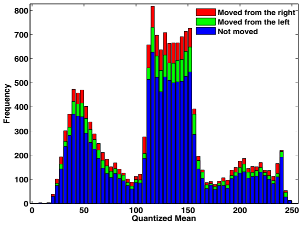
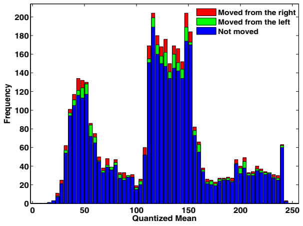
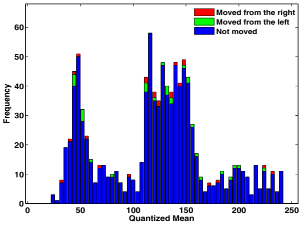
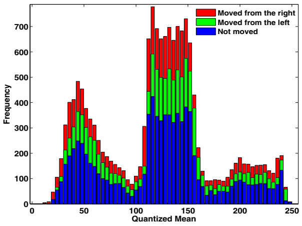
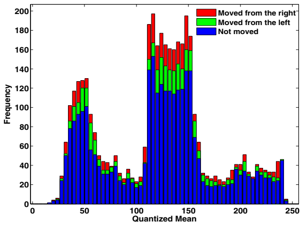
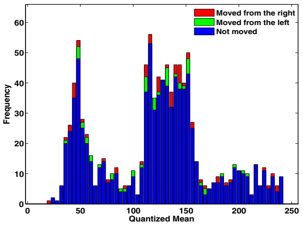
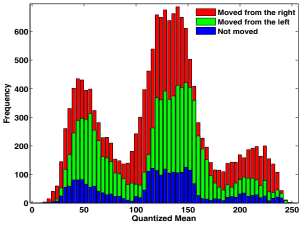
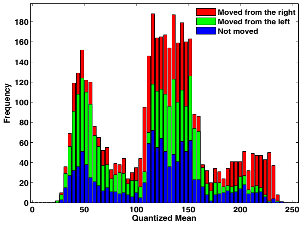
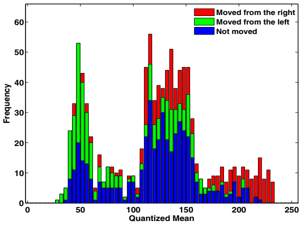
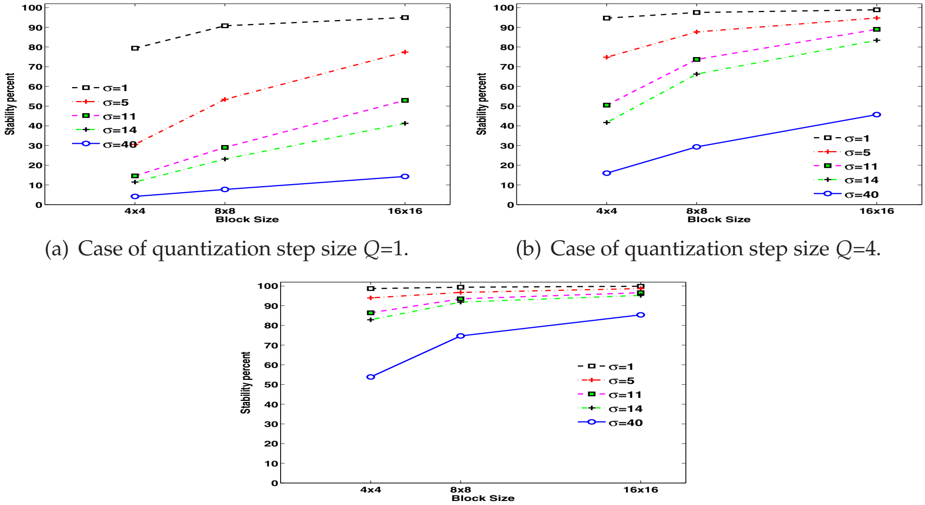
σ	Block size=4×4	Block size=8×8	Block size=16×16
0			
1			
5			
11			
40			

Table 3. Variation of mean distribution for different image block sizes and different levels of additive Gaussian noise in the case of quantization step size $Q = 4$.

consider that an image which undergos a tolerable manipulations equivalent to an additive Gaussian noise whose a standard deviation equal to $\sigma = 5$, we choose a compromise between the percentage of stable features and the size of the blocks image decomposition. For the block size equal 4 × 4 we have to take into account the maximum percent of stable features ≈ 30% and if the block size equals 8 × 8, we take into account the maximum percent of stable features ≈ 54%. The highest percentage of stable features ≈ 77% can be taken if we applied a 16 × 16 in the preprocessing image treatment. We tested our experiment on a large database of grayscale images of size 512 × 512 and we observed that these values presented in Table 4 can be obtained approximatively for others images of the same settings of image blocks decomposition and Gaussian noise addition, also we noted that the percentages of features that moved from the left and those moved from the right approximately equals which coincides with the theoretical study presented in Section 3.2.2. Same remarks of the approximately equalities of the percentages that moved from the left and the right are observed in the cases of Q=4 and Q=16 than in the case of the quantization step size Q=1. These obtained numerical values are almost approximately fixed in the same settings parameters in the block image decomposition and the level of Gaussian noise addition because we tested our experiments on large database grayscale images. These values are obtained for the grayscale image shown in Figure 10(a) and can be obtained for any other grayscale image.

Based on the numerical results presented in Table 4, Figure 14 shows the percentage of the features that remain stable under the additive Gaussian noise for different image blocks decomposition. As we remark, to get a high percentage of of stable features, we have two possibilities: either we apply great size of image block decomposition or the original image undergos small additive Gaussian noise.



(a) Case of quantization step size Q=1.

(b) Case of quantization step size Q=4.

(c) Case of quantization step size Q=16.

Fig. 14. Stability percent of mean features for a fixed quantization step size for different block sizes: (a) Case of quantization step sizes $Q = 1$, (b) Case of quantization step size $Q = 4$ and (c) Case of quantization step size $Q = 16$.

Q	Block Size	σ	**(%)** Not Moved	**(%)** Moved from the Right	**(%)** Moved from the Left
1	4×4	1	79.4128	10.4004	10.1868
		5	30.5237	34.8633	34.6130
		11	14.5569	42.5842	42.8589
		14	11.4258	43.0176	45.5566
		40	4.1382	47.5281	48.3337
	8×8	1	**90.7471**	4.5410	4.7119
		5	53.4180	23.3643	23.2178
		11	29.0283	35.0586	35.9131
		14	23.0957	36.3037	40.6006
		40	7.6660	46.5332	45.8008
	16×16	1	**94.9219**	2.1484	2.9297
		5	77.4414	11.8164	10.7422
		11	52.9297	23.5352	23.5352
		14	41.2109	27.2461	31.5430
		40	14.2578	43.5547	42.1875
4	4×4	1	**94.6960**	2.7710	2.5330
		5	74.7864	12.8540	12.3596
		11	50.4456	24.6826	24.8718
		14	41.6382	28.4851	29.8767
		40	15.9119	41.8457	42.2424
	8×8	1	**97.5098**	1.2939	1.1963
		5	87.6221	6.0547	6.3232
		11	73.7061	12.7930	13.5010
		14	66.2598	15.4297	18.3105
		40	29.2725	35.8154	34.9121
	16×16	1	**98.9258**	0.5859	0.4883
		5	**94.7266**	3.0273	2.2461
		11	88.9648	4.9805	6.0547
		14	83.3984	7.7148	8.8867
		40	45.7031	28.5156	25.7812
16	4×4	1	**98.6694**	0.6714	0.6592
		5	**93.9575**	3.0273	3.0151
		11	86.3953	6.6162	6.9885
		14	82.8918	8.0811	9.0271
		40	53.8086	22.5220	23.6694
	8×8	1	**99.4141**	0.2686	0.3174
		5	**96.7529**	1.5625	1.6846
		11	**93.5059**	3.0518	3.4424
		14	**91.7969**	3.5645	4.6387
		40	74.6826	12.5244	12.7930
	16×16	1	**99.9023**	0.0000	0.0977
		5	**98.7305**	0.7812	0.4883
		11	**96.5820**	1.3672	2.0508
		14	**95.2148**	1.9531	2.8320
		40	85.3516	6.9336	7.7148

Table 4. Numerical results for different levels of the additive Gaussian noise and image block size in the case of the quantization step sizes $Q = 1$, $Q = 4$ and $Q = 16$.

5. Conclusion

In this chapter, we introduced the main aim of the perceptual image hashing field in image security. We presented the important merits and requirements of a perceptual image hash function used for authentication wherein a formulation of the perceptual image hashing problem was given. We dedicated a section to presenting an overview of recent techniques that are used for perceptual image hashing. After, we presented the different quantization techniques used for more robustness of a perceptual image hashing scheme showing their advantages and their limitations. Finally, we presented a theoretical model describing the behavior of the extracted image features to be hashed against content-preserving/content-changing manipulations. In the presented analysis, we simulated the manipulations that may undergo the original image by an additive Gaussian noise. We tested the presented model by several experiments to demonstrate the effectiveness of the proposed theoretical model giving practical analysis for robust perceptual image hashing. The presented model is applied on image hashing based on statistical invariance of mean block features. The obtained results confirms the theoretical study presented in Section 3.2. Some approximations must be done to improve results. The same study can be generalized for other features in block-based image hashing scheme like DCT domain features, DWT domain features, etc.

6. References

Ahmed, F. & Siyal, M. Y. (2006). A secure and robust wavelet-based hashing scheme for image authentication, *in* T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua & L.-T. Chia (eds), *Advances in Multimedia Modeling*, Vol. 4352 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62.

Ahmed, F., Siyal, M. Y. & Abbas, V. U. (2010). A secure and robust hash-based scheme for image authentication, *Signal Processing* 90: 1456–1470.

Bender, W., Gruhl, D., Morimoto, N. & Lu, A. (1996). Techniques for data hiding, *IBM Systems Journal* 35(3-4): 313–336.

Bhattacharjee, S. K. & Kutter, M. (1998). Compression tolerant image authentication, *Proceedings of the IEEE International Conference on Image Processing (ICIP (1))*, pp. 435–439.

Cox, I. J., Miller, M. L. & Bloom, J. A. (2000). Watermarking applications and their properties, *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)*, IEEE Computer Society, Las Vegas, NV, USA, pp. 6–10.

Cox, I. J., Miller, M. L. & Bloom, J. A. (2002). *Digital watermarking*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Fridrich, J. (2000). Visual hash for oblivious watermarking, *SPIE Photonic West Electronic Imaging 2000, Security and Watermarking of Multimedia Contents*, Vol. 3971, SPIE, San Jose, California, pp. 286–294.

Fridrich, J. & Goljan, M. (2000). Robust hash functions for digital watermarking, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'00)*, IEEE Computer Society, Washington, DC, USA, pp. 178–183.

Furht, B., Socek, D. & Eskicioglu, A. M. (2004). Fundamentals of multimedia encryption techniques, *IN MULTIMEDIA SECURITY HANDBOOK*, CRC Press, pp. 93–131.

Guo, X. C. & Hatzinakos, D. (2007). Content based image hashing via wavelet and radon transform, *Proceedings of the multimedia 8th Pacific Rim conference on Advances* *in multimedia information processing*, PCM'07, Springer-Verlag, Berlin, Heidelberg, pp. 755–764.

Hadmi, A., Puech, W., AitEssaid, B. & Aitouahman, A. (2010). Analysis of the robustness of wavelet-based perceptual signatures, *IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA'10)*, Paris, France, pp. 112–117.

Hadmi, A., Puech, W., AitEssaid, B. & Aitouahman, A. (2011). Statistical analysis of the quantization stage of robust perceptual image hashing, *IEEE* 3^{rd} *European Workshop on Visual Information Processing (EUVIP'11)*, Paris, France.

Han, S. H. & Chu, C. H. (2010). Content-based image authentication: current status, issues, and challenges, *International Journal of Information Security* 9: 19–32.

Kerckhoffs, A. (1883). La cryptographie militaire, *Journal des sciences militaires* 9(1): 5–38.

Khelifi, F. & Jiang, J. (2010). Perceptual image hashing based on virtual watermark detection, *IEEE Transactions on Image Processing* 19: 981–994.

Kozat, S. S., Venkatesan, R. & Mihçak, M. K. (2004). Robust perceptual image hashing via matrix invariants, *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, pp. 3443–3446.

Lefebvre, F., Macq, B. & Legat, J. D. (2002). Rash: Radon soft hash algorithm, *Proceedings of the European Signal Processing Conference (EUSIPCO'02)*, Toulouse, France.

Lei, Y., Wang, Y. & Huang, J. (2011). Robust image hash in radon transform domain for authentication, *Signal Processing: Image Communication* 26: 280–288.

Lin, C. Y. & Chang, S. F. (2001). A robust image authentication method distinguishing jpeg compression from malicious manipulation, *IEEE Transactions on Circuits and Systems for Video Technology* 11(2): 153–168.

Lu, C. S. & Liao, H. Y. M. (2003). Structural digital signature for image authentication: an incidental distortion resistant scheme, *IEEE Transactions on Multimedia* 5(2): 161–173.

Memon, N. & Wong, P. W. (1998). Protecting digital media content, *Communication ACM* 41: 35–43.

Menezes, A. J., Vanstone, S. A. & Oorschot, P. C. V. (1996). *Handbook of Applied Cryptography*, 1st edn, CRC Press, Inc., Boca Raton, FL, USA.

Mihçak, M. K. & R.Venkatesan (2001). New iterative geometric methods for robust perceptual image hashing, *Digital Rights Management Workshop*, pp. 13–21.

Mihçak, M. K. & Venkatesan, R. (2001). A perceptual audio hashing algorithm: A tool for robust audio identification and information hiding, *Proceedings of the 4th International Workshop on Information Hiding*, IHW '01, Springer-Verlag, London, UK, UK, pp. 51–65.

Monga, V. (2005). *Perceptually Based Methods for Robust Image Hashing*, Phd dissertation, University of Texas at Austin.

Monga, V. & Evans, B. L. (2006). Perceptual image hashing via feature points: Performance evaluation and trade-offs, *IEEE Transactions on Image Processing* 15(11): 3452–3465.

NIST (2008). FIPS PUB 180-3, Federal Information Processing Standard (FIPS), Secure Hash Standard (SHS), Publication 180-3, *Technical report*, National Institute of Standards and Technology, Department of Commerce.

Ozturk, I. & Ibrahim, S. (2005). Analysis and comparison of image encryption algorithms, *Education Technology and Training & Geoscience and Remote Sensing* 3: 803–806.

Puech, W., Rodrigues, J. M. & Develay-Morice, J. E. (2007). A new fast reversible method for image safe transfer, *Journal of Real-Time Image Processing* 2(1): 55–65.

Rivest, R. L. (1992). The MD5 Message-Digest Algorithm, *Technical Report RFC 1321*, Internet Engineering Task Force (IETF).

Rodrigues, J. M., Puech, W. & Bors, A. G. (2006). Selective encryption of human skin in jpeg images, *Proceedings of the IEEE International Conference on Image Processing (ICIP'06)*, pp. 1981–1984.

Schneider, M. & Chang, S. F. (1996). A robust content based digital signature for image authentication, *Proceedings of the IEEE International Conference on Image Processing (ICIP'96)*, Vol. 3, pp. 227–230.

Seo, J. S., Haitsma, J., Kalker, T. & Yoo, C. D. (2004). A robust image fingerprinting system using the radon transform, *Signal Processing: Image Communication* 19(4): 325–339.

Stinson, D. (2002). *Cryptography: Theory and Practice*, 2nd edn, Chapman & Hall, CRC.

Sun, Q. & Chang, S. F. (2005). A robust and secure media signature scheme for jpeg images, *VLSI Signal Processing* 41(3): 305–317.

Swaminathan, A., Mao, Y. & Wu, M. (2006). Robust and secure image hashing, *IEEE Transactions on Information Forensics and Security* 1(2): 215–230.

Venkatesan, R., Koon, S. M., Jakubowski, M. H. & Moulin, P. (2000). Robust image hashing, *Proceedings of the IEEE International Conference on Image Processing (ICIP'00)*, pp. 664–666.

Wang, S. Z. & Zhang, X. P. (2007). Recent development of perceptual image hashing, *Journal of Shanghai University* 11: 323–331.

Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity, *IEEE Tansactions on Image Processing* 13(4): 600–612.

Zhu, G., Huang, J., Kwong, S. & Yang, J. (2010). Fragility analysis of adaptive quantization-based image hashing, *IEEE Transactions on Information Forensics and Security* 5: 133–147.

3

Robust Multiple Image Watermarking Based on Spread Transform

Jaishree Jain and Vijendra Rai
Mahamaya Technical University
Noida
India

1. Introduction

In this chapter, some multiple watermarking techniques and their limitations are discussed which include both spatial and transform domain methods. Since many algorithms are applied to graphical images, the concept of graphical image perceptibility and measures of PSNR and Bit Error Ratio (BER) are also discussed.

Watermarks are used to keep track of paper provenance and thus format and quality identification in the art of handmade papermaking nearly 700 years ago. In 1993 the term Watermark is used first time. In 1993-1994 the first papers on digital watermarking was published whereas in 1995 the first special session on image watermarking at NSIP95, Neos Marmaras, Greece was held. In 1995 one of the first images watermarking algorithms Patchwork algorithm was proposed. Watermarking has developed basically from two different streams, Cryptography meaning, secret writing and Steganography, which in the Greek language means, cover writing.

This is the digital information revolution era. It has connectivity over the Internet and connectivity through the wireless network. Innovative devices such as digital camera and camcorder, high quality scanners and printers have reached consumers worldwide to create, manipulate and enjoy the multimedia data. The development of high speed computer networks and that of internet, in particular, has explored means of new business, scientific, entertainment and social opportunities in the form of electronic publishing and advertising, real-time information delivery, product ordering, transaction processing, digital repositories and libraries, personal communication etc.

Digital content are spreading rapidly in the world via the internet. It is possible to produce a number of the same one with the original data without any limitation. Copying is simple with no loss of fidelity. A copy of a digital media is identical to the original. This has many instances, led to the use of digital content with malicious intent. The current rapid development of new IT technologies for multimedia services has resulted in a strong demand for reliable and secure copyright protection techniques for multimedia data. One way to protect multimedia data against illegal recording and retransmission is to embed a signal, called digital signature or copyright label or watermark that authenticates the owner of the data. With the ease of editing and perfect reproduction in digital domain, the protection of ownership and the prevention of unauthorized tampering of multimedia data (audio, image, video, and document) have become important concerns. Digital watermarking schemes to embed secondary data in digital media, have made considerable progress in recent years and attracted attention from both academia and industry. Techniques have been proposed for a variety of applications, including ownership protection, authentication and access control. Imperceptibility, robustness against moderate processing such as compression, and the ability to hide many bits are the basic but rather conflicting requirements for many data hiding applications.

Digital watermarking is a technique to embed invisible or inaudible data within multimedia contents. Watermarked contents contain a particular data for copyrights. A hidden data is called a watermark, and the format can be an image or any type media. In case of ownership confliction in the process of distribution, digital watermark technique makes it possible to search and extract the ground for ownership. Many researches on watermarking have been come out in the advanced countries including USA and EU so far, because of its importance of this area in the future.

To avoid the unauthorized distribution of images or other multimedia property, various solutions have been proposed. Most of them make unobservable modifications to images that can be detected afterwards. Such image changes are called watermarks. Watermarking is defined as adding (embedding) a payload signal to the host signal. The payload can be detected or extracted later to make an assertion about the object i.e. the original data that may be an image or audio or video.

Multiple watermarking is an embranchment of digital watermarking which has many desirable characteristics that common singular watermarking does not have, such as robustness to union attacks. For example, employ multiple watermarks to convey multiple sets of information, intended to satisfy differing or similar goals, Used to increase robustness with many different methods, the embedded information is not easily lost, it is possible to support different access levels. To accomplish several goals, one might wish to embed several watermarks into the same image. For example, the owner might desire to use one watermark to convey ownership information, a second watermark to verify content integrity, a third watermark to convey a caption.

The aim of watermarking is to include subliminal information (i.e., imperceptible) in a multimedia document to ensure a security service or simply a labeling application. But existing multiple watermarking has inherent problem such as low validity and high complexity.

In general, any watermarking scheme (algorithm) consists of three parts:

- The watermark (payload)
- The encoder (marking insertion algorithm)
- The decoder and comparator (verification or extraction or detection algorithm)

Each owner has a unique watermark or an owner can also put 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.

1.1 Watermark insertion and extraction

Watermark insertion involves watermark generation and encoding process.

1.1.1 Watermark generation

The watermark can be a logo picture, sometimes a binary picture, sometimes a ternary picture; it can be a bit stream or also an encrypted bit stream etc. The encryption may be in the form of a hash function or encryption using a secret key. The watermark generation process varies with the owner.

1.1.2 Encoding process

Inputs to the embedding scheme are the watermark, the cover data and an optional public or secret key. The output is watermarked data. The key is used to enforce security.

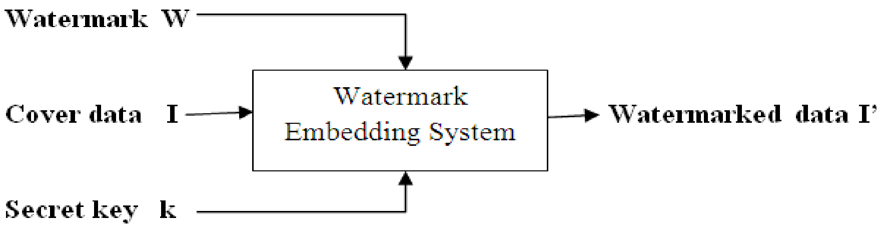


Fig. 1. Embedding Process

1.1.3 Watermark extraction

Extraction is achieved in two steps. First the watermark or payload is extracted in the decoding process and then the authenticity is established in the comparing process.

1. **Decoding process**: Inputs to the decoding scheme are the watermarked data, the secret or public key and depending on the method, the original data and/or the original watermark. The output is the recovered watermarked W.

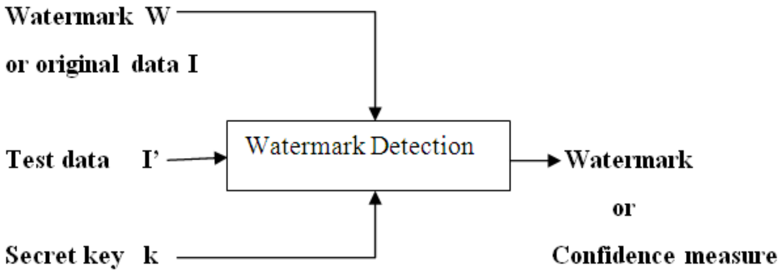


Fig. 2. Extraction Process

2. **Comparison Process:** The extracted watermark is compared with the original watermark by a comparator function and a binary output decision is generated. The comparator is basically a correlated. Depending on the comparator output it can be determined if the data is authentic or not. If the comparator output is greater than equal to a threshold then the data is authentic else it is not authentic. Figure illustrates the comparing function. In this process the extracted watermark and the original watermark are passed through a comparator. The comparator output C is the compared with a threshold and a binary output decision generated. It is 1 if there is a match i.e. C ≥ δ and 0 otherwise. A watermark is detectable or extractable to be useful, depending on the way the watermark is inserted and depending on the nature of the watermarking algorithm, the method used can involve very distinct approaches. In some watermarking schemes, a watermark can be extracted in its exact form, a procedure we call watermark extraction. In other cases, we can detect only whether a specific given watermarking signal is present in an image, a procedure we call watermark detection. It should be noted that watermark extraction can prove ownership whereas watermark detection can only verify ownership [5].

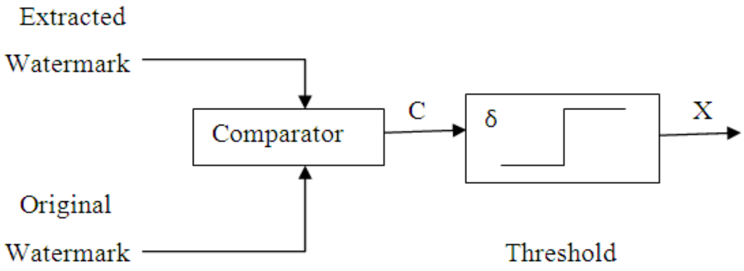


Fig. 3. Comparison Process

1.2 Practical challenges of watermarking

Watermark by itself is not sufficient to prevent abuses unless a proper protection protocol is established. The exact properties that a watermarking algorithm must satisfy cannot be defined exactly without considering the particular application scenario; the algorithm has to be used in. A brief analysis of requirements of data hiding algorithms from a protocol perspective permits to decide whether a given algorithm is suitable for a certain application or not. Each watermarking application has its own specific requirements. Most often than not these requirements have conflicting effects on each other. A good watermarking algorithm obtains optimal tradeoff between these requirements; is not weakened/ destroyed by attacks, both malicious and non-malicious; at the same time unambiguously identifies the owner. These properties can be broadly classified as primary and secondary requirements. The primary requirements include data hiding capacity, imperceptibility and robustness as shown in figure4. However these three characteristics conflict with each other. Increasing fidelity of the watermarked images (i.e. increasing imperceptibility of the mark) would lower the strength of the watermark. Embedding large amount of information reduces the fidelity of the watermark. The secondary requirements include performance i.e. the speed of embedding and of detection of the watermark. These attributes though less commonly discussed are very important for many real world applications. Each of the primary attributes has been discussed in detail below.

1.2.1 Capacity of watermarking techniques

Capacity is a fundamental property of any watermarking algorithm, which very often determines whether a technique can be profitably used in a given context or not. However no requirement can be set without considering the application the technique has to serve in. Possible requirements range from some hundreds of bits in security oriented applications, where robustness is a major concern, through several thousands of bits in applications like captioning or labeling, where the possibility of embedding a large number of bits is a primary need. For copy protection purposes, a payload of one bit is usually sufficient. Capacity requirements always struggle against two other important requirements, watermark imperceptibility and watermark robustness. A higher capacity is always obtained at the expense of either robustness or imperceptibility or both. It is therefore mandatory that a good trade-off be found depending on the application at hand.

1.2.2 Imperceptibility

The watermark should be imperceptible so as not to affect the viewing experience of the image or the quality of the image signal. In most applications the watermarking algorithm must embed the watermark such that this does not affect the quality of the underlying host data. A watermark embedding procedure is truly imperceptible if humans cannot distinguish the original data from the data with the inserted watermark. However even the smallest modification in the host data may become apparent when the original data is compared directly with the watermarked data. Since users of watermarked data normally do not have access to the original data, they cannot perform this comparison. Therefore, it may be sufficient that the modifications in the watermarked data go unnoticed as long as the data are not compared with the original data.

1.2.3 Robustness

Watermark robustness accounts for the capability of the hidden data to survive host signal manipulations, including both non-malicious manipulations, which do not explicitly aim at removing the watermark or at making it unreadable, and malicious manipulations, which precisely aim at damaging the hidden information. The exact level of robustness the hidden data must possess cannot be specified without considering a particular application. Robustness against signal distortion is better achieved if the watermark is placed in perceptually significant parts of the signal. This is particularly evident in the case of lossy compression algorithms, which operate by discarding perceptually insignificant data. Watermarks hidden within perceptually insignificant data are likely not to survive compression. Achieving watermark robustness, and, to a major extent, watermark security is one of the main challenges watermarking researches are facing with.

1.3 Watermarking attacks

Any procedure that can decrease the performance of the watermarking scheme may be termed as an attack. Voloshynovskiy et.al [1] categorizes attacks into four classes' viz.

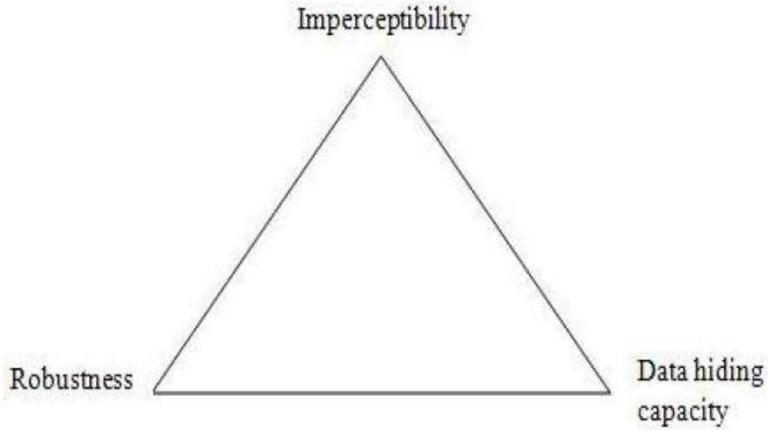


Fig. 4. Primary Requirements of Watermarking Algorithms

removal, geometric, cryptographic and protocol. Removal attack removes the watermark without having any prior knowledge about the watermark, while geometric attacks deal with de-synchronization of the receiver so that watermark detection is distorted. Cryptographic schemes are those that tend to crack the watermarking scheme and protocol attacks exploit invertible watermarks to cause ownership ambiguity. These attacks can be broadly classified as non-malicious (unintentional) such as compression of a legally obtained, watermarked image or video file and malicious (intentional) such as an attempt by a multimedia pirate to destroy the embedded information and prevent tracing of illegal copies of watermarked digital video. Watermarking systems utilized in copy protection or data authentication schemes are especially susceptible to malicious attacks. Non-malicious attacks usually come from common signal processing operations done by legitimate users of the watermarked materials.

1.3.1 Malicious attack

An attack is said to be malicious if its main goal is to remove or make the watermark unrecoverable. Malicious attacks can be further classified into two different classes.

Blind: A malicious attack is said to be blind if it tries to remove or make the watermark unrecoverable without exploiting knowledge of the particular algorithm that was used for watermarking the asset. For example, copy attack that estimates the watermark signal with aim of adding it to another asset.

Informed: A malicious attack is said to be informed if it attempts to remove or make the watermark unrecoverable by exploiting knowledge of the particular algorithm that was used for watermarking the asset. Such an attack first extracts some secret information about the algorithm from publicly available data and then based on this information nullifies the effectiveness of the watermarking system. Examples of malicious attacks: Printing and Rescanning.

1.3.2 Non-malicious attack

An attack is said to be non malicious if it results from the normal operations that watermarked data or any data for that matter has to undergo, like storage, transmission or fruition. The nature and strength of these attacks are strongly dependent on the application for which the watermarking system is devised. For example lossy- compression, geometric and temporal manipulations digital to analogue conversion, extraction of asset fragments (cropping), processing aimed at enhancing asset (e.g. noise reduction), etc. Examples of Non-Malicious Attacks: Lossy Compression: Many compression schemes like JPEG and MPEG can potentially degrade the data's quality through irretrievable loss of data.

Geometric Distortions: Geometric distortions are specific to images and videos and include such operations as rotation, translation, scaling and cropping.

2. Different types of watermarks and watermarking techniques

2.1 Visible watermark

Visible watermarks are the watermarks, existence of which is visible to the user. For example, to indicate ownership of originals, the content owner desires a visible mark that makes clear the source of the materials.

i. Spatial domain visible watermarking

Using patch work algorithm was proposed by N. Memon and P. Wong in 1998 [2].The author has selected n number of patches randomly and make certain statistics to make use of these patches as watermark. This method is more resistant to attempts of data removal by a third party but the scheme is extremely sensitive to geometric transformation. If the patch size is very small with sharp edges then it results in removal of watermark in lossy compressions, also optimal choice of patch shape is dependent upon the expected image modification. Due to the limitations of the spatial domain techniques the visible watermarking is also developed in the transform domain.

ii. Transform domain visible watermarking

A DCT domain visible watermarking technique for images [3] was developed by S. P. Mohanty, et al. The technique modifies DCT coefficients of the cover image and exploits the texture sensitivity of the human visual system. The perceptual quality of the image is better preserved in this technique as compared to the previous one but this technique is not robust for images having very few objects and large uniform areas.

2.2 Invisible watermark

The invisible watermark's existence should be determined only through a watermark extraction or detection algorithm. The invisible watermark falls into three categories:

1. Fragile watermarking

Invisible image watermarks that will change, or disappear, if a watermarked image is altered are called as fragile watermarking. These watermarks are called fragile invisible watermarks because it is desired that they be altered or destroyed by most common image processing techniques. For example, invisible watermarking for a trustworthy camera.

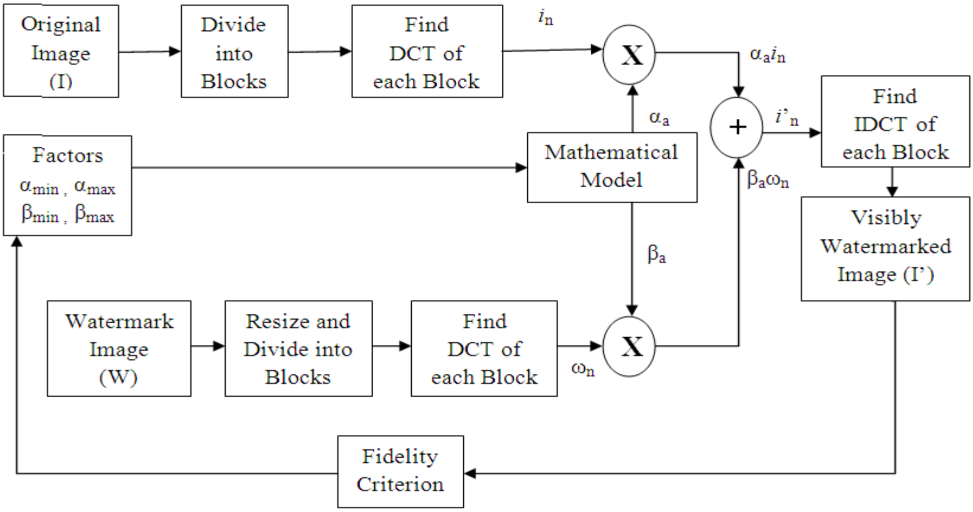


Fig. 5. Watermark Insertion Process [4]

A Fragile Watermarking Scheme for Image Authentication with Tamper Localization Using Integer Wavelet Transform was proposed by M. Venkatesan, et al. in [4] in spatial domain. Watermark is randomly scattered in the LSB of the cover image. The technique is capable of detecting and localizing the malicious changes in the cover image and it has the ability to discriminate watermark and content tampering. The only limitation of the technique is that the relationship between the reliability of tamper detection and the localization accuracy has not investigated.

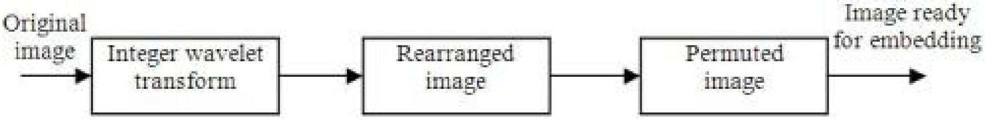
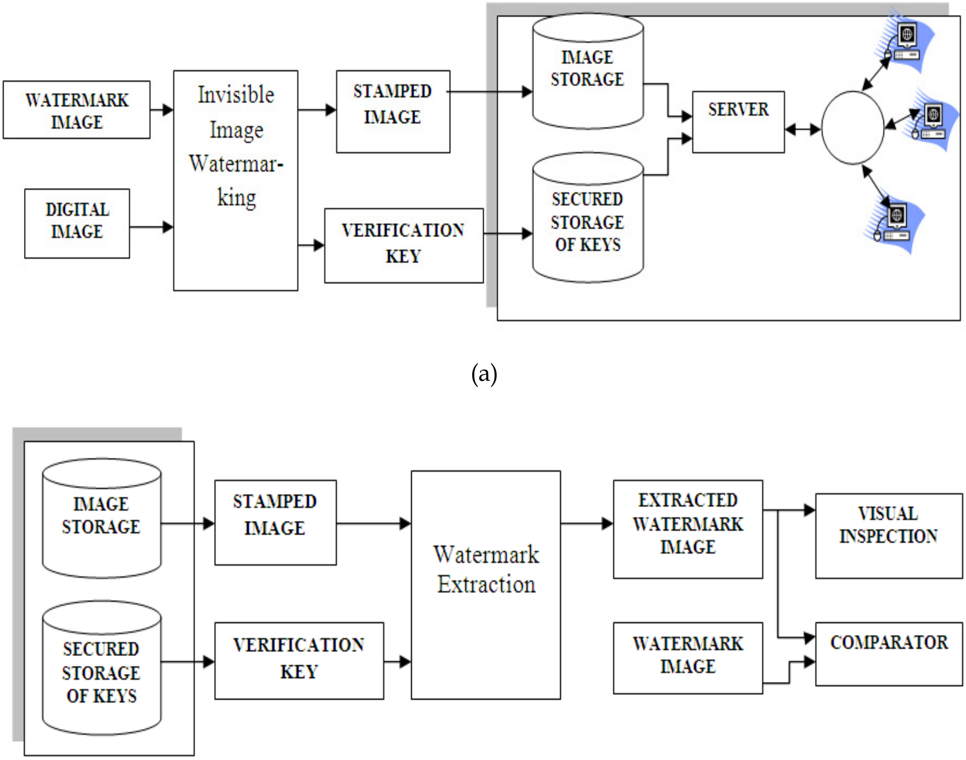


Fig. 6. Preprocessing [4]

2. Semi-fragile watermarking

These are the watermarking systems where content needs to be strictly protected, but the exact representation during exchange and storage need not be guaranteed. Semi fragile watermarking methods validate image content, but not its representation, and are thus made robust against allowable alterations, while being sensitive to non permitted modifications. For example, Semi fragile tamper detection methods are designed to monitor changes in the content and tamper detection is based on the visual assessment of perceived differences by an operator.

An invisible watermarking technique for image verification was proposed by Yeung, M.M and Mintzer F. [5] in spatial domain. The technique is developed using least significant Bit method and the verification key is generated using Look up Table (LUT). The method can localize the regions of image alterations and hence effectively use for tamper detection. The watermarking process does not introduce visual artifacts and retain the quality of the image and provide protection against retention of watermark after unauthorized alterations. As LUT is generated randomly, the pixel values may have to be adjusted by larger amounts to get desired unary value.



(a)

(b)

Fig. 7. The block Diagram of the Image Verification System with Proposed Invisible Watermarking Technique [5]

Semi Fragile Watermarking Based on Wavelet Transform was proposed by Yuichi Nakai [6]. The technique is based on wavelet transform and embeds watermark to wavelet coefficients for evaluating the degree of tampering for each pixel. It embeds MSB of watermarks in low frequency components and LSB in high frequency component. The proposed scheme can evaluate the degree of tampering for each pixel but the number of watermarks that can be embedded without degradation of image quality is less.

3. Robust watermarking

Watermarks that persist even if someone tries to remove them are called as robust watermarking. Since they are desired to survive intentional attacks (e.g. active attack, passive attack etc.), these are called as robust image watermarks. For example, Evidence of ownership.

Van Schyndel, et al. has developed robust watermarking in his paper ''A Digital Watermark'' [7] in spatial domain. The original 8 bit grey scale image data is compressed to 7 bits by adaptive histogram manipulation. The watermark is generated using an m sequence generator. The watermark was embedded to the LSB of the original image and

Cross-correlation based detection was proposed. The method utilizes linear addition of watermark data and is more difficult to decode, offering inherent security. The technique is compatible with JPEG processing. The watermark is not robust to additive noise.

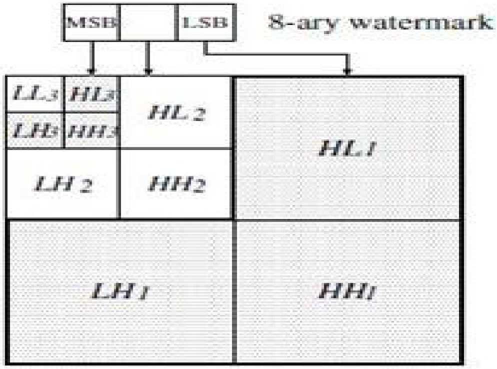


Fig. 8. Embedding 8-ary Watermarks in Several Wavelet Coefficient Level [7]

I.A. Nasir has divided the host image into four different regions each consisting of 128 ×128 blocks in order to hide a watermark [8]. The watermark is a binary image encrypted and embedded into different regions of the blue component of the image by altering intensity values of the selected regions. The watermarks can be extracted by comparing the intensities of the selected region of the original image with the corresponding region of the watermarked image. The proposed watermarking scheme is robust for a wide range of attacks including JPEG compression, rotation, scaling, filtering, etc. The number of watermarks that can be embedded effectively is not statistically proved.

3. Multiple watermarking basics

Multiple watermarking is an embranchment of digital watermarking, which has many desirable characteristics that common singular watermarking does not have. For example, employ multiple watermarks to convey multiple sets of information, intended to satisfy differing or similar goals, used to increase robustness with many different methods, the embedded information is not easily lost, it is possible to support different access levels. To accomplish several goals, one might wish to embed several watermarks into the same image. For example, the owner might desire to use one watermark to convey ownership information, a second watermark to verify content integrity, a third watermark to convey a caption [9].In general, to apply multiple disparate watermarks, ownership watermarks should be very robust, captioning watermarks should be robust, and Verification watermarks should be quite fragile. In general, to apply multiple disparate watermarks, the most robust (ownership) watermark should be embedded first, the most fragile (verification) watermark should be embedded last, and moderately robust (captioning) watermarks should be inserted in between.

Embedding multiple watermarks will then be successful if the robust watermarks are sufficiently robust to withstand all subsequent watermark insertions. After the insertion of multiple watermarks, the watermarked image will possess texture resulting from each watermark. Embedding multiple watermarks also requires that each watermark add less texture than would be permissible.

3.1 Types of multiple watermark

The multiple watermarking is broadly classified into three categories [10] as follows:

i. Composite watermarking

All watermarks are combined into a single watermark which is subsequently embedded in one single embedding step. The composite watermarks are separable if the watermarking patterns are orthogonal (or uncorrelated) in some sense relevant to the watermark detection. Example: Averaged watermarking

ii. Segmented watermarking

The host data is partitioned into disjoint segments a priory and each watermark is embedded into its specific share. If all keys are present the detector can find a watermark in every segment, otherwise it cannot. Example: Interleaved watermarking.

iii. Successive watermarking

It is the most straightforward method to embed the watermarks one after the other.

This method is useful in the applications where retrieval of one watermark should depend on the retrieval of other watermark. For example, it allows us to determine the order in which the watermarks are embedded. The object becomes more degraded with every new watermark inserted into it, both in terms of PSNR and perceived quality. Example: Re-watermarking.

In general, to apply multiple disparate watermarks, the most robust (ownership) watermark should be embedded first, the most fragile (verification) watermark should be embedded last, and moderately robust (captioning) watermarks should be inserted in between. Embedding multiple watermarks will be successful if the robust watermarks are sufficiently robust to withstand all subsequent watermark insertions. After the insertion of multiple watermarks, the watermarked image will possess texture resulting from each watermark. Embedding multiple watermarks also requires that each watermark add less texture than would be permissible.

3.2 Multiple watermarking techniques

The different watermarking techniques are broadly classified between two domains, namely spatial and transform domain.

3.2.1 Spatial domain

The spatial techniques insert the watermark in the underused least significant bits of the image. This allows a watermark to be inserted in an image without affecting the value of the image. Example: Least Significant Bit, Statistical, block based method. The most common implementation of spatial domain watermarking is Least Significant Bit (LSB) replacement method. It involves replacing the n least significant bits of each pixel of a container image with the data of a hidden image. Since the human visual system is not very attuned to small variations in color, the method adjusts the small differences between adjacent pixels leaving the result virtually unnoticeable.

3.2.2 Transformed domain techniques

In the transform domain approach, some sort of transforms is applied to the original image first. The transform applied may be (DCT), (DFT), (DWT), etc. The watermark is embedded by modifying the transform domain coefficients. Example: DFT, DCT, DWT, Spread Spectrum.

Traditional watermarking schemes consisted of visible watermarking. Applications now demand that the watermark being embedded be highly robust to attacks. Techniques of hiding information in images include the use of discrete cosine transform (DCT), discrete

Fourier transforms (DFT) and wavelet transform.

i. Discrete cosine transform

This is the most commonly used transform for watermarking purpose. The DCT allows an image to be broken up into different frequency bands making it much easier to embed watermarking information into the middle frequency bands of an image. In our technique we use middle-band DCT coefficients to encode the message. It avoids the most visual important parts of the image without over exposing themselves to removal through compression and noise-attacks.

I J. Cox have considered watermarking as communications with side information [11].

The DCT allows an image to be broken up into different frequency bands, making it much easier to embed watermarking information into the middle frequency bands of an image.

Algorithm achieves good robustness against compression and other signal processing attacks due to the selection of perceptually significant transform domain coefficients. Robustness and the quality of the watermark could be improved if the properties of the host image could similarly be exploited.

M. Barni has embedded pseudo-random sequence of real numbers having normal distribution with zero mean and unity variance in selected set of DCT coefficients [12]. The watermark is robust to several signal processing techniques, including JPEG compression, low pass and median filtering, dithering etc. But watermark does not resist geometric translations. Mitchell et al. has computed a frequency mask for each block [13].The resulting perceptual mask is scaled and multiplied by the DCT of a pseudo-noise sequence which is different for each block. This watermark is then added to the corresponding DCT block. The watermark is robust to several distortions including white and colored noise, cropping, etc. For JPEG coding at 10% the quality of original image degrades.

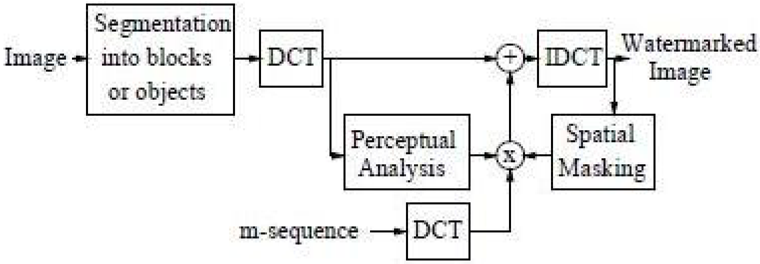


Fig. 9. Diagram of New Watermarking Technique [14]

ii. Discrete wavelet transform

This technique is also called as multiresolution technique. The important aspect of this technique is that watermark is introduced in imperceptibly significant regions of the data in order to remain robust. It decomposes the image into frequency bands using resolution of wavelets. X.Xia in 1997 proposed the concept of Multiresolution Watermark for Digital Images using wavelet transformation [14]. An image can be decomposed into a pyramid structure with various bands information: such as low-low frequency bands, low-high, high-low or high-high frequency bands. Adding watermarks on the large coefficients (HH, LH, HL and LL) is difficult for the human eyes to perceive. If distortion of a watermarked image is not serious, only a few bands worth of information are needed to detect the signature and therefore computational load can be served. This method is robust to all kinds of distortions such as compression, additive noise, etc. If distortion of a watermarked image is more, more bands of DWT are needed to detect watermark and computational load increases.

X. Liang and Wu Huizhong have proposed the multiple perceptual watermarks using multiple-based number conversion in wavelet domain [15]. Multiple watermarks coding and decoding system for image copyright protection is presented. Just Noticeable Difference (JND) threshold in wavelet domain is used to determine the locations for embedding. A multiple-based number system (every digit in number has base bi 0) is proposed to convert the watermark information into values to be embedded in the wavelet coefficient. The method has good robustness to JPEG compression, median filtering, Gaussian noise suppression, cropping and morphing type of distortions. Watermark strength is more as JND is used. The method fails to stir mark attack.

The limitations of wavelet transform have been overcome in dual tree complex wavelet transform. Lan Hong xing et al. in the paper ''A Digital Watermarking Algorithm Based on Dual-tree Complex Wavelet Transform'' [16], has proposed a multipurpose watermarking algorithm based on dual tree complex-wavelet transform. The authors notify the copyright owner with visible watermark and to protect the copyright with an invisible watermark. Dual- tree DWT has relatively high capacity to make the visible watermark hard to remove and invisible watermark robust. The only difficulty is in redesign of watermark with perfect reconstruction properties. It can only bring less visual effects for reconstruction of image in +/-45 sub bands.

iii. Spread spectrum

Spread spectrum watermarking is one of the most popular methods of watermarking. In this technique, the watermark bits are randomly scattered in the cover object. This not only ensures that the watermark is robust to attacks but also simplifies the detection algorithm using correlation analysis. Cryptographers believe that spread spectrum (SS) method of watermarking can incorporate a high degree of robustness because the pseudo-random sequences being used in SS watermarking are very difficult to generate without the prior knowledge of the initial state of the random number generator. This secures decoding or removal of the watermark and also provides resistance to cropping. The major drawback of the SS watermarking scheme is that it requires a high gain value Δ, which sometimes tends to alter the cover data file considerably such that it is noticeable. To overcome this problem, the improved spread spectrum (ISS) technique is used. In this technique a feature vector extraction mechanism has been established which enhances the performance by modulating the energy of the inserted watermark to compensate for the signal interference. The ISS technique using the dither quantization is used to enhance the performance of the embedding procedure and improve the overall performance of the watermarking scheme.

Spread transform dither modulation method is a transform domain method. The transform methods are more complex, but more robust than the spatial methods. The watermark is inserted into the cover image in a spread-spectrum fashion in the spectral domain, thereby making it robust against signal processing operations. In this case, the feature vector extraction process can be seen as an extension of the spread transform technique (a more general method of spreading watermark information over a host signal than spread spectrum) that is frequently employed on multimedia. To this feature vector a quantization based watermarking algorithm is used. Quantization index modulation (QIM) methods are a class of watermarking methods that achieve provably good rate-distortion-robustness performance.

a. Quantization index modulation

The process of mapping a large possible infinite set of values to a much smaller set is called quantization. Since quantization reduces the number of distinct symbols that have to be coded, it is central to many lossy compression schemes. A quantizer consists of two mappings: an encoder mapping and a decoder mapping. The encoder divides the range of source values into a number of intervals. Each interval is represented by a codeword. The encoder represents all the source values that fall into a particular interval by the codeword assigned to that interval. As there could be many possibly infinitely many distinct samples that can fall in any given interval, the encoder mapping is irreversible. For every codeword generated by the encoder, the decoder generates a reconstruction value.

Quantizers, or a sequence of quantizers, can be used to as appropriate-identity functions to embed the watermark information. The number of possible values of m determines the number of required quantizers, m acts as an index that selects the quantizer that is used to represent m. For the case m = 2 we have a binary quantizer. The following figure illustrates the QIM information embedding process. To embed one bit m, m € 0, 1 and image pixel is mapped to the nearest reconstruction point representing the information of m. The minimum distance d min between the sets of reconstruction points of different quantizers in the ensemble determines the robustness of the embedding,

$$d_{\min} = \min_{(i,j):i\neq j} \min_{(x_i,x_j)} \left\| s(x_i;i) - s(x_j;j) \right\|$$

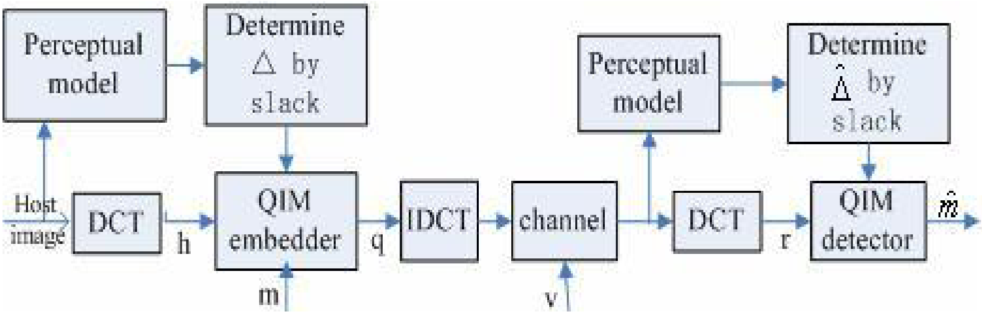


Fig. 10. QIM Scheme

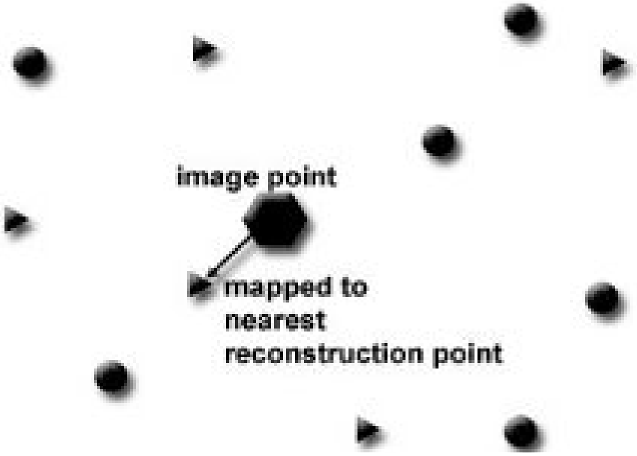


Fig. 11. Quantization Index Modulation

Intuitively, the minimum distance measures the amount of noise that can be tolerated by the system.

b. Dither modulation

A low-complexity realization of QIM called dither modulation which is better than both linear methods of spread spectrum and nonlinear methods of low-bit modulation against square-error distortion constrained intentional attacks. Dither modulation (DM) is the simplest form of quantization index modulation and is the most thoroughly analyzed by its ease of practical implementation. Dither modulation systems embed watermark by modulating the amount of the shift, which is called the dither vector, by the embedded signal. The host signal is quantized with the resulting dithered quantizer to form the composite signal. Dithered quantization (or Dither Modulation) is an operation in which a dither vector d of length L is added to the input x prior to quantization. The output of the subtractive quantization operation is denoted by

$$s_i = Q(x_i + d_i) - d_i;\ 0 \le i < L$$

Or, using the notation introduced above,

$$s(x; m) = Q(x + d(m)) - d(m)$$

For our discussion, we only consider uniform, scalar quantizer with a step size M. The binary dither ensemble can be generated pseudo-randomly by choosing di with a uniform distribution over [–Δ/2; +Δ/2] and assigning di as follows:

$$d_i\,(2) = \begin{Bmatrix} d_i(1) + \dfrac{\Delta}{2},\ if\ d_i(1) < 0 \\ d_i(1) - \dfrac{\Delta}{2},\ if\ d_i(1) \ge 0 \end{Bmatrix}$$

Where, 0 ≤ i < L. For the single embedding case (Figure12 (a)), let the QIM embed ding logic be converting an element to the nearest even/odd multiple of the quantization interval, Δ, to embed 0/1, respectively.

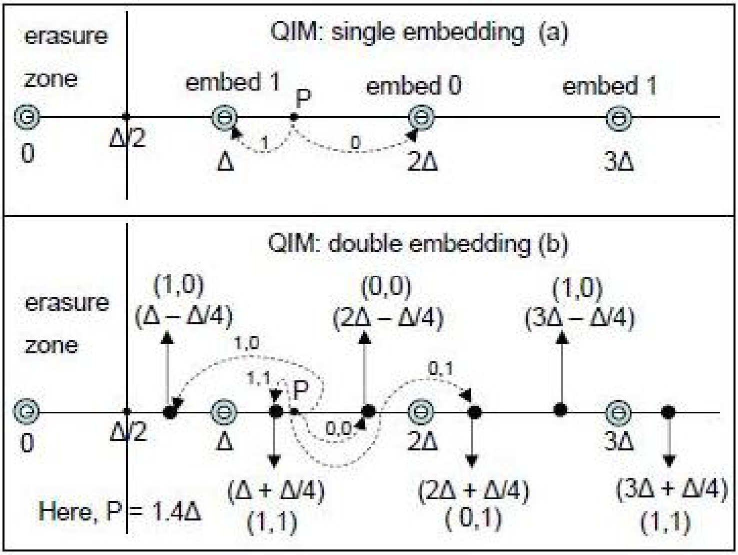


Fig. 12. QIM based Information hiding for single and double embedding

For hiding, we use quantized discrete cosine transform (DCT) coefficients. For perceptual transparency, we do not modify coefficients that are too close to zero; hence, all coefficients in the range [-0.5, 0.5] are mapped to zero and are regarded as erasures.

The two quantizers used for double embedding (Figure 12(b)) have quantization intervals of Δ and Δ/2, respectively. In the example (Figure 12(b)), Δ = 1 and the DCT coefficient (P) equals 1.4. Let the first bit to be embedded be 1 (using the coarser quantizer) and the second bit be 0 (using the finer quantizer).To embed 1, the coefficient (1.4) is changed to the nearest odd multiple of Δ (1). For the second bit, the coefficient is decreased/ increased by Δ/4 to embed 0/1 respectively. To embed 0, the coefficient is changed from 1 to 0.75.

Although it is now well-accepted that binning methods (QIM) are better suited for high-capacity hiding, SS techniques continue to receive a lot of attention because of their perceived advantage for achieving robustness. QIM-based schemes provide robustness against several attacks while embedding large number of bits. The subtractive dither quantization error (SDQE) does not depend on the quantizer input when the dither signal d has a uniform distribution within the range of one quantization bin (d_i € [-Δ/2, Δ/2]), leading to an expected squared error $e^2 = \Delta^2/12$.

c. Spread transform

Spread transform (also called projection) makes the embedding distortion concentrating on one coefficient spread to multiple coefficients. This leads to some advantages, such as the satisfaction of peak distortion limitations. This section presents a multiple watermarking method based on spread transform, in which cover vectors extracted from the cover works are projected to multiple orthogonal projection vectors. Then different watermark signals are embedded in different orientations of these orthogonal projection vectors. The embedding and extracting methods are introduced, and its performances are analyzed.

i. Watermark embedding process

The above discussion suggests the following general procedure for embedding multiple watermarks into the same image.

1. Read the input image to be watermarked.
2. Extract the cover vectors from the cover image by first dividing the image into blocks of 8×8 pixels and compute DCT for each block.
3. Choose L projection vectors to hide L different watermark signals such that number of projection vectors remains orthogonal to each other.
4. Embed different watermarks into corresponding projected data using dither modulation.

The mark is a Watermark sequence of binary values, w_i€ 0, 1.

Coefficient selection

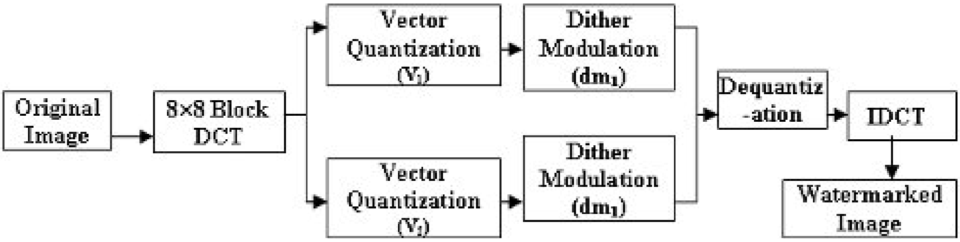
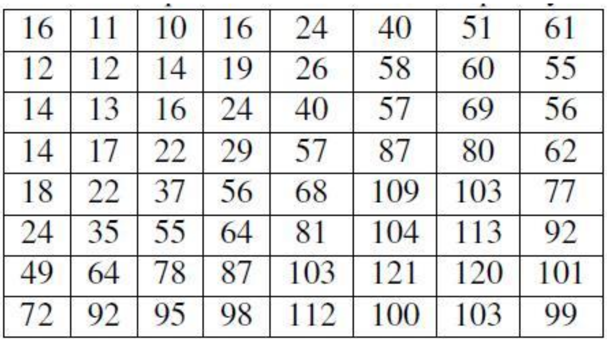


Fig. 13. Watermark Embedding

The proposed algorithm pseudo randomly selects 88 DCT coefficient blocks which are orthogonal to each other. These blocks are considered as a vector and the condition of orthogonality is $V_1. V_2^T=0$.

For embedding firstly, each block is quantized using to the JPEG quantization matrix and a quantization factor Q. Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer. In this step the less important DCT coefficients are wiped out. This (lossy) transformation is done by dividing each of the coefficients in the 8x8 DCT matrices by a weight taken from a quantization table. If all the weights are equal, the transformation does nothing but if they increase sharply from origin, higher spatial frequencies are dropped quickly. Most existing compressors start from a sample table developed by the ISO JPEG committee. Subjective experiments involving the human visual system have resulted in the JPEG standard quantization matrix. With a quality level of 50, the matrix renders both high compression and excellent decompressed image quality. If however, another level of quality and compression is desired, scalar multiples of the JPEG Standard quantization matrix (QM) may be used

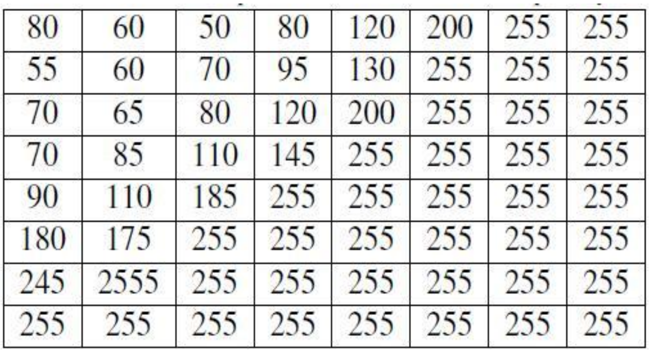


16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	57	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 1. JPEG standard quantization matrix for quality factor (QF) =50

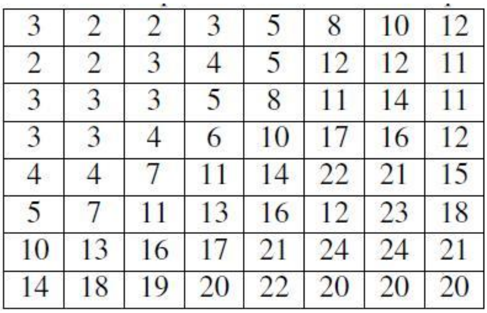
For a quality level greater than 50 (less compression and higher image quality), the standard QM is multiplied by (100-quality level)/50. For a quality less than 50 (more compression, lower image quality), the standard QM is multiplied by 50/quality level. The scaled QM is then rounded and clipped to have positive integer values ranging from 1 to 255. For example, the following QM yields quality levels of 10 and 90.

Then, let f_b denote an 8×8 DCT coefficient block and $f_b(m_1; n_1)$,



80	60	50	80	120	200	255	255
55	60	70	95	130	255	255	255
70	65	80	120	200	255	255	255
70	85	110	145	255	255	255	255
90	110	185	255	255	255	255	255
180	175	255	255	255	255	255	255
245	2555	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Table 2. JPEG standard quantization matrix for quality factor 10



3	2	2	3	5	8	10	12
2	2	3	4	5	12	12	11
3	3	3	5	8	11	14	11
3	3	4	6	10	17	16	12
4	4	7	11	14	22	21	15
5	7	11	13	16	12	23	18
10	13	16	17	21	24	24	21
14	18	19	20	22	20	20	20

Table 3. JPEG standard quantization matrix for quality factor 90

f_b (m_2; n_2) are the selected coefficients within that block. The absolute difference between the selected coefficients is given by:

$$\Delta_b = f_b(m_1; n_1) - f_b(m_2; n_2)$$

In order to embed one bit of watermark information, w_i, in the selected block b_i, the coefficient pair $f_b(m_1; n_1)$; $f_b(m_2; n_2)$ is modified such that the distance becomes where q is a parameter controlling the embedding strength.

$$\Delta_b = \begin{cases} \leq q, \ if \ w_i = 0 \\ \geq q, \ if \ w_i = 1 \end{cases}$$

In this proposed method the two watermarks are embedded using DM method with uniform, scalar quantizer of step size Δ, where Δ is the quantization step used to control the embedding distortion. This method is called double spread transform dither modulation (DSTDM). Figure 14 shows the realization of DM, where, x_0 is the original data, x_w is the watermarked data and $q_\Delta(\cdot)$ is the basic quantizer function, that is

$$q_\Delta(x) = \mathrm{round}(x/\Delta) \times \Delta$$

Where Δ is the quantization step used to control the embedding distortion and each coefficients quantization step can differ from each other, d[m] is the dither value corresponding to the watermark information m.

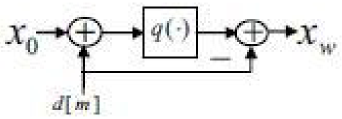


Fig. 14. Watermark Embedding Process of DM

iv. Watermark extraction method

In watermark detection process the embedded watermark signals are extracted using corresponding extraction method and compared with the original watermarked data. Extraction method depends on the embedding method used. The watermark extracting process is the reverse process of the watermark embedding process. Minimum distance decoder is used to extract the watermark which is similar to STDM algorithm. The detailed extracting method of DSTDM is following:

1. Extract the cover vectors by computing DCT in the blocks of 8 × 8 pixels of watermarked image.
2. Project the cover vectors to the same projection vectors used in the embedding process.
3. Apply DM with the same quantization step M.
4. Apply minimum distance decoding rule into the corresponding dither value received by dither modulation.

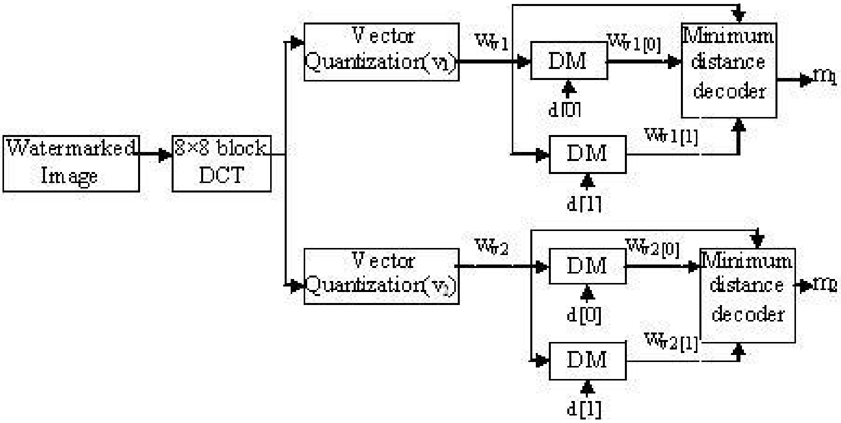


Fig. 15. Watermark extraction

The minimum distance decoding rule is

$$m_i = \arg_{h\text{€}0} \min \ |W_{vi}\,[h] - W_{vi}| \quad i\ \text{€}1,2$$

Where W_{vi} [0] and W_{vi} [1] represent dither modulation result of W_{vi} using d[0] and d[1] as dither value , Vi is the projection vector and mi is the ith extracted watermark signal. During watermark extraction phase, the elements of the signal received at the decoder are quantized using each dither quantizer. The received message is reconstructed from the indices of the sequence of quantizers which contain the reconstruction points closest to the elements. The decoder extracts the embedded information mi based on dither modulation result W_{vi}. It is well known that due to insertion of watermark, there will be degradation in visual quality of the host image (cover image). The degree of deterioration depends on the size of watermark embedded as well as step size used for DM. To achieve that goal, watermark bits are detected using minimum distance decoder and the remaining self-noise due to watermark embedding is suppressed to provide better quality of image. In case of more than two watermark signals, DSTDM can be generalized to multiple spread transform dither modulation (MSTDM). In this situation, the cover vector extracted from the cover work using Rule 1 is projected to multiple (for example, M) projection vectors Vi (i2 1,2,...,M) orthogonal to each other. Then different watermark signals are embedded using DM in different directions, respectively. The extracting method of MSTDM is similar to that of DSTDM.

4. Statistical measures of image robustness

Performance of embedding technique is decided based on some numerical identities such as quality of reconstructed image and extracted information similarity. These are measured with PSNR and bit error ratio respectively.

4.1 PSNR (Peak Signal to Noise Ratio)

The PSNR computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better is the quality of the compressed or reconstructed image. The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower is the error. To compute the PSNR, first calculates the mean-squared error using the following equation:

$$\text{MSE}= \sum_{M,N} \frac{[\text{I}_1(m,n)-\text{I}_2(m,n)]^2}{M \times N}$$

M and N are the number of rows and columns in the input images, respectively. The PSNR is given by the following equation:

$$\text{PSNR} = 10\log_{10}\left|\frac{R^2}{MSE}\right|$$

R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image and the 'noise' is the error in reconstruction. So, if you find a compression scheme having a lower MSE (and a high PSNR), you can recognize that it is a better one. Usually PSNR of more than 35 dB is considered good quality.

4.2 Bit error ratio

Compare the difference between the original binary watermark w and the extracted binary watermark w, and this equals to computing the bit error ratio (BER):

$$\text{BER} = \frac{XOR(w,w^{'})}{L}$$

Where L is length of the binary bit stream of watermark.

5. Summary

The method presented provides effective balance between robustness, complexity, and image quality. Multiple watermark signals are embedded in different orientations of the cover vectors extracted from the cover works, so that different watermark signals will not mutually interfere. Comparing with other relative watermarking techniques, this method yields significant improvements in invisibility and robustness. The proposed method is very flexible and its mathematical background is very clear. Experimental results also show that the presented method can avoid the interference of one watermark signal with another very well, which is one of the most important and difficult problems for a multiple watermarking algorithm and its achieved validity can be 100%.

6. References

[1] Voloshynovskiy, S.et al., ''Attacks on digital watermarks: Classification, estimation based attacks and benchmarks'', IEEE Communications Magazine, vol.39(8), pp.118-126, Aug 2001.

[2] Nasir Memon and Ping Wah Wong, ''Protecting Digital Media Content'', Communications of the ACM, Volume 41, No. 7, pp. 34-43, 1998.

[3] S. P. Mohanty, J. R. Ramakrishnan, and M. S. Kankanhalli, ''A DCT domain visible watermarking technique for images'', IEEE International Conference on Multimedia and Expo, Volume 2, pp. 10291032, 2000.

[4] Hongjie He, Jiashu Zhang, Fan Chen, ''Block-wise Fragile Watermarking Scheme Based on Scramble Encryption'', IEEE International conference on Bio-Inspired Computing: Theories and Applications, PP. 216 220, 2007.

[5] M. M Yeung, F. Mintzer, "An invisible watermarking technique for image verification'', Proceedings of IEEE International Conference on Image Processing, pp. 680 683, 1997.

[6] Yuichi Nakai, ''Semi Fragile Watermarking Based on Wavelet Transform'', Proceeding of the SPIE, Security and Watermarking of Multimedia Contents, pp. 796- 803, 2001.

[7] Van Schyndel, R.G.; Tirkel, A.Z.; Osborne, C.F., ''A Digital Watermark'', Proceedings of IEEE International Conference on Image Processing, pp. 86-90, 1994.

[8] I. Nasir, Ying Weng, Jianmin Jiang, ''Novel Multiple Spatial Watermarking Technique in Color Images'', IEEE International Conference on Information Technology: New Generations, pp. 777 - 782, 2008.

[9] F. Mintzer and G. W. Braudaway, ''If one watermark is good, are more better?'' Proceedings of the International Conference on Accoustics, Speech and Signal Processing, Volume 4, pp. 20672070, 1999.

[10] N. P. Sheppard, R. Shafavi-Naini, and P. Ogunbona, ''On multiple watermarking'' Proceedings of the ACM Multimedia and Security Workshop 2001, ACM Press, pp. 36 , 2001

[11] J. Cox, M. L. Miller, and J. A. Bloom, ''DigitalWatermarking and fundamentals'', Morgan Kaufmann series, San Francisco, 2002.

[12] M Barni, Franco Bartolini, Vito Cappellini, Alessandro Piva, ''A DCT-domain system for robust image watermarking'', Elsevier journal of Signal Processing, Volume 66, No. 3, pp. 357-372, 1998.

[13] Mitchell D. Swanson, Bin Zhu, and Ahmed H. Tewfik, ''Transparent Robust Image Watermarking'', Proceedings of IEEE International Conference On Image Processing, pp. 211-214, 1996.

[14] X. Xia, Charles G. Boncelet, Gonzalo R. Arce, ''A Multiresolution Watermark for Digital Images'' Proceedings of IEEE International Conference on Image Processing, pp.548-551, 1997.

[15] X. Liang; Wu Huizhong, ''Multiple perceptual watermarks using multiple-based number conversion in wavelet domain'', IEEE International Conference on Communication Technology, Volume 1, pp. 213 - 216, 2003.

[16] Lan Hongxing,Chen Songqiao, Li Taoshen, Hu Aina,''A Digital Watermarking Algorithm Based on Dual-tree Complex Wavelet Transform'', IEEE International Conference for Young Computer Scientists, pp. 1488-1492, 2008.

4

Real Time Implementation of Digital Watermarking Algorithm for Image and Video Application

Amit Joshi[1], Vivekanand Mishra[1]
and R. M. Patrikar[2]
[1]Sardar Vallabhbhai National Institute of Technology
Surat
[2]Visvesvaraya National Institute of Technology
Nagpur
India

1. Introduction

Watermarking is the process of hiding a predefined pattern or logo into multimedia like image, audio or video in a way that quality and imperceptibility of media is preserved. Predefined pattern or logo represents identity of an author or rights. In recent years, rapid growth in digital multimedia has been noticed. Digital data (image, audio, and video) is sent through World Wide Web (www) without much effort and money. But security is the main issue in digital multimedia. In the face of these dramatic changes, the entertainment industry has scrambled to adopt a slew of technologies that allow it to retain the copyright controls provided by the law and harness the new world to increase the industry size and enhance the consumer experience.

In recent years, the research community has seen much activity in the area of digital watermarking as an additional tool in protecting digital content and many excellent papers have appeared over the years (Arun Kejariwal,2003). Digital watermarking attempts to copyright the digital data that is freely available on the World Wide Web to protect the owner's rights. As opposed to traditional, printed watermarks, digital watermarks are transparent signatures. They are integrated within digital files as noise, or random information that already exists in the file. Thus, the detection and removal of the watermark becomes more difficult. Typically, watermarks are dispersed throughout the entire digital file such that the manipulation of one portion of the file does not alter the underlying watermark. To provide copy protection and copyright protection for digital image and video data, two complementary techniques are being developed known as Encryption and Watermarking. One more method for data hiding is which is closely correlated with watermarking known as Steganography.Steganography was basically a way of transmitting hidden (secret) messages between allies. There are various data hiding techqniques are available for security. The deatils of each data hiding techniques are presented in next section.

2. Data hiding techniques

Cryptography: It scrambles a message into a code to obscure its meaning. Scrambling of message is done with help of secret key. Scrambling message called as encrypted and it is again decrypted with that secret key only. Cryptography provides security to message. **Steganography:** With Steganography, the sender would hide the message in a host file. The host file or cover message, is the file that anyone can see. When people use this techique, they often hide the true intent for communicating in a more common place communication scenario. In steganography, usually the message itself is of value and must be protected through clever hiding techniques and the "vessel" for hiding the message is worthless.

Watermarking: It is the direct embedding of additional information into the original content or host signal. Ideally, there should be no perceptible difference between the watermarked and original signal and the watermark should be difficult to remove or alter without damaging the host signal. In watermarking, the effective coupling of message to the vessel which is the digital content is of value and the protection of the content is crucial.

In case of steganography, where the method of hiding the message may be secret and the message itself is kept secret; but in watermarking, typically the watermark embedding process is known and the message (except for the use of a secret key) does not have to be secret.Most of the people find difficulty to differentiate term digital watermarking and steognography. Let us take a simple example to understand this difference. If someone gives me a beautiful birthday gift with his name on wraper. Now if I am interested in steagnography approch, I am more willing to see what is inside the wraper so I will open gift without any care of wrapper. While being digital watermarking person, I am interested in wrapper rather then gift provided to me, which gives me a clear indication of the provider. The conecpt of cryptography is totally different then these approaches of data security. Digital content is encrypted at transmitter using a key and can be decrypted at receiver if and only if the correct key is available. Cryptography gives advantage only through the channel. Once encrypted content is decrypted using a key at receiver, no means of security is available for protecting digital content from copyright. Therefore, encryption must be replaced by some method which protects digital content after decryption and there concept of watermarking comes. Another difference between cryptography and watermarking is: cryptography maps the data such that it is unreadable without decryption while, watermarking embeds data maintaining multimedia in its original form.

3. Digital watermarking

These are the parameters important for digital watermarking.

a. Transparency
b. Security
c. Ease of embedding and retrieval
d. Robustness
e. Effect on bandwidth
f. Interoperability

a. Transparency: The most fundamental requirement for any Watermarking method shall be such that it is transparent to the end user. The watermarked content should be consumable at the intended user device without giving annoyance to the user. Watermark only shows up a watermark-detector device.

b. Security: Watermarked information shall only be accessible to only authorized parties. They only have the right to alter the Watermark content. Encryption can be used to prevent unauthorized access of the watermarked data.

c. Ease of embedding and retrieval: Ideally, Watermarking on digital media should be possible to be performed on the fly. The computation needed for the selected algorithm should be least.

d. Robustness: Watermarking must be robust enough to withstand all kinds for signal processing operations attacks or unauthorized access. Any attempt, whether intentionaly or unintentionaly, that has a potential to alter the data content is considered as an attack. Robustness against attack is a key requirement for Watermarking and the success of this technology for copyright protection depends on its stability against attacks.

e. Effect on bandwidth: Watermarking should be done in such a way that it does not increase the bandwidth required for transmission. If Watermarking becomes a burden for the available bandwidth, the method fails.

f. Interoperability: Digitally watermarked content shall still be interoperable so that it can be seamlessly accessed through heterogeneous networks and can be played on various plays out devices that may be aware or unaware of watermarking techniques.

4. Need of hardware implementation

The implementation of watermarking could be on many platforms such as software, hardware, embedded controller, DSP, etc. System performance is a major parameter while designing complex systems. The standard DSP which has Von Neumann style of fetch-operate-write back computation fails to exploit the inherent parallelism in the algorithm. For example, a 30 tap FIR filter implemented on a DSP microprocessor would require 30 MAC (Multiply Accumulate) cycles for advancing one unit of real-time. Further, each MAC operation may consist of more than one cycle as it involves a memory fetch, the multiply accumulate operation, and the memory write back. In contrast, a hardware implementation can store the data in registers and perform the 30 MAC operations in parallel over a single cycle. Thus, high throughput requirements of real-time digital systems often dictate hardware intensive solutions.

FPGAs provide a rapid prototyping platform. They can be reprogrammed to achieve different functionalities without incurring the non-recurring engineering costs typically associated with custom IC fabrication. For commercial applications like movie production, video recording,real on-spot video surveillance,where a real-time response is always required, so a software solution is not recommended due to its long time delay. Since the goal of this research is a high performance encoding watermarking unit in an integrated circuit (IC) for commercial applications, and since FPGAs (field programmable gate arrays) have advantages in both fast processing speed and field programmability, it was determined that an FPGA is the best approach to build a fast prototyping module for verifying design concepts and performance.

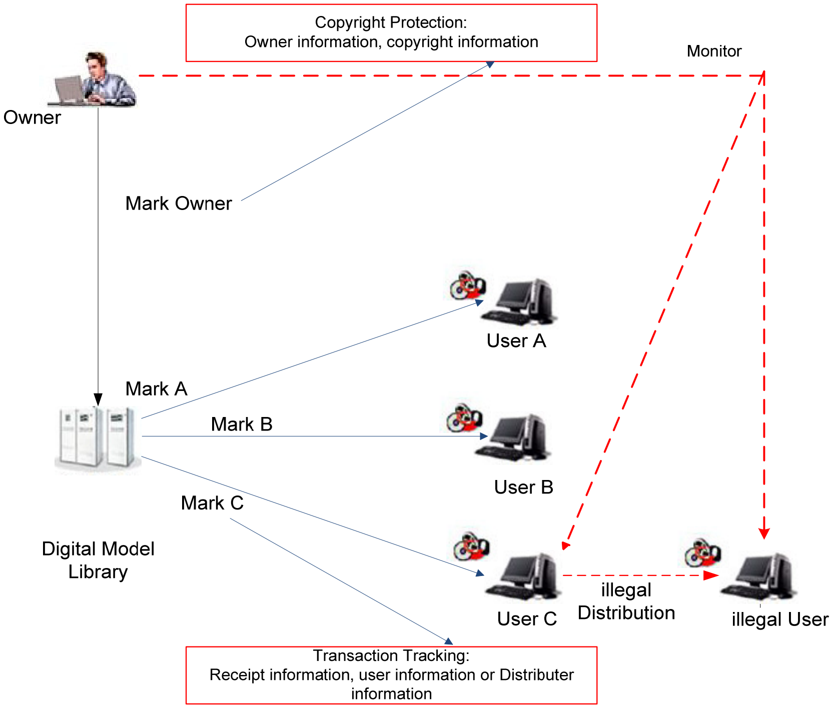


Fig. 1. Copyright Protection service (MarkAny ,2010).

Several software implementations of the watermarking algorithms are available, but very few attempts have been made for hardware implementations. Software implementation of watermarking has been implemented because of their ease of use and flexibility. Mostly software based watermarking works on offline where images are captured through camera and stored on computer and the software for watermarking runs and embeds the watermark and then the images are distributed. This approach has the drawback of certain amount of delay, once images are captured and then watermark is embedded. If attackers would attacks the image before the watermark embedded then it creates issues for ownership of the originator.So there is a need of real-time watermarking where watermark embedding unit reside inside the device (as digital camera) and embedding done directly when image is captured. The hardware implementation of watermarking has advantages in terms of reliability and high performance for area, power and speed. This is very much crucial in some applications like real-time broad casting, video authentication and secure camera system for courtroom evidence. The hardware implementation can have advantage of parallel processin. Since watermarking process deals with processing of watermark and pre-processin of original content before embedding watermark. These two processes are independent and can work in parallel to achieve parallelism to achieve high speed for real-time application.

5. Application of digital watermarking

The digital watermarking technology can be applied to various fields such as copyright protection, transaction tracing, broadcast monitoring and tamper proofing etc.

5.1 Copy-right protection

It is the most common application especially in multimedia object where user inserts copyright information as a watermark or never-copy watermark in a digital content. This watermark can prove his ownership in court when someone has infringed on his copyrights Also number of duplications, manipulations and distribution of digital content can be controlled which are the main sources of illegal process. It is also possible to encode the identity of the original buyer along with the identity of the copyright holder, which allows tracing of any unauthorized copies.

5.2 Transaction tracing fingerprinting

Fingerprint is treated as a transactional watermark. It applies to trace the source of illegal copying of digital content. The owner can embed different unique watermarks for different customers. To trace the source of illegal copies, the owner can use a fingerprinting technique. In this case, the owner can embed different watermarks in the copies of the data that are supplied to different customers. Fingerprinting can be compared to embedding a serial number that is related to the customer's identity in data. It enables the intellectual property owner to identify customers who have broken their license agreement by supplying the data to third party. If misuse of digital content takes place, it is easy to trace out the responsible customer.

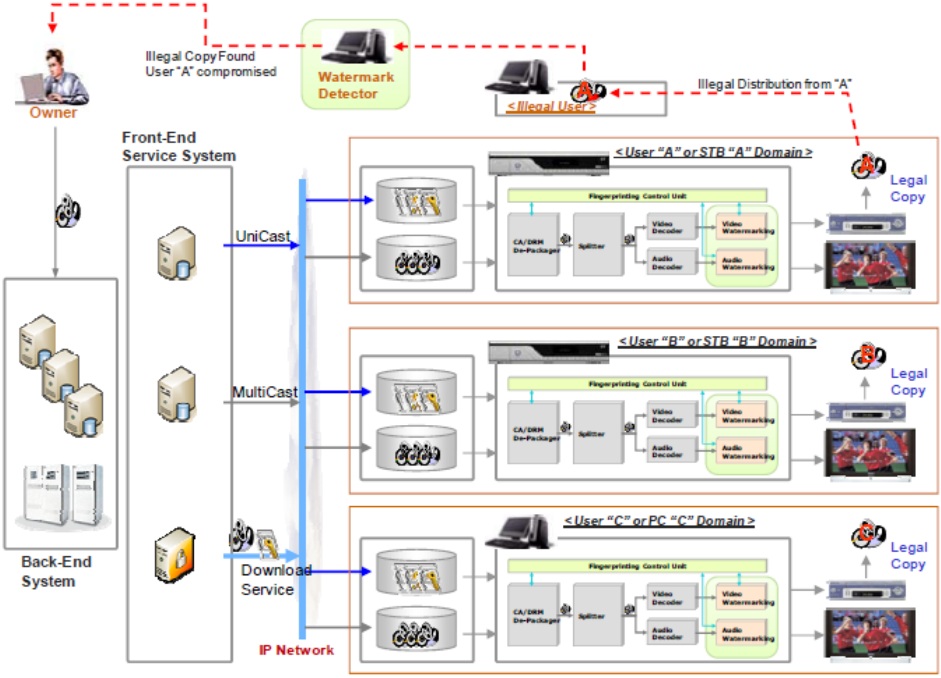


Fig. 2. Video Tracking and finger-printing service (MarkAny, 2010)

5.3 Broadcast monitoring

This application is used by advertisers to broadcast the watermarked information at a specific time and location. Watermarking finds its application to monitor or track the digital content being broadcast, time and location of broadcasting. Specialized equipments are used to track the broadcast channels or radio channels. Upon reception, watermark is detected; content is verified and reported to the broadcasters for true reach of content. It is also useful in finding illegal rebroadcast of copyright information. By embedding watermarks in commercial advertisements an automated monitoring system can verify whether advertisements are broadcasted as contracted. A broadcast surveillance system can check all broadcast channels and charge the TV stations according to their findings. Owners of copyrighted videos want to get their royalties each time their property is broadcasted.

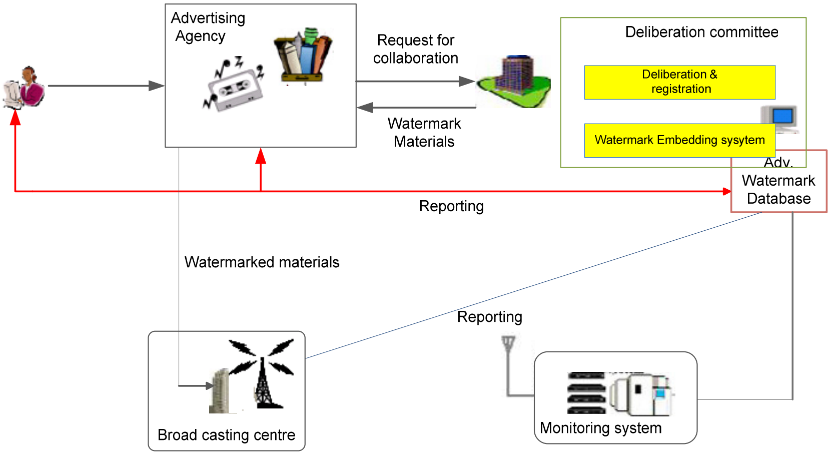


Fig. 3. Broadcast monitoring system (MarkAny, 2010)

5.4 Tamper – proofing

This can be applied to detect existence forgery when contents are forged evil-mindedly and intentionally by embedding watermark information easily damaged by micro-operation. For example, security equipment such as CCTV has been already converted from an analogue system into a digital system, but the data saved by these systems are saved all digitally. However the weakness of digital data that even general users who have personal system can easily operate moving pictures and sound source data which causes a reliability problem for the digital data. A means of judging existence of forgery is necessary to utilize the moving picture data recorded at a digital depository through CCTV as proof data at a court. This can be utilized for a DVR (Digital Video Recorder)/NVR (Network Video Recorder) system, digital camera, camcorder, etc.

6. Implementation of image watermarking

First DCT based semi fragile watermarking algorithm for digital camera with FPGA implementation was developed in paper (Hyum Lim et al,2003). In paper, (Saraju P. Mohanty

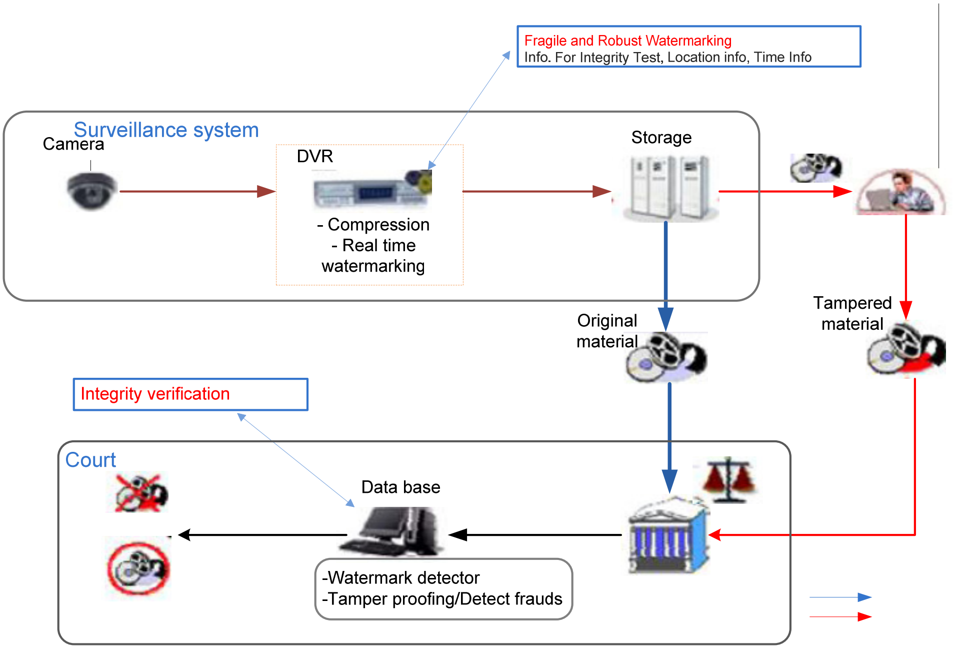


Fig. 4. Tamper proofing Service flow (MarkAny, 2010)

& N. Raganathan 2004) had also developed visible watermarking scheme on DCT. After that algorithms for wavelet based approach were developed to adapt JPEG2000 new millennium standard and to explore multiresolution property of wavelet (Victor V. Hernandez Guzman & Mariko Nankano,2004; Jianyog Huang & Changsheng Yang,2004). In 2005 later part, the proposed watermark (Sammy H. M. Hawk & Edumund Y. Lam, 2002) implementation technique in digital photography with DWT approach for software based implementation. After that next year, development of (Lei Tian, Heng-Ming Tai,2006) secure images captured by digital camera for DWT based approach has been proposed. Another spread spectrum watermarking techniques provides better perceptual transparency and watermark robustness (I.J.Cox et al,1995,1997). This can also developed for secure digital camera application. The watermarking scheme with random binary sequence was developed in paper (A. Lumini & D. Maio,2000). Another watermarking algorithm based on threshold based scheme presnted in paper (Yong-Gang Fu & Hui -Rong Wang,2008). The novel watermarking scheme for block processing method with differential expansion was developed in the paper(Hsien-Wen Tseng & Chin-Chen Chang ,2008). The first attemp to develop simple and efficient watermark technique for JPEG2000 Codec with scattered matrix watermark was presnted in paper (Tung Shou Chen et al,2004). There are many software based implemntation of image watermarking algorithms but very few attempt has been made for hardware implemntation. This will cover in section 6.1 withy detail explanation.

The input image for watermarking algorithm can be either monochrome (black and white) or color image. As with traditional color processing, we first convert a color image from an RGB color space to the YCbCr color space. Then only the Y component of the image is down-sampled to form a grayscale image of resolution of 1 M pixels (assuming the original is between 2 M and 8 M pixels, true for most digital cameras today). Afterwards, a watermark is embedded in the image by quantizing the coefficients of the nth sub band level for DWT of the image. Finally, the Y image plane is converted back to spatial domain by IDWT and a watermark image is formed by up sampling the image and adding it with the original Y, Cb and Cr color components. For extraction process, the user has access to watermark (w), the coefficient selection key (c key) (A.J. Menezes et al,1996; B. Sehneier,1996) and the original image incase of non blind watermarking. Since only the user knows the secret key for the watermarking therefor security against forgery is guaranteed.

As stated earlier, only luminance component Y is down sampled to embed the watermark. The down sampling is a lossy process and thus down sampling of chroma signals (Cb and Cr) are lost that can not be retrieved by reverse process of up sampling at receving end. The complete process of down and up sampling of Lena 256 x 256 color image is shown in this section. First color images which comprises of RGB components are converted to YcbCr, where Y contains luminance information and Cb and Cr contains the chrominance information of the image. Then Y signal is down sampled at factor two to obtain down sample image. This image is used for wavelet processing and to embed the watermark. This image is again up sampled with factor two to obtain recover Y component. The complete process shown in Fig. 5.

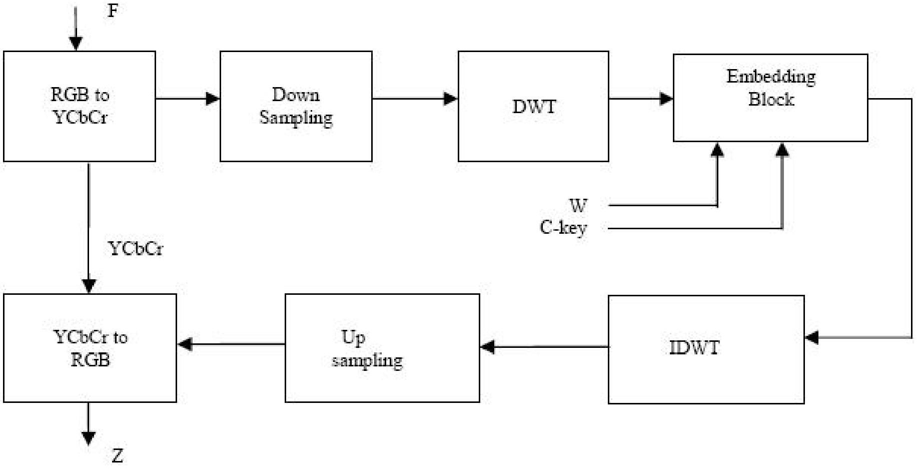


Fig. 5. Watermark Embedding Process

The problem with process that up sampling adds zero's to the images, so the image after up sampling is distorted (recovered Y component) as shown in above Fig 6. To over come this problem, we simply add the original Y component value to the zero paddied values as shown in below in Fig. 7.

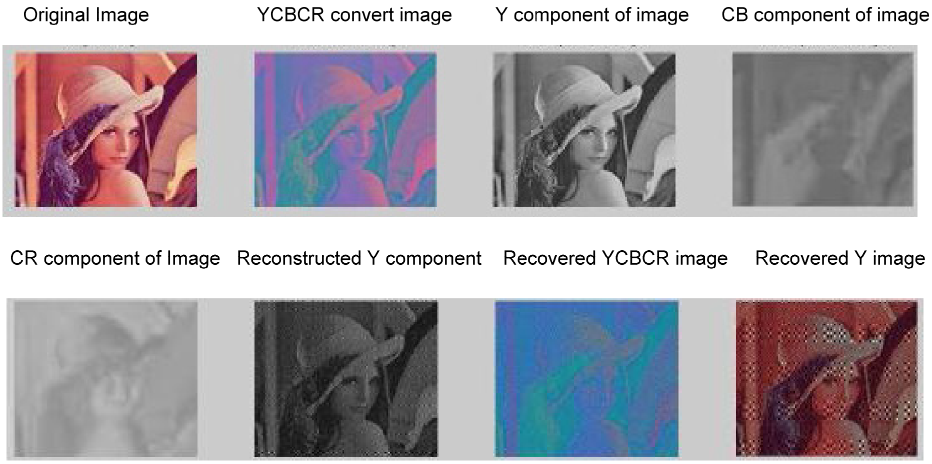


Fig. 6. Down sampling and up sampling process of Y component

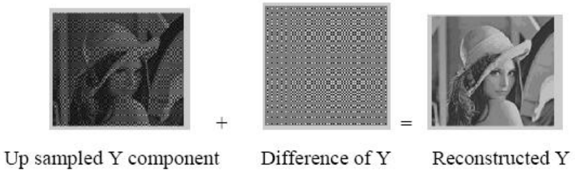


Fig. 7. Reconstructed Y plane after up sampling

The complete modified process is shown in below Fig 8.

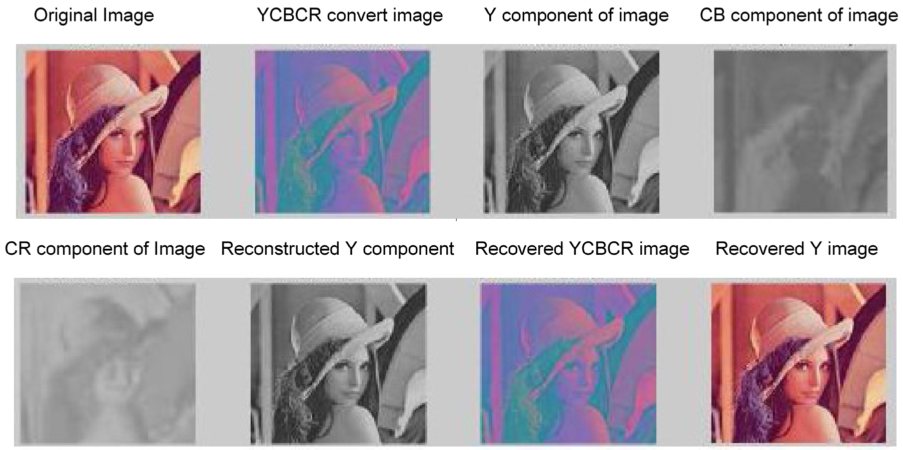


Fig. 8. Modified Down-Up sampling process with proper reconstruction of Image

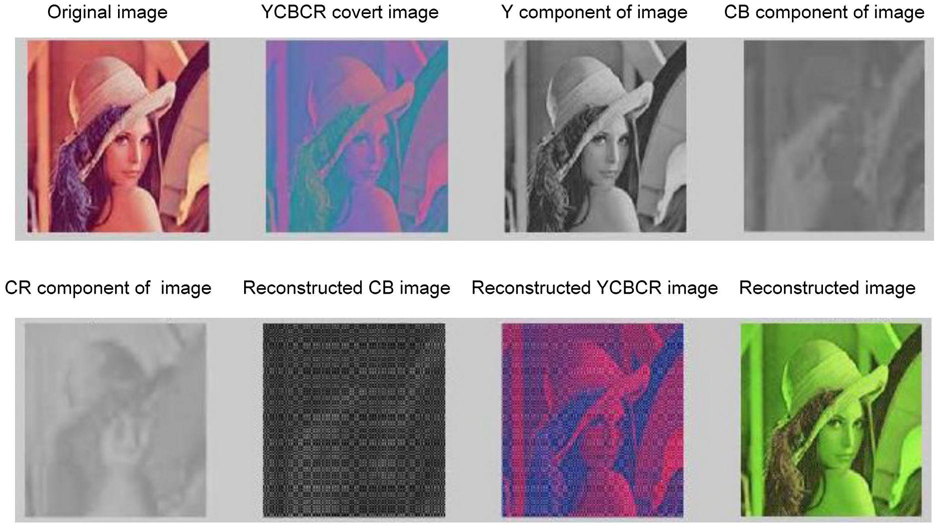


Fig. 9. Down sampling and up sampling process for Cb component

The chromináñce signal can also be down sampled as shown in Fig 9 for Cb component and Fig 10 for Cr component. We can see the difference between original and recovered up sampled image in Fig 9 and Fig 10. As we can see, luminance component Y signal down/up sampling process provides better results compare to chrominance signal Cb and Cr down/up sampling.

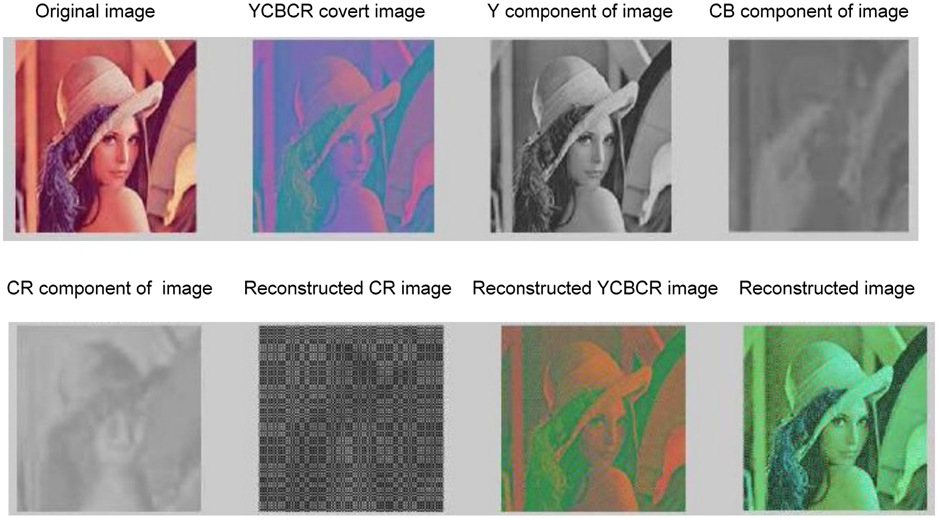


Fig. 10. Down sampling and up sampling process for Cr component

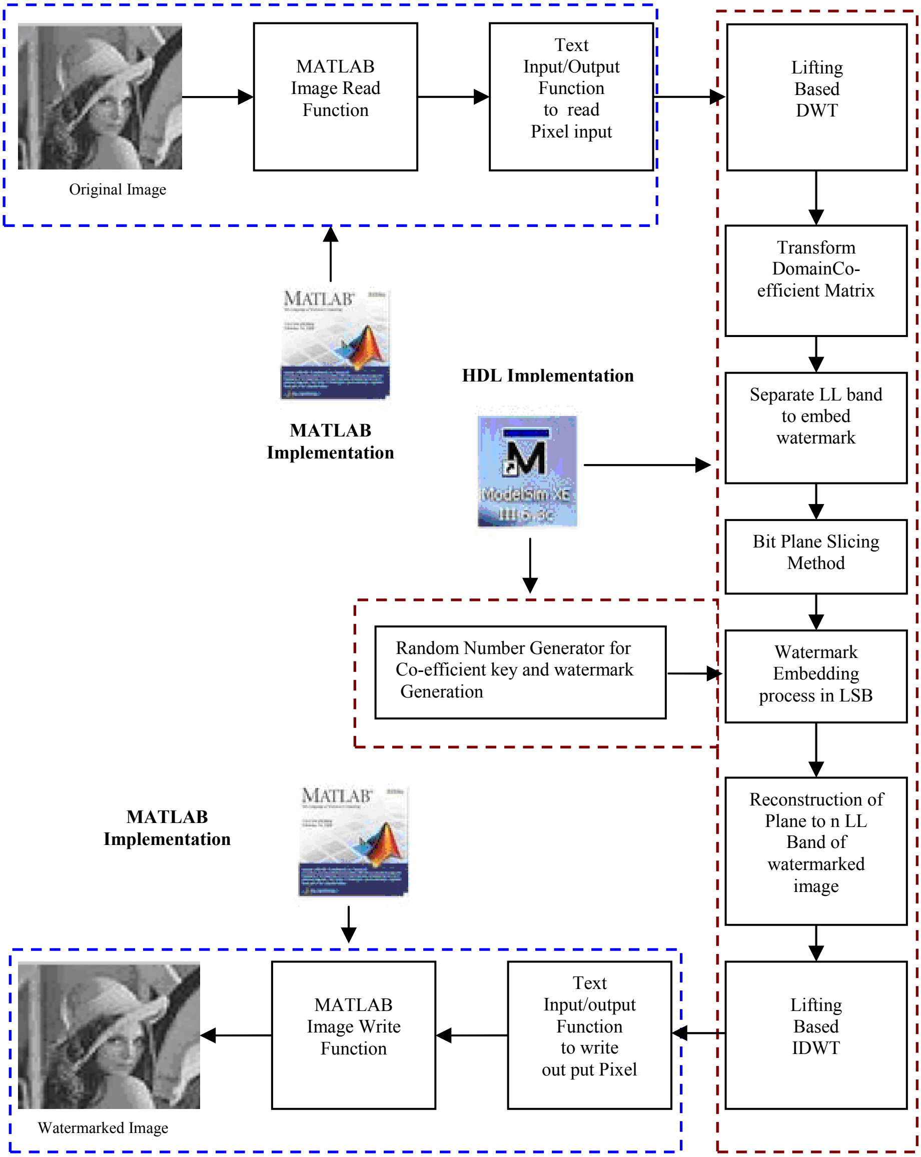

Original Image
MATLAB Image Read Function
Text Input/Output Function to read Pixel input
Lifting Based DWT
Transform DomainCo-efficient Matrix
MATLAB
MATLAB Implementation
HDL Implementation
ModelSim XE III 6.3c
Separate LL band to embed watermark
Bit Plane Slicing Method
Random Number Generator for Co-efficient key and watermark Generation
Watermark Embedding process in LSB
MATLAB Implementation
MATLAB
Reconstruction of Plane to n LL Band of watermarked image
Watermarked Image
MATLAB Image Write Function
Text Input/output Function to write out put Pixel
Lifting Based IDWT

Fig. 11. Embedding scheme for watermarking

6.1 Proposed algorithm

Spatial-domain digital watermarking methods are generally considered as having poor performance after geometric distortion (such as cropping and scaling), common signal processing (such as JPEG and filtering). It is efficient in terms of less computational cost due to their easy operation.On the other hand, frequency-domain watermark techniques, have their high computation complexity, and provide great robustness to different attack. To increase the robustness, we have to increase number of sub band level which require more computational cost. However, we have adopted the combined approach with spatial-frequency domain approach which has advantages of both domains. The frequency domain transformation is done with lifting based wavelet scheme and spatial domain transformation done with bit plane slicing. The steps of algorithms have been described in paper (Amit joshi,2009). The implementation flow for proposed scheme is shown in Fig 11. The image is read through MATLAB and pixel is stored in datain.txt file. With help of text I/O package, the datain.txt file has been read in VHDL and Legall 5/3 based Lifting wavelet applied to obtain transform domain co-efficient matrix. LL band coefficient stored in separate memory to embed watermark. The RTL code of bit Plane slicing has been developed to separate different planes from LSB to MSB. To the selected co-efficient generated by random number generator, then watermark has been added to them. Then all planes are reconstructed with bit plane slicing RTL code to obtain LL band of watermarked image. Then lifting based legall 5/3 IDWT has been applied to obtain pixel values. The MATLAB function is used to construct watermarked image.

6.1.1 Hardware implementation of wavelet

For implementation of hardware efficient DWT based scheme, lifting based scheme obviously far better than traditional convolution scheme. Lifting based wavelet scheme used in various approaches like Daubechies 9/7 and Le Gall 5/3. But Le Gall 5/3 is proven more hardware efficient due to its simplicity and lossless implementation. The odd and even samples values can be calculated by following equations (1) and (2) are

$$y(2n+1) = x(2n+1) - [\frac{x(2n) + x(2n+2)}{2}] \tag{1}$$

$$y(2n) = x(2n) + [\frac{y(2n-1) + y(2n+1) + 2}{4}] \tag{2}$$

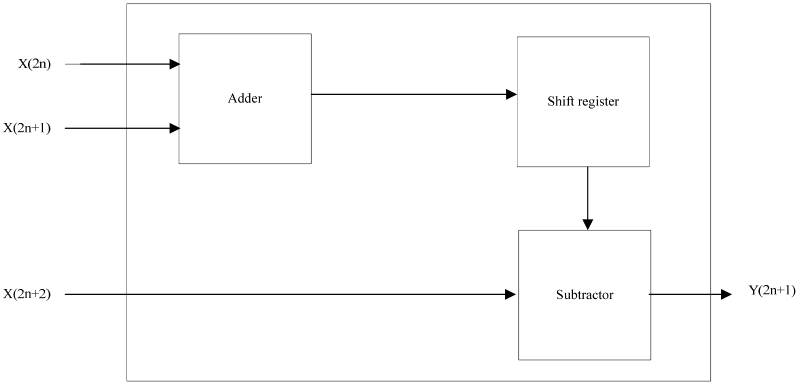


Fig. 12. Predict Phase

To implement this algorithm, the equations stated earlier are utilized. In lifting scheme this algorithm is divided in two phases: predict phase and update phase. In order to find the value for predict phase, simultaneously three inputs are required as per eq. (1). Similarly in the update phase only one even input and two values obtained form predict phase are required as per eq. (2). The 8 bit-gray scale image of LENA is used for performance of Legal 5/3 DWT. The architecture module of predict phase and update phase are shown in Fig. 12 and Fig. 13 respectively.

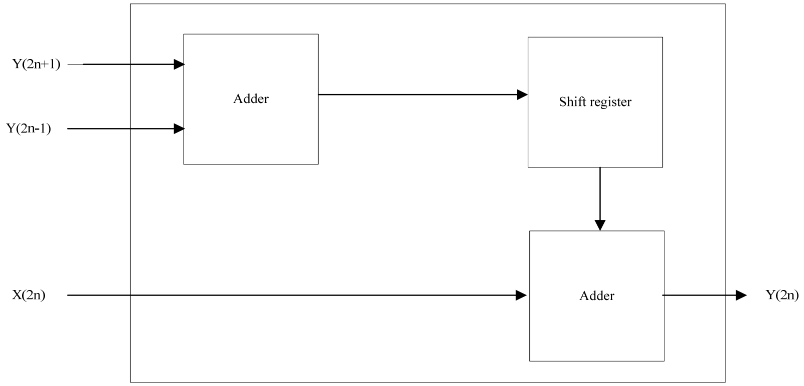


Fig. 13. Update Phase

6.1.2 Memory management

As suggested, to obtain the forward wavelet transform, initially we need to read the three input data. And from these we are get two coefficients detail (high) and approximate (low). One needs to manipulate the co-efficient of the image to obtain the correct output. In VHDL, two dimensional matrixes are not synthesizable. So if we are interested in reading the image having size n X n, then total n^2 memory location are required to store each input pixels. For this, 64 X 64 gray scale image are utilized in the wavelet transform, data is processed row wise and then after columnwise. The memory orgnization is as shown in Fig. 14.

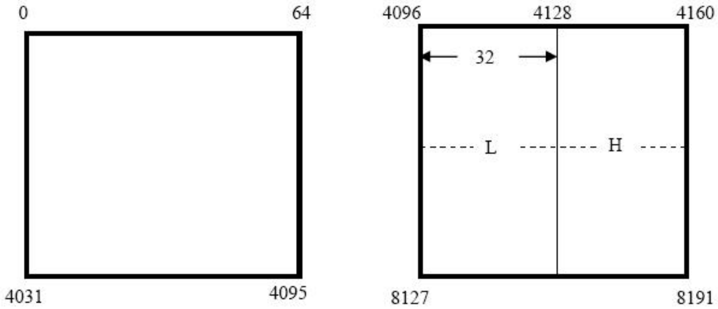


Fig. 14. Memory Management of Wavelet transforms

6.1.3 Watermarking embedding hardware implementation

There are two basic blocks required for watermark embedding process.

a. Bit Plane Slicing scheme implementation
b. Random Number Generator for key selection and watermark generation.

a. Bit Plane Slicing Implementation: It is the spatial domain techniques. In Spatial domain scheme, watermark is directly embedded in the pixel values. Algorithm splits the image into 8 planes from MSB to LSB. The whole concept is explained in details as shown in Fig. 15. Suppose pixel values in binary foramt read from memory as shown here:

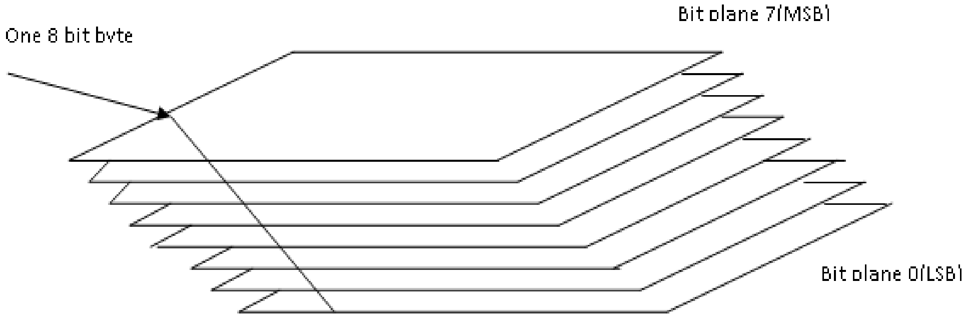


Fig. 15. Bit planes of a Image

01111001 01100101 01001010 00100110 10000100 10000110 10001001 10001101

The values read from memory are taken one by one in temp variable.

To separate out the values in different planes, we will xor the temp values with standard values as follows. Here first value read from memory and stored in temp is **01111001.**

- LSB plane : 01111001 and 00000001 : The resultant values is 0000000**1**.
- Seventh plane: 01111001 and 00000010 : The resultant values is 000000**0**0.
- Sixth plane : 01111001 and 00000100 : The resultant values is 00000**0**00.
- Fifth plane : 01111001 and 00001000 : The resultant values is 0000**1**000.
- Fourth plane : 01111001 and 00010000 : The resultant values is 000**1**0000.
- Third plane : 01111001 and 00100000 : The resultant values is 00**1**00000.
- Second plane : 01111001 and 01000000 : The resultant values is 0**1**000000.
- MSB plane : 01111001 and 10000000 : The resultant values is **0**0000000.

Next time another value of pixel read from memory and stored in temp as 01100101 and same procedure is followed as above. In this way all LSB plane co-efficient are obtained. The reconstruction of the planes are also very simple. Finally we have to just add all the resultant values planes to obtain original value. With addition of all plane values we obtain:

00000001 + 00000000 + 00000000 + 00001000 + 00010000 + 00100000 + 01000000 + 00000000 = **01111001**.

b. Random Number Generator: It is one of the important blocks of watermarking process. Basically its role is to generate the coefficient selection key and embed watermark to original content. As shown in Fig 16, it has 8 bit D-flip flop which are used so that the maximum number of co-efficient selection key from random number generator is 255. The watermark is added according to key generated at the output. The random number generator is started with secret key provided as its initial state. We have used 10101101 as key that serves as the initial seed to start random number generation. The same key has been used with same random number at the detection side which generate same pseudo sequence to retrieve the watermark.

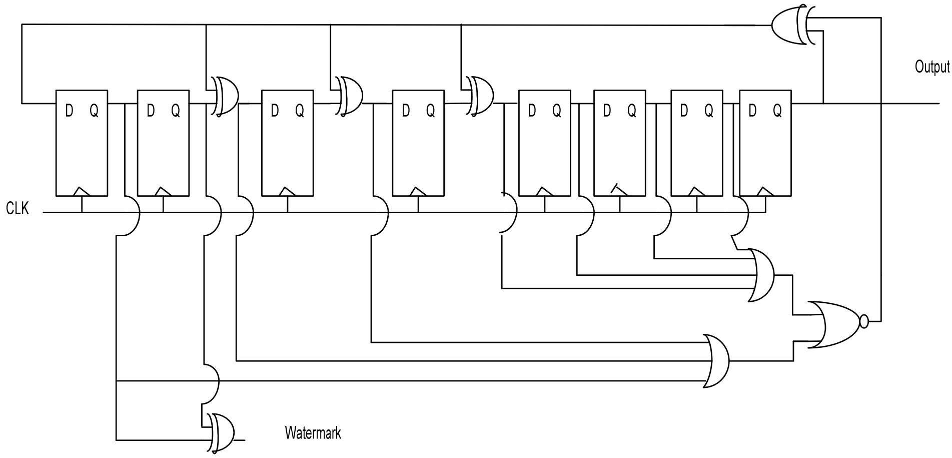


Fig. 16. Random Number Generator

6.1.4 VLSI architecture of image watermarking algorithm

The architecture design proposed for scheme is defined as shown in Fig 17. The main memory comprises the memory space which is twice the size of original image size as it has to store original values and watermarked value. For example, the size of image be 256 x 256, the main memory requirement is 2*256*256=1, 31,072. Now the memory is divided into two parts as RAM1 for original image and second for RAM2 for watermarked image. At the time of detection for non blind scheme, the values in RAM1 are considered as original pixels values and RAM2 are watermarked values. As explained earlier in section 6.1, wavelet scheme based on lifting based legal 5/3 method requires three values to read from RAM1.

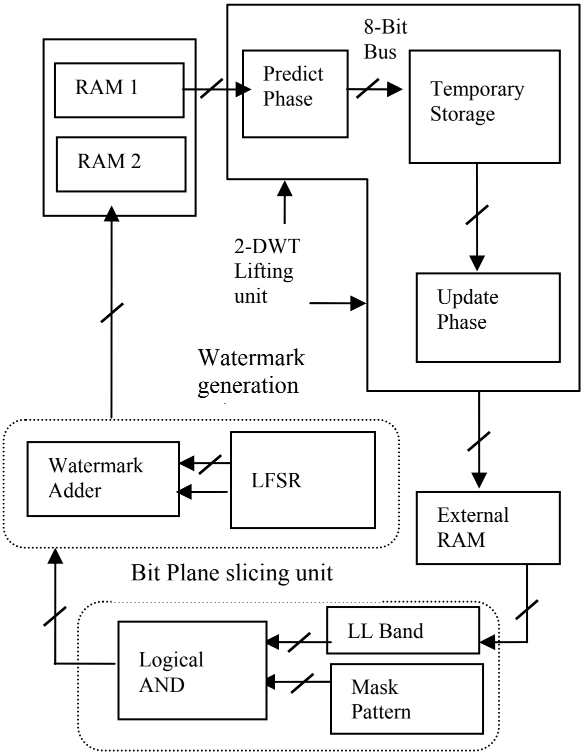


Fig. 17. VLSI Architecture of proposed algorithm

6.1.5 Pin diagram

The pin diagram for wavelet based spatial domain watermarking chip is shown in the Fig. 18. The functional description of each pin is:

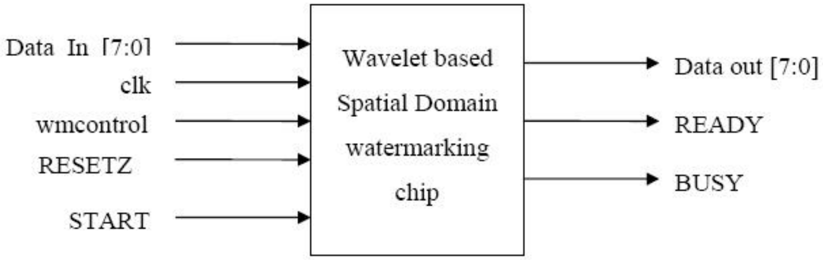


Fig. 18. Pin Diagram

Data In [7:0]: DATA Input bus. Original pixel values which were stored will be input on this bus for operation.

CLK: Clock signal to chip.

Wmcontrol: Enables during embedding the watermark.

RESETZ: It is active low signal to reset the chip

START: It is an active low handshake signal to initiate data transfer operation on Data In bus on every clock edge.

Data out [7:0]: DATA output bus. Watermarked pixel values are output on this bus.

READY: It is active high signal will be activated for one CLK cycle after the completion of watermark embedding operation. It indicates Data out bus has valid out on bus.

BUSY: It is a active high signal. It indicates Watermarking is in progress. When external signal is high which indicates, external access to RAM1's are isolated. The data on data bus out is not valid.

6.2 Hardware implementation results

The simulation results for Legall 5/3 is as shown in Fig. 19.

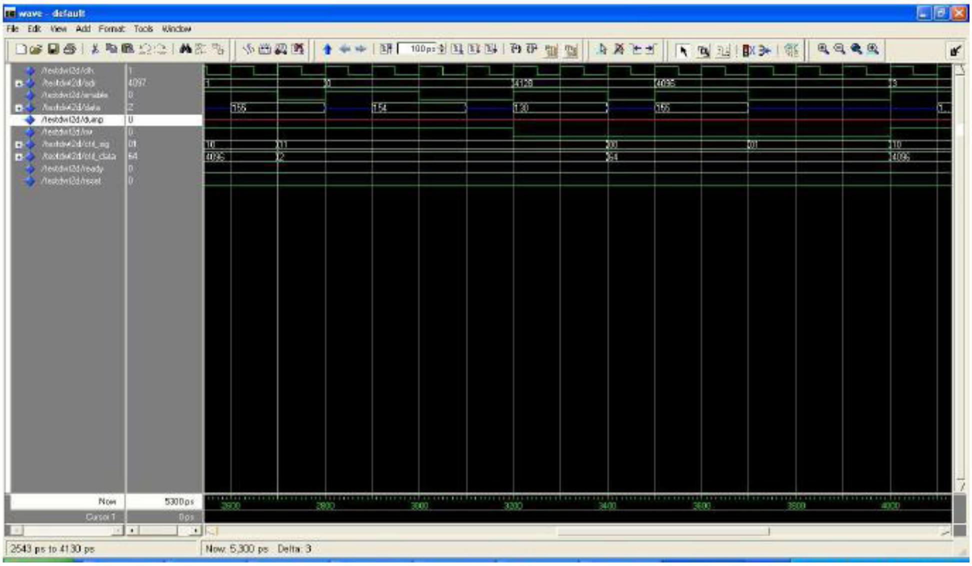


Fig. 19. Simulation of Legal 5/3 wavelet

6.2.1 FPGA results

Synthesis was performed with help of Xilinx project navigator ISE 9.1 software EDA tool. During simulation, textio library was utilized to read the gray scale image (lena 256 x 256) file. After processing, results are stored in text file. This text file is read through

MATLAB to generate image. Synthesis report and device utilization reports for proposed DWT, IDWT and Watermarking Processor is shown in Table 1 and Table 2 respectively. The results are obtained for Xc3s500e-4fg320-SPARTAN 3E FPGA using ISE 9.1 from Xilinx Tool.

Resources	DWT Processor	Watermarking Processor	IDWT Processor
RAM/ROM	3	2	4
Adders/Subtractor	16	2	14
Latches	12	5	10
Muxs	4	3	4
Counters	3	2	3
Registers/Flip Flop	469	62	542

Tt

Table 1. Synthesis Report for Proposed DWT,IDWT and Watermarking Processor

Resources	DWT Processor	Watermarking Processor	IDWT Processor
Slices	(535/4656)	(153/4656)	(570/4656)
Slices FFs	(395/9312)	(117/9312)	(410/9312)
LUTs	982	335	925
I/O	70	25	70
GCLKs	3	1	3

Table 2. Device Utilization Report for Proposed DWT, IDWT and Watermarking Processor

6.2.2 ASIC results

The proposed scheme requires three major blocks to embed the watermark as DWT, IDWT and Watermark process. We have calculated area and power with Design compiler using standard cell library of Farady 0.18 um technology as shown in Table 3. In Table 4, It also has been compared with other existing scheme.

Block	Type	Area (um*um)	Dynamic Power
DWT	1 Dimensional	12196	2.0592 mW
DWT	2 Dimensional	15770	8.6813 m W
IDWT	1 Dimensional	13237	2.8531 mW
IDWT	2 Dimensional	18106	9.3752 mW
Watermarking Processor	Bit Plane Slice	192	23 uW
Watermarking Processor	Binary Number Generator	322	48 uW

yy

Table 3. Area and dynamic power results for proposed scheme

Research Work	Watermarking Type	Processing Domain	Technology	Power	Execution Time
Tsai and Lu,2001	Invisible robust	DCT	0.35 um	107.6 uW	1.494 ms
Garimella et Al,2003	Invisible Fragile	Spatial	0.13um	82 uW	1.3059 ms
Saraju P. Mohanty et Al,2005	Visible	Spatial	0.35um	72uW	0.914ms
Saraju P. Mohanty et al,2006	Invisible Robust	DCT	0.35um	90 uW	1.125ms
Proposed Scheme	Invisible Robust	Wavelet	0.18um	69uW	0.893ms

Table 4. Summary of Watermark Custom IC Hardware Description in the literature of Watermarking

7. Video watermarking

In today's multimedia technology, the most widely used object is a video. Therefor maximum occurrences of copyright infringement and abuse happen for video media content. Video is sequence of frames and each frame is considered as a still image. But the challenges for video watermarking are as follows:

a. Video media is susceptible to increased attacks than any other media.
b. Video content are sensitive to subjective quality and Watermarking may degrade the quality.
c. Video compression algorithms are computationally intensive and hence there is less headroom for Watermarking computation.
d. Video is bandwidth hungry and that is why it is mostly carried in compressed domain. Therefore, Watermarking algorithm shall be adaptable for compress domain processing.
e. For low-bit rate video, Watermarking poses additional challenges, as there is less room for watermark data.
f. During video transmission, frame drops are very usual. If watermark data spreads over many frames, in that case watermark data may become irretrievable. Watermarking should be robust enough against this phenomenon.

The easiest way to embed the watermark in video is consider each of frame of video as still image and apply image watermarking algorithm. So algorithm which described in section 6.1 still holds quite comparable results when applied to video. One algorithm developed for video is presented in paper for wavelet domain (Amit Joshi & Vivekanand Mishra,2011). But with this approach, we are not utilizing the temporal dimension of video. Same way, many algorithms for developing watermarks on images are extended for videos. Some points need to be considered during the extensions. First one is between the frames there exists a huge amount of intrinsically redundant data. So we can explore that before embedding the watermark. Second is there must be a strong balance between the motion and the motionless regions. And another one is strong concern must be put forth on real time and streaming video applications. Video Watermarking mainly done in uncompressed (raw data) domain or in compressed domain. The raw watermarking is classical approach of video watermarking scheme. In this classical approach, to apply a watermark, firstly the compressed video stream is to decompress. Use a spatial domain or transform- domain watermarking technique to apply the watermark, and then recompress the watermarked video. The disadvantages of classical approach is that watermark embedded has no knowledge of how the video will be recompressed and cannot make informed decisions based on the compression parameters. This approach treats the video compression process as a removal attack and requires the watermark to be inserted with excessive strength, which can adversely impact watermark perceptibility. Another issue with classical approach is that compression step is likely to add compression noise, degrading the video quality further. The main drawback is that fully decompressing and recompressing the video stream can be computationally expensive. A faster and more flexible approach to apply watermarking on compressed video is well know as compressed-domain watermarking. In compressed-domain watermarking, the original compressed video is partially decoded to expose the syntactic elements of the compressed bit stream for watermarking (such as encoded transform coefficients). Then, watermark is embedded in the partially decoded bit stream and again reassembled to form the compressed watermarked video. The watermark insertion process ensures that all modifications to the compressed bit stream will produce a syntactically valid bit stream that can be decoded by a standard decoder. The watermark embed process has access to information contained in the compressed bit stream, such as prediction and quantization parameters, and can adjust the watermark embedding accordingly to improve robustness, capacity, and visual quality.

Similar to image watermark implementations, the video watermark system can be implemented in either software or hardware, each having advantages and drawbacks. In software, the watermark scheme can simply be implemented in a PC environment. The watermark algorithm’s operations can be performed as scripts written for a symbolic interpreter running on a workstation or machine code software running on an embedded processor. By programming the code and making use of available software tools, it can be easy for the designer to implement any watermark algorithm at any level of complexity. However, such an implementation is relatively slow and therefore not suitable for real time applications. In practical, video storage and distribution systems, video sequences are stored and transmitted in a compressed format. Thus, a watermark that is embedded and detected directly in the compressed video stream which can minimize computational demanding operations. Furthermore, frequency domain watermark methods are more robust than the spatial domain techniques(Xian Li,2008). Therefore, working on compressed rather than uncompressed video is important for practical watermark applications. There are few standards for video compression. All current popular standards for video compression, namely MPEG-x (ISO standard) and H.26x formats (ITU-T standard), are hybrid coding schemes and are DCT based compression methods. Such schemes are based on the principles of motion compensated prediction and block-based transform coding. Currently, researchers are given more focus on recently developed H.264 based video watermarking standard for low bit rate video application.

7.1 Compressed domain video watermarking

H.264/MPEG4-AVC is the latest video coding standard of ITU-T Video CodingExperts Group(VCEG) and the ISO/IEC Moving Picture Expert Group(MPEG). H.264/MPEG4-AVC has recently become the most widely accepted video coding standard since the deployment of MPEG2 at the drawn of digital television, and it may soon overtake MPEG2 in common use. It covers all common video application ranging from mobile services and video conferencing IPTV,HDTV and HD video storage. The H.264 standard has a number of advantages that distinguish it from existing standards, while at the same time, sharing common features with other existing standards like up to 50 % of bandwidth sharing, high quality video and error resilience.

The paper (Frank Hartung, Bernod Girod, 1998) presented spread spectrum based watermark embedding method for additive digital watermarks into video sequences in uncompressed and compressed video sequences. It adds pseudo-noise signal to the video with invisible and robust against manipulations. For practical applications, watermarking schemes operating on compressed video are desirable. The watermark is processes through discrete cosine transform (DCT) and embedded into the MPEG-2 bit-stream without increasing the bit-rate. The watermark can be retrieved from the decoded video and without knowledge of the original video.

The authors (Karima Ait Saadi et al.,2008) proposes a new block based DCT selection and a robust video watermarking algorithm to hide copyright information in the compressed domain of the emerging video coding standard H.264/AVC. The watermark is first quantized and securely inserted. To achieve invisibility and robustness, the high entropy DCT 4x4 blocks within the macro blocks are elected to minimize the distortion caused by the embedded watermark and then scrambled using Linear Congruential Generator (LCG) technique. This approach provides good robustness against some attacks such as re-compression by the H.264 codec, transcoding and scaling.

The authors (Jing Zhang and Anthony T. S. Ho ,2005) presents a byte replacement watermarking for direct stream marking of H.264/AVC streams. This paper describes a method for applying a watermark directly to an entropy coded H.264/AVC stream. This method can be applied when the stream, or at least the I-frames, is entropy coded with CAVLC. The embedding process involves replacing each identified segment with one of the alternative values from encode VLC table. The choice of alternative is informed by the payload to be embedded.

In this method (Dekun Zou, Jeffrey A. Bloom,2008), a grayscale watermark pre-processing is adapted for H.264/AVC. 2-D 8-bit watermarks such as detailed company trademarks or logos can be used as inconvertible watermark for copyright protection. A grayscale watermark pattern is first modified to accommodate the H.264/AVC computational constraints, and then embedded into video data in the compressed domain. With the proposed method, the video watermarking scheme can achieve high robustness and good visual quality without increasing the overall bit-rate.

They (Jing Zhang.2007) proposes robust MPEG-2 video watermarking techniques , focusing on commonly used typical geometric processing for bit-rate reduction, cropping, removal of any rows, arbitrary-ratio downscaling, and frame dropping. Both the embedding and the extraction of watermarks are done in the compressed domain, so the computational cost is low. Moreover, the watermark extraction is blind. The presented technique is applicable not only to MPEG-2 video, but also to other DCT-based coding videos.

The author (Satyen Biswas,2005) propose a new adaptive compressed video watermarking scheme uses scene-based multiple gray-level watermark that provides more perceptual information. The concept of human vision system (HVS) is employed to find a suitable set of DCT coefficients for watermark embedding. The developed method embeds several binary images, decomposed from a single watermark image, into different scenes of a video sequence. The spatial spread spectrum watermark is embedded directly into the compressed bit streams by modifying discrete cosine transform (DCT) coefficients. The proposed watermarking scheme is substantially more effective and robust against spatial attacks such as scaling, rotation, frame averaging, and filtering, besides temporal attacks like frame dropping and temporal shifting.

7.1.1 Watermarking embedding hardware implementation

a. Integer DCT Transform based watermarking:

The discrete cosine transform (DCT) is a very promising technique used for video/image coding, and widely adopted by most image and video compression standards including latest H.264 standard. Since increasing applications apply these standards to portable systems like hand-held videophone and multimedia terminals, it becomes imperative to develop a high speed and low complexity DCT chip as one key component for such applications. To support low power applications, it is necessary to minimize the computational complexity as much as possible. For the high speed of operation and low delays pipelining structure is used, which also reduces the resource utilization. H.264 supports Integer based DCT for low complexity and high speed. The 2-D integer DCT is obtained on columns and row processor of 1-D DCT.The detailed architecture is shown below in Fig. 20. First input pixels are read from Memory and then 1-D DCT for column processing is done which stores in transpose memory. In transpose memory, transposing of values are done. The values of column DCT is written on horizontal manner while reading of values are read on vertical basis and then applied to 1-D DCT for row processor.

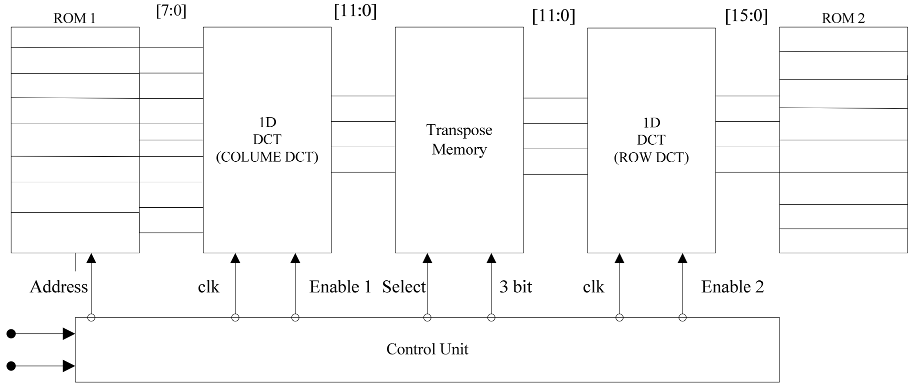


Fig. 20. VLSI Architecture for 2-D DCT.

b. Watermarking Embedding Hardware Implementation

The algorithm presented (Yulin Wang, Alan Pearmain, 2004) is blind watermarking based on scene detection. The algorithm is adapted on hardware where Integer DCT is utilized. The steps for hardware implementation shown in Fig. 21. This algorithm is implemented with simple multiplier, shifter and adder/sub tractor.

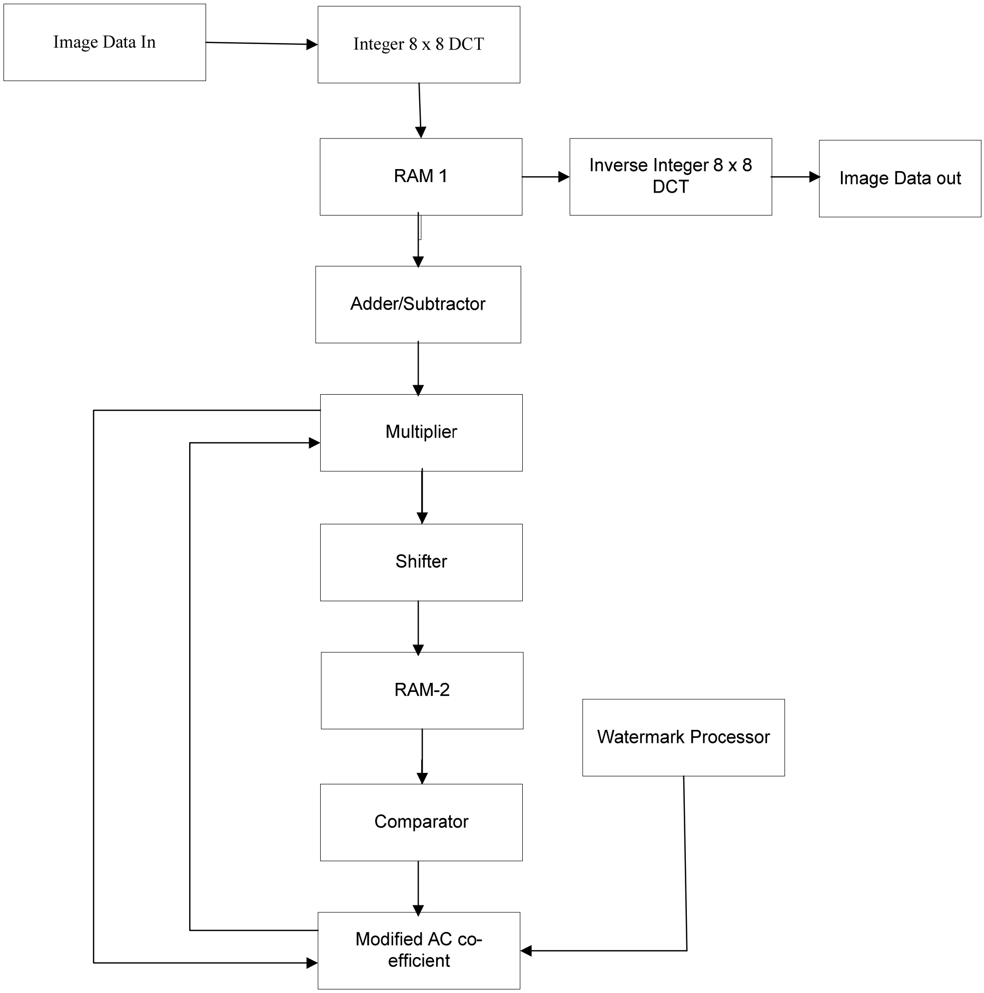


Fig. 21. VLSI Architecture for Integer DCT based watermarking

The input values coming from video capturing devices such as digital camera coming and stored in memory and our DCT column processor run on that stores transformed values in transpose memory and transposing values are given again to row processor. The schematic of video watermarking is as shown in Fig. 22.

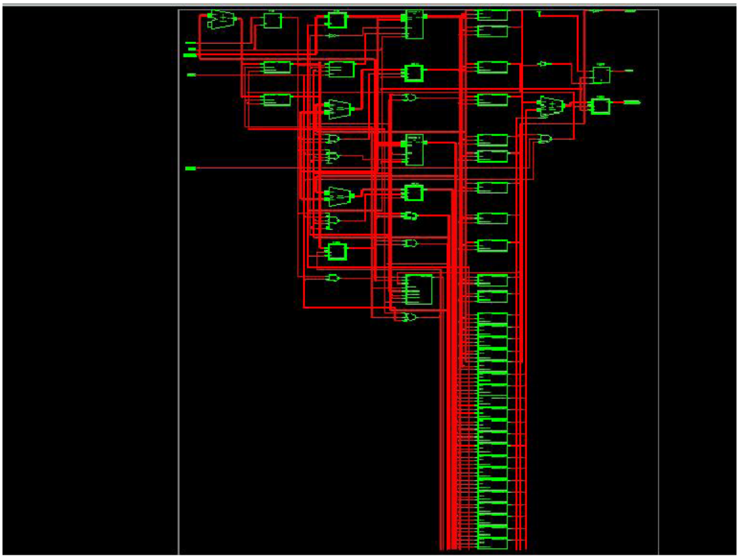


Fig. 22. Schematic Design of Integer based DCT watermarking

The device utilization of above algorithm is as shown in following Table 5.

Resources	Number of Utilization	Percentage of utilization
Number of Slices	70 out of 4656	1 %
Number of Slices Flip Flops	105 out of 9312	1%
Number of 4 input LUTs	129 out of 9312	1%
Number of IOBS	40 out of 158	24%
Number of GCLKS	1 out of 24	4%

Table 5. Synthesis Report of Video watermarking algorithm

8. Conclusion

The proposed algorithm is applicable for image and video application. It has combined approaches of spatial and frequency domain. From Table 1-3, it has been conclude that proposed scheme is suitable for real-time application due to its simplicity. It has overcome the problem of block artifacts of DCT and advantage of both domain properties. It has also lesser computational complexity compare to other algorithms because we embed watermark in Legal 5/3 Integer wavelet transform. From Table 4 of ASIC results taken from design vision from Synopsis also shown that proposed scheme has comparable results for speed and power compare to other existing schemes. From Table 5, it has been verified that propose video watermarking algorithm provides hardware efficient algorithm.

9. Acknowledgment

I am thankful to Prof. Anand Darji who provides me guidance to complete this chapter. would also like to thanks all my friends who directly and indirectly helping me every bit. I am obliged to the in hand support that has been given to me by the my collegaues and especially to the "SMDP Project Lab", which has provided me with the platform like ISE (Xilinx Tool), Modelsim (Mentor Graphics), MATLAB, Design Vision (Synopsys Tool) which could navigate me through the sea of experiments and help me too get the desired level of work satisfaction.. Also quite thankful to SMDP Lab for providing me 180 nm CMOS Standard Cell Library for post synthesis of my design.

10. References

A.J. Menezes, P.V. Oorcscot and S.A. Vanstone (1996), *Handbook of Applied cryptography*, CRC press, Boca Raton.

A. Lumini and D. Maio (2000), A Wavelet-Based Image Watermarking Scheme, *The International Conference on Information Technology: Coding and Computing*, Las Vages, NV, pp. 122-127.

Amit Joshi, Vivekanand Mishra (2011), Blind video watermarking of wavelet domain for copy right protection ,*International Journal of Computing*, Vol 1, Issue 3,pp 291-295.

Amit Joshi, Prof.A.D.Darji (2009), Efficient Dual Domain Watermarking for secure images, *An International Conference on dvances in Recent Technologies in Communication and Computing, 2009. ARTCom '09, pp. 909-914.*

Arun Kejariwal (2003), Watermarking, *IEEE Potential*, October/November, 2003.pp 37-40.

B. Sehneier (1996), *Applied Cryptography: Protocols, Algorithms and Source Code in C*, second edition, John Wiley & Sons, New York.

Dekun Zou, Jeffrey A. Bloom(2008): H.264/AVC stream replacement technique for video watermarking. *ICASSP* 2008: 1749-1752

Frank Hartung, Bernod Girod(1998),"Watermarking of uncompressed and compressed domain Video",*Elseiver,*Vol 66,no. 3,May 1998,pp 283-301

Garimella, A., Satyanarayan, M.V.V., Kumar, R.S., Murugesh, P.S., Niranjan (2003) ,VLSI Implementation of Online Digital Watermarking Techniques with Difference Encoding for the 8-bit Gray Scale Images *In: Proc. of the Intl. Conf.on VLSI Design.,*pp 283–288

Hyun Lim, Soon-Young Park and Seong jun Kang (2003), FPGA Implementation of Image Watermarking Algorithm for a Digital camera, *IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2003. PACRIM. 2003*, pp.1000-1003.

Hsien-Wen Tseng , Chin-Chen Chang (2008), An extended difference expansion algorithm for reversible watermarking ,*Elsiver, Image and vision computing*, pp 1148-1308

I .J. Cox, J. Kilian, T. Leighton and T. Shanon (1995), Secure spread spectrum for Multimedia, *NEC research institute*, prinecton, NJ, technical report pp. 95-10.

I.J.Cox,J. Kilian, F.T. Leighton, and T. Shamoon (1997), Secure Spread Spectrum Watermarking for Multimedia, *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687

Jianyog Huang and Changsheng Yang (2004),"Image Digital Watermarking algorithm Using Multiresolution wavelet Transform", *IEEE International conference on systems, man and Cybernetics*, pp. 2977-2982.

Jing Zhang,Anthony T.S.Ho, Gang Qiu.Robust(2007), "Video Watermarking of H.264/AVC[J]". *IEEE Transactions on circuits and systems-Ti:express briefs*,vol.54,no.2,February 2007.

Jing Zhang and Anthony T. S. Ho(2005), “Efficient robust watermarking of compressed 2-D grayscale patterns for H.264/AVC,” *Proc. of IEEE Workshop on Multimedia Signal Processing*, pp. 1-4, 2005.

Karima Ait Saadi, Ahmed Bouridane, H. Meraoubi(2008):”Secure and Robust Copyright Protection for H.264/AVC based on Selected Blocks DCT”. *SIGMAP 2008,*pp 351-355

Lei Tian and Heng-Ming Tai (2006), Secure Image Captured by Digital Camera, *International Conference on Consumer Electronics, 2006. ICCE '06*, pp.341-342

MarkAny (2010), Watermarking Technology, *Whitepaper*,2010.

Sammy H. M. Kwok, Edmund Y. Lam (2002), Watermarking Implementation in digital photography”, *Proceeding of International Symposium on Intelligent Signal Processing and Communication System*. Hall

Saraju P. Mohanty, N. Raganathan (2004), VLSI Implementation of Visible Watermarking for a secure Digital Camera Design, *Proceeding of 17th International Conference on VLSI design*

Saraju P. Mohanty, N. Raganathan and R.K. Namballa (2005) ,A VLSI architecture for visible watermarking in a secure still digital camera (S2DC) design, *IEEE Transaction on VLSI*, vol 13., pp 1002-1012.

Saraju P. Mohanty, N. Raganathan and K. Balakrishna (2006), A Dual Voltage- Frequency VLSI chip for Image Watermarking in DCT Domain , IEEE *Transaction on circuits and systems II(TCAS-II),* vol. 53,pp 394-398.

Satyen Biswas, Sunil R. Das, Emil M. Petriu, (2005),”An Adaptive Compressed MPEG-2 Video Watermarking Scheme”, IEEE *transaction on instrumentation and measurement*, vol. 54, no. 5, October-2005

Tsai, T.H., Lu, C.Y (2001),A Systems Level Design for Embedded Watermark Technique using DSC Systems, *presented at the IEEE Int. Workshop Intelligent Signal Processing Communication System*, Nashville, TN, pp 20-23

Tung-Shou Chen, Jeanne Chen, Jian –Guo Chen(2004) , A Simple and efficient watermark technique based on JPEG2000 Codec, *Springer, Multi media systems*, pp 16-26.

Victor V. Hernandez Guzman, Mariko Nankano (2004), Analysis of a Wavelet based watermarking Algorithm, Proceeding *of the 14th International Conference on Electronics, Communication and Computers*, pp 283-287.

Xian Li, Yonatan shoshan,Alexander Fish,Graham Jullian,Orly Yadid-Pecht(2010) ,Hardware Implementation of video watermarking , *Information science and computing*, pp-9-16

Yong-Gang Fu and Hui-Rong Wang (2008), A Novel Discrete Wavelet Transform Based Digital Watermarking Scheme, *2nd International Conference on Anticounterfeiting, Security and Identification* ,pp 55-58.

Yulin Wang, Alan Pearmain(2004),“Blind image data hiding based on self reference”, *Pattern Recongnition, Elsevier*, 2004, pp. 1681-1689.

5

Sophisticated Spatial Domain Watermarking by Bit Inverting Transformation

Tadahiko Kimoto
Toyo University
Japan

1. Introduction

Digital watermarking is a technique for embedding additional signals, watermarks, into digital signals such as images and afterward extracting them (Macq (1999)). From the viewpoint of domains to embed watermark signals into, the watermarking techniques are mainly divided into two categories: spatial-domain-based techniques and transform-domain-based ones.

In the watermarking of digital images, the transform-domain-based techniques can usually provide not only good visual quality in the resulting images but also stronger robustness against image modification than the spatial-domain-based ones. However, it is hard to embed watermarks exactly into transform domains. In the embedding procedure, the pixel values of a source image, which are usually quantized levels or integers, are first transformed into frequencies. The frequency coefficients are then modified so as to represent watermarks. The values inversely transformed from such modified transform domain are usually real numbers with fractions. Consequently, the quantization errors occur inevitably when the integral spatial domain is reconstructed. These errors are likely to disturb the watermark that has been embedded in the transform domain.

In contrast, exact watermark signals can be embedded into the spatial domain though they are fragile under signal modification. The traditional method for spatial-domain-based image watermarking is first to select pixels in a source image and then, to modify the levels of the selected pixels so that the watermark can be expressed there (Wang et al. (2009)). The most primitive method for modifying a pixel level is to select a bit in the binary expression of the level and then, to invert the selected bit (Oka & Matsui (1997)). In this method, the bit position selected in the binary expression as well as the pixel location selected in a source image must be kept secret so that the watermark can be protected from unauthorized access.

The embedded watermark distorts the source signal to some extent. The transformation in the image spatial domain has especially direct effects on visual quality. In the bit inverting method, inverting the kth bit of a signal, denoted by the bit k, where the 0th bit is the least significant bit, changes the signal level by 2^k. Hence, the visual distortion caused by the level change increases with an increase in k. On the other hand, when a bit k represents a watermark bit, an attacker searches some range of k's for the correct value to get the bit k. With increasing k, the range to be searched becomes wider, and accordingly, the tolerance to unauthorized access can increase. Thus, determining appropriate values of k is very important to the watermarks involving the bit inverting. Also, the extending of ranges of k and the preserving of visual quality are contradictory subjects.

It is desirable to determine the values of k automatically for given source images to carry out the watermarking efficiently. To choose an appropriate bit k to be inverted in terms of visual quality, a human perceptual model is necessary. The perceptual model means a relationship between quantities representing objective qualities and human subjective qualities of the images being viewed, based on the human visual system (Awrangjeb & Kankanhalli (2004)). In other words, such a perceptual model is a function of objective quality measures that produces a subjective quality measurement as an estimation of human subjective quality. Various kinds of quantities have been proposed for the measurement of objective quality (Wang et al. (2004)). By adaptively determining the embedding parameters such as the values of k by means of the human perceptual model, the watermarking scheme can perform as a *perceptually adaptive system* (Cox et al. (2002)).

In the next section, the function of inverting signal bits is discussed. There a method for inverting a signal bit with making the resultant level change minimum is presented (Kimoto (2005); Kimoto (2007)). In this method, the inversion of the kth bit ($k \geq 1$), where a 0th bit is a least significant bit, results in the change in the signal level by 2^{k-1} or under for any input level. Also, randomly varying signals are added to the transformation outputs so as to improve signal quality (Kimoto (2006)). Both the function of inverting bits and that of randomizing levels are given as a single transformation. The performance of the transformation is analyzed in detail and also demonstrated by some experiments.

The next section treats some subjects regarding the implementation of the bit inverting transformation. First, the transformation domains are considered; a principle for defining domains of the transformation in the input dynamic range under limitations on level changes is formulated. Also, a method for dividing the input dynamic range into a union of transformation domains so that the blind watermarking can be achieved is described. Next, a scheme for embedding watermark bits in every image block using the transformation is presented with some experimental results.

In the next section, the subject of determining a kth bit to be inverted, or the value of k in the same sense, is discussed; a scheme for implementing the bit inverting transformation to embed watermark bits so that a required subjective visual quality can be achieved on the resulting image is developed (Kimoto & Kosaka (2010)). To derive an appropriate perceptual model for such images that are watermarked by the bit inverting transformation, first, objective quality measures are defined based on the properties of the transformation. Then, a subjective visual quality measure based on the objective quality measures is established by subjective evaluations of human observers. A perceptually adaptive image watermarking scheme using the perceptual model is presented. This scheme is aimed at embedding watermark bits in every pixel all over the source images in contrast with the block watermarking scheme described in the preceding section. The performance of the scheme is demonstrated by some experiments.

2. Bit inverting transformation

2.1 Inverting signal bits

Inverting a single bit of a signal can be expressed by a level transformation. Signal levels are supposed to be uniformly quantized to M bits and expressed from the M-bit sequence of

$0 \cdots 0$ to that of $1 \cdots 1$ in natural binary. Each bit in the binary expression is denoted by bit k for $k = 0,\ 1,\ \ldots,\ M-1$, and bit 0 means the least significant bit. For a given level $v \in [0,\ 2^M - 1]$ and a specified value of $k \in [0,\ M-1]$, inverting the bit k of v transforms v to another level v'; this transformation is described as a function t_k of v in the relation

$$v' = t_k(v) = \begin{cases} v + 2^k, & \text{if } n = 2m \\ v - 2^k, & \text{if } n = 2m+1 \end{cases} \tag{1}$$

where, letting $\lfloor x \rfloor$ denote the integral part of the real number x, $n = \left\lfloor v/2^k \right\rfloor$, and m has integers in the interval $[0,\ 2^{M-k-1} - 1]$ as a result. Each input level thus either increases or decreases just by 2^k.

2.2 Minimizing level changes

The amount of level change caused by an inverted bit can be minimized by altering the other bits of the signal appropriately. The principle for transforming the entire bit-pattern of a signal so as to achieve the smallest level change is illustrated in Fig. 1. Let $b(v,\ k)$ denote a binary value of the bit k of a level v, and $\overline{b}$ denotes the 1's complement of a bit value b. Suppose that the kth bit of a level v is to be inverted. Then, the inverting of the bit is performed by changing the level, that is, the entire bit-pattern to one of those patterns of level v'' which satisfy $b(v'',\ k) = \overline{b(v,\ k)}$. Furthermore, the one of the v''s that is closest to v can be chosen as v' to achieve the smallest level change. For the most of v, such v''s exist in both sides of v in the dynamic range as depicted in Fig. 1.

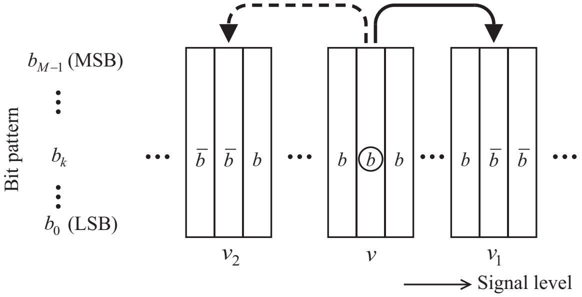


Fig. 1. Changing signal levels for inverting a signal bit: To invert b_k of v, v is changed to either v_1 or v_2.

Both inverting a kth bit and minimizing the resultant level change are performed by a single level transformation. Let $\Delta_k = 2^{k-1}$ where k now assumes values in the range $1,\ 2,\ \ldots,\ M-1$. For a given k, the transformation of a signal level v can be expressed as a function f_k of v, defined by

$$f_k(v) = \begin{cases} 2\Delta_k, & \text{if } n = 0 \\ 2m\Delta_k, & \text{if } n = 2m-1 \\ 2m\Delta_k - 1, & \text{if } n = 2m \\ 2^M - 2\Delta_k - 1, & \text{if } n = 2^{M-k+1} - 1 \end{cases} \tag{2}$$

where $n = \lfloor v/\Delta_k \rfloor$ and m has integers in the interval $[1,\ 2^{M-k} - 1]$. The function f_k satisfies the relation

$$b(f_k(v),\ k) = \overline{b(v,\ k)}. \tag{3}$$

For $k = 0$, the transformation $f_0(v)$ has the special relation

$$f_0(v) = \begin{cases} v+1, & \text{if } v \text{ is even} \\ v-1, & \text{if } v \text{ is odd.} \end{cases} \tag{4}$$

Figure 2 illustrates $f_k(v)$ over the M-bit dynamic range $v \in [\,0,\ 2^M - 1\,]$, comparing with $t_k(v)$ of Eq. (1) depicted by the thick dashed lines, where $k \geq 1$ (the box of thin dashed lines will be explained later). As shown in this figure, $f_k(v)$ possesses the almost staircase relations between input and output levels.

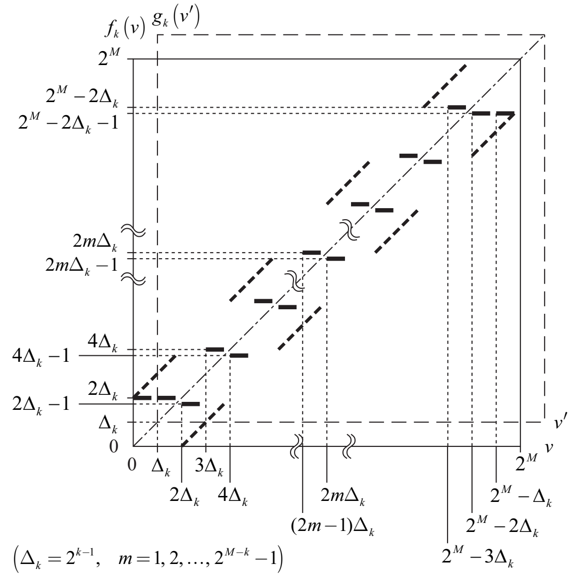


Fig. 2. Transformations for inverting kth bits of M-bit levels ($1 \leq k \leq M-1$): The bold line shows the transformation $f_k(v)$, and the thick dashed line, $t_k(v)$, where $\Delta_k = 2^{k-1}$; the dashed-and-dotted line shows the identity transformation for reference.

The level difference resulting from the level transformation is also a function of the source levels. Figure 3 shows the difference between the transformed level $v_{\text{out}} = f_k(v_{\text{in}})$ and the source level v_{in} in the entire source range, comparing with the difference caused by the transformation t_k. The absolute magnitude of the difference, $|f_k(v_{\text{in}}) - v_{\text{in}}|$, varies in the range from 1 to Δ_k, which is less than or equal to the half of the difference caused by t_k, for the source levels v_{in} over the interval $[\,\Delta_k,\ 2^M - \Delta_k - 1\,]$. On the contrary, for a level v_{in} in the interval $[\,0,\ \Delta_k - 1\,]$, those levels available for bit inversion exist only on the upper side of v_{in} because of the end of M-bit levels. Accordingly, all the source levels in the interval are to be transformed to the smallest one among the available levels, $2\Delta_k$. The resulting level differences consequently get over Δ_k in the interval. A similar end effect occurs for the input levels in the interval $[\,2^M - \Delta_k,\ 2^M - 1\,]$. In this paper we refer to these two intervals as the *end-effect intervals*.

2.3 Modifying level transformation

The M-bit end effects can be removed by translating the coordinate system. Both of the axes of the coordinate system are translated by Δ_k to the positive direction so as to avoid the lower

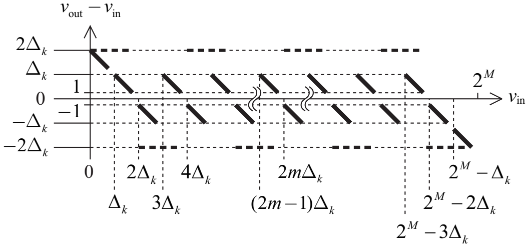


Fig. 3. Characteristics of level change caused by the transformations: The solid lines show $f_k(v_{\text{in}}) - v_{\text{in}}$; the dashed lines show $t_k(v_{\text{in}}) - v_{\text{in}}$.

end-effect interval $[\,0,\ \Delta_k - 1\,]$, that is, letting g_k denote the translated function,

$$\left\{\begin{array}{l} v' = v - \Delta_k \\ g_k(v') = f_k(v) - \Delta_k \end{array}\right. \tag{5}$$

The translated coordinate system is depicted in Fig. 2 by the thin dashed lines. Accompanied with the coordinate translation, both of the upper bounds of the dynamic range with regard to $f_k(v)$ are removed. Eq. (5) directly leads to the relation

$$f_k(v' + \Delta_k) = g(v') + \Delta_k. \tag{6}$$

Using Eqs. (5) and (6) in Eq. (3), we obtain the relation

$$b(g_k(v') + \Delta_k,\ k) = \overline{b(v' + \Delta_k,\ k)}. \tag{7}$$

This equation indicates that the inversion of bit-k holds between $v' + \Delta_k$ and $g_k(v') + \Delta_k$. By carrying out the above addition of Δ_k in modulus of 2^M, we can correspond the interval $[\,2^M - \Delta_k,\ 2^M - 1\,]$ of v' to the interval $[\,0,\ \Delta_k - 1\,]$ of v, and thus, the upper end-effect interval no longer exists in the translated coordinate system.

In the new coordinate system, the function g_k is defined by a level transformation from an input level v_{in}, that is,

$$g_k(v_{\text{in}}) = \left\{\begin{array}{ll} (2m+1)\Delta_k, & \text{if } n = 2m \\ (2m+1)\Delta_k - 1, & \text{if } n = 2m+1 \end{array}\right. , \tag{8}$$

where $n = \lfloor\, v_{\text{in}}/\Delta_k \,\rfloor$ taking values in $[\,0,\ 2^{M-k+1} - 1\,]$ and, as described later, $k \geq 1$. Thus, this transformation maps 2^M consecutive source levels into 2^{M-k+1} discrete output levels.

Figure 4 shows the input-output relationship of the transformation g_k. The transformation is no longer affected by the M-bit end effects. Hence, the level change has the absolute magnitude in the range $[\,1,\ \Delta_k\,]$ throughout the entire source range, as shown in Fig. 5. Thus, the function g_k yields the smallest difference for each input level while achieving the bit k inversion in terms of Eq. (7). Note that it is the bit $(k-1)$ of $g_k(v_{\text{in}})$ that is actually inverted compared with v_{in}, and the level change is not the smallest for each input level in the situation of bit $(k-1)$ inversion. This performance holds for $1 \leq k \leq M-1$. Hence, g_k has been defined in this range of k.

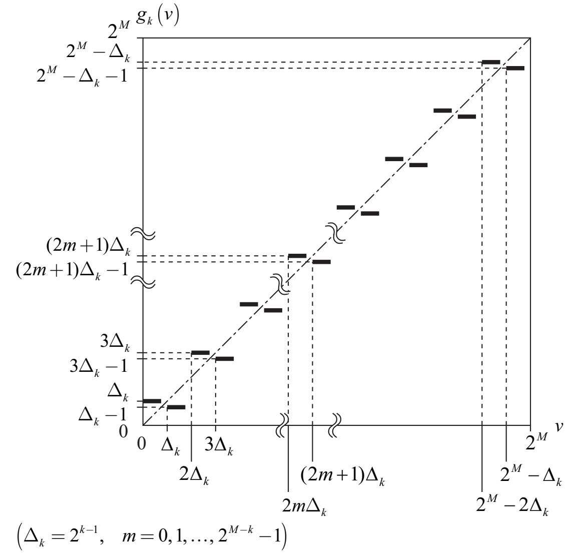


Fig. 4. Modified transformation function g_k.

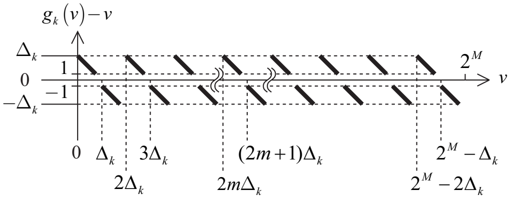


Fig. 5. Characteristics of level change caused by the transformation function g_k.

2.4 Varying transformed levels

The transformation g_k maps 2^M consecutive source levels into 2^{M-k+1} sparse output levels. Furthermore, the output levels consist of pairs of consecutive levels $(2m+1)\Delta_k - 1$ and $(2m+1)\Delta_k$ for $m = 0, 1, \ldots, 2^{M-k}-1$, as shown in Eq. (8). Hence, the resulting images look like coarsely quantized at $M-k$ bits per pixel.

We extend the range of the transformation outputs within the M-bit dynamic range to improve the resulting image quality. In the pulse code modulation (PCM), dithering signals has effects on improving subjective image quality. Then, we consider distributing the transformed levels in the dynamic range by a stochastic process. A transformation from the outputs $v_{\text{out}} = g_k(v_{\text{in}})$ further to v'_{out} is developed so that all the following three conditions can be satisfied:

(i) The inversion of bit-k holds in terms of

$$b(v'_{\text{out}} + \Delta_k,\ k) = \overline{b(v_{\text{in}} + \Delta_k,\ k)}. \tag{9}$$

(ii) The range of v'_{out} extends to the whole of the M-bit dynamic range.

(iii) The amount of level change is limited so as not to increase excessively. Taking into account that the largest change caused by g_k is Δ_k levels, we determine this limitation as

$$|v'_{\text{out}} - v_{\text{in}}| \leq \Delta_k. \tag{10}$$

The above v'_{out} can be obtained by adding an appropriate random level to v_{out}. Generating such v'_{out} from input level v_{in} is performed using a single level transformation expressed in the relation $v'_{\text{out}} = h_k(v_{\text{in}})$: The function h_k is defined as

$$h_k(v_{\text{in}}) = \begin{cases} g_k(v_{\text{in}}) + r, & \text{if } n = 2m \\ g_k(v_{\text{in}}) - r, & \text{if } n = 2m+1 \end{cases}, \tag{11}$$

where $n = \lfloor v_{\text{in}}/\Delta_k \rfloor$ and r has arbitrary levels in the range $[\,0,\ R_k(v_{\text{in}})\,]$ and $R_k(v_{\text{in}})$ is a function of v_{in}, defined by

$$R_k(v_{\text{in}}) = \begin{cases} v_{\text{in}} - 2m\Delta_k, & \text{if } n = 2m \\ 2(m+1)\Delta_k - v_{\text{in}} - 1, & \text{if } n = 2m+1 \end{cases}. \tag{12}$$

Thus, $R_k(v_{\text{in}})$ varies in the range $[\,0,\ \Delta_k - 1\,]$. The value of r is supposed to be determined within the range in a stochastic manner. Thereby, $h_k(v_{\text{in}})$ is a stochastic function whose output range depends on v_{in} as follows:

$$\begin{cases} h_k(v_{\text{in}}) \in [\,g_k(v_{\text{in}}),\ v_{\text{in}} + \Delta_k\,], & \text{if } n \text{ is even} \\ h_k(v_{\text{in}}) \in [\,v_{\text{in}} - \Delta_k,\ g_k(v_{\text{in}})\,], & \text{if } n \text{ is odd} \end{cases}. \tag{13}$$

The number of levels composing the output range varies in $[\,1,\ \Delta_k\,]$. Hence, the level randomization is effective for $k \geq 2$.

Figure 6 illustrates the input-output relationship of $v'_{\text{out}} = h_k(v_{\text{in}})$ by showing the ranges where the output levels vary by the shaded areas. As seen in the figure, the range of v'_{out} spreads all over the M-bit dynamic range, compared with that of $v_{\text{out}} = g_k(v_{\text{in}})$. The difference $|v'_{\text{out}} - v_{\text{in}}|$, however, tends to get larger than its smallest value for each v_{in}, that is, $|v_{\text{out}} - v_{\text{in}}|$, according to the stochastic manner being used.

2.5 Properties of level transformation

2.5.1 Amount of level change

According to the definition of h_k, the output level $v'_{\text{out}} = h_k(v_{\text{in}})$ varies in a stochastic manner, and so do the level differences between v'_{out} and v_{in}. Figure 7 illustrates the level differences for input levels v_{in}.

We evaluate the amount of level change caused by h_k by stochastic analysis. In the analysis, we assume that M-bit levels in the dynamic range $[\,0,\ 2^M - 1\,]$ are uniformly distributed over the input signals. Also, the values of r in Eq. (11) are assumed to be equally likely in the range assigned for each input level.

For a given k, an input level v_{in} is rewritten as $v_{\text{in}} = n\,\Delta_k + w$ using w in the case that $n = \lfloor v_{\text{in}}/\Delta_k \rfloor$ is even (See Fig. 7). Note that w is identical with $R_k(v_{\text{in}})$ in Eq. (12). By using Eq. (8) in Eq. (11), the output level $v'_{\text{out}} = h_k(v_{\text{in}})$ is given by $v'_{\text{out}} = (n+1)\,\Delta_k + r$, where r is randomly chosen in $[\,0,\ w\,]$, and hence, it is a stochastic variable of a uniform occurrence probability of $1/(w+1)$. The difference $d' = v'_{\text{out}} - v_{\text{in}}$ is then written as $d' = \Delta_k - w + r$ for

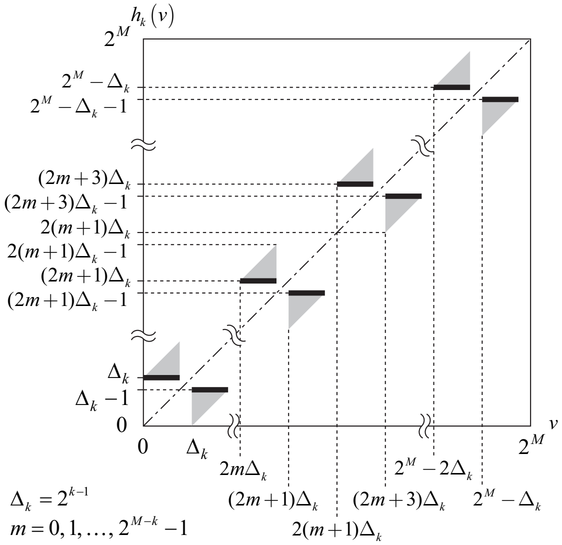


Fig. 6. Transformation function with level randomization, h_k: The thick solid lines show the input-output relationship of g_k for reference.

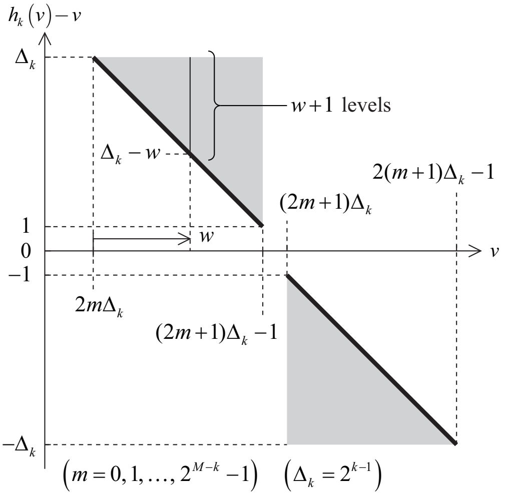


Fig. 7. Characteristics of level change caused by the transformation function h_k: The thick solid lines show the level change caused by g_k for reference.

a certain r. Thus, d' is also a stochastic variable ranging in $[\,\Delta_k-w,\ \Delta_k\,]$ for each w. Then, the expected value of the squared d' are summed over the range of w from 0 to Δ_k-1 in the form

$$\sum_{w=0}^{\Delta_k-1}\left(\sum_{d'=\Delta_k-w}^{\Delta_k} d'^2\cdot\frac{1}{w+1}\right). \tag{14}$$

The same value of sum is obtained in the case that n is odd. Consequently, averaging the above sum over Δ_k levels yields the mean of the squared level differences over the entire dynamic range, denoted by $E_{\mathrm{H}}(k)$, on the above assumptions. As a result, we have

$$E_{\mathrm{H}}(k) = \frac{1}{36}\left(22\,\Delta_k^2 + 15\,\Delta_k - 1\right). \tag{15}$$

2.5.2 Change of level occurrences

The difference in width among the ranges where the values $v'_{\mathrm{out}} = h_k(v_{\mathrm{in}})$ vary for a given v_{in} makes the occurrence probabilities of output levels unequal. We have analyzed the occurrence frequencies of v'_{out} on both the assumption that M-bit input levels are uniformly distributed in the dynamic range and the assumption that output levels for a given input level v_{in} are equally probable in the range of $h_k(v_{\mathrm{in}})$. Suppose that any input level has a frequency of 1. Then, for an output level $v'_{\mathrm{out}} \in [\,(2m+1)\Delta_k,\ 2(m+1)\Delta_k\,)$, by expressing v'_{out} by $v'_{\mathrm{out}} = (2m+1)\Delta_k + w$ $(0 \le w < \Delta_k)$, the frequency of v'_{out} is given as $\sum_{p=w+1}^{\Delta_k} 1/p$. A similar analysis is obtained for the interval $[\,(2m\Delta_k,\ (2m+1)\Delta_k\,)$ of v'_{out}. Hence, for input levels of the uniform frequency distribution, the output levels of h_k has the frequency distribution as illustrated in Fig. 8.

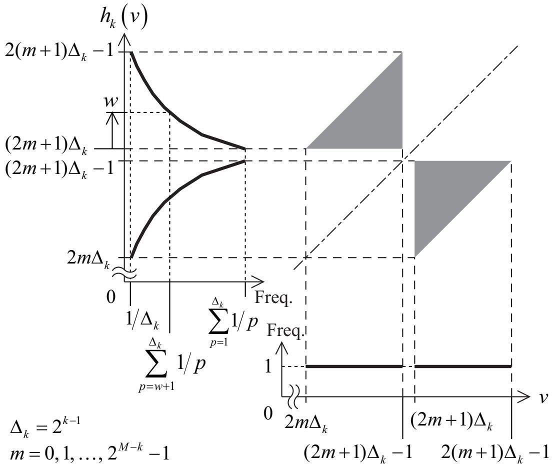


Fig. 8. Occurrence distribution of output levels by the transformation function h_k.

Although the output range of h_k spreads over the entire M-bit range, in every two ranges of Δ_k-level width the order of two ranges are reversed through the mapping, as shown in Fig. 9. These reversed ranges result in distortions on the picture surface.

2.6 Evaluation of level transformation

2.6.1 Amount of level change

We evaluate the amount of level change caused by h_k in terms of the mean squared level difference (MSD). For comparison with the MSD values of h_k in Eq. (15), the expected MSD values of other transformations are derived on the assumption that 2^M levels are uniformly distributed over the M-bit input signals as below. The level change caused by the function

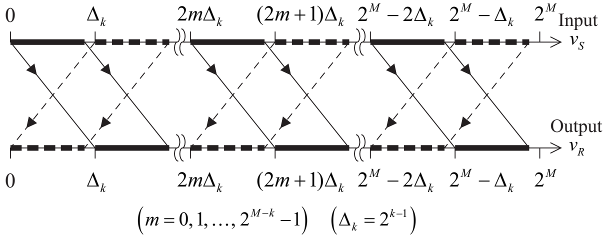


Fig. 9. Correspondence of h_k between input and output levels.

g_k can be evaluated similarly to that caused by h_k. For a given k, the mean squared level difference for the signals transformed by g_k, denoted by $E_{\mathrm{G}}(k)$, is given by

$$E_{\mathrm{G}}(k) = \frac{1}{6}(2\,\Delta_k + 1)(\Delta_k + 1). \tag{16}$$

With regard to the function t_k, for a given k, the magnitude of level change caused by the transformation is $2\Delta_k$ for any input level. Then, the mean squared level difference, denoted by $E_T(k)$, is obviously $4\Delta_k^2$.

For comparison, we consider another method for extending the output levels of g_k within the dynamic range in a stochastic manner; that is, for a given k, all the lowest $(k-1)$ bits of $g_k(v_{\mathrm{in}})$ are replaced with random ones for any v_{in}. We express $g_k(v_{\mathrm{in}})$ and the following bit replacing operation together as a single function of v_{in}, $h'_k(v_{\mathrm{in}})$. The value of $h'_k(v_{\mathrm{in}})$ is a random variable in either range $[\,g_k(v_{\mathrm{in}}),\ g_k(v_{\mathrm{in}}) + \Delta_k - 1\,]$ or $[\,g_k(v_{\mathrm{in}}) - \Delta_k + 1,\ g_k(v_{\mathrm{in}})\,]$ that is determined from the input interval including v_{in}. The whole range of h'_k coincides with the M-bit dynamic range. Also, on the assumption that input levels have a uniform frequency distribution in the dynamic range, the frequency distribution of the output levels gets uniform. As a disadvantage, h'_k tends to increase the resulting level change; the upper bound of the level difference $|h'_k(v_{\mathrm{in}}) - v_{\mathrm{in}}|$ depends on v_{in} and varies in the range $[\,\Delta_k,\ 2\Delta_k\,)$. The expected value of MSD is given on the assumption that output levels for each input level are equally probable in Δ_k levels by

$$E_{\mathrm{L}}(k) = \frac{1}{6}\big(7\,\Delta_k^2 - 1\big). \tag{17}$$

Table 1 lists the MSD values of each transformation for $k = 2,\ 3,\ 4$ and 5. In comparison for a given k, g_k reduces substantially the MSD from that of t_k; the ratio $E_{\mathrm{G}}/E_{\mathrm{T}}$ is, for example, 0.12 for $k = 3$, and approaches approximately to 1/12 for large k's. The transformation h_k increases the MSD from that of g_k due to the level randomizing; the ratio $E_{\mathrm{H}}/E_{\mathrm{G}}$ is 1.52 for $k = 3$, and about 11/6 for large k's. On the other hand, the MSD of h_k is smaller than that of h'_k; we find the ratio $E_{\mathrm{H}}/E_{\mathrm{L}}$ to be 0.62 for $k = 3$ and about 11/21 for large k's.

2.6.2 Experiments of transformation

To visualize the input-output relationship, we have conducted each transformation of t_k, g_k, h_k and h'_k on an 8-bit grayscale ramp image where the pixel level increases at one per pixel from 0 (black) to 255 (white) along the gradation. Figure 10 shows the transformed results with $k = 5$. Also, Fig. 11 illustrates a one-dimensional pixel sequence along the ramp of Fig. 10(d),

	k			
MSD	2	3	4	5
$E_{\mathrm{T}}(k)$	16	64	256	1024
$E_{\mathrm{G}}(k)$	2.5	7.5	25.5	93.5
$E_{\mathrm{H}}(k)$	3.3	11.4	42.4	163.1
$E_{\mathrm{L}}(k)$	4.5	18.5	74.5	298.5

Table 1. Expected values of mean squared difference.

showing an example of the stochastic output levels of h_k. In accordance with Fig. 2, although the results of t_5 has all levels in the dynamic range, there are level gaps every 32 ($= 2^5$) levels as observed in Fig. 10(b). In the gradation transformed by g_5 of Fig. 10(c), we may perceive eight *steps* while the result actually has 16 different levels.

Fig. 10(d) shows the result of adding random signals to Fig. 10(c) by h_5 and has all levels in the dynamic range. For a $v_{\mathrm{in}} = (2m+1)\Delta_k - 1$, the absolute difference between $g_k(v_{\mathrm{in}})$ and $g_k(v_{\mathrm{in}} + 1)$ is one level, but the absolute difference between $h_k(v_{\mathrm{in}})$ and $h_k(v_{\mathrm{in}} + 1)$, in contrast, can reach the utmost $2\Delta_k - 1$ as shown in Fig. 11. Hence, the visible steps in Fig. 10(c) are divided into halves in Fig. 10(d). The result of h'_k in Fig. 10(e) looks similar to that of h_k (Fig. 10(d)) though the level frequency distributions in the dynamic range are different. Note that an extremely large value of k has been used in Fig. 10 so that the input-output relationships get evident.

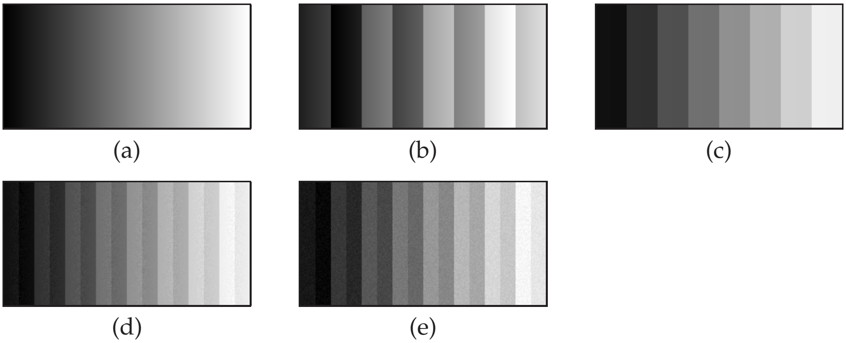


Fig. 10. Experimental results ($k = 5$): (a) Source (8-bit grayscale ramp); (b), (c), (d) and (e) transformed images by t_k, g_k, h_k and h'_k, respectively. All images are printed at 200 pels/inch.

We have also conducted experiments of each transformation on 8-bit monochrome test images. All the pixels in an input image were transformed to evaluate the level changes in terms of visual quality. Figure 12 shows results of the transformations for one of test images, *Lena*. We can observe a distortion pattern similar to that in Fig. 10, in particular, in smooth image areas such as the *shoulder* in each image of Fig. 12. Such distortions can be referred to as *false contours* after those caused by coarsely quantizing pixel levels.

The amount of level change from the source image has been estimated an MSD value for each transformed images. The MSD between a source image f and the transformed image f' is

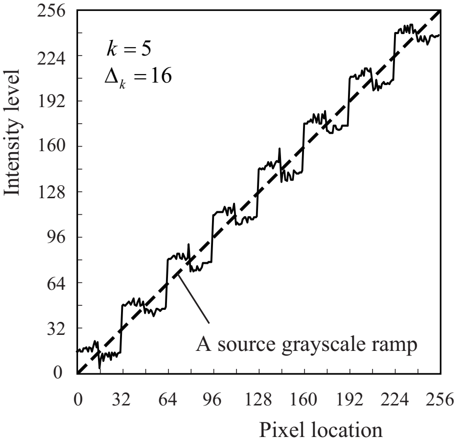


Fig. 11. An example of the transformed 256-grayscale ramp by h_k: levels varying along a horizontal line in Fig. 10(d).

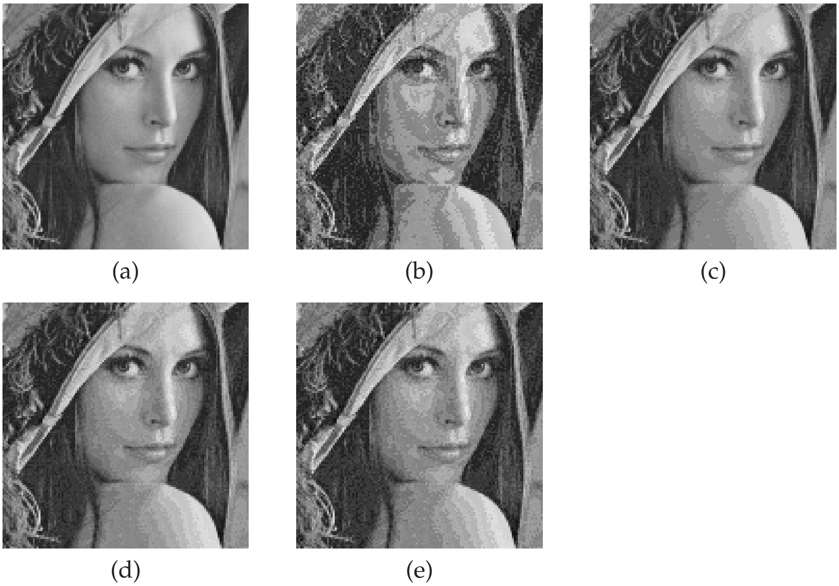


Fig. 12. Experimental result ($k = 4$): (a) zoomed source (128×128-pixel part of *Lena*); (b), (c), (d) and (e) transformed images by t_k, g_k, h_k and h'_k, respectively.

generally defined by

$$\text{MSD} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \{f'(i,\, j) - f(i,\, j)\}^2 \tag{18}$$

where $f(i,\, j)$ and $f'(i,\, j)$ are the respective pixel levels in f and f' at the coordinates of $(i,\, j)$, for $i = 1,\, 2,\, \ldots,\, M$ and $j = 1,\, 2,\, \ldots,\, N$ supposing that both f and f' are $M \times N$ images.

Table 2 lists the measured MSD values for two test images. Although grayscale levels have nonuniform frequency distribution in each input image, the MSD values measured for each transformed image almost agree with the expected values listed in Table 1. Accordingly, the expected values of MSD can be generally used to estimate the actual amount of level change.

(a) Test image *Lena*

	k			
Transformation	2	3	4	5
g_k	2.5	7.5	26.1	91.4
h_k	3.3	11.4	42.8	161.5
h'_k	4.5	18.5	75.7	294.7

(b) Test image *Peppers*

	k			
Transformation	2	3	4	5
g_k	2.5	7.5	25.9	92.5
h_k	3.2	11.4	42.7	162.3
h'_k	4.5	18.5	75.2	296.9

Table 2. Measured values of mean squared difference.

3. Implementation of level transformation

3.1 Limitations on level changes

It is necessary to make level changes caused by a transformation imperceptible both for keeping the embedded watermarks secret and for preserving the image quality. Here, we assume that a range where a pixel level can be changed without being perceived depends only on the own pixel. Then, let us express the upper limit of the range for a source level v as a function only of v, $A(v)$, supposing that $A(v) \geq 0$; that is, a source level v is allowed to change within the range $[\, v - A(v),\ v + A(v)\,]$.

On the above assumption, for a pixel of level v, if the amount of level change caused by a transformation is to be under $A(v)$, the transformation can be actually applied to the pixel. We use the function h_k as the level transformation for watermarking in the rest of this chapter, and let $d_k(v) = h_k(v) - v$. As described in Sec. 2.5.1, for a given v, $d_k(v)$ varies in the range that depends on v, and the upper bound of $|d_k(v)|$ is fixed to Δ_k for any v. Accordingly, we regard $|d_k(v)|$ as the constant Δ_k to determine if the transformation h_k can be applied to a pixel or not. That is, given the function $A(v)$, for a pixel of level v, if $\Delta_k \leq A(v)$ with a certain k, then the level can be changed by $h_k(v)$. Otherwise, the level is left unchanged. The function that keeps v unchanged is expressed by the identity transformation, denoted by $f_I(v)$ such that $f_I(v) = v$.

3.2 Transformation domains

Given the bounding function A of an input level and a value of k, according to the scheme described in the preceding section, each input level v_{in} of the dynamic range is classified into two categories: one that h_k can be applied to, and the other that f_I is to be applied to.

Consecutive input levels of the same category, then, are collected to compose domains of each transformation. The entire dynamic range of input levels, denoted by $U_M = [\,0,\ 2^M - 1\,]$, is consequently expressed as a disjoint union of one or more domains. Let $U_1^{(k)},\ U_2^{(k)},\ \ldots,\ U_{\phi(k)}^{(k)}$ be a sequence of these domains in order, where $\phi(k)$ is the number of domains for the k. Note that the domains of h_k and those of f_I alternate in the sequence.

Next, we modify the domains in U_M so that a blind watermark can be achieve, that is, so that a watermark can be recovered from the transformed image without referring to its source image. Our approach is to make the transformation output ranges disjoint to each other. Let $V_i^{(k)}$ be the range onto which a domain $U_i^{(k)}$ is mapped, for $i = 1,\ 2,\ \ldots,\ \phi(k)$. If these ranges are disjoint mutually, for any output level v_{out}, the range containing it, say, $V_j^{(k)}$, is determined uniquely. This range directly indicates not only the corresponding domain, $U_j^{(k)}$, but also the kind of transformation that is to be applied to the domain. Consequently, it is found whether a pixel is one of those pixels that h_k can be applied to or not. Furthermore, with regard to domains of h_k and the corresponding regions, say, $U_j^{(k)}$ and $V_j^{(k)}$, if that region to which $U_j^{(k)}$ can be mapped by f_I is included in $V_j^{(k)}$, that is,

$$\{f_I(v)\,|\,v \in U_j^{(k)}\} \subseteq V_j^{(k)}, \tag{19}$$

then, for pixels of source level $v_{\text{in}} \in U_j^{(k)}$, we can use either h_k to have the bit-k inverted or f_I to keep the bit-k unchanged.

Based on the input-output relationship of h_k, the domains that have the disjoint relationship among the corresponding regions are defined every $2\Delta_k$ levels in the dynamic range, that is, the boundaries of each domain must be located at the levels of $2m\Delta_k$ $(0 \le m \le 2^{M-k})$. Such domains also satisfy Eq. (19).

3.3 Bit-block watermarking

3.3.1 Data structure for watermarks

As a data structure for representing watermarks, we use bit-planes of a monochrome image or a color component image composed of M-bit pixels, which was also used by Oka & Matsui (1997). As depicted in Fig. 13, first, a source image is divided into pixel-blocks each consisting of N_B pixels. More generally, these blocks can have arbitrary shapes rather than rectangles, and besides, even N_B can be varied in the same image. Then, each pixel-block is regarded as a hierarchy of M bit-blocks. Suppose that a pixel is composed of M bits, a bit $(M-1)$ to a bit 0, that represent the signal level in the natural binary expression. A bit-block k is a set of all the bit k of each pixel in the pixel-block, for $k = 0,\ 1,\ \ldots,\ M-1$.

A watermark bit is represented by a parity value in one of the bit-blocks. Here, the parity value of a set of bits is defined by the sum of the bits in modulo 2. Suppose that to achieve the watermarking, one bit in each bit-block is to be inverted, if necessary, so that the resultant parity value can agree with the watermark bit. The details of the watermarking procedures will be described in the next section.

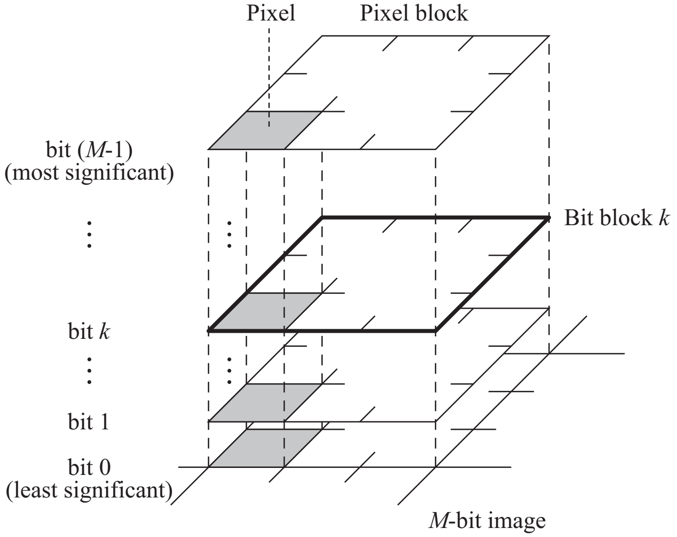


Fig. 13. Data structure for bit-block watermarking.

3.3.2 Watermarking procedures

The procedures for watermarking in the data structure of bit-blocks by using the function h_k are described below. Here, suppose that the domains of h_k in the M-bit dynamic range are given for each value of k. Suppose also that a value of k is assigned to each pixel-block of a source image F. A set of these values of k is associated with the watermarked image and it must be kept secret.

(1) Embedding procedure

Each pixel-block is processed in order. Let $P^{(i)}$ be the ith pixel-block being processed, which is expressed as a sequence of N_B pixels, that is, $P^{(i)} = \{p_1^{(i)}, p_2^{(i)}, \ldots, p_{N_B}^{(i)}\}$, for $i = 1, 2$ and so on. For the value k_i assigned to $P^{(i)}$, the block is processed by the following procedure:

Step 1: Extract from $P^{(i)}$ those pixels whose levels belong to the domains of h_{k_i}. Let $P_e^{(i)}$ be a set of these pixels and n_e be the number of the pixels.

Step 2: Compare n_e with a specified threshold N_E $(1 \leq N_E \leq N_B)$. The result determines how to treat the block with a watermark bit as follows:

(a) If $n_e \geq N_E$, we use this block to represent a new watermark bit. Proceed to *Step* 3.

(b) If $n_e < N_E$, we skip this block without using it to represent any watermark bit. Proceed to the next pixel-block.

Step 3: Collect the bits $b(v_j^{(i)} + \Delta_{k_i}, k_i)$ of the pixel $p_j^{(i)} \in P_e^{(i)}$, where $v_j^{(i)}$ is a level of $p_j^{(i)}$, for $j = 1, 2, \ldots, n_e$. Let the resulting bits compose a bit-block, $Q_{k_i}^{(i)}$. Then, calculate the parity value, y, of $Q_{k_i}^{(i)}$.

Step 4: Fetch a new watermark bit, ω, and compare it with y.

(a) If $\omega \neq y$, choose one pixel from $P_e^{(i)}$, and apply h_{k_i} to it.

(b) Otherwise, no operation is done to $P_e^{(i)}$.

Proceed to the next pixel-block.

After all the pixel-blocks of F are processed, the watermark image F' is obtained.

(2) Extracting procedure

Let $P^{(i)\prime}$ denote the ith pixel-block of F' in the form $P^{(i)\prime} = \{p_1^{(i)\prime}, p_2^{(i)\prime}, \ldots, p_{N_B}^{(i)\prime}\}$. $P^{(i)\prime}$ is processed using the same k_i and N_E as those used for $P^{(i)}$ by the following procedure:

Step 1: Extract from $P^{(i)\prime}$ those pixels whose levels belong to the ranges of h_{k_i}. Let $P_e^{(i)\prime}$ be a set of these pixels and n_e' be the number of the pixels. The disjoint union of the transformation ranges ensures that n_e' is equal to n_e of $P^{(i)}$.

Step 2: Compare n_e' with a specified threshold N_E.

(a) If $n_e' \geq N_E$, proceed to *Step* 3 to find out the watermark bit.

(b) If $n_e' < N_E$, it is found that this block contains no watermark. Proceed to the next pixel-block.

Step 3: Collect the bits $b(v_j^{(i)\prime} + \Delta_{k_i}, k_i)$ of the pixel $p_j^{(i)\prime} \in P_e^{(i)\prime}$, where $v_j^{(i)\prime}$ is a level of $p_j^{(i)\prime}$, for $j = 1, 2, \ldots, n_e'$. Let the resulting bits compose a bit-block, $Q_{k_i}^{(i)\prime}$. Then, calculate the parity value, y', of $Q_{k_i}^{(i)\prime}$. As a result, y' gives the watermark bit directly. Proceed to the next pixel-block.

The threshold N_E can be altered every pixel-block. Furthermore, being kept secret, N_E is expected to improve the concealment of watermarks.

3.4 Experiments

3.4.1 Domains defined by Weber's law

We demonstrate the defining of transformation domains by using a bounding function for level change, $A(v)$, as described in Sec. 3.1. To specify this function, as an example, we here use Weber's law, which is known as a description of human visual properties for luminance contrast (Jain (1989)). This law states the experimental result that $\Delta L/L$ is constant where ΔL is the magnitude just noticeably different from the surround luminance L. Assuming that this law holds true in the entire M-bit dynamic range, we can express it for a luminance level v in the fractional form:

$$\frac{\Delta v}{v + v_0} = \alpha \tag{20}$$

where v_0 is a fixed level equivalent to a light intensity on human eyes at $v = 0$, and α is a constant known as Weber's ratio. Letting $M = 8$, we now consider 8-bit levels in the range $[0, 255]$. From a result of our preliminary experiment for measuring contrast sensitivity of human eyes on a liquid crystal display (LCD), we have observed two pairs of approximate values, $\Delta v = 8$ at $v = 192$ and $\Delta v = 4$ at $v = 64$. By using these values in Eq. (20), then, we obtained $v_0 = 64$ and $\alpha = 0.03$, which is comparable to a typical α of 0.02.

Suppose that we can replace Δv with $A(v)$; that is, Eq. (20) is rewritten as

$$A(v) = \alpha(v + v_0). \tag{21}$$

For simplicity, approximating $A(v)$ by $2^{D(v)}$ where $D(v)$ has integers, we have determined $D(v)$ as

$$D(v) = \begin{cases} 3, & 128 \leq v \leq 255 \\ 2, & 0 \leq v \leq 127. \end{cases} \tag{22}$$

Using $A(v)$ with Eq. (22) in the source range of $[\,0,\ 255\,]$, we have defined the transformation domains according to the manner described in Sec. 3.2. Table 3 lists the respective domains of the level transformation h_k and the identity transformation f_I for each k ($k \geq 1$). Under the limitation given by Eq. (22), the range of k such that one or more domains are available for h_k is found out to be $[\,1,\ 4\,]$. Hence, vales of k are to be chosen from this range for each pixel-block in the bit-block watermarking scheme.

	Domains	
k	h_k	f_I
≤ 3	$[\,0,\ 255\,]$	ϕ
4	$[\,0,\ 127\,]$	$[\,128,\ 255\,]$
≥ 5	ϕ	$[\,0,\ 255\,]$

Table 3. Example of transformation domains based on Weber's law.

3.4.2 Examples

We have carried out an experiment of the bit-block watermarking scheme using 8-bit grayscale test images to evaluate visual quality of the resulting images. In the experiment, the location of an isolated pixel to be transformed was fixed at the center of each block. Generally, a pixel to be transformed can be chosen at random among available pixels in each block. Besides, the transformation h_k was always performed for each block. Note that, assuming that a watermark bit is a random variable, a half of the pixels being processed in the experiment are expected to be actually transformed by h_k.

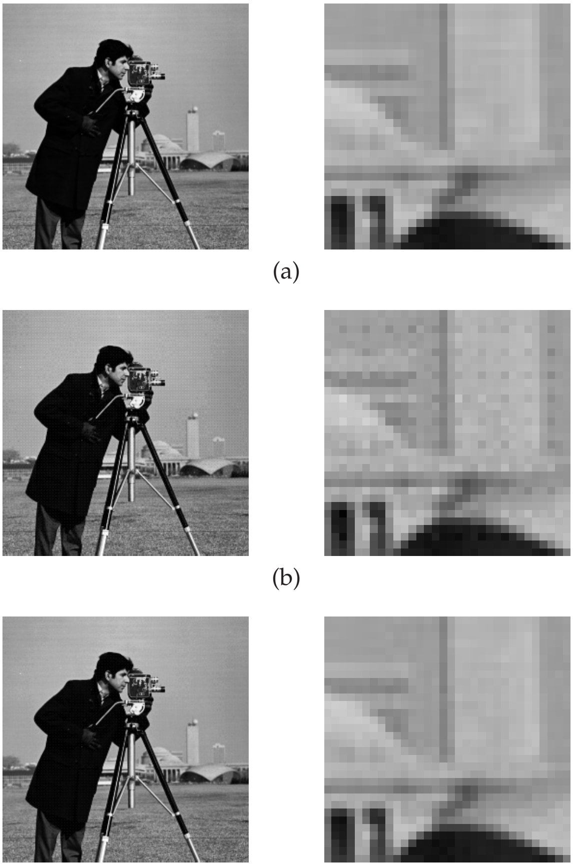
Figure 14 shows an example of the results transformed with pixel-blocks of 3×3 pixels and $k = 4$. Figure 14(b) includes those pixels which were transformed by t_4 and changed by 2^4 levels. These pixels are considerably perceptible. In contrast, those pixels which were transformed by h_4 in Fig. 14(c) have their levels changed by at most 2^3 levels. We can hardly observe any difference between the transformed and source images.

4. Perceptually adaptive watermarking

4.1 Perceptual modeling

4.1.1 Image distortions

Distortions caused by h_k have two phases. The first phase is caused by g_k; although the change in each signal level is made minimum, the number of levels appearing in the transformed



(a)

(b)

(c)

Fig. 14. Results of watermarking with 3 × 3 pixel-blocks and $k = 4$ on a 8-bit test image *Cameraman*: (a) original image; (b) and (c) watermarked image by t_k and h_k, respectively. The left images are of 256 × 256 pixels; the right images are enlarged portion of 32 × 32 pixels of the left images.

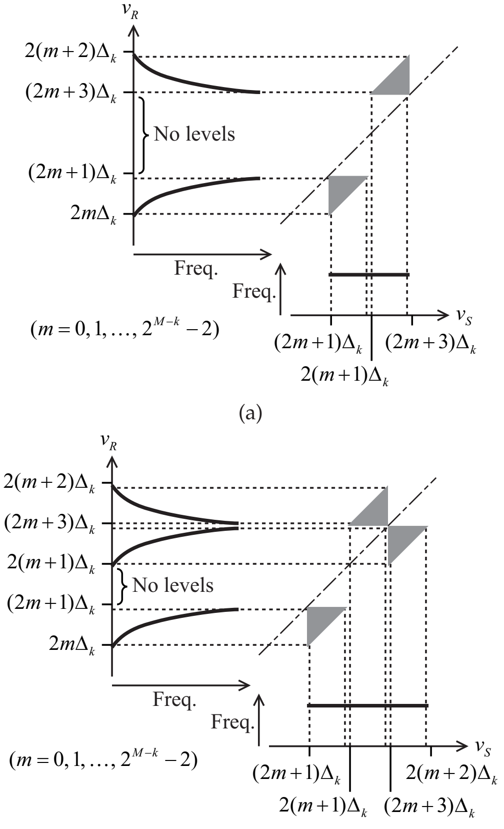
image is reduced. The second phase is caused by the level randomization. In this phase the number of levels appearing in the image increases while the change from the source image increases accordingly. Each phase of distortion affects the image quality in different manners.

(1) Distortion in low-detail image regions

The first phase of the distortion affects particularly the quality of low-detail image areas. Visible false contours are likely to appear in such smooth areas due to the effect similar to coarse quantization. As described in Sec. 2.6.2, randomizing output levels can improve visual quality by making the steps of false contours narrow.

(2) Distortion in high-detail image regions

We consider a local image area where source levels are bounded in a small range. If the source level range is $[2m\Delta_k,\ 2(m+1)\Delta_k)$, the transformed levels lie in the same range, as Fig. 8 has shown. On the contrary, if the source range of $2\Delta_k$ levels is $[(2m+1)\Delta_k,\ (2m+3)\Delta_k)$, the transformed levels lie outside the range and no levels appear inside the original range as shown in Fig. 15(a). If the source range of $3\Delta_k$ levels is $[(2m+1)\Delta_k,\ 2(m+2)\Delta_k)$, there also exists an empty range where no transformed levels appear (Fig. 15(b)). Both the replacement of ranges and the missing of ranges cause distortions in the texture of the local areas.



(a)

(b)

Fig. 15. Transformation h_k from bounded source ranges: Source levels are assumed to occur uniformly in the range.

4.1.2 Objective quality measures

The level transformation h_k causes distortion in the source image, and thus, degrades the image quality. According to the performance analysis of h_k, we have defined two kinds of objective measures to evaluate the distortion.

(1) Change of signal levels

The first measure is the mean squared difference (MSD) between two images, defined by Eq. (18). The MSD value, D_{msd}, of a watermarked image (or an image region) evaluates the mean distortion over the entire area of measurement.

(2) Change of level occurrence distributions

By h_k in every interval of $2\Delta_k$ levels in the input dynamic range, the upper half and the lower half are mapped inversely into the output dynamic range, as already shown in Fig. 9. To evaluate the change in the level occurrence distribution, we define the square variation of level occurrence between a source image X and the transformed image X', denoted by D_{dst}, by

$$D_{\mathrm{dst}} \triangleq \sum_{i=0}^{2^M-1} (\lambda_i' - \lambda_i)^2 \Bigg/ \sum_{i=0}^{2^M-1} \lambda_i^2 \tag{23}$$

where λ_i and λ_i' are the relative occurrence frequencies of a signal level i in the picture X and X', respectively, for $i = 0,\ 1,\ \ldots,\ 2^M - 1$, satisfying $\sum_{i=0}^{2^M-1} \lambda_i = 1$ and $\sum_{i=0}^{2^M-1} \lambda_i' = 1$.

4.1.3 Subjective testing

To find a correlation between the objective qualities and the subjective quality of the transformed images, we have carried out the subjective evaluations by human observers (Kimoto (2008)). In the measurement of image quality, we used a *rating-scale* method (Netravali & Haskell (1988)) where the observers viewed the test images and assigned each image to one of the given ratings. The results were presented by computing a mean value from the numerical values corresponding to the ratings, which is referred to as a Mean Opinion Score (MOS).

The testing materials were prepared in the following way: Here, assuming that $M = 8$, that is, we have only considered 8-bit images.

- The function h_k was implemented with a pseudo-random number generator of a computer in the stochastic process.
- The test images were printed on photographic papers in 200 pels per inch with an image printer, which has a printing resolution of 400 dots per inch.

All the observers, who were all in their twenties, were unfamiliar with the performance of h_k. They were asked to look at the materials sitting at a desk under ceiling lights inside a room.

Two kinds of source images were used in the testing. The first one is a 256-grayscale ramp image where the pixel levels vary linearly from 0 to 255 extending from the top side to the bottom side. Thus, the ramp image represents a low-detail region. The transformation h_k makes the linear gradation of levels distorted in the output image. Accordingly, transforming the source ramp with varying both the value of k and the ratio of pixels chosen to transform in the image, which is referred to as the transformation ratio τ, yields the distorted images of various values of D_{msd}. In the measurement of subjective quality each test image was not compared with the source image, but evaluated from a viewpoint of the appearance of level gradation and assigned one of the five ratings of the absolute rating scale listed in Table 4(b).

The other kind of source image was a granular image representing a high-detail region. The source images used in the measurement were composed of pixels of (pseudo-)random levels ranging uniformly in bounded intervals of a width of a multiple of Δ_k levels for a given k. According to the stochastic analysis of the mapping characteristics, the transformed images have the same D_{msd} (more strictly, almost same D_{msd} with a small difference due to the pseudo-random numbers) and the different D_{dst} depending on the source intervals. Each transformed image was compared with the source image just printed beside on the same paper and then, evaluated according to the degree of perceptible difference with the impairment rating scale listed in Table 4(a). Thus, eleven scores were collected for each test image, and the MOS was calculated from them.

(a)

Value	Rating
5	Imperceptible
4	Perceptible but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

(b)

Value	Rating
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 4. Ratings used in the subjective testing.

4.1.4 Subjective quality measure

In the evaluation of the ramp images, a subjective quality of each test image of a different D_{msd} value was measured in MOS. The result has indicated an approximately linear correlation between the subjective quality and the logarithms of D_{msd}. Consequently, we have observed that D_{msd} has the primary effect on the estimation of MOS for the images transformed by h_k.

In the evaluation of the granular images, those test images transformed with the same k have the same D_{msd} and the different D_{dst}. The result has demonstrated that, for a given k, the MOS values S_{mos} decrease as the D_{dst} increases. Accordingly, we suppose that the correlation between S_{mos} and D_{dst} can be considered approximately linear, while the gradient of linearity varies with the value k.

For a transformed image region, let us consider a subjective quality measure as a function of two parameters, D_{msd} and D_{dst}, that can estimate a subjective quality of the distortion, where D_{msd} and D_{dst} are measured by comparing the distorted image with the source image. According to the above analysis of the measurements, we suppose that a subjective quality measure S_{mos} can be expressed as a linear combination of the logarithm of D_{msd} and D_{dst}: That is,

$$S_{\mathrm{mos}} = \alpha \cdot \ln D_{\mathrm{msd}} + \beta \cdot D_{\mathrm{dst}} + \gamma \tag{24}$$

where α, β and γ are parameters to be estimated by multiple linear regression analysis.

To carry out the regression analysis, we used the triplets of $\{S_{\mathrm{mos}}, D_{\mathrm{msd}}, D_{\mathrm{dst}}\}$ measured from the test granular images described above. Thus, 54 measurements of the dependent variable S_{mos} at 12 different values of the independent variable vector $(D_{\mathrm{msd}}, D_{\mathrm{dst}})$ were obtained and used in the multiple linear regression analysis. As a result, the values of α, β and γ are determined as

$$\hat{\alpha} = -0.46,\ \hat{\beta} = -0.70 \text{ and } \hat{\gamma} = 6.4, \tag{25}$$

respectively. Here, the resulting coefficient of determination, which is commonly denoted by R^2, is 0.86. Let S_e be a predicted value of S_{mos} by Eq. (24) with Eq. (25). Thus, S_e yields values comparable to the MOS values.

To evaluate the effect of D_{dst} on the modeling of S_{mos}, we have also used a simple linear regression model of one independent variable of D_{msd}. In this model, S_{mos} is expressed in the form

$$S_{\text{mos}} = \alpha' \cdot \ln D_{\text{msd}} + \gamma' \tag{26}$$

where α' and γ' are the model parameters. By using the same data as those used in the above multiple regression analysis, these two parameters were estimated from 54 measurements of S_{mos} at four different values of D_{msd} in simple linear regression analysis. As a result, we obtained the estimated values of α' and γ' as

$$\hat{\alpha}' = -0.55 \text{ and } \hat{\gamma}' = 5.7, \tag{27}$$

respectively. Here, the resulting value of R^2 was 0.55. The values of S_{mos} predicted by this model have been compared with the measured values. The relation between the measured and predicted values of S_{mos} of Eq. (24) has higher correlation than that of Eq. (26) as a result. Thus, D_{dst} improves the linear model of S_{mos} as one of independent variables.

4.2 Experiments

4.2.1 Block processing procedures

The subjective quality measure S_e gives an estimate of subjective quality for each image region on the assumption that the entire region is transformed with a given value of k. Let $S_e(k)$ denote the value of S_e resulting from the transformation with the parameter k.

By using the subjective quality measure S_e, we can examine whether the transformation of an image region with a value of k satisfies a given condition of subjective quality. Accordingly, the measure can determine the values of k to transform an image region with so that the desired subjective quality can be achieved. To enhance the difficulty of detecting the values of k in use, the largest one of the available k's is to be chosen. Furthermore, the value of k to use can be changed for each image region.

To implement the above adaptive watermarking, we determine both a threshold value S_T comparable to the subjective quality measure and an upper limit of available k, which is set equal to $M-1$ here for simplicity. The S_T can be determined from the ratings listed in Table 4. Here, let us assume the transformation ratio is 1. Then, the value of k for each image region, k_B, is determined as

$$k_B = \max_{1 \le k \le M-1} \{k \mid S_e(k) \ge S_T\}. \tag{28}$$

The region is actually transformed with k_B. Thereby, the subjective image quality of $S_e(k_B)$ is achieved in the transformed region.

The procedure for implementing Eq. (28) is carried out in each image region, starting from $k = 1$ as follows:

Step 1: Transform the region with the value of k.

Step 2: Calculate D_{msd} by Eq. (18) and D_{dst} by Eq. (23) within the region.

Step 3: Calculate $S_e(k)$ as S_e of Eq. (24) with the specified coefficients of Eq. (25).

Step 4: Compare $S_e(k)$ with the given S_T. If $S_e(k) < S_T$, then, $k_B = k - 1$. Otherwise, increase k by one, and repeat from *Step 1.*

4.2.2 Examples

Simulations of the adaptive watermarking were carried out. The test images of 256 × 256 8-bit pixels, which are well known as *Lena*, *Peppers*, *Cameraman* and so on, were used as the source images. The adaptively watermarking procedure was implemented in each block of 4 × 4 pixels in the simulation with various threshold value S_T.

Fig. 16 shows an example of the values of the parameter k_B that were determined for each block by the adaptive scheme. From this figure it is observed that the adaptive scheme performs such that in the low-detail regions such as the *sky*, where the human visual system is sensitive to distortion (Netravali & Haskell (1988)), k_B's of small values are assigned, and in the high-detail regions such as the lower half area of the image, k_B's of large values can be used.

(a)

(b) k_B values

Fig. 16. The embedding parameters k_B that are determined for each block by the adaptive scheme: The block size is 4 × 4 pels and the threshold $S_T = 3.5$; (a) source image *Cameraman* of 256 × 256 8-bit pels; (b) k_B values, black means $k_B = 1$ and the brighter, the larger k_B.

Table 5 shows for the source image *Peppers* the distribution of the values of k_B that were determined by the adaptive scheme for various S_T. With increasing S_T the ratios of large k_B's decreases, and the average of k_B decreases accordingly.

Table 5 also shows the value of S_e averaged over all the blocks in the image. As this result indicates, the averaged value of S_e was achieved at about 0.5 greater than the given threshold for each of the images in the simulation. The reason why this difference is likely to occur is considered that only integers are available for k.

Examples of the images resulting from the same source image for $S_T = 3.0$, 4.0 and 5.0 are shown in Fig. 17. From these images it is observed that the better visual quality is certainly achieved as the threshold quality S_T is set larger.

4.2.3 Validity of subjective quality measure

We consider the validity of the subjective quality measure in this section. When we look at an image, we usually evaluate the quality of the whole image. Taking account of this fact, we compare the subjective quality measure to human evaluations in terms of the whole image quality.

Using various images produced in the simulation of the adaptive scheme, first, we have carried out the subjective evaluations of visual quality by use of the impairment rating scale listed in Table 4(a). The MOS value $S_{\rm mos}$ was then obtained from the scores of about forty

		Threshold S_T				
		3.0	3.5	4.0	4.5	5.0
Number of blocks	$k_B = 1$	0	0	0	2.3	44.6
(%)	$k_B = 2$	0	0.68	18.7	67.5	52.2
	$k_B = 3$	7.7	54.3	74.9	30.0	3.2
	$k_B = 4$	71.0	44.5	6.5	0.07	0
	$k_B = 5$	21.3	0.51	0	0	0
	$k_B = 6$	0.07	0	0	0	0
Averaged k_B		4.14	3.45	2.88	2.28	1.59
$\overline{S_e}$		3.60	4.10	4.54	4.99	5.46
PSNR (dB)		30.3	34.0	37.3	40.6	44.2

Table 5. Experimental result of perceptually adaptive watermarking for the source image *Peppers*: The block size is 4 × 4 pixels.

people for each image, who were given no information about the making of the images. Note that these S_{mos}'s are the human evaluation of the whole image quality. Next, to estimate subjective quality of the whole image from a collection of the calculated values of block quality, the block values S_e were averaged over all the blocks in each image, and the mean value $\overline{S_e}$ is obtained.

Fig. 18(a)–(c) shows the relationship between the mean value $\overline{S_e}$ and the measured S_{mos} in each of the three test images. A linear correlation between S_{mos} and $\overline{S_e}$ is clearly observed from either result in this figure. However, the slope of the linear regression line is 1.9, 1.8 and 2.2 in Fig. 18(a), (b) and (c), respectively.

This inclined linear correlation results in the incorrect prediction of subjective quality by the developed measure; for example, as Fig. 18 shows, viewers evaluated the quality of the image at the worst MOS of 1, while the mean value of block quality indicates that the image possesses the quality of MOS of 3.

The value $\overline{S_e}$ can be corrected using the corresponding value of S_{mos} by linear regression analysis. The linear regression model is expressed in the form

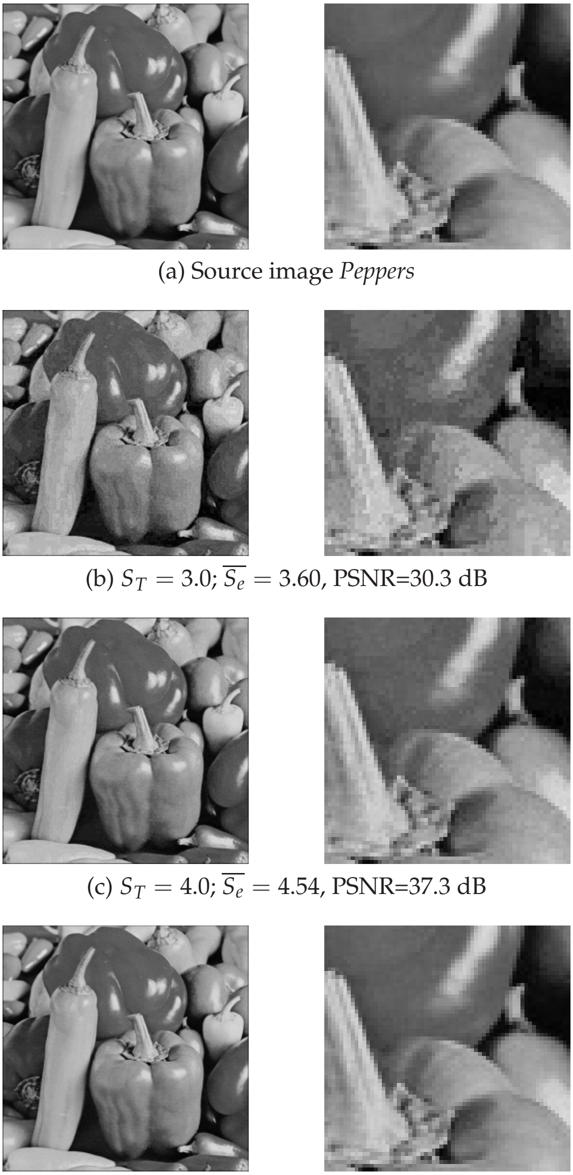
$$S_{\text{mos}} = \mu \cdot \overline{S_e} + \nu \tag{29}$$

where μ and ν are to be estimated by simple linear regression analysis. To carry out the analysis, we collected the pairs of $\{\overline{S_e},\ S_{\text{mos}}\}$, where $1 \leq \overline{S_e} \leq 5$, from the results of three images shown in Fig. 18(a)–(c). As a result, the parameters μ and ν are estimated as

$$\hat{\mu} = 1.9, \text{ and } \hat{\nu} = -4.4, \tag{30}$$

respectively.

Using Eq. (30), the values of $\overline{S_e}$ in Fig. 18(a)–(c) were modified to $\overline{S_e}'$. The resulting relationships between S_{mos} and $\overline{S_e}'$ are shown in Fig. 18(a)′–(c)′. In this figure, each value of S_{mos} is shown with its 95% confidence interval. The slope of any linear regression line in the figure is about 1.1. Furthermore, from the viewpoint of the confidence intervals of MOS, the linear correlation looks almost valid. Consequently, $\overline{S_e}'$ can be used to predict the evaluation of subjective quality for at least these three images.



(a) Source image *Peppers*

(b) $S_T = 3.0$; $\overline{S_e} = 3.60$, PSNR=30.3 dB

(c) $S_T = 4.0$; $\overline{S_e} = 4.54$, PSNR=37.3 dB

(d) $S_T = 5.0$; $\overline{S_e} = 5.46$, PSNR=44.2 dB

Fig. 17. Results of the adaptive watermarking with the various threshold values S_T: Each right column image magnifies a 64 by 64 pel region of the left one, whose size is 256 by 256 pels.

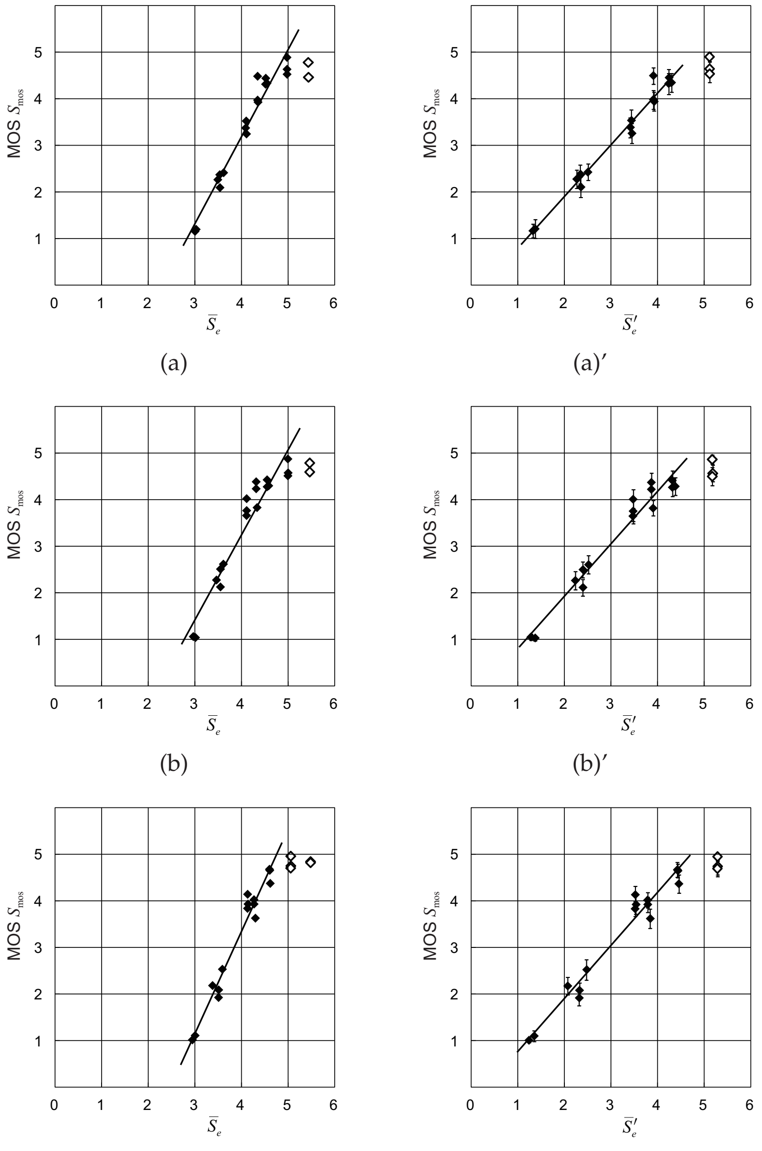


Fig. 18. MOS versus the subjective quality measure averaged over the blocks $\overline{S_e}$ ((a)–(c)) and MOS versus the modified values $\overline{S_e}'$ ((a)′–(c)′) for each source image: (a) and (a)′ source *Lena*; (b) and (b)′ *Peppers*; (c) and (c)′ *Cameraman*; the line in each figure shows the linear regression of the data points; the points painted solid in white are excluded due to out of range $[1,\ 5]$.

5. Conclusion

The first result of this chapter is the bit inverting transformation, h_k. This level transformation performs all the three functions simultaneously: (a) It represents the inversion of a specified bit; (b) it reduces the level change caused by the bit inversion to the minimum; and (c) it adds a random variation to the output levels under limitations on level changes. The transformed level that has both the specified, say, kth bit inverted and the level change minimized includes the lowest $(k-1)$ bits either of all 1's or all 0's. In contrast, for most of the input levels, some of these bits or all of them are replaced with random bits by randomly varying the transformed levels. Accordingly, the transformed pixels are hard to discriminate without any information regarding the locations in the watermarked images.

The properties of the subjective quality measure, which is the second result of this chapter, are summarized below:

- The subjective quality measure is the function of two objective quality measures, D_{msd} and D_{dst}, which are both obtained from each image region. Consequently, the subjective quality measure is essentially to be carried out in block processing.
- The subjective quality measure estimates a value of MOS in the impairment rating scale. Hence, a threshold value for the measurements can be given in terms of subjective evaluation.
- The values of the subjective quality measure are dependent both on the bit position to be inverted and on the image texture. This property enables the bit position to be altered according to the image texture so as to achieve the adaptive scheme.

The subjective quality measure was derived from the measurements of the computer synthesized test patterns. The validity of the measure for images of natural scenery was examined in terms of the whole image quality. We obtained the whole image quality value by averaging the block quality values over the image. In the experiment using three test images, a highly but inclined linear correlation was observed between the mean value and the actually measured MOS. Although the inclined gradient was successfully corrected by simple linear regression analysis for the images used in this experiment, the cause of the difference between the estimated values and measured values may be related to human eye's characteristics (Netravali & Haskell (1988)). Accordingly, we have to consider a method for estimating the whole image quality from block qualities in more detail.

Another remaining subject is related to the block size of the subjective quality measure. The block size of 4×4 has been used in the experiment. The block size is related to the resolution of quality measure. Besides, because a bit position to be inverted is decided at each block in the adaptive scheme, as many bit positions as the blocks must be stored in a secure manner. From these viewpoints, the appropriate block size should be considered.

6. References

Macq, B. M. ed. (1999). Special issue on identification and protection of multimedia information. *Proc. IEEE*, Vol. 87, No. 7, July 1999, pp. 1059–1276.

Wang, F.H., Pan, J.S. & Jain, L.C. (2009). *Innovations in Digital Watermarking Techniques*. Springer-Verlag, Berlin Heidelberg.

Oka, K., Matsui, K. (1997). Signature method into gray-scale images with embedding function. *IEICE Transactions on Information and Systems* , Vol. J80-D-II, No. 5, May 1997, pp. 1186–1191 (in Japanese).

Kimoto, T. (2005). Implementation of level transformations for hiding watermarks in image bit-planes under limited level changes. *Proc. of the IEEE International Conference on Image Processing (ICIP 2005)* , pp. 253–256. Genova, Italy, Sept. 2005.

Kimoto, T. (2007). Modified level transformation for bit inversion in watermarking. *Proc. of the IEEE International Conference on Image Processing (ICIP 2007)* . San Antonio, USA, Sept. 2007.

Kimoto, T. (2006). A sophisticated bit-conversion method for digital watermarking. *Proc. of the 8th IASTED International Conference on Signal and Image Processing* , pp. 139–144. Honolulu, USA, Aug. 2006.

Kimoto, T. (2009). An advanced method for watermarking digital signals in bit-plane structure. *Proc. of IEEE International Conference on Communications (ICC 2009)* , SPC-P2.8. Dresden, Germany, June 2009.

Awrangjeb, M., Kankanhalli, M.S (2004). Lossless watermarking considering the human visual system. In: *Digital watermarking*, pp. 581–592. Kalker, T., Cox, I.J., Ro, Y.M. eds (2004). Springer, 2004.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13 (4) 600–612.

Cox, I.J., Miller, M.L., Bloom, J.A (2002). *Digital watermarking*. Morgan Kaufmann Publishers.

Mikami, D., Shimizu, M., Makabe, S., Kamiyoshihara, Y., Kimoto, T (2008). Measurement of subjective quality of watermarked images made by inverting bits. *Proc. of IEEE TENCON 2008* , O17-7. Hyderabad, India, Nov. 2008.

Netravali, A.N., Haskell, B.G (1988). *Digital Pictures*. Plenum Press, New York, USA.

Jain A.K. (1989). *Fundamentals of digital image processing* Englewood Cliffs NJ, Prentice-Hall.

Kimoto, T., Kosaka, F. (2010). A perceptually adaptive scheme for image bit-inversion-based watermarks. *Proc. of the 6th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS2010)* , pp. 114–122. Kuala Lumpur, Malaysia, Dec. 2010.

6

Performance Evaluation for IP Protection Watermarking Techniques

Tingyuan Nie
Qingdao Technological University
China

1. Introduction

The advance of processing technology has led to a rapid increase in IC design complexity. There are now more than thousand million transistors integrated on a chip, and the increasing trend is expected to continue until 2020 or later. This creates the design productivity gap between IC design (typically 20% per year) and IC manufacturing (over 40% per year), and this gap is becoming wider and wider. To close this gap, IP (intellectual property) reuse emerged as the most significant design technology innovation in the past decades. IP companies, third-party libraries, and industry organizations such as the VSIA (Virtual Socket Interface Alliance) have created high expectations for the value and reusability of design IP.

The IP reuse in the reuse-based design methodology is rather different from other reuses such as media, devices to produce artifacts. The reuse of components, designed for a class of applications, is a method to reduce the design-effort, which is well-known from software design for a long time already. In the field of IC design, the reuse of blocks has been practiced in design houses mainly in form of an evolution of existing products. Due to shorter product cycles and rapidly increasing product complexity, many design companies will more and more refer to module cores from outside. During the process of the transfer of design blocks from the original provider to the integrator, intellectual property issues have to be seriously considered. At the same time, some essential issues for IP reuse are outlined: design quality, documentation, security, support, and integration (Thomas et al., 2001). As suggested in the "Reuse Methodology Manual for System-On-A-Chip Designs" (Keating & Bricaud, 1998), an example process of integrating IPs and doing physical chip design can be broken into the following steps:

Selecting IP blocks and preparing them for integration;
Integrating all the IP blocks into the top-level RTL;
Planning the physical design;
Synthesis and initial timing analysis;
Initial physical design and timing analysis, with iteration until timing closure;
Final physical design, timing verification, and power analysis;
Physical verification of the design.

There are many solved or unsolved issues need to be addressed for IP market: friendly interface between IP provider and IP user, design-for-manufacturing, design-for-test, design-for-reuse, IP standardization, rules for IP exchange, and so on. IP reuse is based on information sharing and integration. Therefore piracy will also have much easier access to the IPs. The IP piracy affects the IP vendors, chip design houses as well as system manufacturers adversely by depriving their revenue and market share. As a result, recent trends of IP piracy have raised serious concerns among the IC design community.

In response to these trends, IP protection becomes crucial to both IP vendors and IP users and becomes one of the key solutions for industrial reuse-based integration. Although sometimes the lack of mechanisms for IP protection becomes barriers to increase design productivity, there have been significant advances from both industry and academic. Especially the VSIA's white paper on IP protection (VSIA, 2000a) and physical tagging standard (VSIA, 2000b) has now been widely adopted by semiconductor and EDA industry. Numerous protection techniques are proposed by researchers both from industry and academia. There exist three forms of IP protection techniques: tagging, fingerprinting, and watermarking. The idea of tagging proposed by Marsh & Kean is to provide a "security tag" for the IP core which can easily be detected off chip using an external receiver called as "wand" (Marsh & Kean, 2007). The approach is vulnerable because the tag can be easily removed by someone if he/her knows some information about the tagging. Bolotnyy & Robins use PUFs (Physically Unclonable Functions) to create aboard RFID (Radio Frequency Identification) tags to protect ICs from cloning (Bolotnyy & Robins, 2007). The security is really improved. However the PUF design is so complicated that the manufacture is hardly reachable. Majzoobi et al. proposed a "Lightweight Secure PUFs" with the new structure in low area, power, and delay overheads. The appeoach facilitates easy security versus implementation cost trade-offs (Majzoobi et al., 2008). There are also other variants in PUF researches, such as implementation of PUFs exploiting physical characteristics other than timing and delay information of silicon circuits. Ravikanth et al. Proposed an optical PUF, which uses the speckle patterns of optical medium for laser light (Ravikanth et al., 2001). Coating PUFs and acoustic PUFs measure the capacitance of a coating layer covering an IC and the acoustic reflections of a token, respectively (Skoric et al., 2005; Tuyls et al., 2005).

Among these techniques, watermarking is the most extensive mechanism implemented at multi-levels of IC design procedure. Primitive watermarking, also known as data hiding, embeds data into digital media for the purpose of identification, annotation, and copyright. The rapid development of digitized media and the internet revolution are creating a pressing need for copyright enforcement schemes to protect copyright ownership. Numerous techniques for data hiding in digital images, videos, audios, texts and other multimedia data have been developed. All these techniques take advantage of the limitation of human visual and auditory systems, and simply embed the signature to the digital data by introducing minute errors. The transparency of the signature relies on human's insensitiveness to these subtle changes. For detail survey, refers to (Gang & Potkonjak, 2003). Especially, watermarking techniques in VLSI domain protects IP cores, CAD tools as well as algorithms from illegal reuse.

CAD tools and algorithms are protected as traditional software by mechanisms such as licensing agreements and encryption. Despite the lack of enforcement of licensing agreements and the security holes of encryption protocols, these protections do not provide the ability to detect IP piracy (Lin et al., 2006). The rare technique that detects possible CAD tool and algorithm piracy is the forensic engineering approach proposed by

Kirovski et al. (Kirovski et al., 2000). It enables the identification of solutions generated by strategically different tools and algorithms. They simply check the given solution for the properties that the algorithm clustering has been performed and claim that the solution is obtained by the algorithm that has the best fit. The poor application to distinguish different algorithms as well as the requirement of candidate algorithms and computing resource is the lack of this technique. So the need for effective CAD tools and algorithms protection becomes vital and urgent. CAD tools and algorithms protection are not in the scope of this book, our work focuses on watermarking techniques for the protection of reuse IP core. We review the representative watermarking techniques and evaluate their performance for both ASIC (Application-Specific Integrated Circuit) and FPGA (Field Programmable Gate Array) designs.

Fingerprinting technology is a complementary to watermarking due to the demand of ensuring the rights of both IP provider and IP users. The main challenge of fingerprinting technique is how to create numerous IP cores with the same function for different IP users. The common approach is to acquire each IP user's signature and repeat embedding it into the entire design to create high-quality solutions from scratch within reasonable amortized design cost.

To the best of our knowledge, the first IP fingerprinting technique is published by Lach et al. (Lach et al., 1998). Their approach is based on the solution by partitioning an initial solution into a large number of parts to provide different fingerprinting realizations (a restricted FPGA mapping problem). Unfortunately, the technique cannot be applied if the design do not have natural geometric structure. Also it has relatively low resilience against collusion attacks due to the identical global structure and the time overhead for creating fingerprinted solutions is relatively high. Andrew et.al proposed a generic fingerprinting methodology that applies to arbitrary incremental optimization/synthesis problems on an watermarked initial "seed" solution to yield different but functionally identical fingerprinted IPs. The approach enhanced collusion resiliency with low runtimes but different solutions are not guaranteed (Andrew et.al, 1999). Gang and Miodrag proposed a fingerprinting technique which uses arbitrary optimization on the problem formulation superimposed additional constraints to produce numbers of distinct solutions with high quality. The run-time overhead for generating many solutions is almost zero (Gang & Miodrag, 2004).

The remainder of this section is organized as follows. We first review the related works of watermarking techniques. Analyze the representative watermarking techniques, introduce watermarking performance evaluation function, and show experimental results for watermarking techniques of ASIC. Followed give a simplified FPGA watermarking investigation and estimation. Finally we have a conclusion for overall work.

2. Watermarking performance evaluation

Referencing viewpoints by VSI Alliance (FallWorldwide Member Meeting, 1997), a state-of-art watermarking-based IPP technique should be:

1. Maintenance of functional correctness.
2. High-credible; coincidence probability, the probability a non-watermarked design might coincide by accident with a watermarked one should be low enough.
3. High-security; watermark should be in the integrity or can be extracted under attack.

4. Low embedding cost.
5. Low overhead.
6. Traceable.

According to the requirements of watermarking, a complete methodology for watermarking performance evaluation should be established. Unfortunately, limited to our knowledge, there is no comprehensive evaluation function for IP watermarking techniques so far. The only literature published for watermarking investigation is accomplished by Abdel-Hamid et al. (Abdel-Hamid et al., 2003). However, they only compared performance of the approaches from their embedding cost, overhead, probability of coincidence, and security. There was no more deeply analysis and evaluation for the watermarking techniques.

In the context, we introduce representative watermarking techniques and evaluate their performance for the two usual IC forms: ASIC and FPGA respectively.

2.1 Watermarking performance evaluation for ASIC

From watermarking construction style, there are almost two methods for watermarking ASIC IP cores. One focuses on introducing additional constraints on certain parts of the solution space of synthesis and optimization algorithms. Another is adding redundancies to the original design.

From VLSI design process, pre-processing watermarking methods and post-processing watermarking methods are discussed. Pre-processing techniques embed watermark before the synthesis tools are applied to solve the watermarked problem. Post-processing techniques firstly solve the original problem without any watermarks. The solved solution will be altered sequentially based on the watermarking constraints. According to design process, watermarking techniques at behavioural-level, structural-level, physical-level, and algorithm-level are proposed.

There may be some shortfalls or defects for a certain watermarking technique. It becomes an important work to evaluate the performance of a watermarking technique because the approaches may bring influences to the origin. So it is impending to build methodologies and functions for watermarking performance evaluation.

2.1.1 Watermarking technique review

In this section, we firstly review a few representative watermarking techniques constructed at different design levels. Then analyze them form a few essential aspects: embedding cost, coincidence probability, security, and tracing cost.

2.1.1.1 Physical-level watermarking

Kahng et al. firstly proposed the constraint-based watermarking methodologies based on the usage of available tools which solves NP-hard problems (Kahng et.al, 1998). The algorithm adds extra constraints to such solutions that would make it yield the new watermarked design. They validated the approaches in pre-processing and post-processing, respectively. The pre-processing flow provides a method that adds constraints by involving segment widths, spaces, and choice of topology. They applied the watermarking by encoding a signature as upper bounds on the wrong-way wiring used to route particular signal nets. The post-processing flow provides a method that encodes a signature as specified parity of the cell row within which particular standard cells must be placed. Narayan et.al provided a method for embedding a watermark by modifying the number of vias or bends of the nets in a design (Narayan, et.al, 2001). There were 12~13% expense in the number of vias and wire length which is unpractical in real life. The author also proposed a post layout watermarking method which smartly changes route directions by setting obstacle and rerouting (Nie, et.al, 2005). There was no extra wire length overhead and the incremental watermarking time is acceptable. Other techniques at physical design level are also proposed (Min & Zhiqiang, 2004; Irby et.al, 2000).

For physical design watermarking, we choose the most representative technique proposed by (Kahng et.al, 1998) for evaluation instance. According to the published results, the extra routing CPU run time for watermarking is about 9.00%; increased wire-length and via number (watermarking overhead) are 0.58% and 0.55% respectively, sum is 1.13%; the coincidence probability geometrically reduced to the constraint number, from 1.1e^{-8} (nearly 10^{-3} for 20 constraints) to less than e^{-85} (nearly 10^{-25} for 320 constraints). From their analysis, the approaches can prevent "ghost signatures" and forging attack due to enough-long constraints and message encoding. They also showed the result from tampering with placement and routing watermark which indicates solution quality degrades much faster than signature strength. It proves that tampering does not appear to be a viable form of an attack.

2.1.1.2 Behavioral-level watermarking

Torunoglu et.al and Oliveira introduced a similar watermarking-based copyright protection technique of sequential functions at behavioral design level (Torunoglu & Charbon,2000; Oliveira, 2001). The algorithm is based on adding new input/output sequences to the finite state machines (FSM) representation of the design. It extracts the unused transitions in a state transition graph (STG) of the behavioral model. These unused transitions are inserted in the STG associated with a new defined input/output sequence, which will act as the watermark. The main advantage of this kind approach is the ability to detect the presence of the watermark at all lower design levels. Torunoglu and Charbon performed exhaustive search only in one case due to the extreme computational complexity of this method. The CPU time in this case was 1.0 second for an area of 2.33-k gates, but it increases exponentially according to their computation formula. The coincidence probability of watermarking is from 10^{-7} to 10^{-34}, averagely 10^{-11}. The watermarking overhead (Extra area of modified FSM) is from 0.2% to 143%, average is 23.77%. It will be much larger if the expected watermark becomes longer. The number of I/O pins which is used to create sequence to insert watermark is not very long, so the approach's resistance to "ghost signatures" attack is not as strong as expected. They proved the "tampering" attack will not successful under various assumptions. Unfortunately, because there is no encryption for the watermarking, the approach is weak to "forging" attack.

2.1.1.3 Structural-level watermarking

There are few watermarking works at structural-level. Kirovski et.al developed a watermarking approach to protect EDA tools and designs at the combinational logic synthesis level. The user-specific watermarking instance is soluted by imposing constraints to the original logic network, where the constraints are uniquely dependent on author's signature (Kirovski et.al, 1998). Cui and Chip-Hong also proposed the similar approach by resynthesizing the "master design" to meet the application constraints.

We select the first appraoch as representative for structural-level watermarking performance evaluation. From their result, the runtime for the watermarking was controlled within ±5% of the program execution runtime. The average likelihood of watermarked solution coincidence is less than 10^{-13} with the overhead of 4%. Because the adopted watermark constraint length is short (5-inputs), its resistance to "ghost signatures" attack is likely low. They proved that the attacker has to perturb great deal of the obtained solution to tamper the watermark while preserving solution quality, like to develop a new optimization algorithm. For "forging" attack, it is less efficient than trying to tamper the signature in a top-down approach and it is more impossible in a bottom-up approach due to the one-way function encoding.

2.1.1.4 Algorithm-level watermarking

There are rare approaches at the algorithmic level. Chapman & Durrani proposed a Digital Signal Processing (DSP) watermarking scheme (Chapman & Durrani, 2000). The algorithm is based on the ability of designers to make minor changes in the decibel (db) requirements of filters. In this approach, the designer of a high level digital filter encodes one character (7 bits) as his/her hidden watermark data. Then the high level filter design is divided into 7 partitions where each partition is used as a modulation signal of one of the bits.

The authors did not discuss the strength of their approach or the probability *Pc* that the design might coincide with a non-watermarked design. The approach as well depends on a very low data rate, just one character (7 bits), which makes it really unpractical to be used in an industrial environment. The approach is also missing a clear way to track and extract the watermark at lower levels. Therefore, we think the approach is incipient and do not evaluate its performance.

2.1.2 Watermarking performance evaluation function

As described in the context, we consider performance evaluation of watermarking techniques from five aspects: embedding cost, coincidence probability, overhead, security, and trace cost. The components of watermarking performance are illustrated in Fig.1. We formulate watermarking technique performace P using the following function:

$$P = F(Em\_Cost, Coin\_pro, Overhead, Security, Trace\_\cos t) \tag{1}$$

Where P is a function with six variables: Em_Cost, Coin_Pro, Overhead, Security and Trace_cost. Em_Cost represents watermarking embedding cost which usually means the additional wire length or vias for watermarking represetation. Overhead represents how long EDA tools runt for watermarking process. Coin_Pro represents the probability that the watermarked design coincided with a non-watermarked one. Security represents strength of watermarking technique resists to various attacks. Trace_Cost displays the cost retrieving watermark from a protected IP design that can be considered almost the same to the embedding cost. Maintenance of functional correctness is not considered as a factor of the function because each watermarking technique in the market should at least satisfy this requirement.

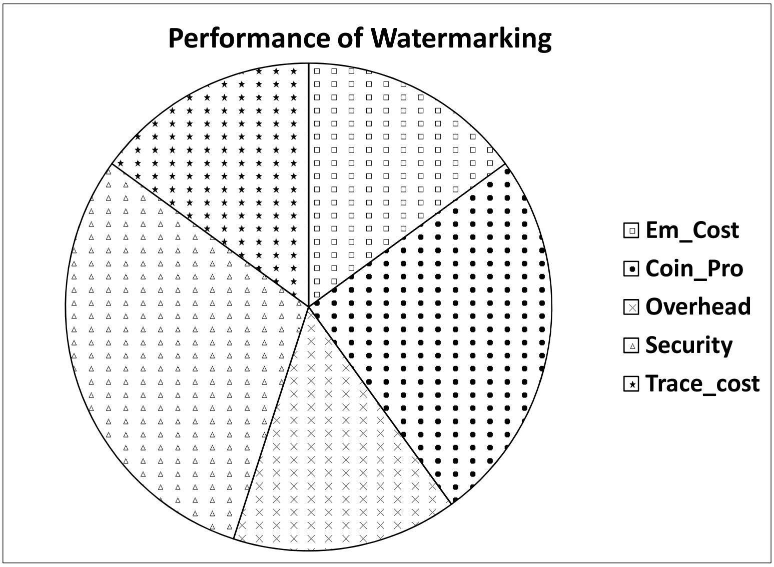


Fig. 1. Watermarking Performance Components

Obviously, lower watermarking cost leads to high-performance watermarking technique. Therefore watermarking performance is reverse to embedding cost. Similarly, watermarking performance is reverse to coincidence probability, overhead, and tracing cost. Instead watermarking performance is proportional to its security which must be concerned. We give a function f_i and a weight to each component, equation (1) can be formulated as:

$$P = \alpha \bullet f_1(Em\_Cost) + \beta \bullet f_2(Coin\_pro) + \gamma \bullet f_3(Overhead) + \lambda \bullet f_4(Security) + \mu \bullet f_5(Trace\_Cost) \quad (2)$$

In practice, watermarking tracing cost is almost equal to watermarking embedding cost, so formula (2) can be simplified as:

$$P = 2\alpha \bullet f_1(Em\_Cost) + \beta \bullet f_2(Coin\_pro) + \gamma \bullet f_3(Overhead) + \lambda \bullet f_4(Security) \quad (3)$$

Each part of formulation (3) is related to both watermark constraints size and watermarking method. Consider process of watermarking-based IP protection, we evaluate the performance of watermarking techniques in such rules:

The watermarking IP protection process is implemented by either intrusive software or an incremental implementation of EDA tool. So the additional CPU run time of the implementation is considered as the embedding cost. Generally, watermarking identification needs some extra circuits. We take the increased wire-length and (or) via number as watermarking overhead. It is considered that the security of watermarking techniques is related with its resistance to attacks. There is a brief introduction of prototypical attacks referred in (Kahng et.al, 1998). The attacks include "ghost signatures" finding, tampering, and forging. To find "ghost signatures", hacker may try a brute-force approach to find a signature that corresponds to a set of constraints that yields a convincing proof of authorship Pc. However, this brute-force attack becomes computationally infeasible if the threshold for proof of authorship is set sufficiently low. e.g., Pc≤2^{-x} (x is the length of constraints). So it is easy to prevent this type attack just by enlarging the length of signature (watermark). As an alternative, attackers may select re-solving every subsequent stage of the watermarking process to forge author's signature. Generally, Specific changes that attacker makes to the final solution will likely correspond to (1) local perturbations of the solution to the watermarked phase, or to (2) global-scale transformations such as those which exploit asymmetry of the design representation. It is critical that common watermarking technique has the resistance to such transformations. Tampering attacks might not be able to ruin the proof of authorship before they significantly degrade the quality of the final solution. Finally, attacker may select to forge author's signature. To finish this work, he needs a signature that he can convince others belong to author. If a signature corresponds simply to a text message, he simply chooses a text message resembling one that author would use. However, such attacks can be easily prevented by using a private key encryption system for watermark generation. We analyze the security of watermarking techniques from the above three aspects, and give a quantitative performance evaluation.

2.1.3 Watermarking analysis summary

Through the investigation and analysis, performance of various watermarking techniques is summarized in Table 1. There are total six columns each display the item of watermarking performance. The first column displays watermarking type. The second and the fifth column are the watermarking embedding cost and tracing cost which represent the increased CPU runtime corresponding to normal IC design process. The forth column of "Overhead" represents the increased percentage of wire length and via number. In the fifth column of "security", there are 3 sub-columns: G, T, and F which represent the resistance to "ghost signatures", "tampering", and "forging" separately. The value of "+" means the method has resistance to such attack, while the value of "-" means no resistance or resistance is really weak.

Watermarking	Em_Cost	Coin_Pro	Overhead	Security			Trace_Cost
				G	T	F	
Physical	9.00%	10^{-3}~10^{-25}	1.13%	+	+	+	9.00%
Behavioral	expensive	avg: 10^{-11}	23.77%	+	+	-	expensive
Structural	5.00%	< 10^{-13}	4.00%	-	+	+	5.00%

Table 1. Performance summary of watermarking techniques

2.1.4 Evaluation results

We evaluate the representative watermarking algorithms from five items: embedding cost, coincidence probability, overhead, security, and tracing cost. According to the investigated results, we calculate each sub-value in the scope (0, 1). Finally we accumulate all the value as the performance evaluation by using formula (3).

Performance of watermarking technique is related with the run time of watermarking process. The more time consumed, the more watermarking technique is ineffective. We define sub-function f_1 as:

$$f_1(Em\_Cost) = 1 - \frac{Em\_\cos t}{Design\_\cos t} \tag{4}$$

Where Design_cost displays the original design cost. If Em_cost (embedding cost) is too expensive to exceed Design_cost, the value of function f1 will be equal to 0.

If the coincidence probability of a watermarking techniques is sufficiently low (for example, less than 10^{-3}), f_2 function can be set as:

$$f_2(Coin\_pro) \equiv 1 \tag{5}$$

The overhead of watermarking (increased wire length, extra via, etc.) degrades the performance of the design. The function f_3 can be written as:

$$f_3(Overhead) = 1 - \frac{Overhead}{Total} \tag{6}$$

Where Total is the total cost for the original design.

There are three factors impact the watermarking security: the resistance to "ghost signature", "tampering", and "forging". We think that no matter which factor is satisfied, the watermarking security gets 1/3 value augment. The f_4 can be written as:

$$f_4(Security) = N \times \frac{1}{3} = \frac{N}{3} \tag{7}$$

Where N is the number of satisfied factors.

Substituting the formulations (4), (5), (6) and (7) into formulation (3) and the of set Design_cost and Total to 1, we have:

$$P = 2\alpha(1 - Em\_\cos t) + \beta + \gamma(1 - Overhead) + \lambda \bullet N / 3 \tag{8}$$

We prepare three schemes to evaluate performance of watermarking techniques: (a) **Balance evaluation** where each item weights are the same, namely $\alpha=\beta=\gamma=\lambda=0.2$; (b) **Cost emphasis evaluation** where the weights of cost and overhead are set double to others, namely $\alpha=\gamma=0.25$ and $\beta=\lambda=0.125$. (c) **Security emphasis evaluation** where security weight is set double to others, namely $\lambda=2/6$ and $\alpha=\beta=\gamma=1/6$. (All the weights obey: $2\alpha+\beta+\gamma+\lambda = 1$)

Based on formulation (8), we calculated the concrete performance value for the several watermarking techniques. The results are shown in Table 2.

Scheme	Physical WM	Behavioral WM	Structural WM
Balance	0.9214	0.4858	0.9053
Cost_Emphasis	0.9268	0.3989	0.8867
Security_Emphasis	0.9345	0.5159	0.8656

Table 2. Performance of watermarking techniques

The first column show the evaluation schemes mentioned above: Balance evaluation, Cost emphasis evaluation, and Security emphasis evaluation. The second, the third and the forth column show evaluated performace value of different watermarking techniques in various test schemes. From the result, we understand performance of physical watermarking representative is high, then structural watermarking representative, and behavioral watermarking representative is relatively low, no matter the scheme. From the curves in Fig.2, we can understand the comparison more intuitively.

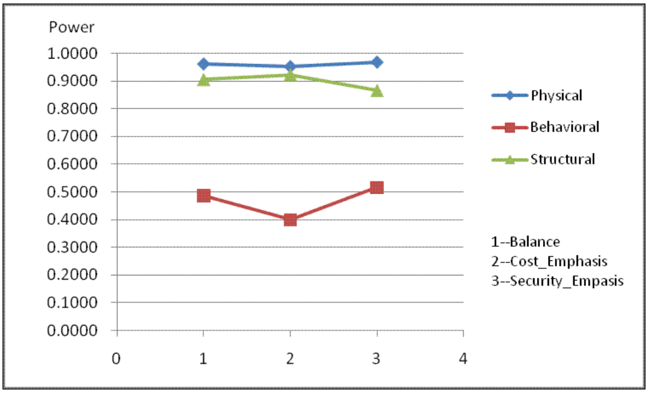


Fig. 2. Performace illustration of watermarking techniques

We introduce functions to evaluate watermarking techniques and hope this work can provide a standard candidate for researchers to evaluate their watermarking techniques. Although performace of various watermarking techniques is different, even the weak technique has its advantages. In future, researchers may develop stronger watermarking techniques by combining the advantages of different level watermarking techniques to prevent any IP piracy attempt from happening.

2.2 Watermarking performance evaluation for FPGA

Before the FPGA being watermarked, a signature should be prepared. The signature may be a short ASCII-text, which identifies the owner of the core. The string is then hashed and encrypted to generate a seed of watermark. Then the watermark is produced from the seed with a pseudo random generator like RC4.

Fig.3 gives an example of FPGA watermarking design flow. As shown in the figure, there are three types of FPGA cores: Source-cores, netlist-cores and bitfile-cores, corresponding to the design levels. Source-cores are delivered in HDL or C language. There are very flexible to synthesize for many target technologies. Netlist-cores have a medium flexibility because they have been fixed on a target technology. Bitfile-cores are very inflexible since they can be used only for a specific device.

Daniel & Jurgen had an accurate evaluation of watermarking methods for FPGA-based IP cores from functional correctness, hardware overhead, transparency, verifiability, difficulty to remove, and proof strength of authorship (Daniel & Jurgen, 2006). They divided watermarking techniques into two categories from their construction: additive methods and constraint based methods. In this chapter, we introduce recent FPGA watermarking techniques and estimate their performance under certain criteria.

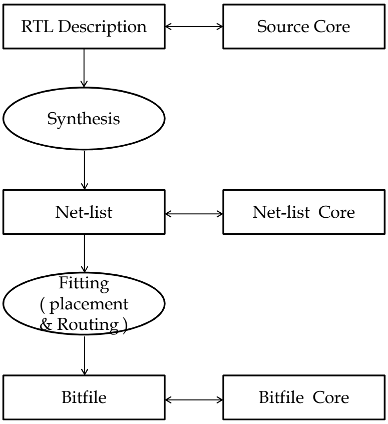


Fig. 3. FPGA watermarking design flow and IP core

2.2.1 Additive methods

Additive methods in FPGA design are watermarking procedures where a signature is added to the functional core. The watermark is not embedded into the functional core yet be masked as a part of the core.

There exist no publications about additive watermarking for source cores protection although it is possible to write an additive source component into the core. However it isn't an applicable watermark strategy because one can also remove this component easily.

Most additive watermarking methods for netlists just watermark the design by introducing redundant logic to the circuit. Moritz et.al presented a novel approach to watermark FPGA designs by converting functional LUTs (Lookup Tables) to LUT-based RAMs or shift registers prevents deletion due to optimization (Moritz et.al, 2009). The resource overhead for watermarking is tiny, generally less than 5%. The method is transparent to EDA tools because the watermarking is performed after the usual netlist generation. The suspected design can be verified only when the extracted bitfile is not encrypted. The authorship can be detected without requesting additional information from the producer. However the watermark can be easily removed by reverse-engineering and the authorship will dispear.

An approach for watermarking bitfile-core is implemeted by embedding the signature into unused look-up tables (John et.al, 1998). The signature will be hashed and coded with an error correction code (ECC) to be able to reconstruct even if some lookup tables are tampered. After the initial placement and routing, the number of unused lookup tables are determined. The ECC code is split into the size of the lookup tables and additional LUTs are added to the design. The watermarked design is obtained after being re-placed and re-routed.

The approach was improved (John et.al, 1999) by using many small watermarks whose size is the exact size of a lookup table. The small watermarks are easier to search relatively. However, the published watermark positions in verification process make the watermarking technique easily attacked. Futhermore, Lach et.al improved the approach to a fingerprinting technology by encoding the fingerprint into the position of the mark in the tile (Lach et.al, 1998).

The watermark consumes low hardware overhead because the unused lookup tables in the original design would remain empty. The approaches provide a strong proof of authorship and are transparency to EDA tools. The methods are verifiable because it is possible to determine the position of the watermark in a tile. On the contrary, the watermark is alos easy to be remoed or overwrited.

2.2.2 Constraint based methods

The constraint based watermarking methods apply to solutions of hard optimization and constraint-satisfaction design problems. It is centered around the use of constraints to "sign" the output of a given design synthesis or optimization. The solutions of a given optimization instance that satisfy these constraints have a watermark embedded in them and provide a probabilistic proof of authorship. The less likely that randomly chosen solutions are to satisfy these constraints, the stronger the proof of authorship is. The coincidence probability Pc is given by the following formula:

$$P_c = n_w / n \tag{9}$$

where n is the number of solutions which satisfy only the original constraints and n_w is the number of solutions which satisfy both the original and the watermarking constraints. If Pc is very small, the solution provide a strong proof of the watermarking existence. A watermark's resistance to attacks is inversely proportional to an adversary's ability to manipulate it without resolving a given optimization problem from scratch.

Darko & Miodrag proposed an approach for a HDL core protection using a watermarked scan chain (Darko & Miodrag, 1998). At first all registers will be sorted to be assigned a sequential number. A pseudo random sequence is generated from author's signature to select registers according to a certain algorithm. The first K selected registers are chosen for the first register in a chain, where K is the number of used scan chains. The variation of the scan chains for different signature can be used to detect the watermark. Unfortunately, an injudicious chosen of test chain could result in more routing resources overhead. The approach is transparent to the synthesis tools because the signature is added to the HDL core. The watermark can be verified easily only when the scan chains can be accessed from outside of the chip. Some deletion of watermark results in corruption of the scan chain. In additional, a strong proof of authorship can be achieved by using a large number of registers in scan chains.

An approach to protect netlist cores is implementing by preserving certain nets in the synthesis and mapping step (Kirovski et.al, 1998). Some nets are chosen from the sorted nets of design according to a signature. These nets are prevented from elimination by the design tools by connecting to a temporary output of the core. Additional logic is inserted to connect the new outputs together to reduce the amount of the additional outputs. The design with new outputs can be seen as the result of constraint based watermarking. The additional logics for watermarking require some resource overheads. This approach is transparent to EDA tools because the choice of preserved nets for watermarking can be done before the synthesis process. The watermark can be verified by comparing the given netlist with the original one. However, it is impossible to verify the watermark from a bitfile. The security of this approach is insufficient because the additional logic is easy to remove by re-synthesizing the design. Furthermore, although the probability of coincidence is really low, forging watermarked design is possible which results in weak authorship proof.

An incremental placement and routing or timing constraint is applied to watermark FPGA bitfile-cores.

As an alternative, a watermark can be embedded by placing configurable logic blocks (CLBs) in even or odd rows depending on the constraints (Kahng et.al, 1998). The resource overhead for watermarking is very low and even tends to zero because the placement is altered marginally. The approach is transparent because the watermarking stage is performed before placement implementation. The CLBs can be corresponded to the signature uniquely by enumerating them form the top left corner. Then the watermarked design can be verified with only the given bitfile. It is nearly impossible to remove watermark from the given bitfile because the CLBs are tightly connected with each other. This approach has a strong proof of authorship due to the large amount of CLB position candidates for watermark embedding.

Another proposed method is to add constraints to the router. The constraints make the router route a net with some unusual routing resources like "wrong way" segments, in which the net goes to a wrong direction and then back in the right direction to form a backstrap. The net can be verified as a watermark net due to its special geometry. The routing resource for watermarking is too minor to be neglected. The approach is also transparent to EDA tools because constraints are added before invoking routing. The watermarked design can be verified with the known strategy and the unique nets. It is easy to remove the mark by wrapping up the constraint nets and rerouting it again if someone knows the routing information and the watermarking algorithm. The proof of authorship is not very strong because the watermark is ambiguous and easy to remove or tamper.

A watermarking approach by setting additional timing constraints between registers is proposed in (Kahng et.al, 2001). The timing constraints for the selected paths may split into two separate constraints, each have a new constraint.

Another approach selects the uncritical paths and adds new timing constraints on them (Adarsh et.al, 2003). The last digit of the time delay is reset depending on the watermark. For example, a path has a delay of 10.64ns. If the corresponding watermark bit is '1', the new time delay of this path is set to10.61ns, if the corresponding watermark bit is '0', the delay is set to 10.60ns.

These approaches for watermarking need no resource overhead. They are transparent because additional constraints are added before invoking the routing tool. These approaches are difficult to verify so that it no use to talk about their authorship proof and attack resistance. However designers can create different bitfiles from the same design which are useful for fingerprinting.

2.2.3 FPGA watermarking validation

As mentioned in (Daniel & Jurgen, 2010), when considering a finished FPGA products, there are five potential information sources can be used for extracting a watermark: configuration bitfile, ports, power consumption, electromagnetic (EM) radiation, and temperature.

The bitfile can be extracted by wire tapping the communication between the PROM and the FPGA. Some FPGA manufactures provide an option to encrypt the bitstream which makes communication monitoring useless. However, it is possible to read out some information stored in RAMs or lookup tables to finish verification. Another approach is to employ unused ports which is limited only at top-level designs and impractical for IP cores.

The method called "Power Watermarking" can force patterns on the power consumption of an FPGA as a covert channel to transmit data to the outside. Related works shown in (Ziener & Teich, 2008) and (Ziener et.al, 2010) indicate the clock frequency and toggling logic can be used to control such a power spectrum covert channel. The resulting change in power consumption can be extracted as the signature from the FPGA's power spectrum.

With almost the same strategy it is also possible to extract signatures by raster scanning electromagnetic (EM) radiation of an FPGA with an EM sensor (Thomas & Christof, 2003). Unfortunately, it becomes unpractical since modern FPGAs are delivered in a packaged shape which decreases the EM radiation.

Finally, a watermark might be read out by monitoring the temperature radiation which is similar to power and EM-field watermarking approaches. There is only one commercial watermarking approach which reads a watermark from an FPGA taking up to 10 minutes (Kean et.al, 2008).

3. Conclusion

In this section, we first reviewed several classical IP protection methods such as tagging, fingerprinting, and watermarking. Then we investigated representative watermarking techniques of ASIC at different design levels. We proposed functions to evaluate watermarking techniques from the under aspects: embedding cost, overhead, coincidence probability, security and tracing cost. The evaluated results show that the performance of physical watermarking technique is high, structural watermarking technique is medium, and behavioral watermarking technique is low. We also summarized watermarking techniques of FPGA core protection and validation methods from three forms of FPGA: source code, netlist, and bitfile.

From this work, we hope it provides a standard candidate for researchers to evaluate their watermarking techniques. In future, researchers may develop stronger watermarking techniques by combining the advantages of different level watermarking techniques to prevent any IP piracy attempt from happening.

4. Acknowledgment

We are grateful to our work team for the contribution to this research. The Project is sponsored by SRF for ROCS, SEM. and Supported by Shandong Province Natural Science Foundation (ZR2009GL007) and A Project of Shandong Province Higher Educational Science and Technology Program (J09LG10). The author also thanks for the support of the family.

5. References

Abdel-Hamid, A.T.; Tahar, S. & El, M.A. (2003). IP watermarking techniques: survey and comparison. *Proceedings of IWSOC2003 3rd IEEE Int. Workshop on System-on-Chip for*

Real-Time Applications, pp.60–65, ISBN 0-7695-1944-X, Calgary, Alberta, Canada, June 30-July 2, 2003

Abdel-Hamid, A.T.; Tahar, S. & El Mostapha Aboulhamid. (2006). Finite state machine IP watermarking. *Proceedings of AHS 2006 1st NASA／ESA Conference on Adaptive Hardware and Systems,* pp.457-464, ISBN 0-7695-2614-4, Istanbul, Turkey, June 15-18, 2006

Adarsh, K.J. ; Lin, Y. ; Pushkin R.P. & Gang Q. (2003). Zero overhead watermarking technique for FPGA designs. *In GLSVLSI '03: Proceedings of the 13th ACM Great Lakes symposium on VLSI,* pp. 147–152, ISBN 1-58113-677-3, USA, 2003

Aijiao, C. & Chip-Hong, C. (2006). Stego-signature at logic synthesis level for digital design IP protection, *Proceedings of 2006 IEEE International Symposium on Circuits and Systems,* pp. 4611-4614, ISBN 0-7803-9389-9, Island of Kos, Greece, May, 2006

Andrew, E. C.; Hyun-Jin, C.; Andrew, B. K.; Stefanus, M.; Miodrag, P.; Gang, Q. & Jennifer, L. W. (1999). Effective Iterative Techniques for Fingerprinting Design IP, *Proceedings of the 36th annual ACM/IEEE Design Automation Conference,* pp. 208-215, ISBN 1-58113-109-7, New York, NY, USA, 1999

Bolotnyy, L. & Robins, G. (2007). Physically unclonable function-based security and privacy in RFID systems. *Proceedings of PERCOM 2007 5th IEEE International Conference on Pervasive Computing and Communications,* pp.211-220, ISBN 0-7695-2787-6, Washington, DC, USA, March 19-23, 2007

Chapman, R. & Durrani, T.S. (2000). IP protection of DSP algorithms for system on chip implementation. *IEEE Trans. on Signal Processing,* vol. 48, No. 3, (March 2000), pp. 854-86 1, ISSN 1053-587X

Daniel, Z. & Jurgen T. (2006). Evaluation of Watermarking methods for FPGA-based IP-cores. *Technical Report 01-2006,* Erlangen, Germany, Mar, 2006

Daniel, Z. & Jurgen, T. (2010). New Directions for FPGA IP Core Watermarking and Identification, *In Proceedings of Dagstuhl Seminar 10281,* 2010

Darko, K. & Miodrag, P. (1998). Intellectual property protection using watermarking partial scan chains for sequential logic test generation, *Proceedings of 1998 International Conference on Computer-Aided Design ICCAD,* 1998

FallWorldwide Member Meeting: (1997). A Year of Achievement (Guidelines Proposed by VSIA Development Working Group on Intellectual Property Protection). VSI Alliance, Santa Clara, CA, 1997

Gang, Q. & Miodrag, P. (1999). Effective iterative techniques for fingerprinting design IP, *Proceedings of Design Automation Conference,* pp. 587–592, ISSN 0278-0070, Los Angeles, CA, June, 1999

Gang, Q. & Miodrag, P. (2003). *Intellectual Property Protection in VLSI Design: Theory and Practice,* Kluwer Academic Publishers, ISBN 978-1-4020-7320-5, USA

Irby, D.L.; Newbould, R.D.; Carothers, J.D.; Rodriguez, J.J. & Holman, W.T. (2000). Low level watermarking ofVLSI designs for intellectual property protection. *Proceedings of IEEE 13th lnt · ASlC/SOC Conferenc,* pp. 136 – 140, ISBN 0-7803-6598-4, Arlington, VA , USA, September, 2000

John L.; William H. M. & Miodrag P. (1998). Signature hiding techniques for FPGA intellectual property protection. *In proceedings of ICCAD International Conference on Computer-Aided Design,* pp. 186–189, ISBN 1-58113-008-2, California, USA, 1998

John L.; William H. M. & Miodrag P. (1999). Robust FPGA intellectual property protection through multiple small watermarks. *In proceedings of DAC99 Design Automation Conference,* pp. 831–836, ISBN 1-58113-092-9, USA, 1999

John, L.; Miodrag P. ; William, H.M. & Miodrag, P. (2001). Fingerprinting Techniques for Field-programmable Gate Array Intellectual Property Protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.20, No.10, (October 2001), pp. 1253-1261, ISSN 0278-0070

Kahng, A.B.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Huijuan, W. & Wolfe, G. (1998). Robust IP watermarking methodologies for physical design. *Proceedings of DAC 35th Design Automation Conference,* pp.782-787, ISBN 0-89791-964-5, San Francico, California, USA, June 15-19, 1998

Kahng, A.B.; Lach, J.; Mangione-Smith, W.H.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Wang, H. & Wolfe, G. (1998). Watermarking techniques for intellectual property protection. *Proceedings of DAC98 35th ACM/IEEE Design Automation Conference,* pp. 776–781, ISBN 0-89791-964-5, San Francisco, CA, USA, June 15-19, 1998

Kahng, A.B.; Lach, J.; Mangione-Smith, W.H.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Wang, H.; & Wolfe, G. (2001). Constraint-based watermarking techniques for design IP protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.e 20, No. 10, Oct. 2001, pp. 1236-1252, ISSN 0278-0070

Kean, T.; McLaren, D. & Marsh C. (2008). Verifying the Authenticity of Chip Designs with the DesignTag System. *In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust,* pp. 59-64, ISBN 978-1-4244-2401-6, Washington, USA, June, 2008

Kirovski, D. ; Liu, D. ; Wong, J.L. & Potkonjak, M. (2000). Forensic Engineering Techniques for VLSI CAD Tools, *Proceedings of 37th ACM/IEEE Design Automation Conference,* pp. 581-586, ISBN 1-58113-187-9, Los Angeles, CA, June, 2000

Kirovski, D.; Yean-Yow Hwang; Potkonjak, M. & Cong, J. (1998). Intellectual property protection by watermarking combinational logic synthesis solutions. *Proceedings of ICCAD 1998 IEEE/ACM International Conference on Computer-Aided Design,* pp. l94-l98, ISBN 1-58113-008-2, San Jose, CA, USA, November 8-12, 1998

Keating, M. & Bricaud, P. (1998). *Reuse Methodology Manual for System-on-a-Chip Designs,* Kluwer Academic Publishers, ISBN 0792385586, Boston, USA, 1998

Lach, J.; Mangione-Smith, W.H. & Potkonjak, M. (1998). FPGA Fingerprinting Techniques for Protecting Intellectual Property, *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference,* pp. 299-302, ISBN 0-7803-4292-5, Santa Clara, CA, May, 1998

Lin Y.; Qu, G.; Ghouti, L. & Bouridane, A. (2006). VLSI Design IP Protection: Solutions, New Challenges, and Opportunities, *In Proceedings of Adaptive Hardware and Systems 2006,* pp. 469-476, ISBN 0-7695-2614-4, NY, USA, June 15-18, 2006

Lin, Y.; Gang, Q. ; Lahouari, G. & Ahmed, B. (2006). VLSI design IP Protection: Solutions, New Challenges, and Opportunities, *Proceedings of AHS 2006 1st NASA/ESA Conference on Adaptive Hardware and Systems,* pp. 469-476, ISBN 0-7695-2614-4, Istanbul, Turkey, June 15-18, 2006

Majzoobi, M.; Koushanfar, F. & Potkonjak, M. (2008). Lightweight secure PUFs, *Proceedings of Computer-Aided Design 2008,* pp. 670-673, ISBN 978-1-4244-2819-9, San Jose, CA, 2008

Marsh, C. & Kean, T. (2007). A security tagging scheme for ASIC designs and intellectual property cores. *Proceedings of IP-SoC 2006 IP Based SoC Design Conference & Exhibition,* pp. 6-7, France, January 2007

Min, N. & Zhiqiang G. (2004). Constraint-based watermarking technique for hard IP core protection in physical layout design level. *Proceedings of IEEE 7 Int · Conf · on Solid-State and Integrated Circuits Technology,* pp.1360-1363, ISBN 0-7803-8511-X, Beijing, China, October, 2004

Moritz, S. ; Daniel, Z. & Jurgen, T. (2008). Netlist-Level IP Protection by Watermarking for LUT-Based FPGAs. *Proceedings of FPT 2008 International Conference on ICECE Technology 2008,* pp. 20 -216, ISBN 978-1-4244-3783-2, Taipei, China, Dec. 2008

Narayan, N.; Newbould, R.D.; Carothers, J.D.; Rodriguez, J.J. & Holman, W.T. (2001). IP Protection for VLSI Designs Via Watermarking of Routes. *Proceedings of 14th Annual IEEE International ASIC/SOC Conference,* pp.406-410, Washington, DC, USA, September, 2001

Nie, T.; Kisaka, T. & Toyonaga, M. (2005). A watermarking system for IP protection by a post layout incremental router. *Proceedings of DAC 42th Design Automation Conference,* pp.218-221, ISBN 1-59593-058-2, San Diego, CA, USA, June 13-17, 2005

Oliveira, A.L. (2001). Techniques for the creation of digital watermarks in sequential circuit designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* VOL. 20, NO. 9, September, 2001, pp.1101-1117, ISSN 0278-0070

Ravikanth, P.; Ben R.; Jason, T. & Neil, G. (2001). *Physical One-Way Functions,* PhD thesis, Massachusetts Institute of Technology

Skoric, B.; Tuyls, P. & Ophey, W. (2005). Robust key extraction from physical unclonable functions, *Proceedings of the Applied Cryptography and Network Security Conference 2005,* pp. 407-422, ISSN 0302-9743, berlin, 2005

Thomas, H.; Zebo, P. ; Raimund, U. ; & Manfred, G. (2001). Challenges for Future System-on-Chip Design. *Proceedings of ECCTD15th European Conference on Circuit Theory and Design,* pp.173-176, Espoo, Finland, August 28-31, 2001

Thomas W. & Christof P. (2003). How Secure Are FPGAs in Cryptographic Applications. *In Proceedings of International Conference on Field Programmable Logic and Applications (FPL 2003),* Lecture Notes in Computer Science Volume 2778, pp. 91-100, Sept. 2003

Torunoglu, I. & Charbon, E. (2000). Watermarking based copyright protection of sequential functions. *IEEE Journal of Solid-State Circuits,* vol. 35, No. 3, (May 1999), pp.434-440, ISBN 0-7803-5443-5, 2000

Tuyls, P.; Skoric, B. ; Stallinga, S. ; Akkermans, A. & Ophey, W. (2005). Information theoretical security analysis of physical unclonable functions. *Proceedings of Conference on Financial Cryptography and Data Security 2005,* pp. 141-155, ISSN 0302-9743, berlin, 2005

Virtual Socket Interface Alliance (2000a). Intellectual Property Protection White Paper: Schemes, Alternatives and Discussion Version 1.0. September 2000

Virtual Socket Interface Alliance (2000b). Virtual Component Identification Physical Tagging Standard (IPP 1 1.0). 2000

Ziener, D. & Teich, J. (2008). Power Signature Watermarking of IP Cores for FPGAs. *Journal of Signal Processing Systems,* VOL. 51, 2008, pp.123-136

Ziener, D.; Baueregger, F. & Teich, J. (2010). Using the Power Side Channel of FPGAs for Communication. *In Proceedings of the 18th Annual International IEEE Sympo- sium on Field-Programmable Custom Computing Machines (FCCM 2010),* pp. 237-244, ISBN 978-0-7695-4056-6, Carolina USA, May, 2010

Using Digital Watermarking for Copyright Protection

Charlie Obimbo and Behzad Salami
University of Guelph
Canada

1. Introduction

Without a doubt, the Internet has revolutionized the way we access information and share our ideas via tools such as Facebook, twitter, email, forums, blogs and instant messaging. The Internet is also an excellent distribution system for digital media. It is inexpensive, eliminates warehousing and delivery, and is almost instantaneous. Together with the advances of compression techniques such as JPEG, MP3 and MPEG; the Internet has become even faster, easier and more cost effective to distribute digital media such as audio, video, images and documents over the World Wide Web.

In addition to existing web sites and shared networks, the recent development of peer-to-peer (P2P) file distribution tools such as Kazaa, Limewire, Exceem or eMule enables a copious number of web users to easily access and share terabytes of digital media across the globe. These technologies also significantly reduce the efforts of pirates to illegally record, sell, copy and distribute copyright-protected material without compensating the legal copyright owners.

Today, content owners are eagerly seeking technologies that promise to protect their rights and secure their content from piracy, unauthorized usage and enable the tracking and conviction of media pirates. Cryptography is probably the most common method of protecting digital content [Koch & Zhao, 1995], where the content is encrypted prior to delivery and a decryption key is provided to those who have purchased legitimate copies. However, cryptography cannot help the content providers monitor their goods after the decryption process; a pirate could easily purchase a legit copy and then re-sell it or distribute it for free over a shared network.

It is therefore important to find a way to protect these digital media with a more stringent method, which would enable the vendors and artists / photographers / directors get confidence in placing and distributing their material over the Internet. Watermarking could be such a vehicle.

2. Overview

Digital watermarking is a field that refers to the process of embedding digital data directly onto multimedia objects such that it can be detected or extracted later.

It has three unique advantages over other techniques such as cryptography. First of all, it is imperceptible and does not affect the aesthetic of the digital data. Secondly, watermarks become fused with the actual bits of the work, unlike headers they do not get removed when the work is displayed, copied or during format changes. Lastly, they undergo the same transformation as the work itself and sometimes the extracted mark can be used to learn about the history of transformations that the work has undergone.

In general any watermarking system consists of three components

a. Watermark generation stage,
b. encoding and
c. decoding [12].

Watermarking can be applied to various digital multimedia such as images [Wolfgang et. al, 1999 & Hartung & Kutter, 1999], videos [Ren-Hou et. al., 2005 & Lie et. al., 2006], audio [Liu & Innoue, 2003 & Berghel, 1997], or text [Huang & Wu, 2004]. Image watermarking is either perceptible or imperceptible to the human eye and can be designed to be robust, fragile or semi-fragile [Koch & Zhao, 1995].

An example of a basic visible watermark would be placing a text or logo onto an image to identify it's rightful copyright owner (see Figure 1). As seen in Figure 1, an image can be placed on the web in low resolution as an advertisement. The purchaser would then receive a copy minus the watermark, on completion of the purchase, from the vendor.

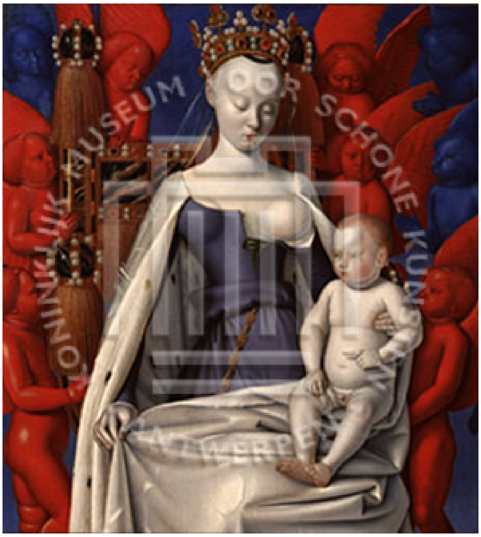


Fig. 1. Example of a visible watermark

Visible marks are usually embedded in the spatial domain, that is, directly onto the pixel values of an image. Clearly, this method is fragile and can easily be compromised by cropping or replacing the text using either a basic image processing tool such as Microsoft Paint, advanced software such as Adobe Photoshop or sophisticated Algorithms such as Huang & Wu's [Huang & Wu, 2004 & Baaziz, 2005].

As a result various other domains have been proposed. In current literature, the watermark is added to the image either in the spatial domain or in a transform domain [Leighton et. al. 1997]. Example of transform domains are discrete Fourier transform (DFT), the full-image discrete cosine transform (DCT) [Bartolini et. al., 2001], the block-wise DCT [Wolfgang et. al, 1999], the discrete wavelet domain (DWT) [Cappellini et. al., 1998], fractal domain [Puate et. al., 1996 & Shahraeini and Yaghoobi, 2012], the redundant contourlet transform [Leighton et. al., 1997], the Hadamard domain, Fourier- Mellin domain or the Radon domain [Lie et. al., 2006]. It has been shown that embedding the mark in the mid-frequencies of a transform domain is advantageous in terms of visibility and security over the spatial domain [Cheng et.al., 1999].

In the embedding stage visibility artifacts must be avoided and thus the Human Visual System (HVS) must be taken into account. The watermark is generally shaped using spatial or spectral shaping to reduce it's energy in areas where the mark would become visible [Lie et. al.]. An image adaptive watermarking scheme uses the local or global characteristics of the original image to determine the maximum strength that can be achieved in each area without introducing visible artifacts [Cappellini et. al., 1998]. Image-adaptive watermarking Algorithms have been proposed in [Hartung et. al., 1999, Podilchuk et. al., 1998, Cappelini et. al., 1998].

Watermarking techniques that do not require the original image for verification or extraction of the watermark are called "blind" watermarking as opposed to "informed" watermarking [Liu & Innoue, 2003, Cappellini et. al., 1998, Anderson & Petitcolas, 1998, Koch & Luo, 1998.].

The functions of the digital watermarking technology can be classified in four broad categories [Miller et. al.]:

a. Copyright Protection,
b. Monitoring,
c. Authentication, and
d. Secure and Invisible Communications.

Each individual application area desires its own set of special requirements with regards to robustness, fidelity and capacity [Lie et. al.].

In spite of the fact that digital watermarking has been an active area of research for decades, there is still a lot of room for improvements. One main reason for this is the limitations associated with each technique and the need to find the best balance between the three conflicting requirements (robustness, fidelity and capacity).

Robustness calls for the watermark to be as strong as possible where the fidelity requirement asks the watermark to be invisible.

It is difficult to satisfy all the requirements to their maximum at the same time. In current systems image watermarks are typically a pseudo-random signal with much lower amplitude, compared to the original image amplitude and usually with distribution of each bit into a group of pixels [Wolfgang et. al]. The pseudorandom signal is generally generated with Gaussian, uniform or bipolar probability density distribution using a secret seed.

Watermarks could also be a string of bits or a pseudo-randomly generated set of real numbers or a small image such as a company logo.

These watermarks however, often carry no extra information and are not very useful. On the other hand, multi-bit watermarks typically include a second signal used as error correction and thus decrease the amount of useful information or the payload that can be embedded.

Below are some Watermarking applications.

2.1 Watermark applications

Copyright Protection: Content providers such as individual artists or large-scale broadcast companies are interested in enforcing copyright protection of digital media [Koch & Luo, 1998, Berghel, 1997]. Authors wish to be ensured that their products are not commercially used without the payment of royalties. Another branch of this technology is fingerprinting. A product is marked with a unique label or fingerprint and then distributed to the rightful customer. Fingerprinting and Copyright applications require a high degree of robustness, and should be imperceptible but may have low capacity.

2.2 Monitoring

Digital watermarks can also be used to track and monitor digital content. In medical applications, watermarks might be used for identification and accessing of individual patient records. This particular application may prevent human errors such as record mismatching therefore preventing fatal mistakes [Koch & Luo, 1998]. In broadcast monitoring, companies like to confirm that their advertisements receive the full amount of airtime purchased. They have a desire to ensure that their product is broadcasted with the full duration, at the most optimal time of the day, and at preferred strategic frequencies [Cox, 2008]. Also, companies may wish to monitor the advertisement of the competition to predict future business strategies or explore competitive marketing techniques.

2.3 Authentication

For proof of authentication watermarks can be used not only to identify if a digital file has been tampered with, but also to determine how it has been tampered with. Such information can possibly give clues on how to reverse the malicious tampering to recover the original data. Authentication of surveillance cameras can be of importance if authorities question the reliability of such evidence in courts [Koch & Luo, 1998].

2.4 Communication

The idea of covert or secret communication is as old as communication itself [Hartung & Kutter, 1999] and is used frequently by defence and intelligence sectors. Digital watermarking continue to exist even after the receiver has obtained the information. If sensitive data is leaked out to unauthorized personal, the digital watermark contained in them can be used to trace back to the original owner or the intended receiver [Koch & Luo, 1998].

Digital watermarking used as covert communication adds an extra level of security compared to cryptography. In cryptography, the data is encrypted and can only be decrypted using a secret key. However, the attacker is aware of the existence of such data and can be certain that with enough time, he can decrypt the data, where as in digital watermarking, the attacker can never be certain that secret information is being transmitted.

Another advantage of digital watermarks is that it continues to exist even after the receiver obtains the information. Digital watermarking combined with cryptography is highly desired.

In this Chapter we will describe a watermarking algorithm for digital images for the purpose of copyright protection.

3. The watermarking process

In general any watermarking system consists of three components Watermark generation stage, encoding and decoding [Bartolini et. al., 2001].

3.1 Watermark generation

The watermark signal is typically a pseudo-random signal with much lower amplitude, compared to that of the original image and usually with distribution of each bit into a group of pixels [Hartung & Kutter, 1999]. The pseudo-random signal is generally generated with Gaussian, uniform or bipolar probability density distribution using a secret seed. Watermarks could also be a string of bits or a pseudo-randomly generated set of real numbers or a small image such as a logo.

3.2 Watermark encoding

The general idea is to embed a unique mark into a digital image such that it cannot be perceived by the Human Visual System but can be extracted at a later time using the content owner's secret key to prove ownership. Figure 2 shows the general example of encoding and decoding of a 4096 bit mark into the image "Lena". The mark is a binary image that has been uniquely generated by the watermarking system.

The cover image is first transformed into a domain that facilitates data embedding. The watermark can be embedded or encoded generally by adding or multiplying the signal to the cover image's luminance channel, the colour channels or both. For increased security and invisibility a spread spectrum coding with combination of a shaping technique is applied. In spread spectrum coding the watermark signal is spread over another known signal and then added to the image. Shaping can be done by increasing and decreasing the watermark's energy in some areas to adapt (become less visible) to the original work. In the DCT-Block domain, coefficients are modified according to the watermark content either by re-quantization, substitution or modification to impose a relationship [Bartolini et. al., 2001], [Koch & Zhao, 1995]. A General Watermarking encoding is described in Figure 3.

Watermark decoding

In the extraction stage some watermarking techniques need the original host image for subtracting the watermarked images, such techniques are referred to as "Informed" or

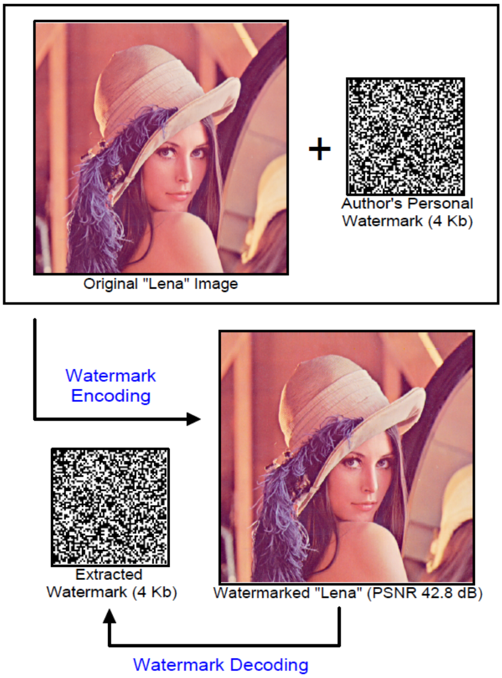

+
Author's Personal Watermark (4 Kb)
Original "Lena" Image
Watermark Encoding
Extracted Watermark (4 Kb)
Watermarked "Lena" (PSNR 42.8 dB)
Watermark Decoding

Fig. 2. General example of watermarking an image

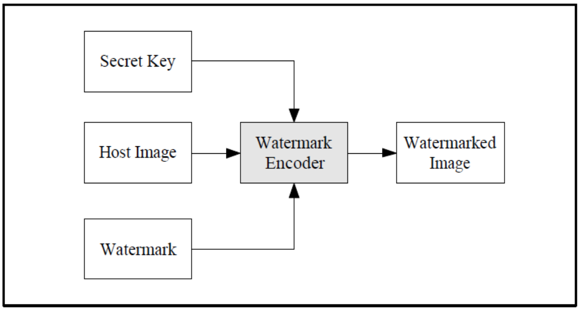


Fig. 3. General watermark encoding diagram

private watermarking [Miller et. al., 2002]. Other techniques do not need the original host image but need a secret seed to generate the original watermark for comparison. Such systems are referred to as “blind” watermarking. A watermarking system is “semi-blind” if it relies on some data or features derived from the original host image.

It is important to distinguish between watermark verification and watermark extraction. In most of literature the watermark is only verified, that is a correlation between the potential watermarked image and the original watermarked image is performed using the normalized correlation defined in Equation 1.

$$SIM(W,W^*) = \frac{W^* \cdot W}{\sqrt{W^* \cdot W^*}} \tag{1}$$

The output of a verification system is a yes/no answer. Extraction of the watermark is performed by reconstructing the watermark bit by bit from a potential watermarked image and comparing it with the original watermark. A threshold is defined for the percentage of similarity (Bit Error Rate) between the two. The basic process is depicted in Figure 4. An image marked with a watermark and a secret key are used by the watermark decoder to extract the original watermark signal.

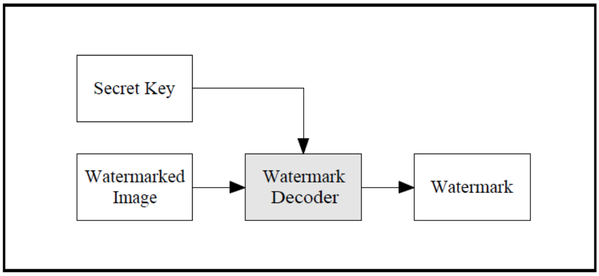


Fig. 4. General watermark decoding diagram

3.3 Watermark generation algorithm

Watermarks can take many shapes such as a company logo, image of a text or a pseudo-randomly generated sequence of bits or real numbers. We propose a new watermark with properties of self-correction. The Error Correction stage performs without any additional sources or reference marks. The author of the host image has the ability to specify personal information such as name, creation date, transaction ID or image ID as a human readable string of characters.

The provided information string is denoted as *S*, where S_i represents the *i*th characters in the string. First, each character S_i is converted to it's binary representation B_i and all B_i's are concatenated to form a sequence of bits denoted as B. For example, the binary representation of the string "Ben" is "01000010 01100101 01101110", where the spaces are only added for ease of visual distinction.

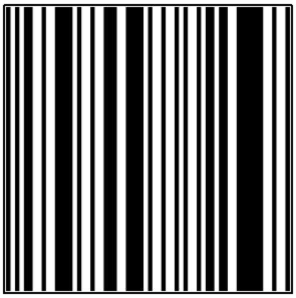


Fig. 5. Personal watermark of the string "Salami06" before encryption

In the next stage the sequence B is converted to a binary image, where a "0" represents a white pixel and a "1" represents a black pixel. The sequence is repeated vertically generating a barcode like image, illustrated in Figure 5. The mark uses a 64 bit information, duplicated 64 times, resulting in 4096 individual bits or 4 kilobytes. The vertical dimension of the mark depends on the height of the host image I , the larger the image dimensions the more the string can be repeated, thus increasing the robustness.

The dimensions of the image is determined by the Equation

$$W_h = \frac{I_h \times I_w}{\text{Strlen}(S) \times 8} \tag{2}$$

where W_h is the height of the watermark image, I_h and I_w are the dimensions of the host image and Strlen(S) is a function that returns the number of characters in the string S provided by the owner or author.

The watermark image is further encrypted using a user specified seed K_{mark} into a fast uniform pseudo-random number generator called "Mersenne Twister" with a period of $2^{19937} - 1$. The algorithm was developed by M. Matsumoto and T. Nishimura [Matsumoto & Nishimura, 1998] in 1998 and improved in 2002 [Matsumoto & Nishimura, 2002]. The generator is implemented to generate fast output by completely avoiding divisions and multiplications. It generates an array at one time and takes the full advantage of cache memory and pipeline processing if supported. Figure 6 depicts an example of an encrypted watermark

Fig. 6. Example of an encrypted watermark

Experts consider this an excellent random number generator. Using the seed Kmark, a sequence of N long-integer values ranging from 0 to $N - 1$ is generated where $N = W_h \times W_w$. The result is a 1-Dimensional array of pseudo-randomly generated values denoted as R, where R_i denotes the ith value in the list.

In order to encrypt the watermark the forward-scrambling Algorithm 1 is used, where each individual pixel W_i is exchanged with the corresponding pixel WR_i defined by R_i. The

Decryption method is very similar except that the shuffling is performed in the reverse order, for more details see Algorithm 2.

Algorithm 1 (Encrypt) *Encrypts the Watermark using Mersenne Twister*

```
ENCRYPT-MARK($W$, $K_{mark}$, $N$)
R ← GENERATERANDOMS($N$,$K_{mark}$)
for $i \leftarrow 0$ to $N$
    do $Temp \leftarrow W_i$
        $W_i \leftarrow W_{Ri}$
        $W_{Ri} \leftarrow Temp$
DELETE($R$)
```

Algorithm 2 (Decrypt) *Decrypts the Watermark using Mersenne Twister*

```
DECRYPT-MARK($W$, $K_{mark}$, $N$)
R ← GENERATERANDOMS($N$,$K_{mark}$)
for $i \leftarrow N - 1$ to $0$
    do $Temp \leftarrow W_i$
        $W_i \leftarrow W_{Ri}$
        $W_{Ri} \leftarrow Temp$
```

3.4 Watermark encoding algorithm

We embed the watermark into the DCT-Block domain of the host image. The DCT-Block has the advantage of revealing the local image characteristics [Cox & Li, 2005] and unlike using the full frame DCT, the watermark strength can be adapted to each local frequency content. This method proves to achieve maximum watermark fidelity [De Rosa et. al., 2000].

At first, the general encoding procedure is briefly described to allow the reader a broad conceptual view of the algorithm. Then in subsequent sections the algorithm is disassembled in individual components and each is further described in greater detail. The algorithm can be divided in three general stages. Image Preparation: The image is segmented into individual non-overlapping blocks, the colour space is converted from RGB to YCrCb (YUV) and each 8 × 8 block is transformed from the spatial to the frequency domain.

Watermark Encoding: The properties of the Human Visual System is explored and image adaptive strengths are determined for each block, the blocks are checked for potential edges before the pixels of the watermark image can be embedded. A testing mechanism ensures that the pixel was correctly embedded. Image Finalization: This stage is exactly the same as "Image Preparation" only in the reverse order. Each block is transformed back from the frequency to the spatial domain and the colour space is converted back from YCrCb to RGB. Lastly, all blocks are re-assembled to form the final watermarked image.

A pseudo-code of the encoding method is described in Algorithm 3 and for a more visual representation please refer to Figure 7.

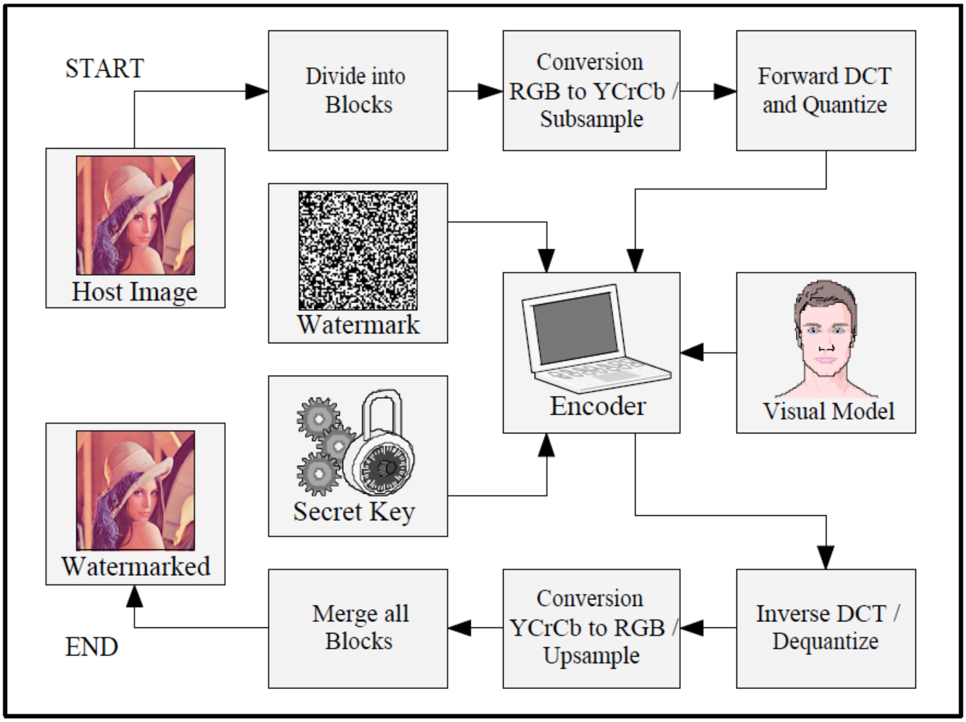


Fig. 7. Proposed watermark encoding diagram

Algorithm 3 (Encode) *Encodes the watermark W in image I*

```
Encode$(W, I, K_{image}, \alpha, Target, ET)$
    Prepare-Image$(I, N, Y, B, I^{crcb})$
    ShuffleBlocks$(Y, K_{image})$
    CalcWatsonsSlacks$(Y, S)$
    Encode-Watermark$(N, Y, W, S, ET, \alpha, Target)$
    UnShuffleBlocks$(Y, K_{image})$
    $I^W \leftarrow$ Finalize-Image$(Y, I^{crcb}, N, B)$
    return $I^W$
```

In Algorithm 3 *W* is the encrypted watermark and I is the original host image. K_{image} is used for shuffling of blocks, α is the user defined watermark strength, Target is a value that can be toggle between 0 and 255 to minimize the number of changes that the encoding algorithm must perform. ET is the edge threshold used in edge classification of blocks. The image I is first segmented via a call to Prepare-Image(I, N, Y, B, I^{crcb}) where *N* luminance blocks denoted as *Y* together with the chrominance components I^{crcb} are extracted.

Algorithm 4 (Prepare) *Prepares image I for encoding*

```
$\textsc{Prepare-Image}(I, N, Y, B, I^{crcb})$
$N \leftarrow 0$ and $B \leftarrow 0$
$I^{Ycrcb} \leftarrow$ NIL and $Y \leftarrow$ NIL
for each $16 \times 16$ $Block$ in $I$
    do $\textsc{Convert-ColorSpace}(I_B^{RGB}, I_B^{Ycrcb})$
       $I_B^{crcb} \leftarrow I_B^{Ycrcb} - I_B^{Y}$
       $B \leftarrow B + 1$
       for each $8 \times 8$ $Luminance\ Block$ in $I_B^{Y}$
           do $\textsc{ForwardDCT}(I^{DCT})$
              $Y^N \leftarrow \textsc{Quantize}(I^{DCT})$
              $N \leftarrow N + 1$
```

The image I is first segmented into 16 × 16 non-overlapping *RGB* blocks, and for each the colout space is converted from *RGB* to *YCrCb* with a subsampling ratio of (4:1:1) obtaining I^{Ycrcb}. The chrominance I^{crcb} and luminance I^{Y} components are separated. Then the Block-DCT is applied to transform each 8 × 8 luminance block from the spatial domain to the frequency domain, followed by a lossy quantization step similar to JPEG compression.

The final quantized luminance blocks are saved in the set Y for the embedding procedure. In the next subsection and the discrete cosine transform (DCT) is described in more detail.

3.4.1 Discrete cosine transform / quantization

After the colour conversion, the luminance (Y) component is extracted and a 2- Dimensional Discrete Cosine Transform (DCT) is performed on every 8 × 8 (Y) block. The DCT is an invertible function that transforms the data from the spatial domain to the frequency domain and helps to separate the image into parts (spectral sub-bands) of differing importance with respect to the image's quality. The JPEG, MPEG-1, MPEG-2 and MPEG-7 encodings use the DCT domain to discard high frequency information that are not important to the human perception. The Forward DCT is defined in Equation 3 and it's inverse in Equation 3.3.

$$C(u,v) = \alpha(u,v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} S(x,y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{3}$$

where $C(u, v)$ is the resulting DCT coefficient at the coordinates (u, v), $\alpha(u, v)$ is defined by Equation 5, S is the two dimensional square array of size $N \times N$ and in this case $N = 8$.

$$S(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha(u,v) C(u,v) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{4}$$

where N, S and C are as described in Equation 3.

$$\alpha(u,v) = \begin{cases} 1/N & \text{for } u = 0 \text{ and } v = 0 \\ 2/N & \text{otherwise} \end{cases} \tag{5}$$

The first transform coefficient in the block is the average value of the sample so at location (0, 0) in the two dimensional 8 × 8 block the value for (u, v) is $1/N$. This value is referred to as the "DC" coefficient. All other transform coefficients are called the "AC" coefficients and have $\alpha(u, v)$ equal to $2/N$.

The lossy JPEG compression uses an 8×8 quantization matrix of step sizes (quantums), one element for each DCT coefficient, to further increase the compression ratio by discarding the high frequency coefficients. In this watermarking algorithm it is important to ensure that the coefficients used for embedding are not affected by JPEG's lossy-quantization step, therefore the embedding will occur after the lossy-quantization. Great care was taken not to affect the compression ratio. The luminance quantization matrix "Q" obtained from the (Independent Jpeg Group) IJG JPEG library is shown below.

$$\mathbf{Q} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Each DCT coefficient in the block is divided by the associated element in the quantization matrix and rounded to the nearest integer. The higher frequency coefficients which are located towards the lower part of the block are divided by higher values forcing them to become 0's. The lower frequencies (upper left) which are the perceptually significant part of the image are divided by smaller values, maintaining their accuracy. After quantization, usually more than half of the DCT coefficients are equal to zero. Therefore, it is impractical to modify any of the high frequency coefficients during watermark embedding for two main reasons. The first and most important reason is that JPEG compression will wipe out these values, destroying the watermark completely. The second reason is that it will disallow JPEG to perform an optimal compression using run-length coding of the zero coefficients.

3.4.2 Pixel encoding

The next step in the encoding process is the call to ShuffleBlocks(Y,Kimage) in which quantized luminance blocks are shuffled using a user defined key, very similar to the algorithm described in Algorithm 3.1. This step adds extra security to protect the watermark from intentional removal so that it will be very difficult for an attacker to guess or statistically show in which block which pixel of the encrypted watermark has been embedded.

Finally the encoding algorithms are given below. Algorithm 5 embeds the mark into the entire image and Algorithm 6 embeds on bit into one DCT block.

Algorithm 5 (Encode-Watermark) *Encodes W in Luminance component Y*

```
Encode-Watermark($N, Y, W, S, ET, \alpha, Target$)
for $i \leftarrow 0$ to $N - 1$
    do GetRandomIndices($C_X, C_Y, Y_i$)
       GetAssociatedSlacks($i, S_{C_X}, S_{C_Y}$)
       $\delta \leftarrow \min((S_{C_X} + S_{C_Y} + 1) \times \gamma\ ,\ \alpha)$
       $ID \leftarrow i$
       SetCoefficients($C_X, C_Y, C_1, C_2, Target, W_{(i\%W_{Dim})}, ID$)
       Encode-InBlock($ID, Y_i, ET, C_1, C_2, \delta$)
```

Algorithm 6 (Encode-InBlock) *Encodes one DCT block β with strength*

```
Encode-InBlock($ID, \beta, ET, C_1, C_2, \delta$)
if HasRelationship($\beta_{C_1}, \beta_{C_2}, \delta$)
  then return
if IsEdgeBlock($\beta, ET$)
  then AddToBucket($ID$) and return
  else repeat
              $ForceRelationShip(\beta_{C_1}, \beta_{C_2}, \delta)$
              Copy($\beta, \mu$)
              IDCT($\mu$) and DeQuantize($\mu$)
              FDCT($\mu$) and Quantize($\mu$)
              if $|\mu_{C_1}| > |\mu_{C_2}|$
                then Bit Embedding Is Confirmed
                else Copy($\mu, \beta$)
         until Bit Embedded
```

In Algorithm 5, Y is the set of *N* luminance blocks, W is the encrypted watermark, *ET* is the threshold used for edge detection, α is the maximum watermark strength and Target is a value that can be toggled between 0 and 255 for minimizing the number of changes a single run of embedding creates.

3.5 Watermark decoding algorithm

The watermark decoding stage is very similar to the procedures described in the encoding except that now the original image and the original watermark are not available. The watermark bits are constructed bit by bit using the watermarked image. In order to extract the watermark successfully, several requirements must be met. One of the requirements is that the author's key file must be present. A key file is an encrypted binary file that has been written at the time of watermark encoding. This file contains the secret keys used by the author, the target value used for embedding, several templates such as image patches that facilitates in synchronization of the image in the spatial domain and several indices of rejected DCT blocks that can be included for a more robust decoding.

Furthermore, if the dimensions of the image have been altered, the image must be rescaled to the exact same dimensions as when it was marked. The algorithm will also need to be informed by a human user if a potential cropping has occurred, which in that case the synchronization templates provided in the key file will be matched against the cropped image. If the matching has been successful, the remaining (uncropped) image is pasted against a black background in the exact same position that the templates suggest.

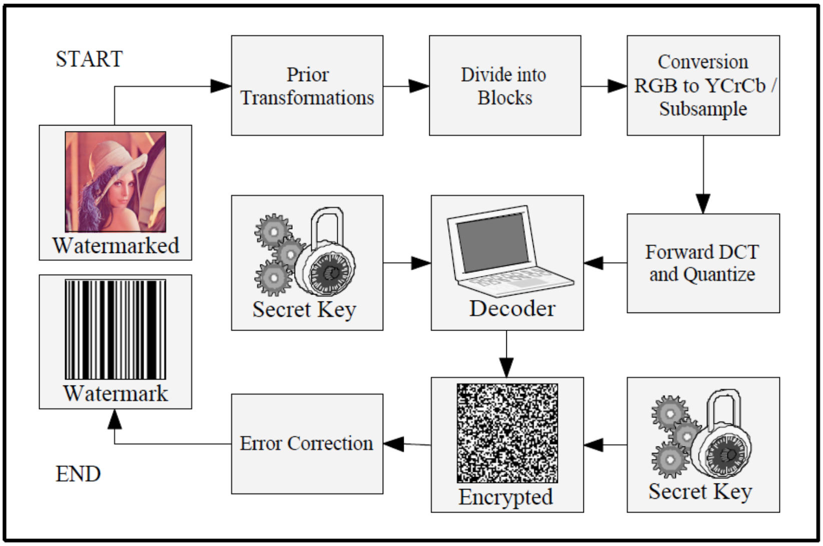


Fig. 8. Proposed watermark decoding diagram

The watermark is constructed pixel by pixel according to the relationship between specific DCT coefficients in a block. Before the relationships can be tested for, the image is segmented into blocks of 16 × 16 for a colour space conversion from RGB to YCrCb (YUV). Next, all blocks from the luminance (Y) component are extracted and a Forward DCT is performed on each 8 × 8 non-overlapping block using Equation 3.

The DCT blocks are shuffled using a secret key K_{image} obtained from the key file. Now the DCT blocks are ready for the watermarking extraction routine outlined in Algorithm 7.

The Target value is read from the key file and can either be 0 or 255, the value for AntiTarget is always calculated to be the opposite of the Target. The decoding procedure in Algorithm 7 traverses each block β in the luminance component Y, choosing the same two random coefficients C_1 and C_2 as used in the encoding procedure described in Algorithm 6. The structure Bucket read from is used to check if the current block has been previously discarded by the encoding algorithm, in which case the sign of the value in $Bucket_j$ will decide if the pixel W_i is equal to 255 or 0. After all blocks have been processed, the watermark W has to be decrypted to reveal the original Bar-code like image created by the author.

Algorithm 7 (Decode) *Extracts the embedded watermark from image I*

```
$\text{DECODE}(Y, K_{image}, K_{mark}, W, Target, Bucket)$
$Y \leftarrow \text{SHUFFLEBLOCKS}(Y, K_{image})$
$AntiTarget \leftarrow |Target - 255|$
$j \leftarrow 0$
for $i \leftarrow 0$ to $N - 1$
    do if $Bucket$ not empty and $|Bucket_j| = i$
         then if $Bucket_j \geq 0$
                then $W_i \leftarrow Target$
                else $W_i \leftarrow AntiTarget$
              $j \leftarrow j + 1$
         else $\beta \leftarrow Y_i$
              $\text{GETRANDOMINDICES}(C_1, C_2, \beta)$
              if $|\beta_{C_1}| \geq |\beta_{C_2}|$
                then $W_i \leftarrow Target$
                else $W_i \leftarrow AntiTarget$
$W \leftarrow \text{DECRYPTMARK}(W, K_{mark}, N)$
```

Watermarked images are usually posted on web sites (internet) or distributed to individual customers sometime after the encoding process. Between the time of encoding and the time of decoding the watermarked image may undergo many possible manipulations. Some of these attacks are intentional such as cropping and others are unintentional like the collection of channel noise. In addition, the lossy quantization step during JPEG compression and decompression is a major source for error in the decoding process. Therefore, the decoded watermark may not always appear 100% identical to the original embedded mark. One method of determining the similarity of two given signals is known as Bit Error Rate (BER). The BER is the ratio of the total bit error to the total number of bits embedded and is given by:

$$BER = \left(\sum_{I=0}^{N-1} B_I^* \oplus B_I \right) \; / \; N \tag{6}$$

where B_I^* and B_I are the bits at the i^{th} position of the decoded watermark and the original watermark respectively. The symbol ⊕ is the binary XOR operation and N is the total number of bits in B.

For a perfect decoding step the BER would equal to 0, indicating no lost bits. Since no algorithm can claim to be perfect it is important for every watermarking system to expect such bit errors and facilitate a method of Error Correction.

4. Experiments and results

This Section presents the results found using the methodology described in Section 3. One Thousand color images are used to test and obtain results from the proposed system "Digital Image Copyright Protector" (DIGI-COP). In Section 4.1 we explore the fidelity of the marked images, in Section 4.2 Error Rates are analyzed and in the subsequent sections various attacks are explored as listed below.

Compression

4.1 Fidelity

One of the major requirements of digital watermarking systems is the ability to hide the mark within the cover work such that it becomes perceptually invisible to a human observer. The proposed image watermarking method (DIGI-COP) presented in this Chapter makes use of local characteristics of the image to achieve higher invisibility rates than it's base algorithm (BWM). Consider an image I and it's watermarked version IW, then the standard deviation between I and IW is defined by the MSE (Mean Square Error):

$$\text{MSE}\ (I, I^W) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I(i,j) - I^W(i,j)||^2 \tag{7}$$

Peak Signal-to-Noise Ratio (PSNR) is often used for global evaluation of the quality of reconstruction in image compression techniques. It is expressed in terms of the logarithmic decibel (dB) scale and defined as:

$$\text{PSNR}\ (I, I^W) = 10 \log_{10} \left(\frac{255^2}{MSE} \right) = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \text{dB} \tag{8}$$

Thus PSNR is the ratio between the maximum possible power of a signal and power of corrupting noise that affects the fidelity of its representation. The original image and the watermarked image are denoted by I and by I^W respectively, and M and N are the dimensions of the images. A lower value of MSE means less distortion, and therefore the higher the PSNR value is, the better or closer the watermarked image is to the original. Generally a PSNR larger than 32 dB means invisible visual degradation and a human observer perceives both images as indistinguishable [Wilson & Martinez 97].

For the fidelity test 1000 RGB images were marked with various watermark strengths, α ∈ {10, 20, 30, 40}, using both the base method (BWM) and the proposed watermarking method (DIGI-COP). The PSNR values obtained are graphically produced in Figure 9 and the average PSNR values together with the standard deviations σ are presented in Table 1.

α (strength)		Digi-Cop			BWM	
		PSNR (dB)	σ		PSNR (dB)	σ
10		41.76	2.37		39.97	2.75
20		41.24	2.07		38.71	2.22
30		40.72	1.92		37.39	1.75
40		40.35	1.83		36.12	1.38

Table 1. Average PSNR of 1000 watermarked images DIGI-COP BWM

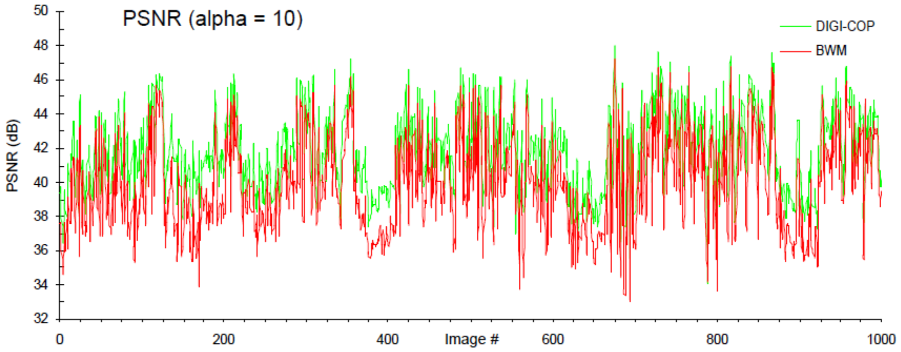


Fig. 9. PSNR values of 1000 watermarked images with α = 10

It is clear from the results that DIGI-COP achieves higher PSNR values than the BWM even when the watermark strength (α) is increased. Next, the PSNR results for classical images marked with α = 40 are illustrated in Table 2 and Figure 10. The PSNR values are calculated in the YCrCb domain and reflects the similarities in the luminance components.

Image	DIGI-COP PSNR (dB)	BWM PSNR (dB)
Boys	42.56	37.67
Boat	39.11	35.73
Peppers	40.31	36.67
Baboon	36.78	34.60
Plane	40.48	36.05
Tiffany	40.99	37.15
Drop	42.12	37.35
Lena	41.30	37.09

Table 2. PSNR values of classical images with α = 40

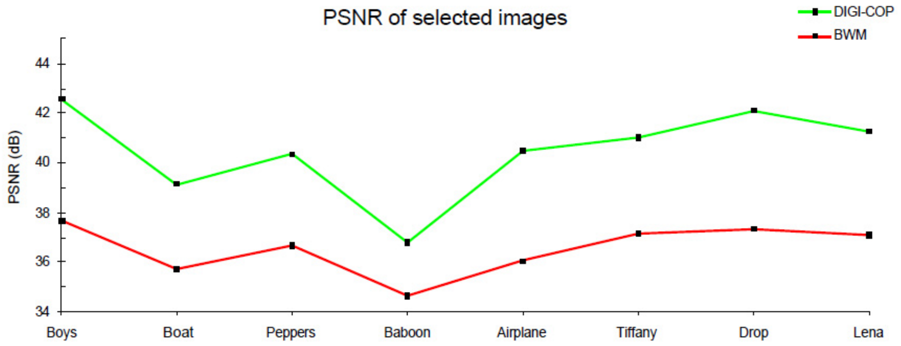


Fig. 10. PSNR values of selected watermarked images with α = 40

In literature [Meerwald 2001, Jellinke 2000, Mohanty 1999, Guo 2003], it has been previously reported that the base watermarking method (BWM) occasionally produces small spatial defects around edges and an undesired blocking effect in smooth regions of the image. Figure 11 demonstrates that DIGI-COP protects the image from such unwanted artifacts.

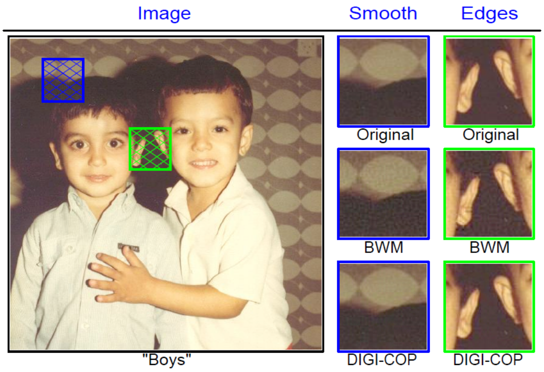


Fig. 11. Smooth and edge regions of the image “Boys” after watermarking

In Figure 11 the shaded blue and green regions in the image represent the selected locations of smooth and edge regions. The enlarged views of the smooth and edge region are presented towards the right of the image and compared to the original unwatermarked areas.

4.2 Error rates

There are two types of errors that can occur during the watermark extraction stage. The first error is a false-negative (FN), in which a watermark decoder fails to identify a watermarked image as a legit or “marked” copy. The decoder’s ability of correctly extracting the watermark *W* from a marked image *I* is calculated in terms of a BER (Bit Error Rate) value described in Equation 6 on Page 17. The FN test was performed on 1000 watermarked images with α of 20 using the decoder of BWM and DIGI-COP. The BER values are illustrated in Figure 12.

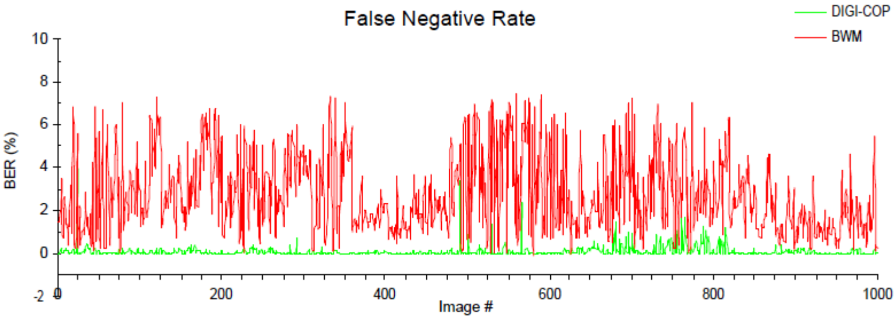


Fig. 12. False negative rate

The graph indicates that DIGI-COP achieves a lower BER in the decoding process and extracts the watermark with higher precision.

The second type of a decoding error is a false-positive, where the watermark decoder incorrectly detects the presence of a watermark in an image. There are two types of false-positives, the first type (FP-I) occurs when a watermark decoder extracts a watermark in an image I that has not been marked previously. The second type (FP-II) of false positives is when a decoder extracts watermark W from an image I that has been marked with a different mark W . Both types of false-positives are undesired in a reliable system. The results for both types of false-positives tests are presented in Figure 4.9.

In Figure 13, the FP-I results suggests that the BWM has a higher accuracy in determining unwatermarked or incorrectly marked images. On the other hand, FP-II results show clearly that both watermark decoders can correctly extract the watermark W from an image marked with W placed at location 500. Further, BWM and DIGI-COP show high BER values when attempting to extract watermark W from images marked with different watermarks such as W*.

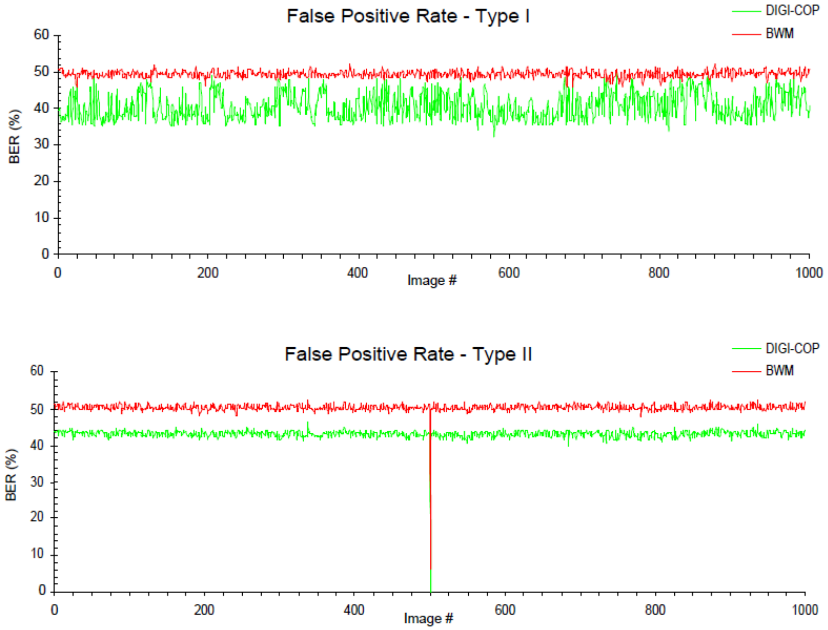


Fig. 13. False positive rates types I and II

4.3 Image processing attacks

4.3.1 JPEG compression

JPEG compression is one of the main reasons for the success of the internet and must be taken into account when designing an image watermarking system. JPEG compression will attempt to remove the perceptual unimportant elements from an image and may render the imperceptible watermark undetectable. Image compressions are considered one of the strongest enemies of digital watermarking techniques today.

The JPEG compression resistance test presented in this thesis is performed on 1000 color images over 11 different JPEG quality factors ranging from 100% to 0%. The BER value is calculated for each individual image and averaged over the entire image set for that particular quality as illustrated in Table 3.

	DIGI-COP		BWM	
JPEG Quality	BER (%)	σ	BER (%)	σ
100	0.29	1.61	3.74	4.25
90	2.53	1.78	10.38	4.78
80	3.81	2.14	10.86	4.97
70	6.91	3.49	17.05	6.71
60	9.30	4.82	17.09	6.64
50	11.36	5.57	14.32	5.88
40	14.25	6.28	11.70	4.57
30	19.88	7.37	18.63	5.50
20	26.85	7.69	29.29	5.88
10	33.14	6.66	38.41	3.90
0	37.84	4.41	44.44	0.83

Table 3. Effects of JPEG compression on BER

Both the BWM and DIGI-COP decoders show high resilience against this attack up to a JPEG quality factor of 40. In addition, an Error Correction (EC) technique is used to achieve a much lower BER value than the original proposed method in it's essence. The illustration in Figure 14 is the direct result of a total of 50,000 individual decoding operations and shows the advantage of the Error Correction technique.

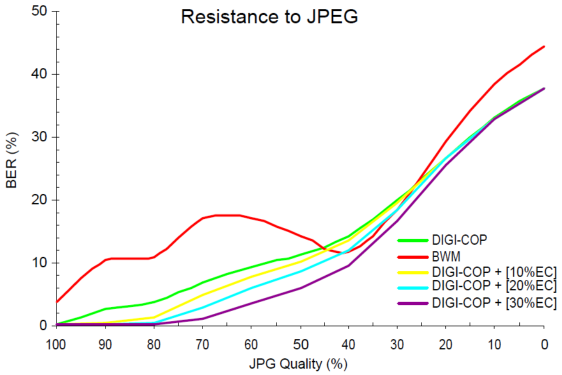


Fig. 14. JPG Compression with Error Correction (EC)

The Error Correction was set to 10%, 20% and 30% to see the changing effects of the BER value over the entire image set. In Figure 4.10 the BER value is plotted against the various JPEG qualities, and it can be seen that as images are compressed at higher rates (lower qualities), the Bit Error Rate also increases. However, DIGICOP's Error Correction feature significantly reduced the decoding BER.

5. Conclusion and future research

A new adaptive and invisible digital watermarking system ("DIGICOP") in the DCT-Block domain is discussed as a method for protecting copyright for digital images. The performances of DIGI-COP and the classical DCT-Block technique are compared.

Extensive results show that DIGI-COP is preferred over the classical method in terms of it's fidelity and robustness. The method can embed large quantities of data into the cover image without any noticeable changes. The new embedding technique facilitates embedding the right amount of watermark at the most advantageous locations in the image without causing visual artifacts. The improvement is achieved by exploiting the characteristics of the cover image in the DCT-Block domain, as well as the sensitivity of the HVS to small changes in smooth regions and edges of the image. The fidelity test of DIGI-COP achieved on average 41 dB over one thousand encodings where the classical method achieves on average 38 dB on the same set of images.

In addition, both false-negative and false-positive rates of the two algorithms were compared over a set of thousand images. DIGI-COP's false-negative rates show to be more reliable than the classical algorithm. It correctly extracts 99.9% of the data with a standard deviation of 0.26 as compared to the classical method with 97.2% decoding rate and a deviation of 1.8. False-positive rates were compared and both algorithms show good performance. Although both algorithms can identify a legit watermark from a set of thousand randomly marked images and deny all unmarked or incorrectly marked images, the classical algorithm shows a slight better performance. This is due to DIGI-COP's feature of storing discarded bits in the key file and assuming it's presence in unmarked or incorrectly marked images.

6. References

Anderson, R. J. & Petitcolas, F. On The Limits Of Steganography. (1998). IEEE J. Select. Areas Communication. (Special Issue on Copyright and Privacy Protection), 16, 474–481, May 1998.

Baaziz, N. (2005). Adaptive Watermarking Schemes Based On A Redundant Contourlet Transform. IEEE International Conference on Image Processing ICIP 2005, 1, , 11–14 September 2005, pp 221–224.

Bartolini, F.; Barni, M.; Podilchuk, C. I. & Delp, E. J. Watermark Embedding: Hiding A Signal Within A Cover Image. IEEE Communications Magazine, 39, August 2001. pp 102–108,

Berghel. (1997). Watermarking Cyberspace. Communications of the ACM, 40, 1997, pp 19–24.

Cappellini, V.; Barni, M.; Bartolini F.; & Piva, A. (1998). A DCT Domain System For Robust Image Watermarking, Signal Processing (Special Issue on Watermarking), 66, May 1998. pp 357–372

Cappellini, V.; Piva, A.; Barni, M.; Bartolini F. & Rigacci, F. (1998). A M.A.P. Identification Criterion For DCT-Based Watermarking, Proceedings Europe: Signal Processing Conference (EUSIPCOŠ98), September 1998.

Cheng, L.L.; Ng, K.S.; Cheng L. M. & Wong, M.K. (1999). Adaptive Watermarking By Using Pixel Position Shifting Technique IEEE Transactions on Consumer Electronics, 45, November 1999. pp 1057–1064.

Cox, I. J. Digital Watermarking and Steganography. Morgan Kaufmann, 2008.

Cox and Q. Li. Using Perceptual Models To Improve Fidelity And Provide Invariance To Value-Metric Scaling For Quantization Index Modulation Watermarking. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2005' (ICASSP), March 18-23 2005.

De Rosa M. Barni, F. Bartolini and A. Piva. Capacity of Full Frame DCT Image Watermarks. In IEEE Transactions on Image Processing, vol. 9, August 2000, pp 1450–1455.

Guo, H. Digital Image Watermarking for Ownership verifciation. PhD thesis, University of Ottawa, 2003.

Hartung, F. & M. Kutter, Multimedia Watermarking Techniques. (1999). Proceedings of the 1999 IEEE, 87, July 1999, pp 1079–1107.

Huang, C.-H. and Wu, J.-L. Attacking visible watermarking schemes. In IEEE Transactions on Multimedia, volume 6, pages 16–30, February 2004.

Jellinek, B. Invisible watermarking of digital images for copyright protection. Master's Thesis, University of Salzburg, January 2000.

Koch, E. & Zhao, J. (1995) Towards Robust And Hidden Image Copyright Labeling. Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image Processing, pages 452–455. Halkidiki, Greece, June 1995.

Koch E.; Zhao, J. & Luo, C. (1998). In business today and tomorrow Communications of the ACM, 41, July 1998. pp 66–72.

Leighton, F. T.; Cox, I. J.; Kilian, J. & Shamoon, T. (1997). Secure Spread Spectrum Watermarking For Multimedia. IEEE Transactions Image Processing, v6, December 1997. pp 1673–1687.

Lie, W. & Chang, L. (1997). Robust And High-Quality Time-Domain Audio Watermarking Based On Low-Frequency Amplitude Modification, IEEE Transactions on Multimedia, vol 8, February 2006. pp 46 – 59.

Liu, Z. & Inoue, A. (2003). Audio Watermarking Techniques Using Sinusoidal Patterns Based On Pseudorandom Sequences. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, August 2003, pp 801–812.

Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transactions on Modeling and Computer Simulation, vol 8, January 1998, pp 3–30.

Matsumoto, M. & Nishimura, T. (2002). A nonempirical test on the weight of pseudorandom number generators. Monte Carlo and Quasi-Monte Carlo methods, 2002, pp 381–395.

Meerwald, P. Digital image watermarking in the wavelet transform domain. Master's Thesis, University of Salzburg, January 2001.

Miller, M. L.; Cox, I. J. & J. A. Bloom. (2002). Digital Watermarking. Academic Press, 2002.

Mohanty, S. P. Watermarking of digital images. Master's Thesis, University of Salzburg, January 1999.

Podilchuk, C. I.; & Zeng, W. (1998) Image-Adaptive Watermarking Using Visual Models, IEEE Journal on Selected Areas in Communications, 16, May 1998. pp 525–539.

Puate, J. and Jordan, F. "Using fractal compression scheme to embed a digital signature into image," in Proc. SPIE Photonics East Symp., Boston, MA, Nov. 18–22, 1996. Available http://iswww.epfl.ch/~jordan/watermarking.html.

Ren-Hou, L.; Lian-Shan, L. & Qi, G. (2005). A Robust Video Watermarking Scheme Based On DCT. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 8, 5176-5180, August 2005. pp 18–21.

Shahraeini, S. and Yaghoobi, M. A Robust Digital Image Watermarking Approach against JPEG Compression Attack Based on Hybrid Fractal-Wavelet. Advanced Materials Research, 403-408, 2012.

Wilson, D. R. and Martinez, T. R. Improved heterogeneous distance functions. In Journal of Artificial Intelligence Research., 1997.

Wolfgang, R. B.; Podilchuk, C. I. & Delp, E. J. (1999). Perceptual Watermarks for Digital Images and Video, vol 87:7, July 1999, pp 1108–1126.

2D Watermarking: Non Conventional Approaches

Hassen Seddik
*ESSTT Higher Sciences and Technical School of Tunisia, Tunis
Tunisia*

1. Introduction

The growth of new image technologies and data exchanges, in addition to the ever-increasing use of multimedia content through online services, has created the need for new techniques capable of assuring copyright protection and data owner identification. Watermarking is now considered as an efficient means for resolving these problems. Watermark embedding techniques depend on the representation domain of the image (spatial, frequency, and multiresolution). Every domain has its specific advantages and limitations. Moreover, each technique in a chosen domain is found to be robust to specific sets of attack types. In addition all the techniques developed in theses domain are widely known and can be defeated to break the used algorithm and target the embedded watermark to destroy it or to put it out. So we need to develop another robust domain that defeats these limitations and respects all the watermarking criterions (capacity, invisibility and robustness). In this chapter, new watermarking methods are presented using new domains for the image representation and watermark embedding. These domains are based on different mathematical transformations of the image matrix. The applied transformations that process the image coefficients must dispose of three indispensable proprieties: no data loss, reversibility and preservation. Theses domains are found to be robust against a wide range of synchronous and asynchronous STIRMARK attacks. The robustness of the techniques in preserving and extracting the embedded watermark is proved after various attacks types. It is also improved when compared with other methods in use. In addition, the proposed methods are blind and the use of the host image is not needed in the watermark detection process.

2. A Blind image watermarking method based on the Hessenberg transformation

2.1 Introduction

The advent of the Internet brought about a sudden increase in the use of digital media in electronic commerce and various other services. Because of the ease of reproducing or falsifying digital media, it's very easy for the manufacturer to incur financial losses. To counter such problems, watermarking methods have gained significantly in popularity as they protect the ownership rights and simplify proprietor identification. To that end, various techniques have been developed, each ultimately aiming at pinpointing equilibrium between imperceptibility and robustness of the watermark against wide attacks kinds, depending on the image domain representation. Many researchers have focused their efforts on security and robustness, as well as on the watermarking capacity that are essential to obtain an irremovable and inappreciable watermark with regards to the image processing domain. Each domain presents its robustness face to particular kind of attacks and its limitation to others, but no one is able to resist to a wide set of synchronous and asynchronous attacks gathering the robustness of the different domains. In addition, many of these techniques require the presence of the original image to read the inserted watermark. To satisfy these watermarking obligations, the necessity of either finding a blind watermarking method robust to a large set of attacks kinds, or a new image domain representation more robust than the known domains, is more and more urgent. In this chapter, these two constraints are satisfied. In fact we propose a new watermarking method using a new domain of the image representation based on the mathematical Hessenberg transformation. Using this method, both robustness and security criteria are fully met, and the embedded mark is fully invisible and present in all cases after different signal processing distortions. The Hessenberg Image Transformation brings a new representation domain with remarkable watermarking possibilities exceeding the limits of the domains cited above face to different attacks. The image is represented by the triangular part of the Hessenberg matrix used in the watermarking process. A study of this matrix is conducted with a view to identify the sectors or zones that can hold the watermark according to the three criteria mentioned above. Once the watermark is embedded in a chosen sector, inverse transformation is applied to return to the spatial representation holding the embedded mark.

If we explore the field of watermarking we find that the most commonly used watermarking technique domains are Spatial, DFT, DCT and Wavelet domains. There are many ways the spatial domain can be used in watermark embedding, for example: substituting the least significant bit in a chosen image pixel, coding by texture blocks, changing paired pixels, etc. However, various approaches that defeat the limitations of the spatial domains have been developed and can be used in the frequency domain. These include, the spread spectrum, content-based approaches, JPEG-based approaches, etc. For this purpose, the transformations used are the Discrete Cosine Transform (DCT), the Discrete Fourier Transform (DFT), and the Discrete Wavelet Transform (DWT). The DCT is the main transform used in JPEG image compression. It eliminates DFT high frequency components induced by the sharp discontinuities at the boundary between the consecutive periods in the time. To represent sharp value changes, it needs non-zero high frequency DFT components. For purpose of compression, all high frequency DFT components are deleted, causing a distortion of the original image. To overcome this difficulty, the DCT concatenates a period with the mirrored image of its an-adjacent period. The common DCT form is derived from a class of discrete Chebyshev polynomials. Whereas the advantage of DFT is its ability to describe the frequency responses of a signal even as allowing the possibility of extracting different signal characteristics from this frequency domain. While it's notable disadvantage is the absence of any information concerning the occurrence time of these frequency components. However, a particular frequency response that occurs in a certain interval can be detected with the Short Time Fourier Transform which splits the signal into fixed-length intervals where the Fourier analysis is applied. But if the cycle of the frequency response exceeds the length of the fixed interval, it becomes impossible to describe it. To overcome this problem, the discrete wavelet transform is necessary by the use of base functions and windowing operations. In the case of data compression, the implementation of the DWT is similar to that of sub-band coding, where at each stage, a coarse overall shape and the details of the data obtained from the previous stage are derived. When encoding is performed in the DWT domain, two processes are applied: decomposition by separating data into frequency bands using high-pass and low-pass filtering, and down-sampling, which consists in removing unneeded data for future reconstruction. Decoding for its part, involves up sampling in order to adjust dimensionality and recombine data from different bands. Many methods are developed based on DWT schemes. One of the interesting is the use of the SA-DWT (Shape Adaptive Discrete Wavelets Transform) for developing a blind watermarking algorithm and HVS characteristics to achieve the best trade-off between invisibility and robustness. This method is found to be robust against some attacks such as lossy compression (JPEG, JPEG2000 and MPEG-4), scaling and filtering.

As explained, different domains are used for watermark embedding with various developed techniques. The spatial domain is found to be more robust against different kinds of asynchronous attacks such as rotation, rescaling, affine transforms etc. and less than others, whereas the frequency domains is well known for its robustness against synchronous attacks such as lossy compression (using DCT transform), filtering, noise adding or a specific kind of geometrical transforms such as scaling, rotation and translation (using Fourier-Melin transform or the multi-resolution domain: DWT). These limitations created the necessity to develop specific techniques in each domain to cover the robustness loss face to some attacks. In the following, we propose a new image watermarking domain: called the Hessenberg domain, which is able to counter a large set of different attacks kinds with high robustness, by the use of a substituting technique. We show that this domain can cover the limitations of the spatial, frequency and DWT domains. In addition, the robustness of this watermarking technique is improved when compared with many recent techniques in use.

2.2 Proposed Hessenberg watermarking technique

2.2.1 Mathematical Hessenberg transform overview

Any image can be transformed by an orthogonal transformation; we will illustrate the relationship between our proposed method and other transformations. In fact an orthogonal transform applies a rotation to the representation space. The data of the image passes from a space where they are highly correlated into a space where this correlation is minimized. The less correlated coefficients of the transformed image gather the image characteristics. If this information is classified in way of significance, the data can be compressed by eliminating the data which valuesare null or near to be null and by quantization of the selected coefficients to be transmitted. Let's note $T_{k,l}(i,j)$ a set of function of orthogonal discrete bases and $T^{*}_{k,l}(i,j)$ its complex conjugate. An image I of size M×N has a transformed form I_T given:

$$I_T(k,l) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j) T_{k,l}(i,j) \text{ Where } 0 \le k \le M-1 \text{ and } 0 \le l \le N-1 \quad (1)$$

$$I(i,j) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I_T(k,l)T_{k,l}^{*}(i,j) \text{ Where } 0 \le i \le M-1 \text{ and } 0 \le j \le N-1 \tag{2}$$

From these transformations an important characteristic must be illustrated which is the energy distribution. In Figure 1, we show an example of the energy distribution in the spatial and frequency DCT domains. We will illustrate the importance of the energy distribution of the image through the Hessenberg transform and its affect in the watermarking process.

The proposed method uses the mathematical Hessenberg transformation. The used algorithm is based on the LAPACK routines for computing the Hessenberg form of the processed matrix. Perhaps the most successful numerical algorithm for computing the complete eigensystem of a general square matrix A is the implicitly shifted QR algorithm. One of the keys to the success of this method is its relationship to the Schur decomposition:

$$\mathrm{A} = \mathrm{U\,T\,U}^{*} \tag{3}$$

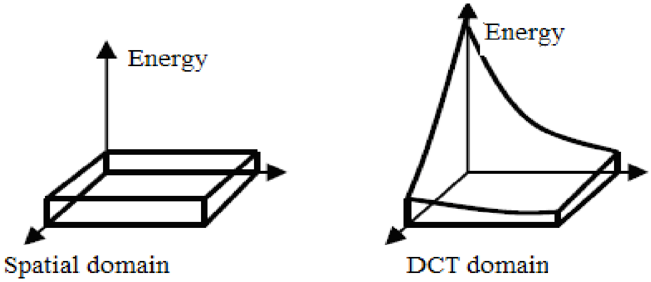


Fig. 1. Energy distribution in the spatial and transformed domains.

This well-known decomposition asserts that every square matrix A is unitarily similar to an upper triangular matrix T. The QR algorithm produces a sequence of unitary similarity transformations that iteratively reduce A to upper triangular form. In other words, it computes the Schur decomposition. A practical implementation of the QR algorithm begins with an initial unitary similarity transformation of A to the condensed form $\mathrm{Q}^{*}\,\mathrm{A\,Q} = \mathrm{H}$ where H is upper Hessenberg (almost upper triangular') and Q is unitary. Then the following iterations to set the lower triangular to zero is performed as follows:

$$\mathrm{Q}^{*}\,\mathrm{A\,Q} = \mathrm{H} \text{ is first factorized} \tag{4}$$

For j=1, 2, 3… until convergence. Select a shift μ then the QR factorization is as follows:

$$\mathrm{VR} = \mathrm{H} - \mu I \tag{5}$$

Then the matrices Q and H are assigned the following values:

$$\begin{cases} \mathrm{H} = \mathrm{V}^{*}\,\mathrm{H\,V} \\ \mathrm{Q} = \mathrm{QV} \end{cases} \tag{6}$$

In this scheme, V is unitary and R is upper triangular (i.e., the QR factorization of $H-\mu I$). It is easy to see that H is unitarily similar to A throughout the course of this iteration. The iteration is continued until the sub-diagonal elements of H converge to zero, i.e., until the Schur decomposition has been (approximately) obtained. To summarize these mathematical steps, we can say that in finding the eigenvalues of a matrix using the QR algorithm, the matrix is first transformed by a unitary similarity transformation to upper Hessenberg form. The QR algorithm then iteratively generates a sequence of upper Hessenberg matrices by unitary similarity transformations. For general real matrices, Q^* becomes Q^T and the routine reduces the real general matrix A to the upper Hessenberg form H by this orthogonal similarity transformation:

$$A = Q\,H\,Q^T \text{, where Q is a unitary matrix so that } Q^TQ = I\ (size(A)) \tag{7}$$

I: denotes the identity matrix and Q^T represents the unitary matrix transpose. The general matrix structure produced by the routine is presented by the following equations as $H(i,j)=0$ unless $i=j$ or $i=j\text{-}1$ and:

$$H = \begin{bmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ 0 & a_{22} & & & a_{2n} \\ . & & & . & . \\ . & . & & . & . \\ 0 & . & & 0 & a_{nn} \end{bmatrix} \tag{8}$$

The form of H will be tri-diagonal if the processed matrix is symmetric or Hermetian. In general, the mathematical application of this transformation serves two purposes: the first is to obtain a matrix having the same eigenvalues as the original one, but which requires less computation to reveal them; conversely, the second is the packed storage. In fact, a triangular matrix may be stored more compactly, if the relevant triangle is packed by columns in a one-dimensional array.

The a_{ij} elements of H are stored in the one-dimensional array W as:

$$a_{ij} = W\left(i + j\left(j\text{-}1\right)/2\right) \quad \textit{for} \quad i \le j \tag{9}$$

In this work, the produced triangular Hessenberg matrix is exploited as a new representation domain of the image where the watermark is embedded. By analyzing this matrix, we discovered the existence of an exploitable zone. Embedding a watermark in this zone produces no effect on the original matrix values after applying the inverse Hessenberg transform. This zone provides imperceptibility and robustness to the watermark. To take advantage of this characteristic, we apply this transformation to the image matrix to obtain the Hessenberg triangular representation of the image. This matrix is processed to find in which zones we can embed a watermark without any change produced on the original image. This means that we can change and increase the values of this zone in the Hessenberg matrix, without any effect being produced on the original image after the inverse Hessenberg transform is applied. The following section presents the study applied to this matrix to identify this zone called insensitive zone.

2.2.2 Hessenberg matrix study

Successive iterations as shown in the following figure decreases the original matrix values from upper region to the lower one.

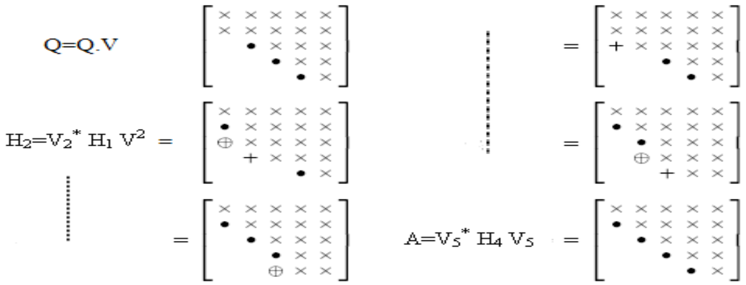


Fig. 2. Successive iterations of the H matrix.

These iterations tends to set the values of the lower triangular part of the Hessenberg matrix that we will call H matrix to zero. As a consequence, the values of the upper part of the Hessenberg matrix follow the sense of this decrease. The values of the matrix H decrease from the upper left sub-diagonal to the lower right sub-diagonal as shown in Figure 3, 4 and 5, containing a pick in the upper left part which values are in the range of 2 to 3.10^4 as shown in figure 6. The first figures 3, 4 and 5 are illustrated without this pick.

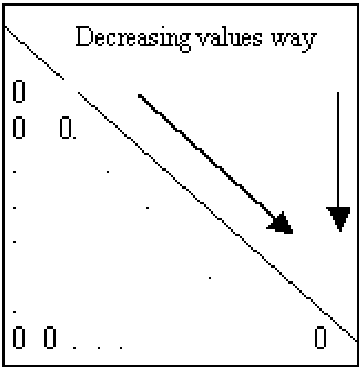


Fig. 3. Direction of decreasing matrix values.

The unidirectional value decrease is an important characteristic of this transformation. In fact, it is helpful in the matrix zones study. It allows the upper triangular part of the matrix to be divided into different blocks, in the way of the decreasing values. In analogy with the image matrix, the energy of the image is concentrated in the mid-high and high part of the transformed Hessenberg matrix. The mid-low and lower part of this transformed matrix contains a low energy of the image while its values are the weaker in the matrix. While the Hessenberg transformation is an orthogonal transformation similarly to DCT, we will exploit the previous detailed characteristics to determine different Hessenberg matrix bands that can characterize the Hessenberg transform with respect to their effect on the image quality if a watermark is embedded. In this study the processed blocks are rectangles of 5 pixels high and 20 pixels wide. They are respectively examined with an overlap of 10 pixels from left to right and 2 pixels from top to bottom. In order to determine the bands or zones that produces the same effect on the image if they are processed, the study consists of substituting each selected block B by a supposed small watermark W and then applies the inverse Hessenberg transform in order to come back to the spatial representation and view the effects of each block change on the image distortion. After sliding over the entire matrix and testing all the blocks, we set the limits of the insensitive zone. This zone represents the matrix sectors for which values change or increase does not affect the original image. It is exploited for watermarking purpose. This study revealed the existence of other three zones detailed in the next section.

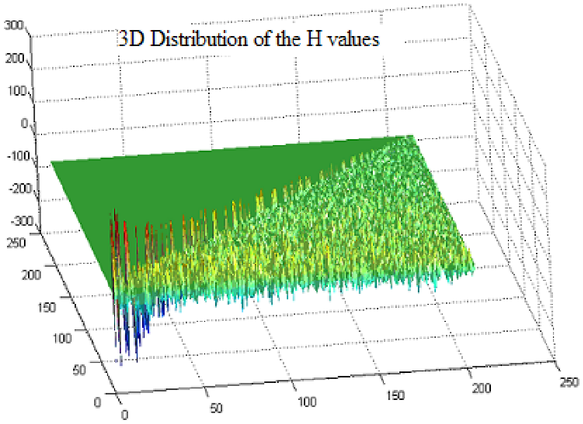


Fig. 4. A 3D distribution of the H matrix values

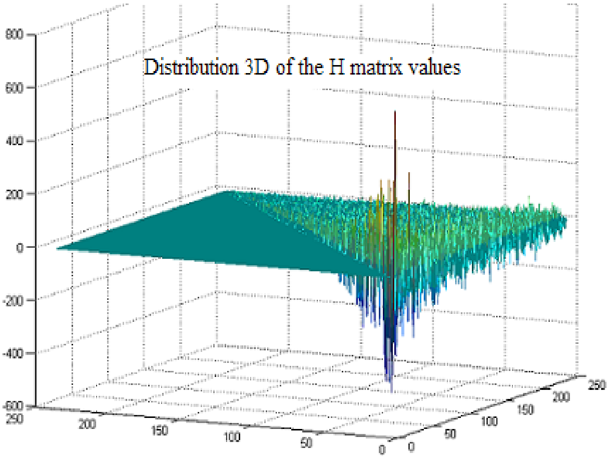


Fig. 5. Representation of the decreasing H values.

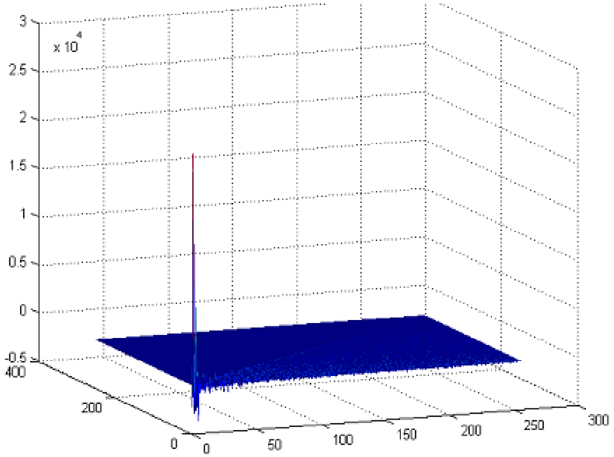


Fig. 6. Representation of the pick in the H matrix.

2.2.3 The watermarking process

The watermarking process consists in substituting one or multiple blocks in the Hessenberg matrix with a watermark. Once these blocks are substituted, the inverse transform is applied to the watermarked matrix in order to return back to the original image hiding the inserted watermark. To simplify the watermarking process, the watermark consists in applying a non-random permutation function on the chosen block values. The same function turns the obtained values to the nearest integer. The embedded watermark is then multiplied by a gain factor. This factor is chosen with respect to the limits of the image quality preserve. The substitution procedure is shown in Figure 6 and in equations (10)-(14).

Let B be the designed block, H the Hessenberg matrix, n the number of blocks that partitions the matrix and B'$_{k}$ the watermark block:

$$H = \{\ B_i\}\ as\, i \in [1,n] \tag{10}$$

H is composed by a set of n associated blocks:

$$\mathrm{H} = \{\mathrm{B}_1, \mathrm{B}_2, \mathrm{B}_3, \ldots, \mathrm{B}_k, \ldots, \mathrm{B}_n\} \tag{11}$$

$$W = A(B_k) \tag{12}$$

A is a function used to apply a non-random mixture on the block values and turning them to the nearest integer value, W is the obtained watermark having the same size as B$_{k}$.

$$B_k^{'} = K\ w \tag{13}$$

K is a gain factor used to increase the watermark values in order to add more resistance against attacks. The transformed Hessenberg matrix $H^{'}$ that holds the substituted watermark becomes:

$$H^{'} = \{B_1, B_2, B_3, \ldots, B^{'}_k, \ldots, B_n\} \tag{14}$$

To preserve the image quality and to guarantee that the embedded watermark is kept imperceptible, it is important to choose "very well" the matrix zone where blocks are substituted. Indeed from this zones study, a partition of the Hessenberg matrix in four zones is carried out with respect to the image quality change due to the block substitution, as shown in Figures 7 and 8.

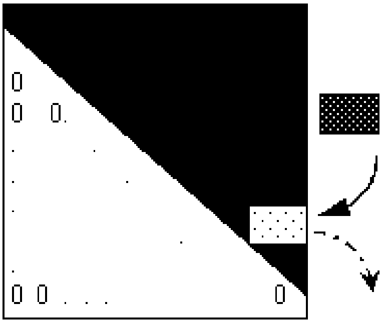


Fig. 7. a) The matrix blocks substitution.

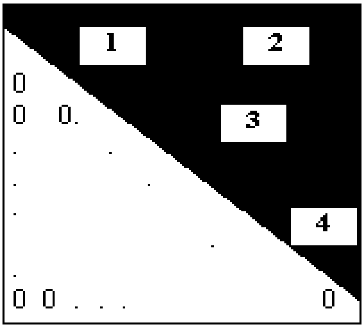


Fig. 7. b) The matrix partition.

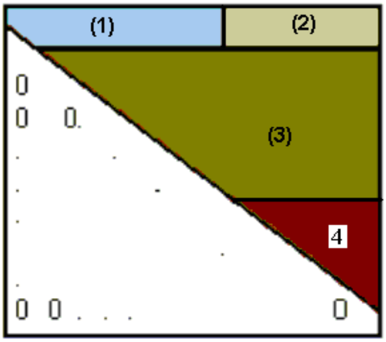


Fig. 8. Matrix delimited zones.

The Hessenberg transform is an entirely reversible transform that brings the image from the spatial representation to the H matrix and come back without any information loss or change. If this transform is applied on the image several times the same matrices are generated. This data preservation allows us to apply it on the image and use its transformed matrix for watermarking purposes. The Hessenberg matrix is divided in four zones. Each zone produces a specific distortion effect on the image if it is modified. The first zone containing the highest values in the transformed matrix and then contains the main part of the image energy. This zone is represented by the first twenty lines from the top and 150 columns from the left. Substituting blocks in this region of the Hessenberg matrix, affects considerably the image quality by adding to it a distortion look as presented in section 4. The second zone is situated between the first twenty lines from the top and a width between columns 150 and 256; this is a sensitive zone for image watermarking because any change in its values damages the original image considerably by adding to it a blurred effect. The intensity of the blur depends on the gain factor strength used. The first and second zones at the top and the fourth zone at the bottom delimit the third zone shown in Figure 8. In this zone, any watermark embedding followed by an inverse Hessenberg transform, damages the image by adding a noise effect to it. In fact the forms and shapes in the obtained image do no change, but a noise speck appears locally or in the entire image depending on the position of the substituted blocks. The intensity of this noise depends essentially on the magnitude of the gain factor used. Of course in these three mentioned zones if a low gain factor is used as it preserves the range of the watermark values near to the original matrix coefficients no change appears in the image. The fourth zone, which is red, is delimited between lines 160 to 256 in width and between columns 155 to 256 in height, excluding the lower triangle zero values. This zone contains the lower values of the matrix as presented in figure 1c; week energy of the image is transformed and spread in this zone. Because of the decreasing values direction in the Hessenberg matrix, this zone has the smallest values in the entire matrix. It is found to have an insignificant effect on the image quality whatever the change in its values and the increase in the gain factor. In this zone, different sectors can be substituted by a watermark. In our simulations, the sector delimited between lines 220 to 253 and columns 230 to 256 is used. As a result, the watermark insertion will consist in embedding a watermark block in this fourth Hessenberg zone as shown in figure 9 and 10. The gains factor increases up to 35 without any distortions to the watermarked image. Then, embedding a watermark in this zone allows us to increase the watermark robustness against different attacks. The visual imperceptibility threshold is not exceeded and the image quality is preserved. To supply more security, a secret key is provided to the copyright owner, used to determine the position of the watermarked sector.

Based on the detailed previous study, the choice of the embedding zone and the selected block to be substituted with the watermark block is chosen. It must increase the robustness of the embedded watermark against a large set of attacks and decreases the distortions introduced to the watermarked image. The fourth zone in the Hessenberg matrix is found as it satisfies these essential constraints to develop the watermarking algorithm. After watermarking the Hessenberg matrix, an inverse process shown by the following equation is applied to come back to the spatial representation of the image as follows:

$$\ddot{I}_W = inv(Q)\ \mathrm{H}'\ \mathrm{inv}\left(\mathrm{Q}^{\mathrm{T}}\right) \tag{15}$$

Where $\ddot{I}_W$ is the watermarked image, H′ the watermarked Hessenberg matrix and Q is the unitary matrix. The watermark-embedding algorithm is presented in Figure 5.

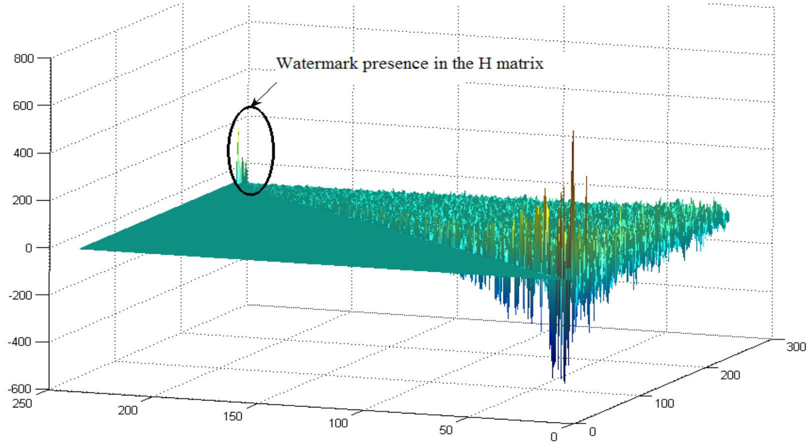


Fig. 9. The embedded watermark in the Hessenberg matrix.

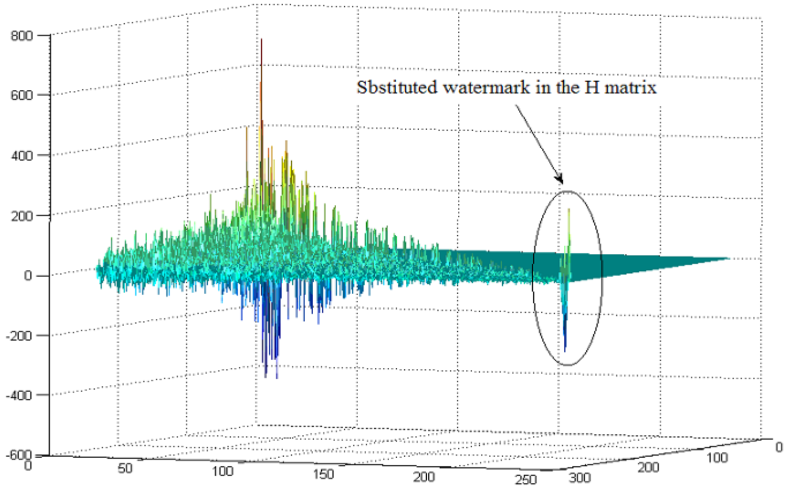


Fig. 10. Block watermark substituted in the fourth zone of the H matrix.

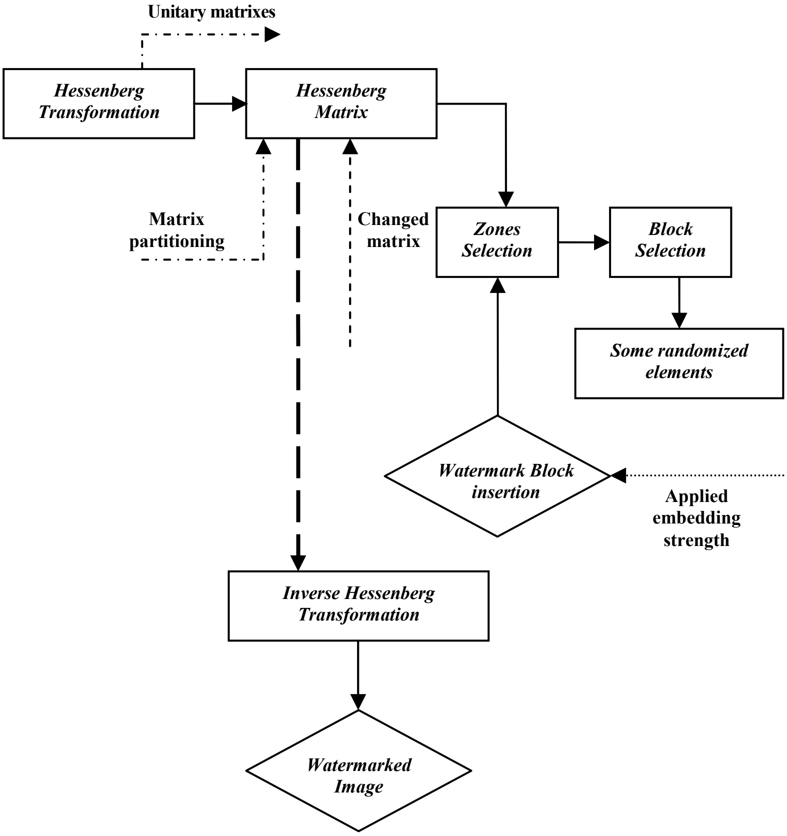


Fig. 11. The Hessenberg watermark embedding algorithm.

2.3 Watermark recovery and tests

In order to test the watermark presence, the Hessenberg transformation is applied on the watermarked image to handle the transformed Hessenberg matrix. The secret key is used to detect the position of the watermarked sector. Once this sector is located, the similarity between the extracted and the original watermark (W′ and W) is determined. The watermark extraction procedure is detailed in figure 6. Two measures of performance can be computed, the first is the (BER) bit error rate as false positive or false negative detection errors. The false positive detection is a false alarm of an incorrect detected watermark. While the false negative detection error consists in a missed detection of an existent watermark. The probabilities of these two types of errors are derived based on a first-order autoregressive image model as shown by:

$$P_{fp} = \frac{1}{2}\, erfc\left(\frac{T\sqrt{N}}{\sigma_w\ \sigma_I\ \sqrt{2}}\right) \text{And } P_{fp} = \frac{1}{2}\, erfc\left(\frac{\left(\sigma_w^2 - T\right)\sqrt{N}}{\sigma_I\ \sqrt{2}}\right) \tag{16}$$

Where $erfc\left(x\right) = \frac{1}{\sqrt{2\Pi}}\int_{x}^{\infty} e^{-t^2/2}\,dt$, σ_w and σ_I are the watermark and image variances, N is the total number of pixels in the watermarked image and T is the detection threshold over which the watermark is set as detected. While the location of the watermark in the Hessenberg matrix is known by the use of the secret key, the second measure based on the normalized correlation is more appropriate in this work. This measure is presented in (17), where n and m are the watermark block size.

$$Corr = \frac{\sum_{1}^{n}\sum_{1}^{m} W\,W'}{\sqrt{\left(\sum_{1}^{n}\left(\sum_{1}^{m}W^2\right)\ \sum_{1}^{n}\left(\sum_{1}^{m}W'^2\right)\right)}} \tag{17}$$

To test the robustness and the improvement that our method offers, different STIRMARK attacks are applied on the watermarked image. Once the image is attacked, a correlation is computed between the original watermark block and the attacked one. A threshold T_h fixed equal to 0.85 is applied to decide whether the watermark is detected or not. This threshold is chosen as the mean of the total correlation values corresponding to all synchronous and asynchronous attacks applied on the watermarked image in the simulation study. Some of these tests that provide week results corresponding to some geometrical attacks are not displayed in Table 1. The used attacks and gathered correlation results are detailed in Table 1. The chosen sector and applied gain factor, giving the best correlation result without causing any visible distortions between the original image and the watermarked one, are shown in the same table. The results prove the high resistance of this method against different attacks, especially JPEG compression, noise adding and convolutions filtering Stirmark attacks. In addition, an improvement against other attacks is noted when compared with other current techniques as shown in figures 18 and 19. Various examples of those attacks are simulated below on the original cameraman image followed by the corresponding peak detection of the watermark block between 1000 other random blocks. The watermark embedding capacity depends on the required embedding procedure. For a robust watermark embedding only the fourth Hessenberg zone is used. We can embed in the other zones for a fragile watermark embedding method using a low gain factor that avoids damaging the image quality. To establish a more quantitative measure of imperceptibility, we make use of the peak signal to noise ration (PSNR) metric. This measure serves generally as a good rule for the watermark visibility estimation, given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(I_i - \dddot{I}_{Wi}\right)^2 \tag{18}$$

$$PSNR = 10\log_{10}\frac{255\times 255}{MSE} \tag{19}$$

Where MSE is the mean square error, N is the total number of pixels in the image, I and $\dddot{I}_W$ are the original and watermarked image. With K is used equal to 35 and embedding in the fort zone. The PSNR of the watermarked image is maintained in the range of 40–50 dB (so that $\ddot{I}_W$ is visually indistinguishable from I). Generally if the distortions between two images output a PSNR higher than 35 dB, no differences are visually detected. In this work, the computed PSNR from the experimental study is equal to 44.55 dB using the cameraman image, 49.70 dB using the image door and 44.62 dB using the image blood.

	STIRMARK ATTACKS	Correlation values		STIRMARK ATTACKS	Correlation values
1	Convolution filter 1	0.9138	**2**	Adding Noise 0	1
3	Convolution filter 2	0.9994	**4**	Adding Noise 20	0.9662
5	Compression JPEG 20	0.9603	**6**	Adding Noise 40	0.8811
7	Compression JPEG 40	0.9704	**8**	Adding Noise 80	0.7671
9	Compression JPEG 60	0.9802	**10**	PSNR 10	1
11	Compression JPEG 70	0.9891	**12**	PSNR 50	1
13	Compression JPEG 80	0.9925	**14**	Remove lines 60	0.8412
15	Compression JPEG 90	0.9999	**16**	Rotation 2°	0.7789
17	Compression JPEG 100	1	**18**	Rotation 45°	0.8641
19	MEDIANCUT 5	0.7538	**20**	Rotation 90°	0.9124
21	MEDIANCUT 7	0.8634	**22**	Affine 7	0.9876
23	MEDIANCUT 9	0.7565	**24**	Affine 5	0.9941

Table 1. Watermark detection responses values (Substituted Block dimension and location: [220:253,230:256], Gain factor: K = 35).

JPEG quality factor	20	40	60	70	80	90	100
PSNR (dB)	23.943	24.581	24.979	25.168	25.373	25.607	25.760

Table 2. PSNR variation against different JPEG quality factor attacks.

Median Filter size	5×5	7×7	9×9
PSNR (dB)	21.528	20.831	20.381

Table 3. PSNR variation against Median filtering attack.

Noise level	20	40	60	80
PSNR (dB)	25.763	8.374	6.826	6.015

Table 4. PSNR variation against different noise level adding attack.

The PSNR is also computed between the original image and the different watermarked attacked images. The PSNR variation against different attacks level as JPEG compression, median filtering and noise adding are present in tables 2, 3 and 4. The watermark extracting using the inverse procedure is shown by figure 12.

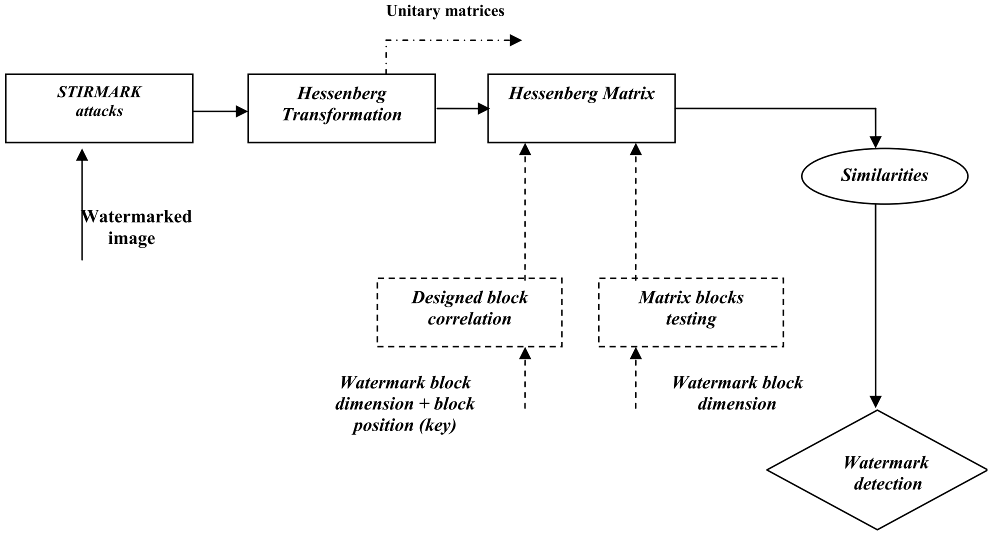

Unitary matrices
STIRMARK attacks
Hessenberg Transformation
Hessenberg Matrix
Similarities
Watermarked image
Designed block correlation
Matrix blocks testing
Watermark block dimension + block position (key)
Watermark block dimension
Watermark detection

Fig. 12. The watermark blind detection algorithm after the applied attacks.

2.4 Experimental results and discussion

The results prove the high resistance of this method against different kinds of attacks such as: JPEG compression, noise, rotation, affine transform and other Stirmark attacks. Using this method we exploit the advantages of being robust face to different kind of attacks in the same time we have the guaranty of a secure and undetectable watermark with a rapid embedding and extraction algorithm. After applying the Hessenberg transform on different attacked images we note the values of the H matrix are nearly invariant if the fixed gain factor in the embedding procedure is not exceeded. This explains the fact that the watermark is always detected. The maximum error values resulting from the difference between the H matrix of a watermarked image and this of a watermarked and attacked image by JPEG compression indexed 40 are contained in the range between 7 and 9 in the entire matrix excluding the upper left matrix corner. The matrix resulting of this difference is shown in figures 13. This error pick value varies in the range of 2.10^3. With regard to this error value occurring in the upper part of the first zone, the error resulting in the zones 2 and 3 represents a ratio limited between 4 and 5.10^{-3}. The fourth zone presents an error band between the watermarked image and the attacked one contained in the range between -2 and 3 as shown in figure 14. This means that the numerical difference resulting between the watermarked image and the watermarked attacked one in this zone is too week. This difference presents the error resulting from the applied attack. This resulting error presents a ratio of 1.10^{-3} when compared with the error pick in the first zone. This week variation in the embedding zone with regards to the other zones composing the H matrix preserves the watermark from loss. On the other hand, the mathematical characteristics concerning the successive iterations of the Hessenberg transform bringing the values of the lower triangular part to zero and transforms the image matrix into a triangular matrix which absolute values varies from 2.10^4 to 0. We can say that the energy of the image is concentrated in the upper and middle zones. That is why embedding in the fourth zone introduces a week energy to the image and then a high embedding strength is needed to introduce distortions to the watermarked image.

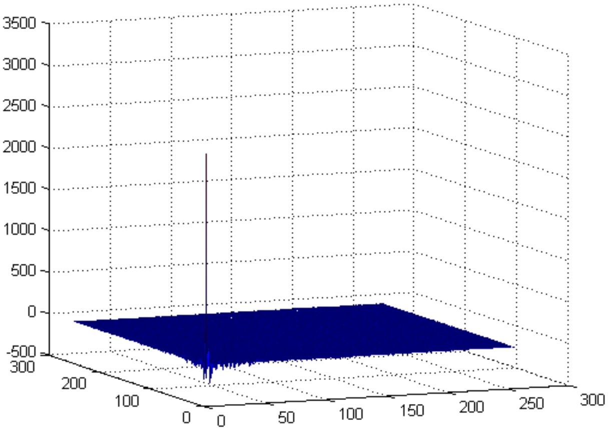


Fig. 13. Difference resultant between two *H* matrices belonging respectively to a Watermarked image and JPEG 40 lossy compression attacked image.

Figure 15 and 16 illustrate the presence of the watermark in the fourth zones after the watermarked image has been attacked by JPEG 40 compression and convolution 2 filtering. Because of the very week variation of the Hessenberg values in the watermarking zones if compared with the others matrix zones, the watermark is always preserved from loss after attacks.

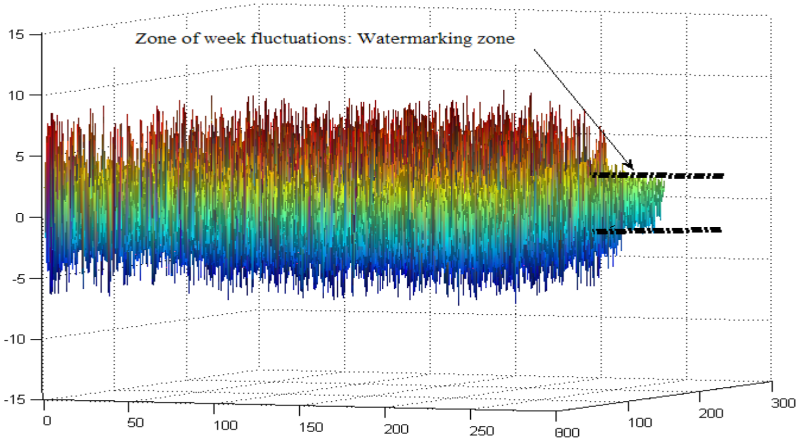


Fig. 14. Error band between the H matrices corresponding to zones 2, 3 and 4.

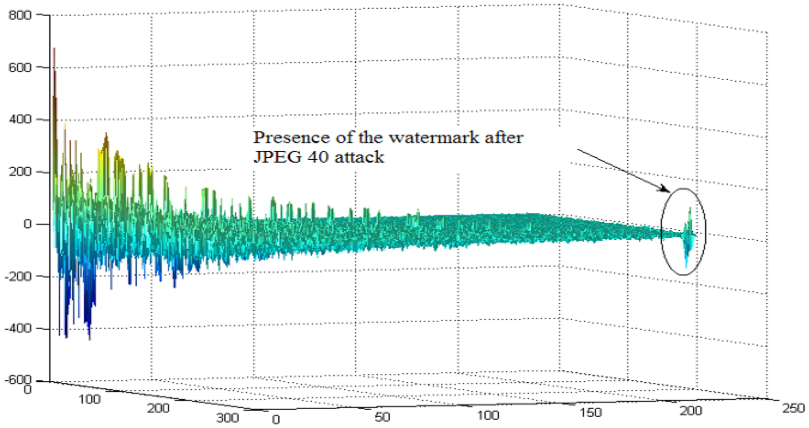


Fig. 15. Watermark presence in the H matrix after JPEG 40 attack

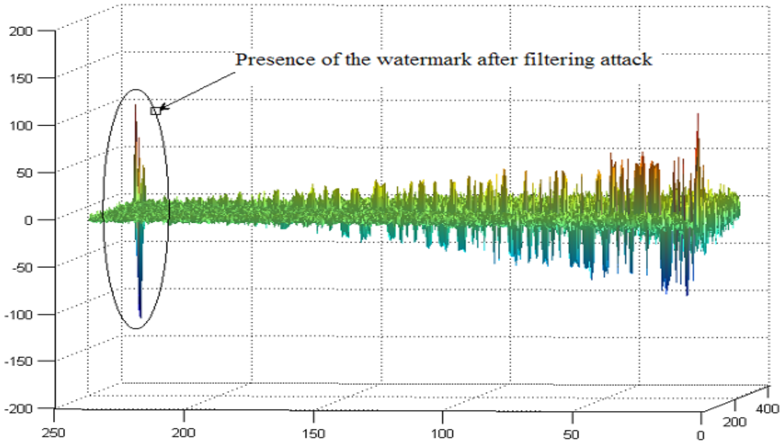


Fig. 16. Watermark presence after Convolution filtering 2 attack.

As shown in figure 18, the proposed method is also more robust to JPEG compression than recent algorithms. From JPEG 70 to JPEG 50 the CHEN algorithm is slightly higher then our algorithm, Nevertheless, when using high compression rates such as JPEG 20 and JPEG 10 the proposed method maintain its robustness by preserving the embedded watermark. The other methods lose their resistance face to this destructive attack. In the same time, when compared with other methods using the DWT domain, high robustness against noise adding is presented. The additional robustness to convolution filtering and different geometrical distortions is also presented. The proposed Hessenberg method provides also a better robustness face to median filtering then many other techniques as shown in figure 19.

Figure 17, presents the correlations values computed between the extracted watermark after seven JPEG compression attacks and the original one, with different quality factors from JPEG 100 to JPEG 20.

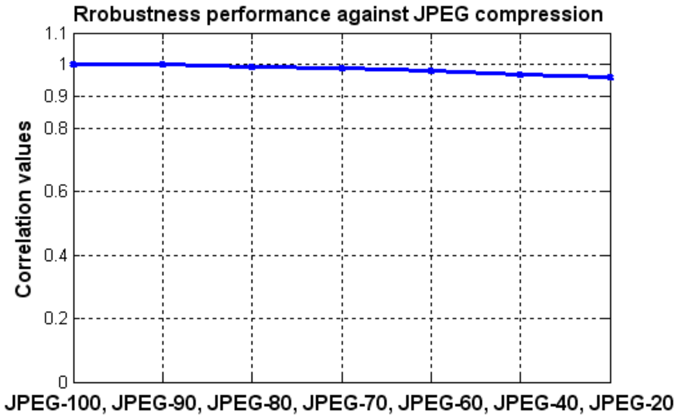


Fig. 17. Robustness of the Hessenberg technique against JPEG compression quality.

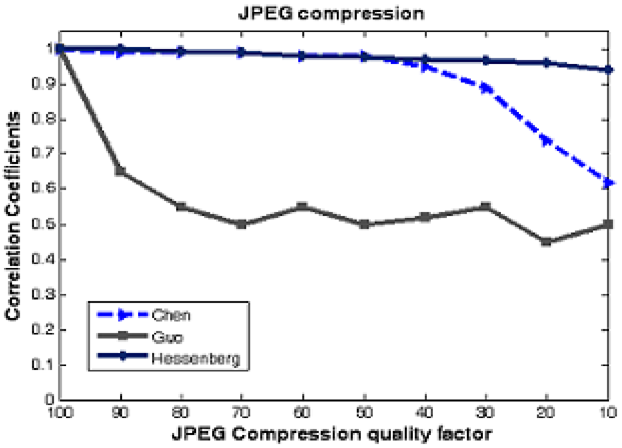


Fig. 18. Robustness variation face to JPEG compression.

As we shown in table 1 this technique is visibly more robust against synchronous attacks than geometrical ones. In fact the synchronous attacks applied on the image modify the values of its intensity values. After applying the Hessenberg transform on it we will extract the same H matrix with a variation of its values from a zone to another. Of course as we demonstrated in the previous section, the fourth zone has the less variation in the entire matrix and that is why we find high robustness if we embed in this zone. Conversely, if an asynchronous attack is applied on the watermarked image a change in the pixels position will happen. Since the Hessenberg transformation is a block-based orthogonal transform, the change in the image pixels position will be reflected on the Hessenberg matrix. The position of the values of the H matrix corresponding to the location of the watermark will be modified by this geometrical transformation increasing the error on the watermark extraction. As the watermark block will be extracted from the same location in the H matrix, it will contain some wrong values introduced by the change of the values position. For this reason, this method is more robust against synchronous attacks than asynchronous ones presenting high degrees and levels of distortions. Using the Hessenberg domain, we can provide more robustness against different sets of intentional and malicious possible attacks. Various examples of these attacks are simulated below on the original cameraman image followed by the corresponding correlation values of the detected watermark block between 1000 other matrix blocks. For a robust watermark embedding only the fourth Hessenberg zone is used. We can embed in the other zones for a fragile watermark embedding method using a low gain factor to avoid damaging the image quality. In addition the image appearance can be changed by modifying the Hessenberg blocks matrix with respect to the zone location.

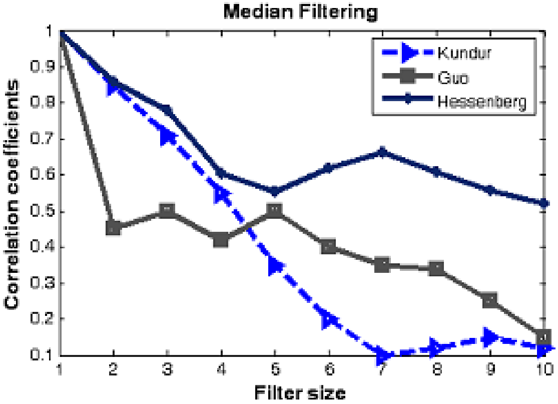


Fig. 19. Robustness against Median Filtering.

2.4.1 Experiment 1: Convolution filtering 1 and 2 attacks

Figures 20a, 20b, 21a, and 21b show applied convolution filters attack and the responses of the watermark detector to 1000 random blocks of the watermarked Hessenberg image matrix. The positive response due to the correct watermark block is much stronger than with incorrect blocks. The detection rates of these attacks are very high when compared

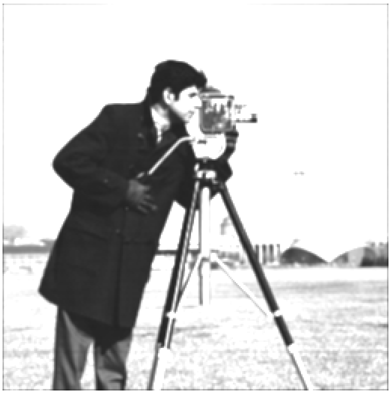


Fig. 20. a) Convolution 1 attack.

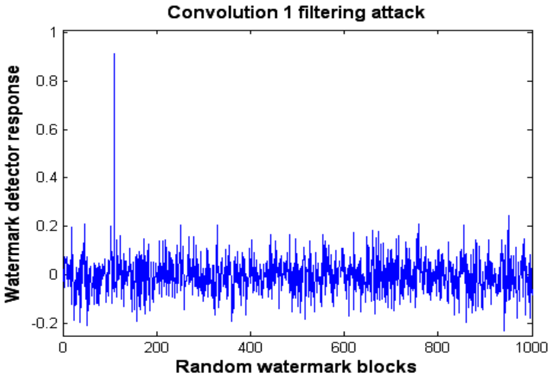


Fig. 20. b) Watermark detection result.

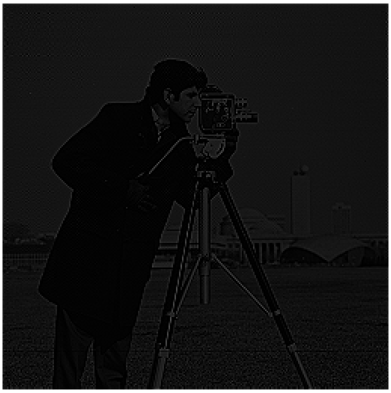


Fig. 21. a) Convolution 2 attack.

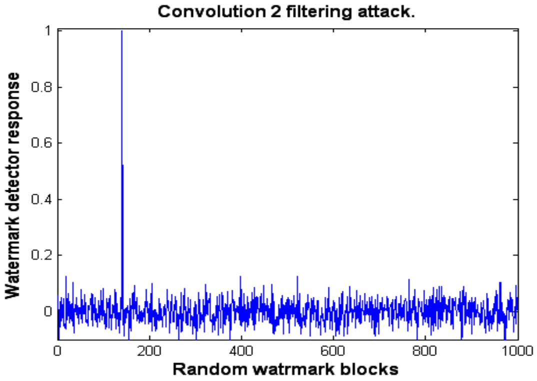


Fig. 21. b) Watermark detection result.

with other watermarking domains, such as the spatial domain [8]. Two filtering attacks are proposed: the convolution 1 and 2 filters, where the first represents a gaussian filter and the second a sharpening filters. The parameters of the filters applied on the cameraman image as shown in Fig.20a and Fig.21a, is given as the following: CONV 1 filter = 3, 3, 9. Where the two first numbers corresponds to the filter width and high and the third number is the division factor. The matrix filter is $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. The parameters of CONV 2 filter are the same, and its matrix is: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$.

2.4.2 Experiment 2: JPEG compression attacks

Figures 22a and 22b show the results obtained after respectively applying a JPEG 20 and JPEG 60 compression on the image. This technique is found to be robust against this kind of signal processing distortion. A series of different JPEG compression rates are applied, as shown in Table 1, with high rates of watermark block detection. Using this method, the watermarked images are safe face to these unintentional signal processing attacks and the watermark can be entirely get back after this lossy compression.

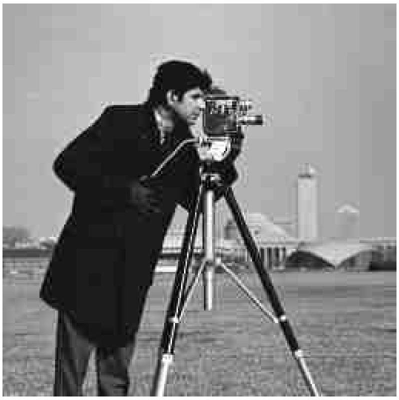


Fig. 22. a) JPEG 20-compression attack.

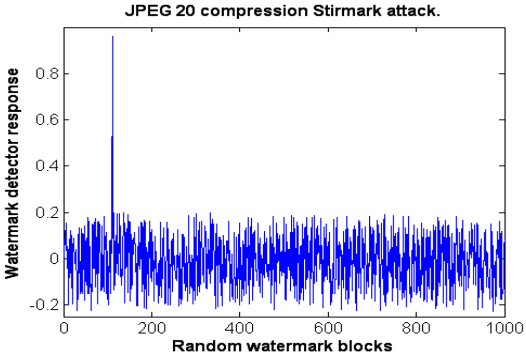


Fig. 23. b) Watermark detection result.

2.4.3 Experiment 4: Noise attacks

Gaussian noise with zero mean and varied variances σ .Different noise magnitudes are added to the watermarked image from (0 to 80), as shown in Table 1, with the corresponding watermark detector responses. In all the cases, the watermark was not removed, and the correlations with the real watermark block were higher. Figures 23a and 23b illustrate two noise attacks (80), with the corresponding normalized watermark detector responses.

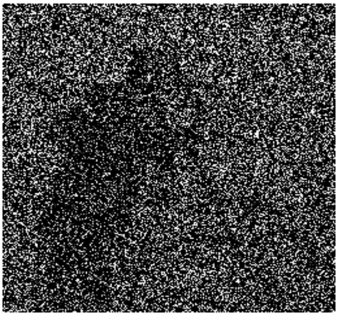


Fig. 23. a) NOISE 80 attack.

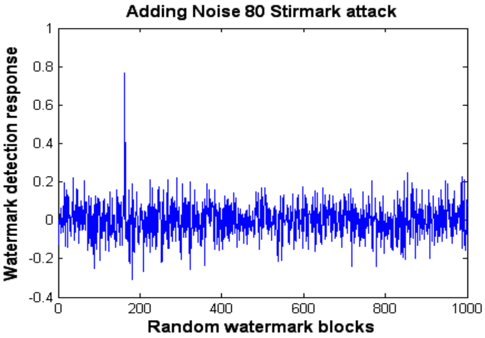


Fig. 23. b) Watermark detection result.

2.4.4 Experiment 5: Geometrical distortions

Different geometrical distortions are applied as attacks to the watermarked image such as rotations and affine transforms. The rotation attacks change the position of the image pixels and break the correlation between the image and the embedded watermark. Many rotation degrees are applied between which three are showed in the table above: two, forty-five and ninety degrees rotations attacks. The second kind of geometrical distortions attacks are the affine transform. Two kinds of affine attacks are applied indexed as affine 5 and affine 7.This geometrical transform is given by the equation (14).

$$\begin{vmatrix} X' \\ Y' \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \cdot \begin{vmatrix} X \\ Y \end{vmatrix} + \begin{vmatrix} d \\ e \end{vmatrix} \qquad (20)$$

The variables a, b, c, d, e changes with the affine transform index and fixed by the Stirmark tool.

The proposed technique is found to be robust against this kind of signal processing distortion as shown by the correlation results in the table 1. Figures 24a and 25b show the response value of the watermark detection corresponding to the computed correlation between the original watermark block and the detected one after the applied attack, and the attacked watermarked image, respectively. The watermark block resists this image processing with a high correlation value, compared with the other matrix blocks used in the test.

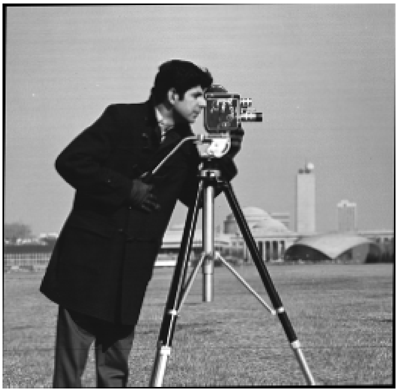


Fig. 24. a) Affine 7 transform attack.

2.4.5 Effect of changing blocks in different Hessenberg matrix zones

As detailed in section 2, the Hessenberg matrix is divided in different zones. The choice of the zone in which the watermark block is substituted, is very important in order to avoid a possible image characteristics and appearance change. In this section, the influence of each matrix zone is illustrated. The fourth zone, which is indifferent with regard to the image quality when a watermark is embedded in, is also shown, and different images are

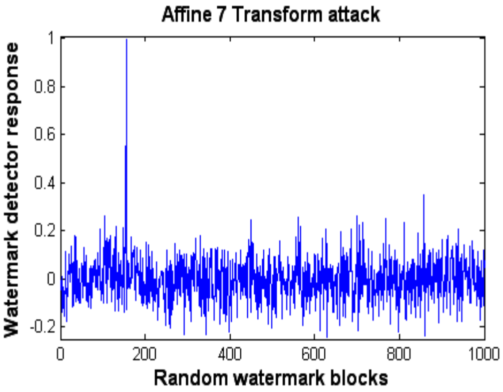


Fig. 24. b) Watermark detection result.

watermarked with various gain factors. All figures belonging to these simulations are detailed below in different sets. In all the sets of figures, the limits of the blocks processed are presented by [lines-limits, Columns-limits], and K represents the gain factor used.

Figures 25: "Image BLOOD", 26: "Image DOOR" and 27: "Image RICE" are the original images used throughout the simulation experiments, in addition to the "Cameraman" image shown in section 3. The images, from 28a to 28h, illustrate the results of changing blocks in the first zone applied to the "cameraman, door and blood" images with different gain factors, varying from 5 up to 35. The examples show the effect of operating in different sectors of zone 1. The damage caused to these images by changing blocks in this first zone results in local or general variable image blurring. In the set of figures, from 29a to 29f, the second zone of the transformed image matrix is changed. The simulation is applied to the different used images with different gain factors, as shown below. When changing a sector belonging to this zone, a string effect appears locally or on the entire image. The intensity of this string effect varies with the level's value of the gain used. Figures from 30a to 30i represent the results of simulations where the third zone is processed. In fact, this is an interesting zone. The result of changing blocks in zone 3 is shown in Figures 30a, 30b, 30c, 30d and 30g. Embedding a watermark in this zone affects these images by adding a non-uniform noise appearance. The intensity of this noise varies from one image region to another. Figures 30e, 30f, 30h and 30i show the effect of a uniformly distributed noise by changing the sectors detailed with the images.

In all these simulations, dealing with the fourth zone is the most interesting in this proposed Hessenberg watermarking method. From Figures 31a to 31i, the fourth zone of the transformed image matrix is changed. In fact, as we will detail in these figures, it is clear that this zone is the least sensitive to watermark embedding, and can be totally insensitive in some cases to the blocks changing. Figures 31a, 31b, 31c and 31d clearly show that changing the blocks in the fourth Hessenberg matrix zone does not affect the image quality where no visible changes are observed in the watermarked image even though the gain increases from 1 up to 35. As shown in Figure 31e when using the "Cameraman" image, some visible changes begin to appear in the upper left corner of the watermarked image as indicated by the arrow if the applied gain factor reaches the value 38. The same distortions are shown in Figure 31f, with a gain factor that reaches 50. In the figure 31g, the same gain factors is used, and the same block is changed, we note no visible changes appear in the watermarked

"Blood" image. Some other images require higher gain factors to be affected by certain changes. Figures 31h and 31i show the "Door" and "Rice" images, where a gain factor of 210 and 250 respectively, is applied. Until the gain factor reaches these high values, some visible changes begin to appear, as indicated by the arrows on the regions affected by the changes. It is clear that the gain factor used, and which is capable of causing some damage or changes to the watermarked image differs with the image type and characteristics. Of course, using a high gain factor implies higher correlation values and watermark detection between the attacked watermarked image and the original one. The Table 5 presents the different PSNR corresponding to the figures from 31a to 31i. The computed PSNR shows the distortion magnitude introduced to these test images watermarked in the fourth Hessenberg zone.

Figures number	21a	21b	21c	21d	21e	21f	21g	21h	21i
PSNR (dB)	44.72	44.68	44.60	49.70	44.41	44.32	49.65	42.72	43.12

Table 5. PSNR variation against different watermarked test images with variable embedding strength.

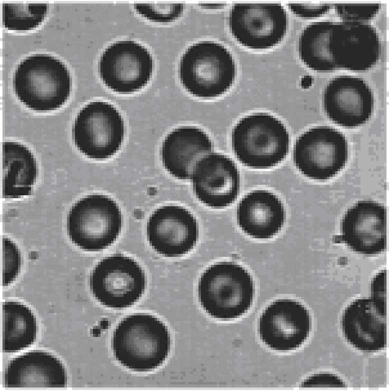


Fig. 25. Original image “Blood”.

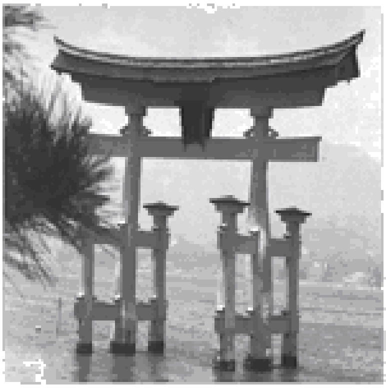


Fig. 26. Original image “Door”.

Original image "Rice".

Fig. 27. The original images used in the simulations.

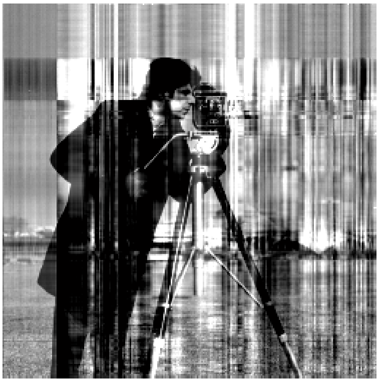


Fig. 28. a) [1:10,10:20], K = 5.

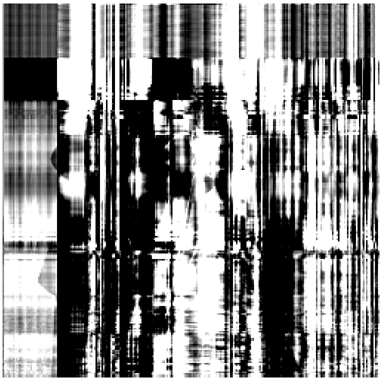


Fig. 28. b) [1:10,10:20], K=20.

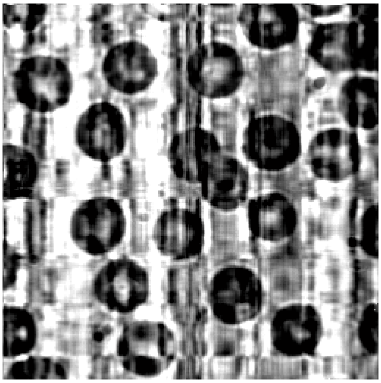


Fig. 28. c) [1:10,10:20], K=5 (Blood).

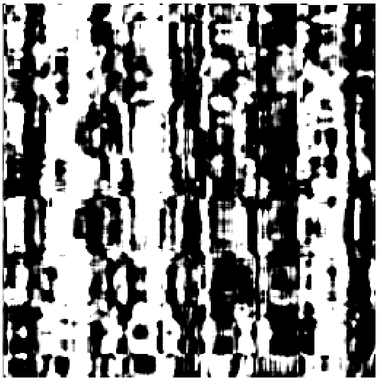


Fig. 28. d) [1:10,10:20], K=35 (Blood).

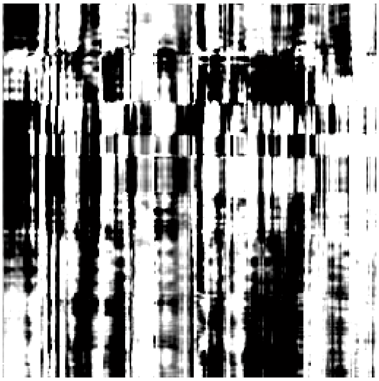


Fig. 28. e) [1:10,10:20], K=20 (door).

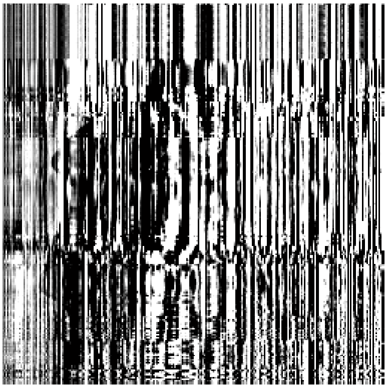


Fig. 28. f) [1:10, 50:100], K=5.

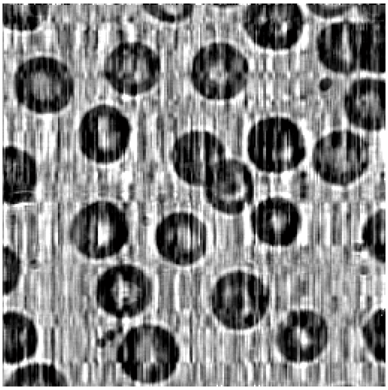


Fig. 28. g) [1:10, 50:100], K=5 (blood

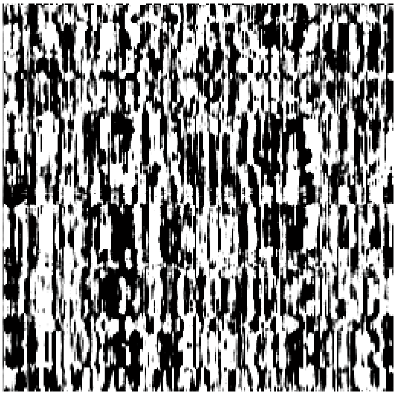


Fig. 28. h) [1:10, 50:100], K=35.

Fig. 28. Result of changing blocks in the first zone on the image.

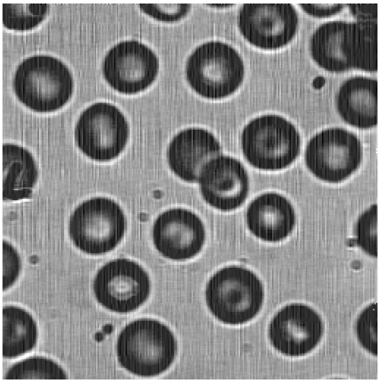


Fig. 29. a) [1:10,230:256], K=5.

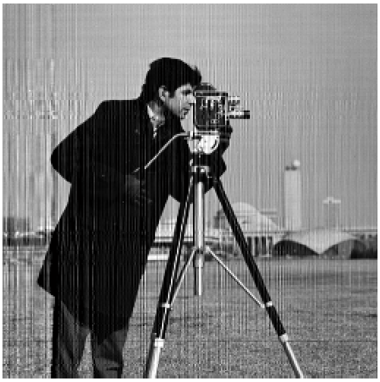


Fig. 29. b) [1:10,230:256], K=5.

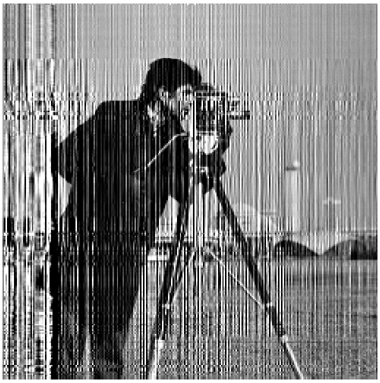


Fig. 29. c) [1:10,230:256], K=20.

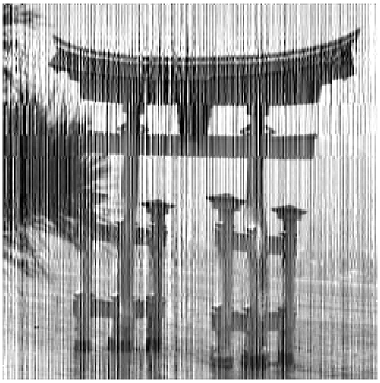


Fig. 29. d) [1:10,230:256], K=20.

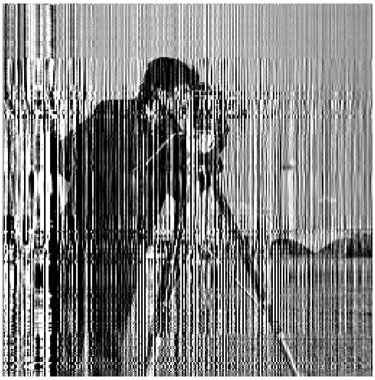


Fig. 29. e) [1:10,230:256], K=35.

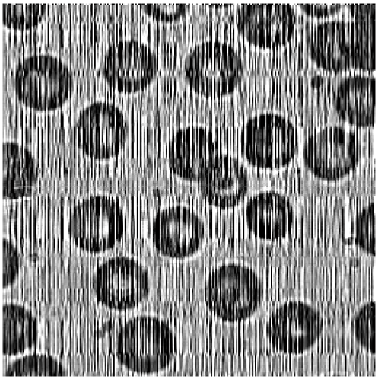


Fig. 29. f) [1:10,230:256], K=35.

Fig. 29. Result of changing blocks in the second zone on the image.

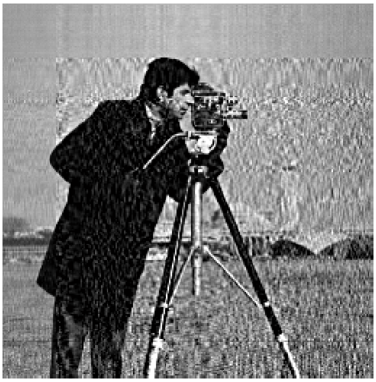


Fig. 30. a) [30:60, 60:100], K=5.

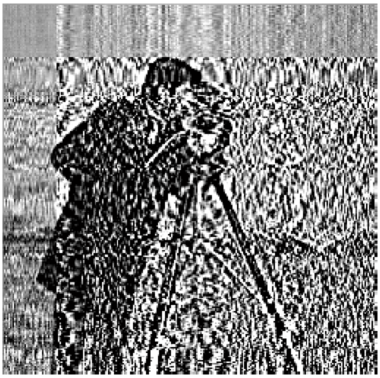


Fig. 30. b) [30:60,60:100], K=25.

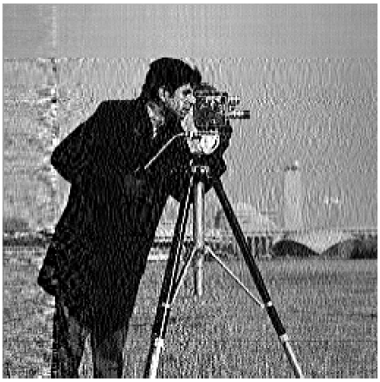


Fig. 30. c) [30:60,150:256] K=5.

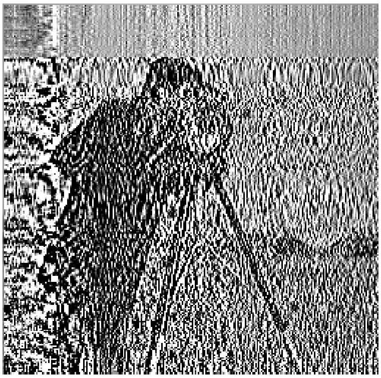


Fig. 30. d) [30:60,150:256], K=35

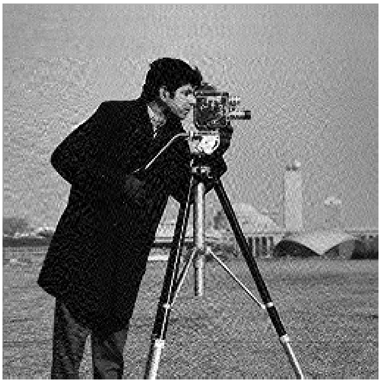


Fig. 30. e) [100:180,100:180], K=5.

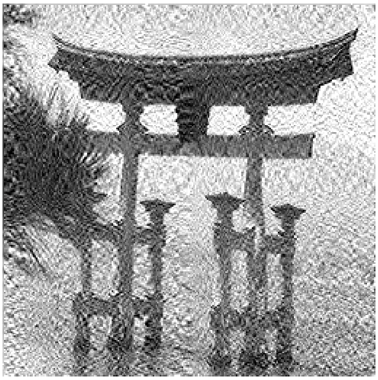


Fig. 30. f) [100:180,100:180], K=20.

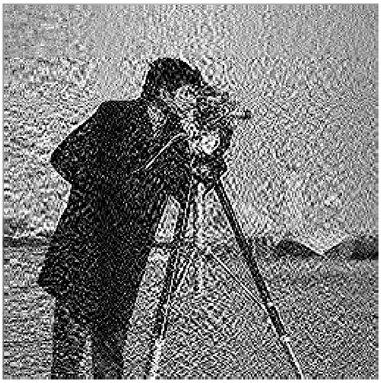


Fig. 30. g) [100:180,100:180], K=20.

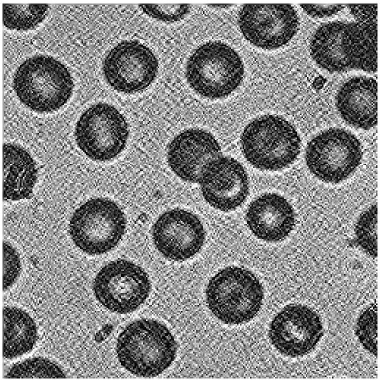


Fig. 30. h) [100:180,100:180], K=35

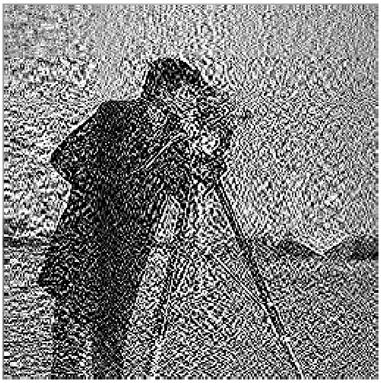


Fig. 30. i) [100:180,100:180], K=35.

Fig. 30. Results of changing blocks in the third zone on the image.

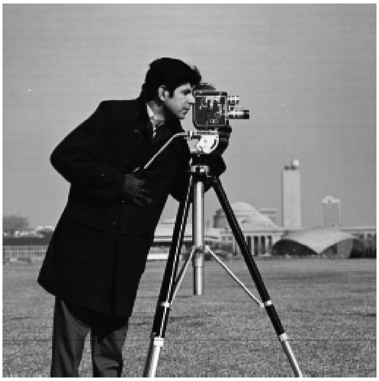


Fig. 31. a) K=5.

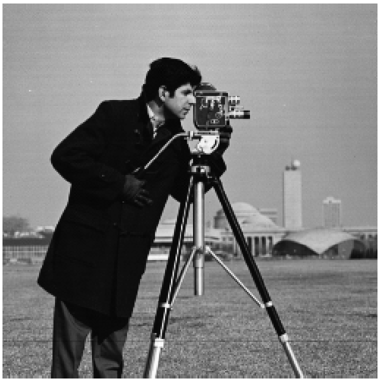


Fig. 31. b) K=10.

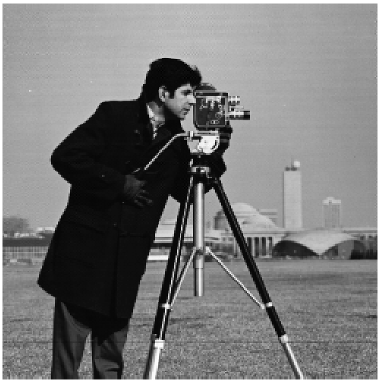


Fig. 31. c) K=20.

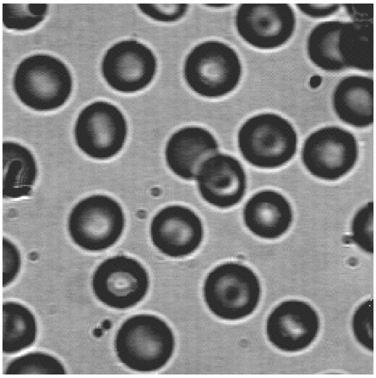


Fig. 31. d) K=35.

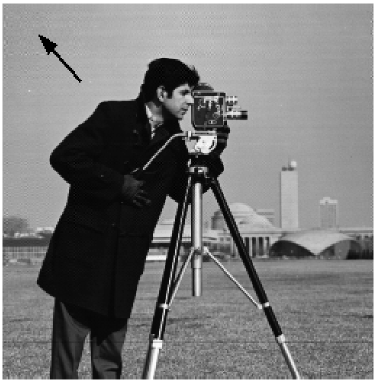


Fig. 31. e) K=38.

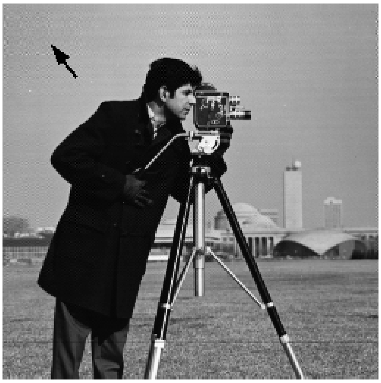


Fig. 31. f) K=50.

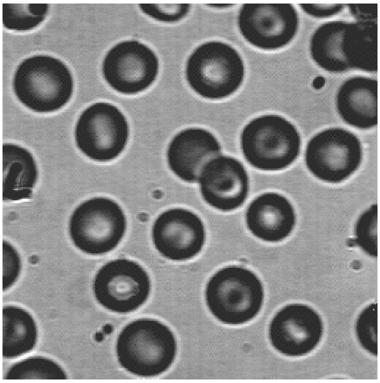


Fig. 31. g) K=50.

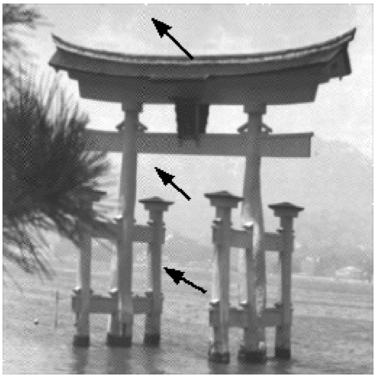


Fig. 31. h) K=210.

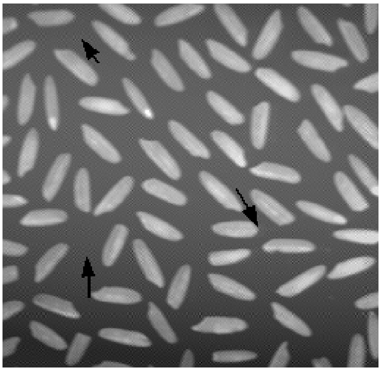


Fig. 31. i) K=250.
Fig. 31. Result of changing blocks in the fourth zone on the image.

4. A new watermarking method using the parametric hough transform domain

4.1 Introduction

Different constraints are required in a watermarking method, such imperceptibility and robustness. Besides, lossy JPEG compression remains the most unintended used attacks with data exchange in Internet, for size reduction. It can seriously affect the embedded watermark if the compression rate is high and the used scheme presents a weakness against this attack. So, the best solution resides in exploiting the DCT domain used in the JPEG algorithm in order to dispose of the robustness against this compression or the multi-resolution domain as in. But acting to be robust against this attack reveals automatically the domains of watermark embedding and than increase the possibility of its detection. In this section, a novel watermarking method is proposed. It consists in using the parametric space of the mathematical Hough transform as a watermarking domain. The technique consists in selecting specific maximums in the Hough matrix with respect to a secret key. The peaks are found to be invariant points in the proposed Hough domain especially against lossy JPEG compression. Two signatures are considered; the first is hold in the Hough domain by the transformed space and consists in the locations of the specific chosen invariant points. Whereas, the second is represented by the use of end points of the correspondent detected lines. These end points are used as centers to embed similarities blocks in. The watermarking in this domain is found to be extremely robust against JPEG compression and some geometrical transforms. All these attacks are generated by the STIRMARK tools. This section is organized as the following: In section 2, an overview of the Hough transform is presented. Section 3 details the proposed method in the Hough domains: the carried study and the proposed solutions. In section 4, we study the robustness of this technique against different STIRMARK attacks by testing its capacity to detect the embedded watermark. The privileges offered by this approach are also detailed, and finally we conclude this work.

4.2 Hough transform overview

The Hough transform is a mathematical algorithm used in images processing to detect the presence of parametric forms as ellipses or lines in the image. This technique uses the principle of evidences accumulating to prove the existence of a particular form in the image. For this aim, this transform uses a parametric domain or space to characterize these forms. Each form is represented by its proper parameters in this space. In our work, this transform is coded to be used as lines detector where its parametric space is exploited. It's important to note that the Hough transform is found to have the capacity to detect the same segments or broken lines in the image, before and after being compressed. This detection invariance is due to the fact of the invariance properties of its parametric space in the case where the image is subjected to JPEG compression and some asynchronous transforms. In the case of lines detecting, the Hough transform is presented as follows:

Each line can be described in the orthonormal space by the equation (1) or (2)

$$y = a \,.\, x + b \tag{1}$$

$$\rho = x \cdot \sin\theta + y \cdot \cos\theta \tag{2}$$

The parametric space is than composed by two parameters: ρ and θ that forme a space matrix as shown in figure 1.

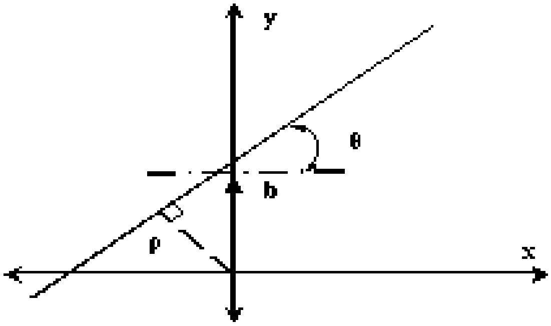


Fig. 1. The parametric space (ρ, θ).

An infinity of lines can pass through a fixed point called P having (x,y) as coordonates. But if we consider a second point P_1 having (x1,y1) as coordinates, only one line can pass through P and P_1 satisfying the same couple of (ρ and θ). If this principle is applied to the image, the Hough transform of an image generates a parametric space matrix as presented in (3):

$$H\ (M)\ =\ \mathrm{A}(\rho,\theta) \tag{3}$$

Where H is the Hough transform, M is the image matrix and A is the space parametric resulting matrix. Since this matrix contains a limited number of elements, the number of possible detected lines is with respect to the quantization step of ρ and θ in their respective variation domains. Peaks contained in this space represent an accumulation of evidences indicating the possibility of lines presence with respect to a specific position and orientation.

4.3 The proposed method

In this work, we propose to apply the philosophy of the Hough transform on the image in order to process and manipulate it in the parametric Hough space, and use it as a watermark-embedding domain. If we consider an (N×M) image; in order to accumulate evidences and define the parametric space, the information source is gathered from the pixels composing the image. More the evidences are accumulated and put in the parametric space matrix; more the chance to identify a real line in the image is high. In the following, we will define the parametric space matrix generated by the Hough transform as the Hough space or Hough domain. The first step consists in applying a high pass filter in order to extract the image edges. In each point belonging to this edge, infinity of lines can pass through it. Accumulating evidences in the parametric space provides the unique position and orientation (ρ and θ) for witch one line can pass through this point. In our case the positions and the orientations are quantified by a step computed with respect to the required precision. The Hough space is than a two-dimensional matrix or map. The size of this map depends on the quantification step as shown in Figure 2.

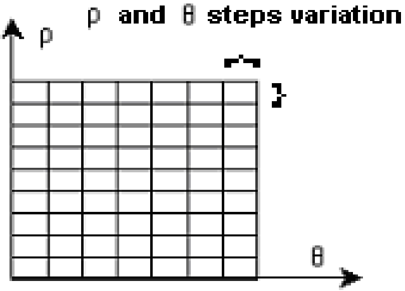


Fig. 2. The parametric space map.

The quantification steps are computed as follow:

Consider an image with size N×M, θ can vary in the interval range of [0, 2π], the value of ρ is maximum when it's computed in the image diagonal. The steps and variation domains of ρ are than described by the equations (4, 5, 6 and 7):

$$\rho_{MAX}^2 = \left(\frac{\mathrm{N}}{2}\right)^2 + \left(\frac{\mathrm{M}}{2}\right)^2 = \frac{\sqrt{N^2 + M^2}}{2} \tag{4}$$

$$\rho_{MAX} \in \left[0\,, \frac{\sqrt{N^2 + M^2}}{2}\right] \tag{5}$$

More the quantification steps are decreased, better the resolution is; but the Hough matrix size increase. In order to attend equilibrium between: resolution, computing time and parametric space dimension, we will fix the orientation step depending on the image size. If θ varies as

$$\theta \in \left[0\,, \frac{2\pi}{\sqrt{N \cdot M}}\right] \tag{6}$$

If the image is square the resolution will be as:

$$\theta \in \left[0\,, \frac{2\pi}{N}\right]; \text{and } \rho_{MAX} = \frac{\sqrt{2}}{2} \cdot N; \ \textit{with } \Delta\,\rho = \sqrt{2} \tag{7}$$

In the following, we will consider the ρ step as:

$$\Delta\ \rho = \frac{\rho_{MAX}}{100} \tag{8}$$

$$\Delta\rho = \frac{\sqrt{N^2 + \mathrm{M}^2}}{200} \tag{9}$$

These steps provide an acceptable precision to browse the entire image as shown in Figure 3.

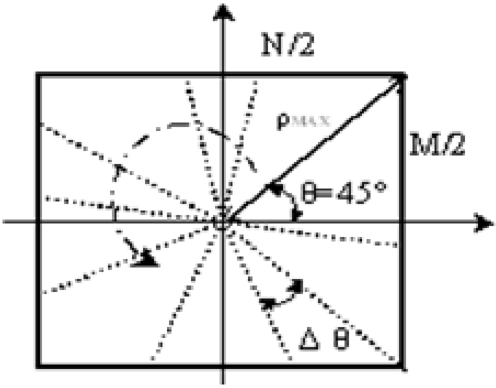


Fig. 3. Image browsed by ρ and θ variation.

The operation of accumulating evidences in the Hough matrix for potential presence of lines in the image is characterized by the appearance of maximums in the matrix. A threshold is previously chosen to characterize since witch values we can consider maximums in the space parametric matrix as peaks. The number of picks and their position in the Hough map is used as secret key. By finding and fixing the peaks number, we extract the correspondent lines and respectively their end points. These end points are used as centres of blocks similarities embedding. In fact in each end point we extract a block of size $(2n+1) \times (2n+1)$. All these blocks are substituted with similarities as shown by the equations (10) and (11). If we consider B_i as the chosen block, W_i the watermark and B_{wi} the watermarked block:

$$W_i = \text{dyn}\left(\text{B}_\text{i}\right) = \frac{B_i - \dddot{\text{B}}_i}{\max\left(B_i - \dddot{\text{B}}_i\right)} \tag{10}$$

$$\text{B}_\text{wi} = B_i \cdot \left(1 + \alpha \cdot W_i\right) \tag{11}$$

Where $\dddot{\text{B}}_i$ is the indexed block mean and α is the watermark embedding strength. In the experimental results, as will be shown in the next section, the selected peaks (maximums) in the Hough space matrix corresponds to the embedded watermark location in the image. The peaks position in the Hough space and their respective end points in the spatial representation, are completely invariant when the image is attacked by JPEG compression or some geometrical transforms.

4.4 Experimental configuration

In our experiments, the cameraman image is used to simulate the applied method and the chosen attacks. This image is chosen because of its content variation. In fact it contains lines in addition to homogeneous and textured zones. A binarizing method is applied to convert this image into a binary image. An edge extractor filter is then applied to extract the image edges. Once theses edges are taken out, the Hough transform is coded and then applied on the resultant image to browse it and then accumulates evidences in the transformed parametric matrix to decide witch maximums corresponds to real lines in the image. In this matrix, the number of chosen peaks and their respective position represents the secret key used to select the ends of the correspondent lines where the similarities blocks are embedded. The position of the peaks are returned and saved to be compared with the same peaks position after the attacks are applied on the image and view if they can be considered as unvaried points with respect to the applied attack. The peaks are defined as the entire matrix maximum that exceeds a fixed threshold. In this work, in order to obtain a better precision, the threshold is fixed as $T_h = 0.7$ and then:

$$P_K = T_h \cdot \max(H) \tag{12}$$

Where P_k represents the returned peaks, H is the Hough matrix. Different peaks can be selected and then view theirs corresponding lines and end points in the image as shown in Figures 4, 5, 6 and 7.

4.5 Simulation results

In the following, the number of peaks is chosen equal to one. The location of the peaks in the Hough parametric space is represented by the respective position in the matrix lines and columns as (L, C).

The first peak is selected and its position is returned as (-23, 89.2472) in the Hough matrix space. That means that $\theta = -23°$ and $\rho = 89.2472$. Figures from 8 to 11 present the detected peaks in the Hough space and their respective positions shown in Table 1. The correspondent detected lines and respectively their end points where the similarities blocks are embedded are shown in these figures. The Table 1 presents the attacks applied on the cameraman image; the JPEG compression and the rot-scale transform. The extracted peaks after the attacks have been applied presented. A total invariance is remarked concerning the peaks positions against lossy JPEG compression.

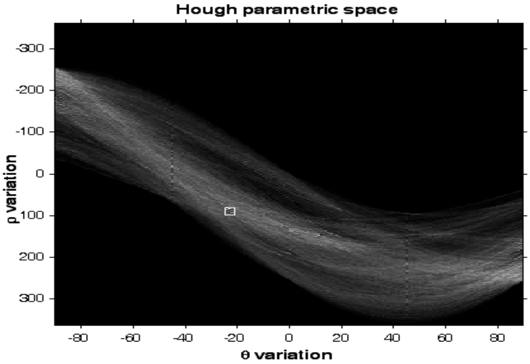


Fig. 4. The first pick in the parametric Hough space.

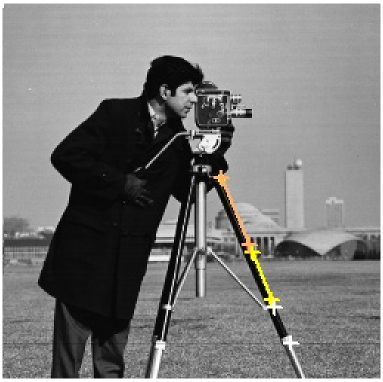


Fig. 5. Three detected segments corresponding to the first selected peak presented in Fig.4.

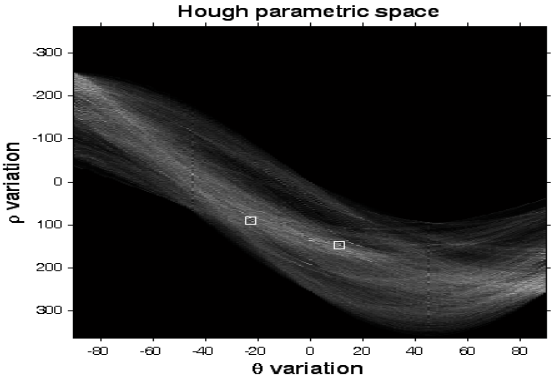


Fig. 6. Two first picks in the parametric Hough space.

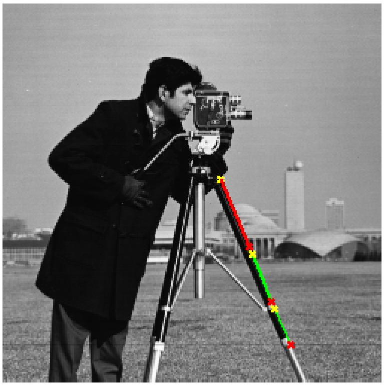


Fig. 7. Detected segments corresponding to the first selected peak presented in fig.6.

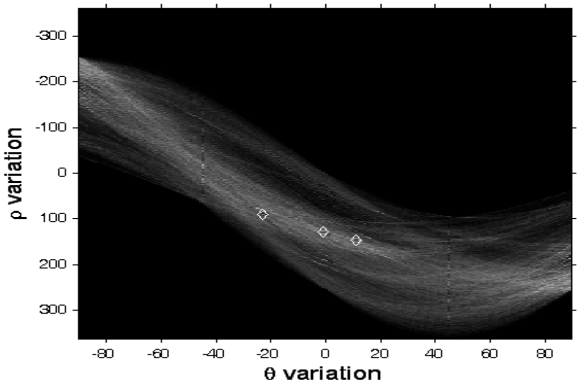


Fig. 8. The three first selected picks.

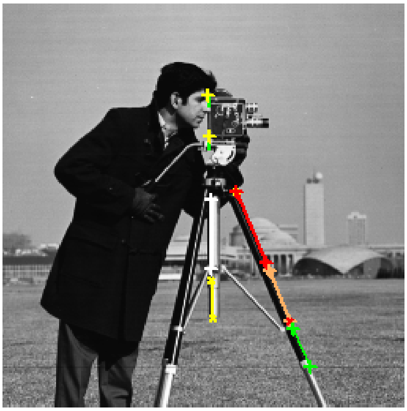


Fig. 9. Detected segments corresponding to the three first selected peaks presented in Fig.8

4.5.1 Experiment 1: JPEG compression

The figures from 8 to 11 show respectively the cameraman image attacked by the JPEG 10, 20 and 40 compression and the obtained result of the detected peaks position leading to the watermark detection. The proposed method based on the Hough parametric space is found to be highly robust against lossy compression. A series of different JPEG compression rates from JPEG 100 to JPEG 10 are applied as shown in Table 1. The distortion caused to the watermarked image by all these attacks hasn’t changed the position of the selected peaks in the Hough space.

APPLIED ATTACK	PEAK POSITION IN THE HOUGH SPACE
JPEG 100	(-23, 89.2472)
JPEG 90	(-23, 89.2472)
JPEG 80	(-23, 89.2472)
JPEG 70	(-23, 89.2472)
JPEG 60	(-23, 89.2472)
JPEG 40	(-23, 89.2472)
JPEG 30	(-23, 89.2472)
JPEG 20	(-23, 89.2472)
JPEG 10	(-23, 89.2472)
ROTSCALE –0.25	(-23, 89.2472)
ROTSCALE –0.5	(-23, 89.2472)
PSNR 100	(-23, 89.2472)

Table 1. Invariance of the selected peak position against applied attacks.

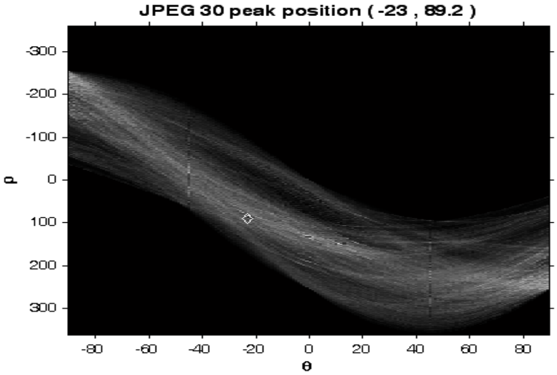


Fig. 10. The position of the detected peak in the JPEG 30 compressed image.

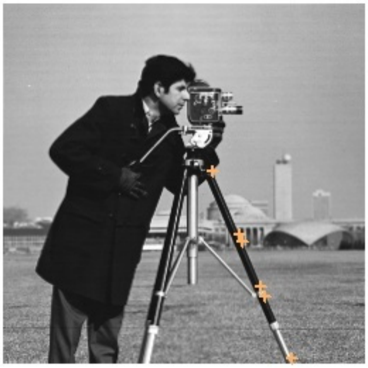


Fig. 11. End points of the detected lines correspondent to the peak in Fig.10.

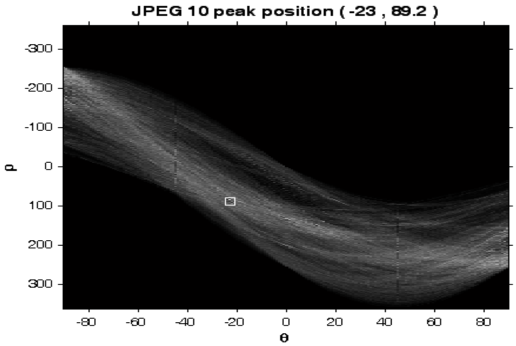


Fig. 12. The position of the detected peak in the JPEG 10 compressed image.

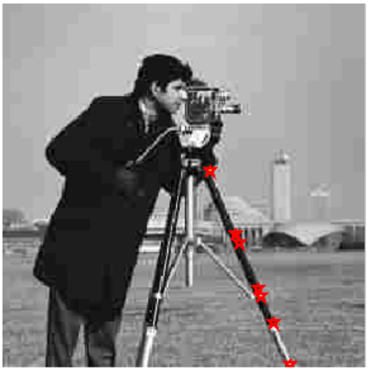


Fig. 13. End points of the detected lines correspondent to the peak in Fig.12.

4.5.2 Experiment 2: Asynchronous attacks

Figures 14 and 16 show the rotation and scale attack, and the correspondent detected peaks position in the Hough space. As shown in the Table 1 and the figures below. The use of this space provides high robustness against these attacks and grant invariance properties to the selected peaks if the image is attacked, especially when dealing with small distortions the invariance of the peaks position is kept unchanged.

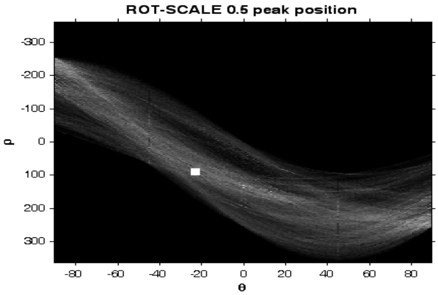


Fig. 14. The position of the detected peak in the ROT-SCALE 0.5 attacked image.

4.6 Evaluation and comments

In this section, we comment the results and compare our proposed methods to other ones. As shown previously, this method is highly robust against JPEG compression. The Peaks positions are unvaried whatever the compression rate used. The image is represented by the Hough parametric space as a new representation domain where the signature is characterized by the unvaried positions of the selected peaks and hold in it. The robustness of this method is picked out from the robustness of this new Hough domain face to JPEG and other attacks. In fact, the parametric Hough space holding the signature cannot be modified by synchronous attacks as JPEG, filtering, i.e. it doesn't modify the pixels position and then the ends of lines. As a result, the detected peaks after the Hough transform is applied remain unvaried. Conversely, the asynchronous attacks that modify vastly the pixels position change the position of the lines ends and then modify the positions of the Hough space peaks. Figures 16 and 17 show the asynchronous rot-scale attack with two degrees and the correspondent detected peaks.

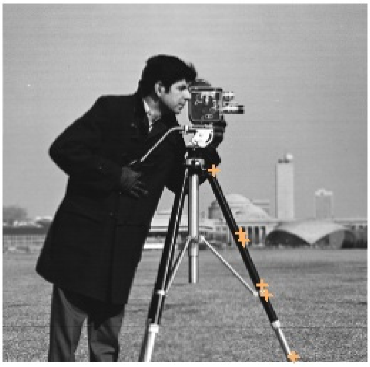


Fig. 15. End points of the detected lines correspondent to the peak in Fig.14.

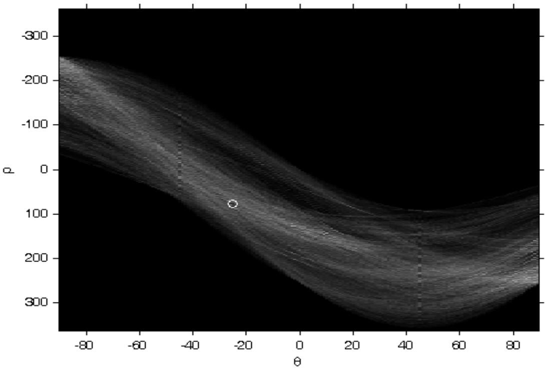


Fig. 16. Rot_scale 2° peak position.

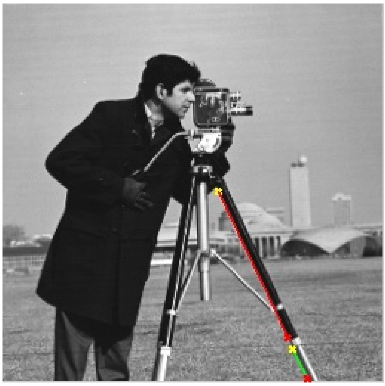


Fig. 17. Peaks detected corresponding to Fig.16.

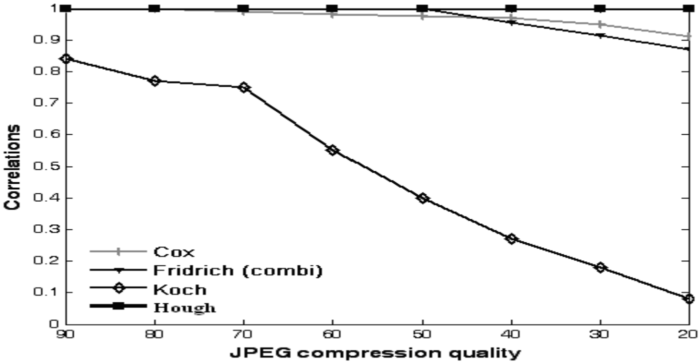


Fig. 18. Robustness of the Hough algorithms.

Figure 18 shows the robustness of the proposed Hough algorithm when compared with the well known and most robust algorithms proposed in the DCT domain to defeat the JPEG compression attacks. It's evident that our algorithm is the most robust. This method is found to be better than those actually in use, due to the fact that the image is not processed similarly to the methods in use that embed the watermark by modifying the image either in the spatial domain or in the frequency and multi-resolution domains.

5. Conclusion

A new domain and watermarking techniques are proposed in this work. In the first presented approach, sing the mathematical Hessenberg transformation, the original image is transformed in the Hessenberg domain as a triangular matrix which values present specific characteristics. The embedding procedure is applied to a transformed image in a non-sensitive zone that has no effect on the image quality after an inverse transformation is applied. Processing the lowest values in the entire matrix, by modifying them we impose a low variation on the original coefficients of the image and no distortions appears on the watermarked image. A study was carried out to show how the Hessenberg matrix can be partitioned, and the effect of each matrix zone on the image perception. Many advantages are proposed by the use of this method. In fact it allows the use of a high embedding strength which allows being more robust against attacks than DCT, DFT or spatial methods. Its robustness against lossy JPEG compression exceeds this allowed by the well known and used until now DCT domain. In addition, by choosing the appropriate zone and changing some of its values, this technique is able to give the appearance of a noised, banded or blurred image without really applying these signal processing operations on the image. This technique is found to be very resistant against simultaneous a large set of synchronous and asynchronous signal processing attacks, and the watermark is always present in the entire set of the attacked image. The watermark detection process and the similarities computing presented in this approach are obtained from tests applied on the "Cameraman" image with a gain factor of 35. Evidently, the embedding strength can be highly augmented without exceeding the watermark imperceptibility when dealing with certain kinds of other images presenting different characteristics that allow high gains value without any changes, as shown in Figures 31h and 31i. Markedly, this high gain increases the robustness of the embedded watermark and improves vastly the results of the proposed method. We finally note that we propose a blind watermarking technique; the presence of the host image is not required for watermark detection procedure.

In the second approach, a new watermarking method is presented. Based on the mathematical hough transform, a parametric space matrix is obtained and used as a new space where the image is processed. A secret key is characterized by the number of selected peaks in this space matrix is chosen. These peaks are found to be invariant and robust against jpeg compression and some asynchronous transforms. They are also used to determine the correspondent lines end points where similarities blocks are embedded. The embedded watermark is carried in hough space by the invariant peaks and their position that corresponds to the embedded similarities. This method proposes higher resistance against lossy compression than the previous algorithms based essentially in the DCT domain.

6. References

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, “LAPACK User's Guide”, Third Edition, SIAM, Philadelphia, 1999.

M. Barni, F. Bartolini, A. De Rosa and A. Piva, “ Capacity of the watermark channel: how many bits can be hidden within a digital image”, Proc. SPIE 3657, 1999, pp. 437-448.

P. Bas, J.M. Chassery and B. Macq, “Image watermarking: an evolution to content-based approaches”, Pattern Recognition 35 (2002), pp.545-561.

P. Bas and J.M. Chassery, Tatouage d’image résistant aux transformées géométriques, 17éme colloque GRETSI, Vannes, France, 13-17 Septembre 1999.

G. Caronni, “Assuring ownership rights for digital images”, Proceeding of reliable IT Systems, VIS' 95, Viewveg Publishing Company, Germany, 1995.

L. Chang, “Issues in Information Hiding Transform Techniques”, Storming Media, Computers: Cybernetics, 20 May, 2002, http://www.stormingmedia.us/84/8491/A849104.html.

L.H. Chen and J.J. Lin, "Mean quantization based image watermarking", Image and vision computing vol. 21, No. 8, 1 August 2003, pp. 717-727.

P. Chun Chen, Y. Sheng Chen, and W. Hsing Hsu, "A communication system model for digital image watermarking problems", International conference on Information Systems Analysis and Synthesis , ISAS, Vol.6, pages 2935, USA,1999.

I. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia", IEEE Trans. Image Process. No. 6, Vol. 12, June 1997, pp.1673-1687.

F. Davoine and S. Pateux, "Tatouage de documents audiovisuels numériques", Hermes science, Lavoisier 2004.

L.Diane, Cours de traitement d'images, Laboratoire I3S Informatique, Signaux et Systèmes, Université de Nice Sophia Antipolice, Rapport de recherche ISRN I3S/ RR, 22 janvier 2005.

A. Fabien, P. Peticolas, "Watermarking schemes evaluation", IEEE Signal Processing Magazine, Vol. 17, no. 5, pp. 58-64, September 2000.

A. Fabien, P. Petitcolas, M. Steinebach, F. Raynal, J. Dittmann, C. Fontaine and N. Fatès, "A public automated web-based evaluation service for watermarking schemes: StirMark Benchmark", In Ping Wah Wong and Edward J. Delp, editors, proceedings of electronic imaging, security and watermarking of multimedia contents III, vol. 4314, San Jose, California, U.S.A., 20-26 January 2001.

J. Fridrich, Combining low-frequency and spread spectrum watermarking, In Proceedings of the SPIE Symposium on Optical Science, Engineering and Instrumentation, San Diego, USA, July 1998.

G. H. Golub, and C. F. Van Loan, "Matrix Computation", Johns Hopkins University Press, 1983, pp. 384.

M. Van.droo. Genbroeck, "Acquisition et traitement d'image", Université de Liège publication, Institut Montefiori, Service de télécommunication et d'imagerie, Septembre 2001, version 4.14.

H. Guo and N. D. Georganas, "Multi-resolution Image Watermarking Scheme in the Spectrum Domain", IEEE Canadian conference on electrical and computer engineering, pp125, may 2002.

F. Hartung and M. Kutter, Multimedia watermarking techniques, Proc. IEEE Vol. 87, No. 7, 1999, pp. 1079-1107.

J. Huang, Yun Q. Shi and Yi Shi, "Embedding image watermarks in DC components", IEEE Trans. Consumer Electron. 46 3 (2000), pp. 415-421.

N. Kaewakamnerd and K.R. Rao, Wavelet based image adaptive scheme, Electronics letters Vol. 36, 2000, pp. 312-313.

S. Katzenbeisser, F.A.P. Petitcolas, Information hiding techniques for steganography and digital watermarking, Artech house, December 1999.

E. Koch and J. Zhao, Towards robust and hidden image copyright labeling, Proceedings of the IEEE International Workshop on Nonlinear Signal and Image Processing, pp. 452-455, Halkidiki, Marmaras, Greece, June 1995.

X .Kong, Y. Liu, H. Liu and D. Yang, "Object watermark for digital image and video", Image and vision computing journal, Vol.22, Issue.8, August 2004, pp. 583-595.

D. Kundur and D. Hatzinakos, "Diversity and Attack Characterization for Improved Robust Watermarking", IEEE Transactions on Signal Processing, Vol. 49, no. 10, Oct 2001.

P. Lan, "Robust transparent image watermarking system with spatial mechanisms", Journal of systems and software, 15 February 2000, 107-116.

G.C. Langelaar, I. Setyawan, and R.L. Lagendijk, "Watermarking in digital image and data: A state of the art overview", IEEE Signal Processing magazine, September (2000), pp. 20-40.

M. Laug, "Traitement optique du signal et des images", Ecole Nationale Supérieure de l'Aéronautique et de l'Espace SUP'AERO, Edition Cépaduès, France, 1980.

P. Moulin, M.K. Mihcak, "A framework for evaluating the data-hiding capacity of image sources", IEEE Transactions on Image Processing, September 2002, vol. 11, no. 9, pp. 1029-1042.

A. Natarajan, "Discrete cosine transform", IEEE Trans. on Computers, 1974, Vol. c-23, pp 90-93.

N. Nikolaidis and I. Oitas, "Robust image watermarking in the spatial domain", Signal Processing, vol. 66, no. 3 (1998), pp. 385-403.

I. Pitas, T. Kaskalis, "Applying signatures on digital image", Workshop on Nonlinear Signal and Image Processing, IEEE, Neos Marmaras, June 1995, pp 460-463.

C.I. Podichuck and W. Zeng, image adaptive watermarking using visual models, IEEE journal on selected area in communication, Special Issue on Copyright and Privacy Protection, Vol. 16, 1998, pp. 525-538.

V. Rouilly, Présentation de la transformée de Hough, Rapport interne, ENST, Raris, France, http://www.tsi.enst.fr/tsi/enseignement/ressour ces/mti/ellipses/Hough.html .

J. O'Ruanaidh and T. Pun, "Rotation, Scale and Translation Invariant Digital Image Watermarking", Proc. IEEE international conference on image processing, Vol. 1, pp. 536-539, 1997.

K. Sayood, "Data compression", Maurgan Kaufmann Publishers, San Francisco, CA, 2000.

H. Seddik, M. Sayadi and F. Fnaiech, "A New Watermarking method using the parametric Hough Transform Domain", WSEAS Trans. on Information Science and Application, no. 9, Vol. 2, Sep. 2005, pp. 1277-1284.

H.Seddik, E.Ben.Braiek, "Color Medical Images Watermarking" ICGST International Journal on Graphics, Vision and Image Processing, Vol.6 Special Issue on Medical Image Processing, pp.81-86, March 2006.

J.S. Seo and C.D. Chang Yoo, Localized image watermarking based on features points of scale-space representation, Pattern Recognition, Vol. 37, No. 7, July 2004, pp. 1365-1375.

F.Y. Shih, S.Y.T. Wu, "Combinational image watermarking in the spatial and frequency domains", Pattern Recognition, vol. 36, Issue 4, April 2003, pp. 969-975.

P. Su, C.J. Kuo and H.M Wang, Blind digital watermarking for cartoon and map images, SPIE conference on security and watermarking of multimedia contents, San Jose, CA, USA, January 1999 pp. 296-305.

T.L. WANG and W.B. GRAGG, "Convergence of the shifted QR algorithm for unitary Hessenberg matrices", Mathematics of computation, Volume 71, Number 240, pp. 1473-1496,November 30, 2001.

R. Wolfgang, E. Delp, "A watermarking technique for digital imagery: further studies", International Conference on Imaging Science, Systems and Technology, Los Vegas, Nevada, July, 1997.

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

9

Audio Watermarking for Automatic Identification of Radiotelephone Transmissions in VHF Maritime Communication

Oleksandr V. Shishkin and Vitaliy M. Koshevyy
Odessa National Maritime Academy
Ukraine

1. Introduction

Audio watermarking (AW) corresponds to digital information imperceptibly embedded into the audio signal. AW for maritime VHF (Very High Frequency) communication is inspired first of all by the ability of implementation an automatic identification of radiotelephone transmissions in the channels of maritime (156…174) MHz mobile radio communication service. Applied to VHF radiotelephony, a watermarking system could overcome existing limitations, and ultimately increase safety and efficiency of maritime communication. The same application of AW may be implemented in the aeronautical (118…136) MHz mobile service. In the mentioned services analogue broadcasting channels with frequency/phase and amplitude modulation correspondingly are utilized.

For the meanwhile the identification of the sea vessels is realized by means of verbal calling of ship's call sign or numerical identification. However on account of different reasons such verbal identification may be absent, transmitted with delay, or understood with errors. This problem is illustrated in Fig. 1. Motor vessel "Arcona" transmits a certain message to all stations. But one of the receiving vessels missed the name and call sign of the transmitting ship, and another ship interpreted the name of transmitting ship as "Gargona" instead "Arcona".

It is obvious that false, incorrectly interpreted or delayed verbal identification negatively affects maritime navigation. Automatic identification could avoid misidentification and call sign confusion. Strictly speaking from the time the Global Maritime Distress and Safety System (GMDSS) (Brehaut, 2009) came in force in 1999, each radiotelephone exchange should be preceded by digital selective calling (DSC) on special calling channel 70 and appropriate acknowledge by means of DSC. After such calling procedures the radiotelephone transmission should be started on the assigned in DSC working channel. Meanwhile DSC and radiotelephone exchange are two independently executed by the navigational officer operations. Correct execution of these operations under Radio Regulation completely depends on human factor. In practice, however, DSC is often ignored, and navigators at once use radiotelephone channel 16. Especially this is typical for urgent communication. In such circumstances timely, clear and authentic identification is extremely necessary. Automatic identification would exclude the human factor and increase an efficiency of VHF radiocommunication and maritime safety in the whole.

Identification in DSC is produced by means of so called maritime mobile service identity (MMSI). MMSI is a unique combination of nine decimal digits. In binary representation MMSI occupies 36 bits sequence in DSC format. The same identification by means of MMSI may be applied to radiotelephone identification.

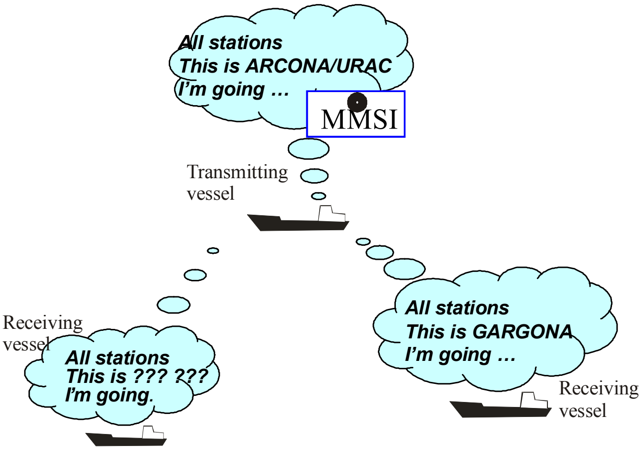


Fig. 1. Audio watermarking makes possible guaranteed identification of VHF maritime radiotelephony

Verbal identification doesn't protect against illegal radio transmission. Illegal transmissions are especially harmful on the VHF distress channel 16. Of course, violators are transmitting anonymously. Reliable identification of such transmissions could avoid the violation of radiotelephone regulation.

Another advantage of automatic identification becomes apparent in the ability of digital information inputting to another ships' navigational and information systems, for example ECDIS (Electronic Chart Display and Information System). ECDIS makes visualization of neighboring vessels in the range of VHF radio (i. e. approximately 30 nautical miles). However the transmitting vessel by no means is marked in an electronic map. Automatic identification would enable to mark on the electronic chart the transmitting vessel. Such an innovative function of ECDIS would be useful for clearing of current navigational environment. It is obvious that in such visual presentation navigator officer decision could be accepted more quickly and correctly. Such application of AW would again reduce risks of human factor demonstration.

One more application of AW is a covered information transmission in the special applications (for example, facing the threat of terrorist aggression).

It is essential that AW doesn't require altering an existing radio installation and operational procedures. AW identification keeps standard equipment and procedures. Only new telephone receiver (or headset) with embedded processor at the transmitter side and processor with mini-display switched to common audio output at the receiver side are to be mounted. Automatic identification starts right away press-to-talk switching and runs during all transmitting period independently from verbal signal occurrence. No additional time and frequency channel recourses are required.

2. Watermarking as communication problem

Watermarking may be examined in the frame of common communication problem (Cox et al., 2008), especially taking into account numerous algorithms of signal processing and technologies acquired in the field of conventional communication.

2.1 Watermarked communication over a channel with side information

Model of communication system with additive embedding of digital watermarks is presented in Fig. 2. Watermark signal w is formed on the base of embedded data m. Encoder may use information about carrier signal (or host signal) x, that is reflected with dotted line. Then carrier signal x is added by watermark w. Power σ_{w}^2 of w is limited by the acceptable level of introduced distortions of carrier signal because watermark w should be imperceptible on the background of carrier signal x.

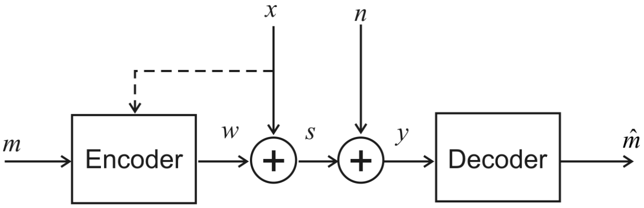


Fig. 2. Watermarked communication over a channel with side information

In the channel two interferences act against watermark w: the first interference is itself the carrier signal with the power σ_{x}^2, and the second one – a noise n with the power σ_{n}^2. Watermarking channel is characterised by its capacity - the maximum achievable code rate. Assuming the both interferences are white Gaussian noises the capacity C (bit/sample) of watermarked channel with noniformed encoder, when host signal is not available to encoder, is defined by formula:

$$\mathrm{C} = \frac{1}{2}\log_2\left(1 + \frac{\sigma_{\mathrm{w}}^2}{\sigma_{\mathrm{x}}^2 + \sigma_{\mathrm{n}}^2}\right). \qquad (1)$$

Practically $\sigma_{\mathrm{w}}^2 << \sigma_{\mathrm{x}}^2$, and capacity is limited mainly by the host itself.

At the same time using information about the carrier signal x, it is possible to increase C (this is a case of informed encoder). An idea of informed encoder goes back to Kuznetsov & Tsybakov, 1974 and is known as writing in memory with defective cells. Gel'fand & Pinsker, 1980 and Costa, 1983 shown that assuming the host is known at the transmitter, the capacity is defined by the formula:

$$\mathrm{C} = \frac{1}{2}\log_2\left(1 + \frac{\sigma_{\mathrm{w}}^2}{\sigma_{\mathrm{n}}^2}\right). \qquad (2)$$

The Eq. (2) shows that carrier signal doesn't influence on watermark transmission and the capacity is determined only by the second noise, which is unknown at the encoder. Capacity for such channel is increasing very much. The signal doesn't act as a noise source, that's why an informed encoding (i.e. "writing on dirty paper") is an attractive method for watermarking on account of potential capacity.

Noninformed encoder watermarking techniques exploit spread spectrum (SS) methods. In SS schemes the embedded bit flow is modulated by an SS sequence and added to the signal in the time or frequency domain. In schemes using SS the signal itself is seen as a source of interference and for reliable watermark restoration at the receiver the length of spreading sequence should be rather long to accumulate sufficient watermark energy. SS methods are traditionally considered as the most resistant against various attacks.

Capacity according Eq. (1) for SS watermarking is valid for assumption that the host signal is additive white Gaussian (AWGN) process. In practice, speech signal is highly correlated process and characteristics of SS watermarking may be remarkably improved. In paragraph 6 we consider adaptive whitening procedure for decreasing interfering influence of the host signal on watermark detection.

Informed encoding techniques are based on quantization the host signal directly or its certain transformation. The most popular method for this mode of embedding is quantization index modulation (QIM), proposed by Chen & Wornell, 2001. QIM-methods are free from the host signal interference, but commonly are more sensitive to attacks in watermarking channel.

Our investigations are based on classical communication approaches, applied to watermarking. It is known another view to the problem, for instance Hofbauer, et al., 2009 presented a blind speech watermarking algorithm that embeds the watermark data in the phase of non-voiced speech by replacing a certain voice segments on watermarked signal. However, the proposed method is based only on speech as a carrier signal and cannot involve any audio signals.

A comprehensive review of state-of-art methods for watermarking and data hiding is done by Moulin & Koetter, 2005. From the variety of watermarking method we focused on three candidates: SS, QIM and improved SS (ISS). The main goal of the paper consists in applying the modern communication technologies to audio watermarking.

2.2 Interferences in VHF radio channel

In this section we consider interferences which are important for watermarking in VHF analog radiochannel.

AW uses the common radiotelephone channel. The main interferences in through audio-radio-audio channel that affect AW are:

1. intersymbol interference (ISI) caused by low frequency circuits in the transceiver and multipass radio waves propagation;
2. flat amplitude fading;
3. external additive noise;
4. nonlinear distortions (clipping);
5. resampling and desinchronization.

Hofbauer & Kubin, 2006 proposed to take into account also Doppler effect that is actual for aeronautical applications.

Watermarking channel model is presented in Fig. 3.

Watermarking is concerned with the reliable transmission of information embedded into a host signal. The main difference from classical communication situation comes from the restriction of the host signal distortion. Digital watermarking can be viewed as a communication problem: information m to be sent from point A to point B is encoded into a signal w using information on host signal x. It is clear that making use of x calls for some time delay in x transmission. The time delay depends on complexity of processing at the encoder. But practically delay less then 100 – 200 msec in the master channel determines nothing but gives the ability to eliminate interfering influence of host on watermark signal w. Standard audio-radiotelephone channel in Fig. 3 is denoted C – D.

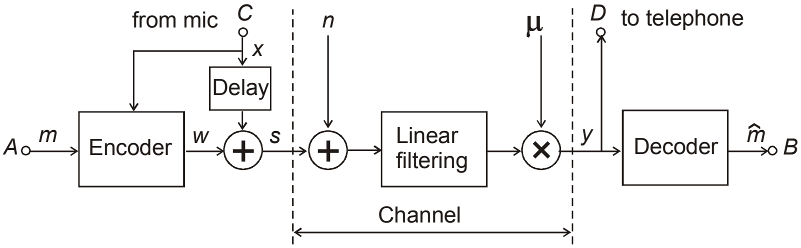


Fig. 3. Watermarking channel model

Watermarked signal $s = x + w$ is then passing through a common channel, which is unknown by nature. The embedded watermark should be reliably decodable even after further processing of the marked signal, which is also denoted as attack against the embedded watermark. Consider that malicious attacks are absent.

The attacks result unavoidable signal degradation. For robust watermarking we need restoration of embedded information m until speech communication in the master channel is possible. Another words watermarks robustness should monotonically fall with the quality of radio transmission in the master channel.

2.2.1 Intersymbol interference (ISI)

ISI forms distortions of a signal in which current symbol interferes with the previous one. Previous symbol has influence on the current symbol like noise, thus making communication less reliable. In the radio (or wireless) channels ISI is usually caused by multipath propagation. The transmitting medium in VHF radio communication is the atmosphere, in which radio signal is transferred by means of electromagnetic waves. The received electromagnetic signal is usually a superposition of a line-of-sight path signal and multiple waves coming from different directions. This phenomenon is known as multipath propagation. It is clear that reflected waves have to pass a longer distance and therefore arrive with a time-delay compared to the line-of-sight signal. The received signal is spread in time and the channel is said to be time dispersive. The time delays correspond to phase shifts in between superimposed waves. The phase shifts vary depending on frequency and signal frequency component may be cancelled or reinforced. This effect is known as frequency selective fading and gives rise to notches in the frequency response of the channel.

Another physical cause of ISI is nonuniformity of frequency response of a channel. Analog low-frequency circuits of the transceiver are composed from reactive elements. These elements (including spurious effects) are bases for channel frequency band limitation. Frequency dependent elements cause nonuniformity of frequency response within audio signal spectrum. When frequency response is explicitly nonuniform within signal spectrum output signal highly differs from input one. Distortions caused by bandlimited low-frequency channel also represent ISI.

From the signal processing point of view the two physically different causes (presence of reactive elements in audio circuits and multipath radio wave propagation) lead to the same final result in the form of ISI.

For watermarking ISI may be simulated by linear filtering with appropriate frequency or impulse response. In Fig. 3 the two above mentioned sources of ISI are incorporated in one block denoted "Linear filtering".

2.2.2 Flat fading

Coming back to multipath propagation, one can analyze a variant when the different path lengths are very similar compared to the wavelengths of the signal components. Then the phase variations between components will be small and they will all undergo very similar amounts of cancellation or reinforcement. This case is usually termed flat fading.

In watermarking flat fading is simulated by amplitude scaling attacks. In Fig. 3 flat fading is shown as multiplicative interference μ .

2.2.3 Additive noise

Additive noise is imposed onto the signal during transmission. The noise results from thermal noise in electronic circuits, from atmospheric noise or from other radio stations. Quantization noise from analog-to-digit converter may be attributed to additive noise. Commonly recognized model of an additive noise is additive white Gaussian noise, denoted in Fig. 3 by n .

2.2.4 Nonlinear distortions

Nonlinear distortions appear in amplitude limitations caused, for example, by the overload in audio circuits. Overload arises from redundant power of transmitting station. The simplest model of nonlinear distortions is clipping. AW in any case should be resistant against such distortions. Source of nonlinear distortion is not shown in Fig. 3.

2.2.5 Desynchronization and resampling

At the transmitter and receiver sampling processes are not synchronized. It means that sampling instants are mutually shifted. Consider sampling frequencies are equal at the transmitter and receiver. At the receiver beginning of watermark is unknown. For watermark restoration beginning of watermark should be first detected and then all decision points are counted from the starting point. Analog radiotelephone channel by all means leads to resampling and loss of the watermark beginning. Desynchronization and resampling attacks are not reflected in Fig. 3.

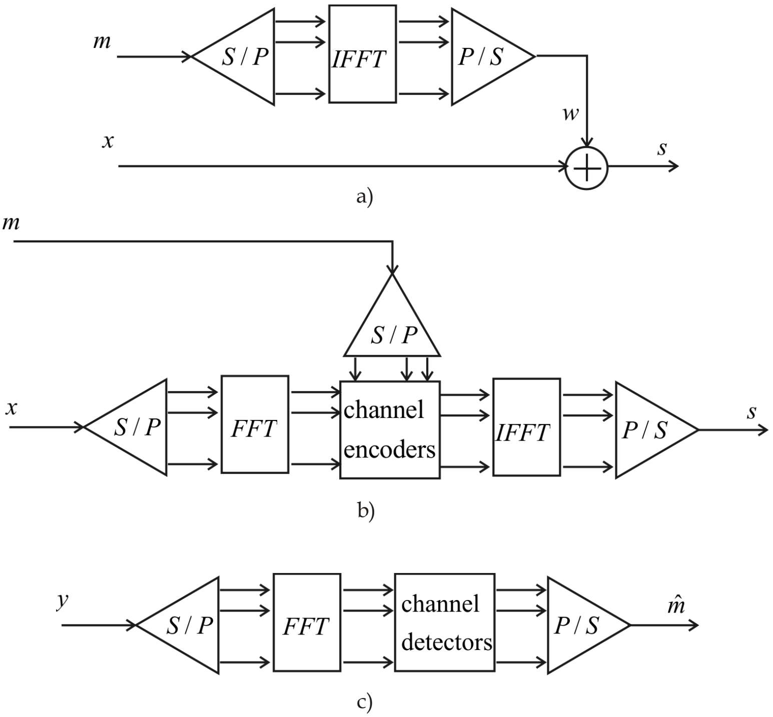


Fig. 4. OFDM application for watermarking: a) noninformed encoding;
b) informed encoding; c) decoding

3. OFDM for watermarking

Orthogonal frequency division multiplexing (OFDM) is well known multi-carrier modulation method and is used in various wireless communication systems (Ipatov, 2005). It has been shown to be an effective technique to combat multipath fading in wireless channels.

3.1 OFDM based watermarking schemes

In basic OFDM scheme data symbols modulate a parallel collection of regularly spaced sub-carriers. OFDM is simple to use on channels with time delay spread or, equivalently, frequency selectivity. OFDM converts one frequency selective channel into a parallel collection of frequency flat sub-channels. Techniques that are appropriate for flat fading channels can then be applied in a straight forward fashion to every sub-channel.

A general system for SS watermarking based on OFDM principle is shown in Fig. 4 a). Sequence of message bits m first is split in serial-to-parallel (S/P) block into some "slow" flows. Relation between a dimension of inverse fast Fourier transform (IFFT) and a number of slow flows depends on what Fourier coefficients are subjected for watermarking. So that some of inputs of IFFT block may be set to zeros. Then parallel-to-serial (P/S) block combines slow flows into one sequence which represents watermark it the time domain. Commonly SS methods use a certain type of transform (discrete cosine transform (DCT), discrete Fourier transform (DFT), Wavelet, etc.) for subsequent watermarking of transformation coefficients.

Application of any transform demands storing of carrier signal and strictly speaking leads to some delay in signal transmission. OFDM scheme produces inherently the same watermarking of fast Fourier transform (FFT) coefficients without any delay. Encoder turns out really noninformed.

General system for informed OFDM encoding is presented in Fig. 4 b). Signal x is splited into slow flows which are subjected to FFT. Message flow m is splited into some, suppose B, $(\mathrm{B} < \mathrm{N} / 2)$ slow flows. Channel encoders use B Fourier coefficients for watermarking independently in each channel. In the simple case one bit may watermark one coefficient. For more complex variants one embedded bit may be distributed among $\mathrm{L} > 1$ coefficients. In general NL samples of x are needed for embedding B message bits. Algorithms for each channel encoders are identical. Watermarked coefficients and all the rest undisturbed coefficients are then used for IFFT. Again P/S block combines slow flows into one sequence s in the time domain.

Watermarked signal s to resist against intersymbol interference may be added with prefix $\mathrm{s_p}$ (see below) so the watermarked signal becomes $[\mathrm{s_p}, \mathrm{s}]$. Prefixed signal presents so called OFDM symbol and is then transmitted through the channel.

At the receiver (Fig. 4 c) OFDM symbol is primarily cleared from the prefix, which is mostly corrupted by ISI. This operation is not shown in the figure. Then signal y is splited into N flows which are transformed in Fourier coefficients. These coefficients are processed according to demodulation algorithm for extracting a watermark message bits $\hat{\mathrm{m}}$. Receiving scheme is general for informed and noninformed encoding.

Attractive features of OFDM-like watermarking are good vectorisation for encoding and decoding algorithms and application of standard FFT and IFFT procedures.

3.2 Resistance to ISI

Let us show how to apply the main principles of OFDM for resistant to ISI watermarking.

According to inverse DFT signal block $\mathrm{x_n}$, $\mathrm{n} = 0,1,\ldots,\mathrm{N}-1$ may be composed from N harmonics with complex amplitudes $\dot{\mathrm{X}}_\mathrm{k}$, $\mathrm{k} = 0,1,\ldots,\mathrm{N}-1$.

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} \dot{X}_k \exp\left(j\frac{2\pi nk}{N}\right), \quad n = 0,1,\ldots,N-1. \tag{3}$$

At the decoder on account of ISI we have harmonics on the same frequency grid, but with another complex amplitudes $\dot{Y}_k \neq \dot{X}_k$, where $\dot{Y}_k$ are obtained from DFT:

$$\dot{Y}_k = \sum_{n=0}^{N-1} y_n \exp\left(-j\frac{2\pi nk}{N}\right), \quad k = 0,1,\ldots,N-1. \tag{4}$$

According to OFDM principle each harmonic should be cyclically extended with P samples. Then we obtain so called OFDM symbol:

$$x'_n = \frac{1}{N}\sum_{k=0}^{N-1} \dot{X}'_k \exp\left(j\frac{2\pi nk}{N}\right), \quad n = 0,1,\ldots,N+P-1\,, \tag{5}$$

where $\dot{X}'_k$ - complex amplitudes undetermined for the present.

It obvious that $x'_{N+i} = x'_i, \quad i = 0,1,\ldots,P-1$.

However we couldn’t insert $N+P$ samples instead of N samples without sampling frequency alteration. We are forced to utilize $N+P$ samples x'_n from Eq. (5) instead of N samples x_n from Eq. (3) plus subsequent P samples from the future block. Prefix results in the host signal distortions and serves as a penalty for ISI decreasing. The length of prefix comes from the channel impulse response. Reasonable P may be accepted to suppress ISI while keeping introduced distortions.

Mathematically equations (5) presents an overdetermined system with $N+P$ equations and N unknowns $\dot{X}'_k$.

For the least squares criteria $\min\|x_n - x'_n\|$ mathematic methods are well designed and implemented in MatLab.

Solution of system (5) is inaccurate in general and consists from complex numbers $\dot{X}'_k$, $k = 0,1,\ldots,N-1$. Therefore samples x'_n may come up to complex values also. Only if $\dot{X}'_k = \dot{X}'_{N-k}{}^*$, $k = 1,\ldots,N/2-1$ and $\dot{X}'_0, \dot{X}'_{N/2}$ are both real, the sequence x'_n is guarantied to be real.

To obtain real values x'_n let us represent system (5) in trigonometric form. Denote $\dot{X}'_k = a_k + jb_k$ in Eq. (5). After evident transformation one may get the system in required form:

$$x'_n = \frac{2}{N}\left(\frac{a_0}{2} + \sum_{k=1}^{N/2} a_k \cos\frac{2\pi}{N}kn - b_k \sin\frac{2\pi}{N}kn\right), \quad n = 0,1,\ldots,N+P-1 \tag{6}$$

As before we have $N+P$ equations and N unknowns: $a_0, \quad (a_k, b_k), k = 1,2,\ldots,N/2-1, \quad a_{N/2}$. Least squared solution will be in real field numbers and sequence the x'_n is assured to be real also.

Coefficients $\dot{X}'_k$, or part of them, are subjected for watermarking. In general we get the watermarked coefficients:

$$\dot{S}_k = \dot{X}'_k + \dot{W}_k, \quad k = 1,2,\ldots,B. \tag{7}$$

In Eq. (7) it is supposed that watermarked coefficients are from 1 to B.

First N samples of watermarked sequence are calculated by means IDFT: $s_n = \mathrm{IDFT}\{\dot{S}\}$, $n = 0,1,\ldots,N-1$ and supplemented with repeated prefix.

Because of ISI received block $y = [y_0, y_1, \ldots, y_{P+N-1}]$ even providing zero additive noise differs from transmitted block. Initial samples are especially corrupted by ISI. For processing decoder removes the first P samples and computes coefficients $\dot{Y}_k = \mathrm{DFT}\{y\}$, utilising the last samples $y = [y_P, y_{P+1}, \ldots, y_{P+N-1}]$, which are free from ISI.

Prefix appending results in matching linear $y = s * h$ and cyclic $\tilde{y} = \tilde{s} * \tilde{h}$ convolutions at interval $\left[P+1, N+P\right]$. Here $\tilde{s}, \tilde{h}$ - are periodic sequences which are formed from the sequences $[s_0, s_1, \ldots, s_{N-1}]$, $[h_0, h_1, \ldots, h_{P-1}, 0, 0, \ldots, 0]$ correspondingly and the second sequence is added by zeros up to N samples.

If $\dot{Y}, \dot{S}, \dot{H}$ - are DFTs of $\tilde{y}, \tilde{s}, \tilde{h}$, then

$$\dot{Y} = \dot{S}\,\dot{H}. \tag{8}$$

In the last Eq. (8) $\dot{H}$ - is the frequency response of general channel.

According to basic idea of OFDM an audio signal can be split into the some of sub carriers on frequencies $f_0, f_1, \ldots, f_{N-1}$. Every sub carrier has its own complex amplitude, say $\dot{X}_i$. The amplitudes vary from slot to slot and are constant within every time slot.

From point of view of watermarking every sub carrier forms separate sub channel, and every channel operates independently from each other. Amplitudes $\dot{X}_i$ are subjected to watermarking. Assume they are altered to $\dot{S}_i$ according certain algorithm. Since sub carriers are orthogonal the amplitudes $\dot{S}_i$ will not have influence on each other. Composed watermarked audio may by presented in the form:

$$\dot{S}(t) = \sum_{i=1}^{M} \dot{S}_i \exp\left(j2\pi f_i t\right). \tag{9}$$

Every sub channel occupies a narrow frequency band. It is reasonable to assume that within one sub channel frequency response of the general channel is constant.

Thanks to prefix ISI is eliminated and received complex amplitude in sub channel will be defined by Eq. (8).

For slow fading it is considered $\dot{H}\left(j2\pi f_i\right) = const$ within one time slot.

4. Quantization Index Modulation (QIM)

Chen & Wornell, 2001 introduced a class of data-hiding codes known as dither modulation codes, or quantization index modulation (QIM) codes. These methods are based on quantization techniques.

4.1 Scalar and complex QIM

The simplest implementation of quantization-based watermarking employs scalar quantizer for embedding one bit into one host sample. The watermarking rule for this case is expressed through the quantization function $s = Q(x)$, where scalar quantization function may be presented in the form:

$$Q(x,m) = \Delta \mathrm{round}\left(\frac{x}{\Delta} + \frac{m}{2}\right) - \Delta\frac{m}{2}. \tag{10}$$

In Eq. (10) denoted: Δ - quantization step, $m = \{0,1\}$ - embedded bit, $\mathrm{round}(\cdot)$ - rounding operation.

Quantization step is chosen depends on distortions-robustness trade-off.

Scalar QIM translates real numbers x into lattices $\Lambda_m = \Delta Z - m\Delta/2$, Z - is set of integer numbers.

The QIM decoder operates as a minimum-distance decoder. It finds the quantizer node closest to y and forms the estimation of extracted bit

$$\hat{m} = \arg\min_{m\in\{0,1\}} \left\| y - Q(y,m) \right\|. \tag{11}$$

Let us introduce quantization on the complex plain and denote it QIM2. Complex quantization function for QIM2 is written in the form:

$$\dot{s} = \tilde{\Delta}\mathrm{round}\left(\frac{\dot{x}}{\tilde{\Delta}} + \frac{m}{2}(1+j)\right) - \tilde{\Delta}\frac{m}{2}(1+j), \tag{12}$$

where $j = \sqrt{-1}$.

Step of quantization $\tilde{\Delta}$ represents in general a complex number. Rounding of complex magnitude is done independently according to real and imagynary axes.

Complex quantization QIM2 is illustrated in Fig. 5. QIM2 uses two-dimensional lattices

$$\Lambda_0 = \tilde{\Delta}\, Z,\ \Lambda_1 = \tilde{\Delta}\,(Z + 1/2), \tag{13}$$

where $Z = \ldots,-2,-1,0,+1,+2,\ldots$ - set of integer numbers.

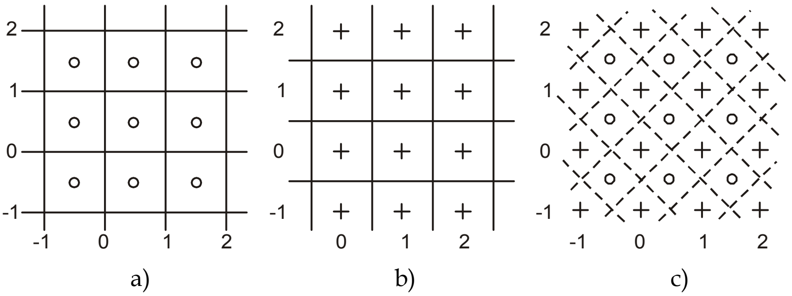


Fig. 5. Lattices on the complex plain: a) lattice for $m = 1$, b) lattice for $m = 0$,
c) partition to decision areas

Advantage of QIM2 over scalar QIM consist in increasing the of distance between the nearest concurring decision points. It is possible to show that, assuming equal distortions the minimal distance is $\sqrt{2}/2$ times greater for QIM2 procedure comparatively to scalar QIM (see Fig. 5).

4.2 Invariance to amplitude scaling and phase shift

The main drawback of QIM is sensitivity to amplitude scaling and filtering.

Perez-Gonzalez et al., 2005 proposed techique for images processing against value-metric scaling attack, named as rational dither modulation (RDM). The main idea of RDM is snaping of quantization steps to watermarked and received signals at the transmitter and receiver correspondingly. For the first order for RDM scheme steps quantization for the current samples X_k and Y_k are produced on the base of previous samples S_{k-1} and Y_{k-1}. Assuming noise absence, the influence of constant multiplier μ in the channel is fully eliminated on decoding process.

This scheme works until $|S_{k-1}| \neq 0$. Performance of the first order RDM scheme may by improved by high order schemes. Stepsize estimation at the transmitter is produced on the base of l previous watermarked samples and is given by the p norm of vector S:

$$g = \left(\frac{1}{l} \sum_{i=1}^{l} |S_{k-i}|^{p} \right)^{\frac{1}{p}}. \tag{14}$$

Analogous procedure is produced at the receiver with vector Y.

Let us extend idea of double stepsize calibration on complex quantization that will be useful for enhancing watermark resistance to filtering. For this purpose apply RDM approach on the case of complex multiplicative interference $\dot{\mu} = \mu \exp(-j\varphi_0)$. This interference adds constant unknown phase shift φ_0. We want to quantize complex amplitudes on a plain lattice invariantly to this shift. For that it is necessary to make QIM process transparent not only to amplitude scaling but to phase alterations also.

Suppose vector $\dot{S}$ presents complex amplitudes of narrowband signal. Passing the channel all amplitudes $\dot{S}$ changes their amplitudes by μ : $|\dot{Y}_i| = \mu |\dot{S}_i|$ and corresponding phases by φ_0 : $\arg(\dot{Y}_i) = \arg(\dot{S}_i) + \phi_0$. Phase of the resulting vector at the transmitter is given by

$$\phi_S = \arg\left(\sum_{i=1}^{l} \dot{S}_i \right). \tag{15}$$

Thanks to narrowband nature phase of resulting vector $\dot{Y}$ at the receiver will be turned on phase ϕ_0 compared to transmitter: $\phi_Y = \phi_S + \phi_0$. Adding phase multiplier $\exp(-j\phi_S)$ in Eq. (14) we get complex estimation for quantization step.

Appropriate equations in vector form are presented below:

$$\mathbf{G}_{k-1} = \mathbf{g}_{k-1} \exp(j\mathbf{\Phi}_{k-1}), \tag{16}$$

$$\mathbf{g}_{k-1} = \left(\frac{1}{l}\sum_{i=1}^{l}\left|\mathbf{S}_{k-i}\right|^{p}\right)^{\frac{1}{p}} \tag{17}$$

$$\boldsymbol{\Phi}_{k-1} = \arg\left(\sum_{i=1}^{l}\mathbf{S}_{k-i}\right) \tag{18}$$

Vector for step quantization in Eq. (12) is then given by the relation

$$\tilde{\Delta}_{k-1} = \Delta\,\mathbf{G}_{k-1}\,. \tag{19}$$

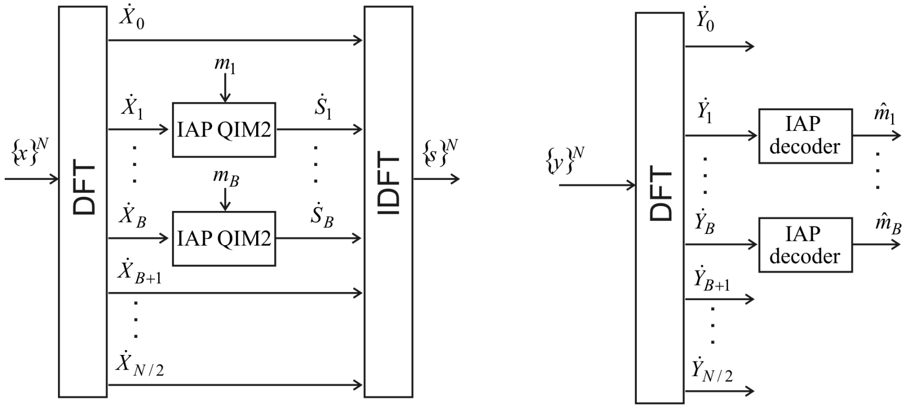
where Δ is certaine stepsize chosen apriory.

The same ralations shold be used at the receiver with the substitution vector $\mathbf{Y}_{k-i}$ instead of $\mathbf{S}_{k-i}$. Note that all bold-marked vectors have length B. At the receiver detection of embedded bits is performed by using a minimum Euclidian distance rule, i.e.

$$\hat{\mathbf{m}} = \underset{\mathbf{m}\in 0,1}{\arg\min}\left\|Q\left(\mathbf{Y}_k,\mathbf{G}_{k-1},\mathbf{m}\right)-\mathbf{Y}_k\right\|. \tag{20}$$

Relations (16) - (20) make possible to eliminate flat fading and realize QIM process fully imperceptible to slowly amplitude scaling and phase shift.

From Eq. (8) it is clear that invariant amplitude scaling and phase shift procedure, abbriviate it as IAP, can be applied to each narrowband channel. Multi channel encoder implementation is presented in Fig. 6.



a) b)

Fig. 6. OFDM invariant to amplitude scaling and phase shift QIM2 system:
a) transmitter; b) receiver

Coming through the channel signal is subjected to linear filtering. Thanks to prefix the received signal after linear filtering may be considered in the frequency domain according to

Eq. (8). At the receiver prefix $\left[y_1, y_2, ..., y_P\right]$ is removed and the remaining samples $\left[y_1, y_2, ..., y_N\right]$ are used for performing DFT. Then decoding IAP procedures are performed under coefficients $\dot{Y}_1, \dot{Y}_2, ..., \dot{Y}_B$ for detecting embedded bits $\hat{m}_i$, $i = 1, ..., B$ in each channel

In Fig. 7 simulation results are presented for filtering attack. For channel simulation Butterworth low pass filter of order 2 an cut off frequency 0.3 was chosen. Function representing that filter in MatLab is: [b,a]=butter(2,.3). Length of impulse response for that filter is 8. Additive noise is absent. Number of watermarked channel is $B = 8$.

It is seen that even for zero prefix received complex amplitudes $\dot{Y}$ are scattered because of ISI influence (Fig. 7 a). Increasing P leads to concentration of $\dot{Y}$ around centroids $\dot{S}$, and for $P = 8$ dispersion is nearly zero: $\dot{Y} = \dot{S}$.

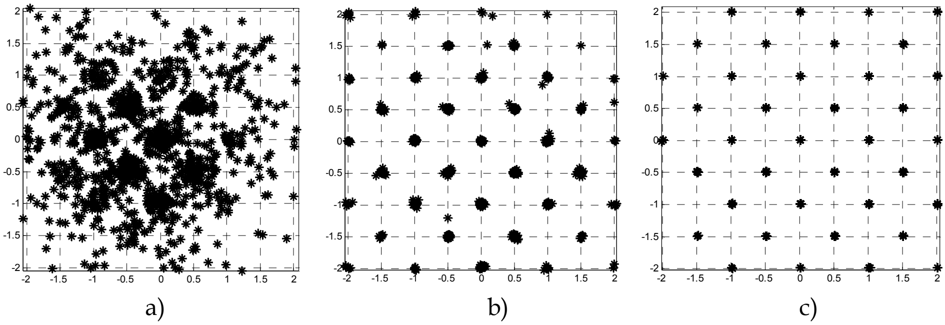


Fig. 7. Influence of prefix on scattering for OFDM IAP QIM2 algorithm:
a) no prefix $P = 0$; b) moderate prefix $P = 4$; c) full prefix $P = 8$

5. Improved Spread Spectrum (ISS)

Malvar & Florencio, 2003 introduced ISS method for robust watermarking. In spite of the title ISS radically distinguishes from common SS due to utilization of information about carrier signal. When compared with traditional SS, the signal doesn't act as noise source. Thanks to that ISS encoding algorithm refers to informed encoding and may be treated as binary QIM. The main idea of ISS is to look ahead across the carrier signal and relying on it generate the appropriate watermark signal. Just as for standard SS chip sequence $\mathbf{u}$ is used but with the coefficient $\mu(\tilde{x}, b)$:

$$\mathbf{s} = \mathbf{x} + \mu(\tilde{x}, m)\mathbf{u}, \tag{21}$$

where $\tilde{x} = (\mathbf{x}, \mathbf{u})$, $m = \{-1, 1\}$ - embedded bit.

Here inner product is defined as:

$$(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{L} x_i u_i . \tag{22}$$

The main idea of ISS is to generate watermark vector $\mathbf{w} = \mu(\tilde{x}, m)\mathbf{u}$ in such mode that inner product $(\mathbf{s}, \mathbf{u})$ would give the value not less than certain threshold ρ if $m = 1$ and not more than $-\rho$ if $m=-1$. It is obvious that there are situations when inserting of watermark signal is not needed at all. In this case distortions due to watermarking are absent. When do the modification of carrier signal is necessary, total correction expressed by the relation $\tilde{w}=\rho m-\tilde{x}$ is distributed along the carrier signal.

Taking into account that after embedding process according to Eq. (21) the watermarked amplitude may take negative value, the following algorithm for calculating $\mathbf{w}$ was applied:

$$\mathbf{w}=\begin{cases}\tilde{w}\mathbf{u}^{-}/\left\|\mathbf{u}^{-}\right\|, & \text{if} \quad \tilde{w}\geq 0,\\ \tilde{w}\mathbf{u}^{+}/\left\|\mathbf{u}^{+}\right\|, & \text{otherwise}\end{cases}. \tag{23}$$

where symbol $\|\cdot\|$ denotes unit vector norm.

Sequences $\mathbf{u}^{+}=\{0,1\}$ and $\mathbf{u}^{-}=\{0,-1\}$ are composed from sequence $\mathbf{u}$ according the following rules:

$$u_i^{+}=\begin{cases}1, & \text{if} \quad u_i=1,\\ 0, & \text{otherwise}\end{cases} \quad u_i^{-}=\begin{cases}-1, & \text{if} \quad u_i=-1,\\ 0, & \text{otherwise}\end{cases} \tag{24}$$

Eq. (23), (24) eliminate in any case a negative amplitude values for watermarked signal according Eq. (21).

Detected bit is restored by means of sign function

$$\hat{m}=\operatorname{sign}(\tilde{y}) \tag{25}$$

where, $\tilde{y}=(\mathbf{y},\mathbf{u})$.

Comparative detection characteristics for ISS and SS according Shishkin, June 2008 are presented in Fig. 8. Characteristics are plotted as error probability function versus watermark-to-signal ratio (WSR) and watermark-to-noise ratio (WNR) both expressed in dB. Graphics are plotted for identical parameters signal-to-noise ratio (SNR) and WSR accordingly. Teoretical bounds, when host signal is completely availablee at the receiver, i.e. nonblind watermarking are shown with dashed lines.

Almost vertical slope of $p_{er}(\mathrm{WSR},\mathrm{SNR}=\mathrm{const})$ is explained by the independency of watermarking channel capacity from the carrier signal (see Eq. (2)). Gain for ISS method is within (15 … 20) dB in comparison to SS Fig. 8 a)).

It is worth to notice that OFDM-like processing is completely applicable to ISS. Characteristics in Fig. 8 are given for zero prefix.

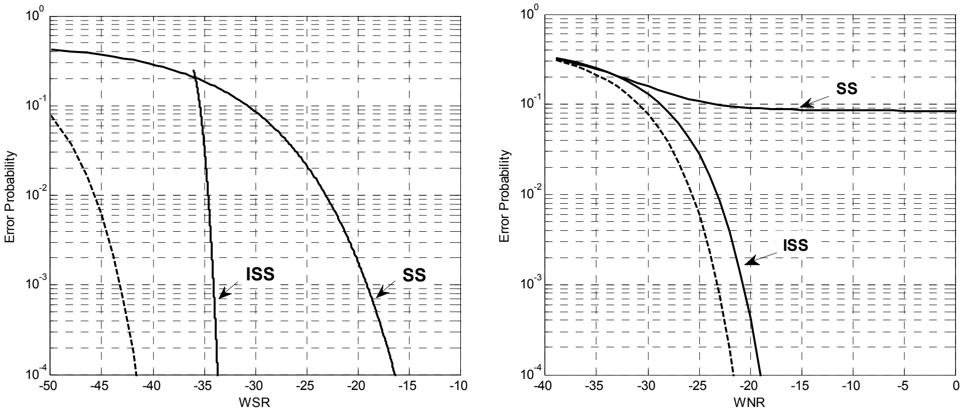
6. Adaptive cancelling of carrier signal at the receiver

SS-based watermarking algorithm embeds one bit of information in a vector $\mathbf{s}$ of L samples:

$$\mathbf{s}=\mathbf{x}+\sigma_{w}\, m\,\mathbf{u}\ , \tag{26}$$

where σ_w is wanted root mean square deviation of watermark, $m = \{-1, 1\}$ - information bit to be embedded, $\mathbf{u} = [u_1, u_2, ..., u_L]$ - binary pseudo random sequence, $u_i = \{-1, 1\}$.

If the channel modeled as additive noise, the received signal is: $\mathbf{y} = \mathbf{s} + \mathbf{n}$.



a) b)

Fig. 8. Comparative error probability functions for ISS and SS methods: a) p_{er} versus WSR, $SNR = 20$ dB, b) p_{er} versus WNR, $WSR = -30$ dB

Detection at the receiver is performed by the correlator that computes inner product

$$\tilde{y} = (\mathbf{y}, \mathbf{u}) = (\mathbf{x}, \mathbf{u}) + \sigma_w mL + (\mathbf{n}, \mathbf{u}). \tag{27}$$

Detected bit is estimated according to Eq. (25). One can see that interferences for watermark are terms $\tilde{x} = (\mathbf{x}, \mathbf{u})$ and $\tilde{n} = (\mathbf{n}, \mathbf{u})$ in Eq. (27). Suppose vectors $\mathbf{x}$ and $\mathbf{n}$ are uncorrelated random processes. Therefore terms in Eq. (27) $\tilde{x}$, $\tilde{n}$ grows proportionally $\sqrt{L}$, meanwhile useful term $\sigma_w mL$ is proportional to L. Processing gain is $\sqrt{L}$. It is possible to yield desired probability of detection through the appropriate chip length L.

Another possibility for achieving the goal appears when $\mathbf{x}$ represents a correlated process. Practically x is a really highly correlated process. And just this process has a maximal influence on watermark.

In communication optimal receiving algorithms in the presence of correlated noise are well developed (Van Trees, 1968). Scheme for optimal receiver on the background correlated noise is based on whitening filter (WF) and presented in Fig. 9. WF is based on predictor and forms at the output an error signal $e(i)$ between an actual sample and predicted one. For good predictor $e(i)$ looks like white noise with less power compared to power of input signal $y(i)$.

Verbal signal is a nonstationary random process and it requires adaptation of transfer function. For WF implementation let us apply linear prediction method. Basic principle of linear prediction consists in presentation of predicting sample through the linear combination of previous samples. Weighting coefficients in the linear combination are calculated on the basis of mean squared error minimization for prediction, i.e. differences between signal samples and theirs predicted values.

Linear prediction method in the view of so called linear predictive coding (LPC) is widely used in the audio compressing algorithms. LPC algorithm assumes partition of audio signal on frames of duration approximately 20 msec. For every segment weighting coefficients of WF, which would minimize mean squared error of prediction are calculated. Prediction at step i is expressed through the preceding samples:

$$y_{pr}(i) = h_1 y(i-1) + h_2 y(i-2) + \ldots + h_p y(i-p) . \tag{28}$$

Prediction error comes to

$$e(i) = y(i) - y_{pr}(i) = \sum_{k=0}^{p} h_k y(i-k) . \tag{29}$$

Coefficients h_k, $k = 1,2,\ldots,p$ in Eq. (29) are subject of adaptation, $h_0 = 1$.

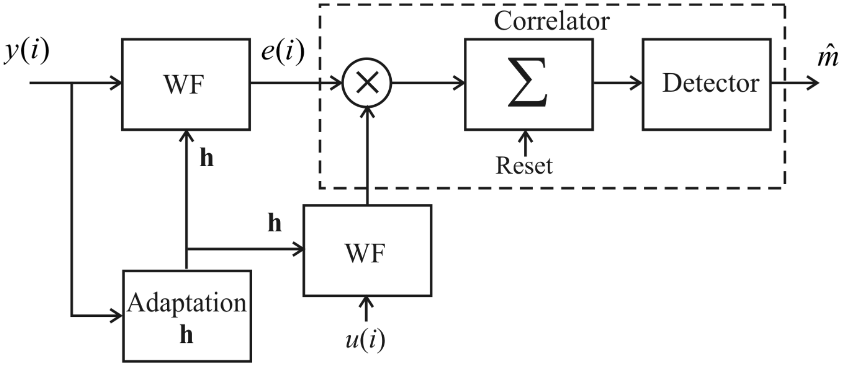


Fig. 9. Optimal receiver for watermark on the background of correlated noise

Algorithm for finding coefficients h_i is well elaborated mathematically. It is based on Yule-Walker equation and computing procedure by the Levinson-Durbin algorithm (O'Shaughnessy, 2000). In MatLab function lpc(y,p) finds the coefficients of a p -order linear predictor (finite impulse response filter) that predicts the current value of the real-valued time series y based on past samples.

LPC algorithm is block and applies the samples from fixed time interval. During filtration coefficients of WF doesn't vary within the frame. Coming to next frame the coefficients should be recalculated. Other prediction procedures may use continuous adaptation algorithms, for example, Least Mean Square (LMS) or Recursive Least Square (RLS) algorithm. Shishkin, October 2008 simulated above mentioned algorithms for SS-watermarked speech signals (Fig. 10 a)). In Fig. 10 a) time axes are marked in sample numbers, assuming sampling frequency $F_s = 22050\text{ Hz}$, WF order $p = 7$.

Waveforms shows that prediction error is less for LPC algorithm compared to LMS and RLS. Time segments for samples 1 – 2000, 2000 – 4000 and 8000 – 10000 correspond to vowel sounds and have quite negligible prediction error because appropriate signal is highly correlated. On the other hand consonant speech sound (samples 4000 – 6000) is closer to white noise and that's why it is predicted worse.

Comparative detection characteristic for SS watermarking in AWGN channel are presented in Fig. 10 b). Watermarking was executed in the time domain. It is seen that additional processing at the receiver effectively suppress correlated speech signal. Processing gain makes approximately 15 dB.

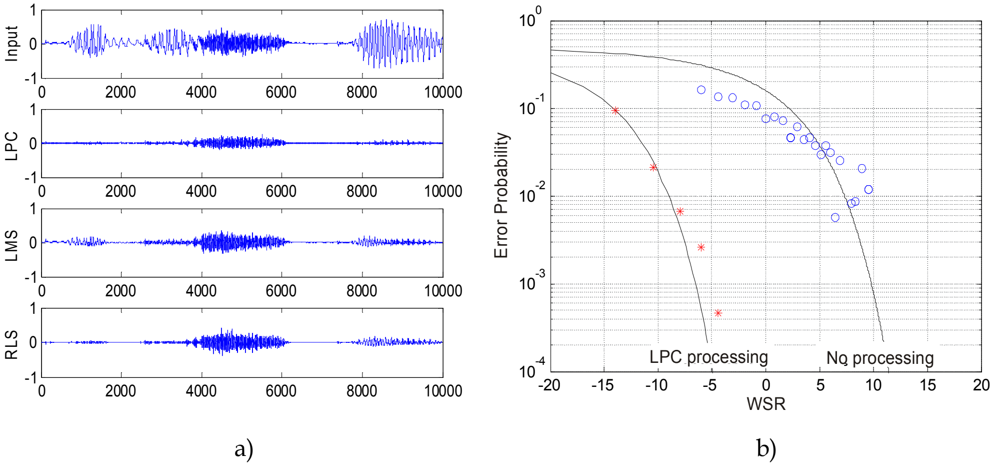


Fig. 10. Simulation results for adaptive carrier signal cancellation: a) signal waveforms after whitening according LPC, LMS, RLS algorithms; b) error probability as a function of WSR (Shishkin, October 2008)

7. Conclusion

Proposed OFDM technology application to QIM watermarking makes possible to resist against intersymbol interference, that is induced by multipath propagation and bandlimited nature of VHF radiotelephone channel. OFDM-QIM integration makes QIM watermarked signal to be invariant against amplitude scaling and phase shift. Reasonable prefix length about $P = 2...4$ to cancel ISI is acceptable for the total OFDM symbol size about 500 samples. The main restricted factor is reliable estimation for step quantization on the background of additive noise. One of the possible ways to resist against noise is one bit distribution on numerous samples, i.e. application of the principle one bit – many samples (or coefficients) instead of principle one bit – one sample (coefficient).

Improved SS watermarking inherently resist to amplitude scaling. Noise immunity may be achieved by means of exchange embedding rate and noise robustness. For existing radiotelephone channel of frequency band (300 … 3000) Hz and signal-to-noise ratio about 15 dB realistic embedding rate forms about 60 bit/sec. This rate is quite sufficient for reliable identification. Presently ISS watermarking appears to be the most acceptable method.

Traditional SS method with additional processing at the receiver gives about 40 bit/sec rate. Absolute advantage of this method is absence of delay and simple encoding at the transmitter. OFDM-like pre-processing procedure makes possible to watermark in the frequency domain without any delay for transmitting signal.

8. References

Brehaut, D. (2009). *GMDSS: A User's Handbook* (Forth edition), Adlard Coles Nautical, ISBN 978-1408114933, London

Chen, B. & Wornell, G. (2001). Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, Vol. 47, No. 4, pp. 1423-1443

Costa, M. (1983). Writing on dirty paper. *IEEE Transactions on Information Theory*, Vol. IT-29, pp. 439 – 441

Cox, I.; Miller, M.; Bloom, J.; Fridrich, J. & Kalker, T. (2008). *Digital Watermarking and Steganography*, (Second edition), Morgan Kaufmann Publishers, ISBN 978-0-12-372585-1, Burlington, MA, USA

Gel'fand, S. & Pinsker, M. (1980). Coding for channel with random parameters. *Problems of Control and Information Theory*, Vol. 9, No.1, pp. 19 – 31

Hofbauer, K. & Kubin, G. (2006). Aeronautical voice radio channel modelling and simulation—a tutorial review, In: *Proceedings of the International Conference on Research in Air Transportation (ICRAT)*, Belgrade, Serbia, Available from http://www3.spsc.tugraz.at/people/hofbauer/papers/ Hofbauer_ICRAT_2006.pdf

Hofbauer, K. et al. (2009). Speech watermarking for analog flat-fading bandpass channels, *IEEE Transactions on Audio, Speech, and Language Processing*, 2009, Vol.17, No.8, pp. 1624 - 1637, ISSN 1558-7916

Ipatov, V. (2005). *Spread Spectrum and CDMA: Principles and Applications,* John Wiley & Sons, Ltd, ISBN 0-470-09178-9, Chichester, England

Kuznetsov, A. & Tsybakov, B. (1974). Coding in a Memory with Random Parameters. *Probl. Peredachi Inf,* Vol. 10, No.2, pp. 52-60, UDC 621.391.15

Malvar, H. & Florencio, D. (2003). Improved Spread Spectrum: A New Modulation Technique for Robast Watermarking, *IEEE Transactions on Signal Processing*, Vol.51, No.4, pp.898 – 905

Moulin, P. & Koetter, R. (2005). Data-Hiding Codes. *Proceedings of the IEEE*, Vol.93, No.12, pp. 2083 – 2126

O'Shaughnessy, D. (2000). *Speech communication: human and machine,* (Second edition), IEEE, Inc. New York, ISBN 0-7803-3449-3

Perez-Gonzalez, F. et al. (2005). Rational Dither Modulation: A High Rate Data-Hiding Method Invariant to Gain Attacks. *IEEE Transactions on Signal Processing*, Vol.53, No.10, pp.3960–3975

Shishkin, A. (June 2008). Digital Watermarks with Spectrum Spreading for Audio Signals Using the Signal Carrier Information. *Radioelectronics and Communication Systems,* Vol. 51, No.6, pp.308-315, ISSN 0735-2727

Shishkin, A. (October 2008). Adaptive Algorithms Application in Sound Steganographic Systems with Signal Spectrum Broadening. *Radioelectronics and Communication Systems,* Vol.51, No.10, pp.524-530, ISSN 0735-2727

Van Trees, H. (1968). *Detection, Estimation, and Modulation Theory*, (First edition), John Wiley & Sons Inc, ISBN 978-0471899556

10

Robust Watermarking Framework for High Dynamic Range Images Against Tone-Mapping Attacks

Jiunn-Lin Wu
Dept. of Computer Science and Engineering
National Chung Hsing University, Taichung
Taiwan

1. Introduction

1.1 Background explanation

As digital cameras become more and more popular recently, it is very easy for us to take many digital photos. Unfortunately, they are rarely true measurements of relative radiance in the scene due to the limited dynamic range in the image acquisition devices. High dynamic range (HDR) images emphasis in image processing fields because they can accommodate a greater dynamic range of luminance between the brightest and darkest parts of an image. Dynamic range is the ratio between the brightest and darkest luminance values of a scene. In general, human eyes can handle a very large dynamic range of approximately 100000:1 in a single view. However, a standard photo taken with a standard camera with film or an electronic imaging array always has a limited dynamic range [1]. A standard image, called a LDR image, cannot reproduce the luminance ratio observed in the real world. A scene containing very bright highlights and deep shadows always loses some detail if the exposure time is not suitably determined. Over the past decade, many researchers have developed HDR imaging techniques (Debevec & Malik, 1997)(Reinhard *et al*, 2005) (Reinhard *et al*, 2007). Debevec and Malik proposed a method to recover the single high dynamic range radiance map from multiple images with different exposure times (Debevec & Malik, 1997), this method has been implemented in many HDR software.

The reconstruction of a high dynamic range image is a complex process. Producing an HDR image is by capturing multiple images of the same scene with different exposure levels and merging them into a single HDR image (Debevec & Malik, 1997). The photographers can use a tripod in order to capture the same scene and avoid image registration problems. However, if the differently exposed image sequences are took hand-held, an image registration method which is robust to the illumination changes and moving objects should be used to align the multiple input images before HDR image composition. In addition, the user must be able to use the exposure bracketing technique to ensure the pictures are properly exposed.

1.2 Research motivation

Obviously, it is quite an achievement to create a high dynamic range image containing pixel values that span the whole tonal range of real world scenes. It takes efforts not only to capture the differently exposed photographs as input, but also to reconstruct high dynamic range image by the techniques of image registration and image composition. The copyright protection for HDR images has become increasingly important.

Image watermarking is a common method of proving ownership or determining origin (Tsang & Au, 2001). Unintentionally destroyed watermarks happen when transmitting an image. Since pirates may also seek to remove the watermark or make it undetectable, the watermark must be robust to common attacks. Some of the common problems include noising, blurring, cropping and geometric distortions. Several watermarking schemes embed the watermark in the transformed domain (Piva *et al*, 1997)(Barni *et al*, 1999)(Wang *et al*, 2002) (Suhail. & Obaidat, 2003), which is robust to common image processing attacks, such as low-pass filtering or JPEG compression. However, the watermarking methods in the literature paid attention on the conventional LDR images, they can not be applied to the high dynamic range images.

1.3 The purpose of research

The challenge of the watermarking techniques for high dynamic range images is the tone mapping operators in which we usually use them to convert a high dynamic range image to the conventional low dynamic range image. Tone mapping is necessary for rendering an HDR image on low dynamic range devices such as standard screens or printers. This chapter presents a new watermarking framework for HDR images to alleviate the problem of tone mapping distortions. To demonstrate the powerfulness of the proposed method, a simple DCT-based watermarking technique for conventional LDR images is used. We embed the watermarking in the middle frequency DCT coefficients of the tone-mapped LDR image, the ratio image is then multiplied to recover the HDR values, where the ratio image is computed by dividing the original HDR image at each pixel by the tone-mapped luminance. Experimental results shows the watermarked HDR image keeps high visual quality and the embedded watermark using the proposed technique is robust to varying degree to tone mapping distortions, low-pass filtering, noise contamination and cropping.

2. HDR watermarking technique for HDR images

This section presents an efficient and robust watermarking algorithm for high dynamic range images. Using this blind watermarking algorithm, the watermark extraction is without the original image. The most common process for HDR images is tone mapping. Tone mapping HDR images to LDR images reveals highlights and shadow details on standard LDR devices. The aim of the proposed method is to develop a watermarking scheme against the process of tone mapping.

2.1 Watermark embedding

The key idea of the proposed method is triggered by a sub-band encoding algorithm for high dynamic range images (Ward & Simmons, (2004), the new lossy HDR high dynamic range image format is backwards compatible with existing JPEG software. A tone-mapped version of HDR original image is accompanied by restorative information in the standard 24-bit RGB format. This sub-band in JPEG format contains a compressed ratio image, which can be used to recovers the original high dynamic range image by multiplying the tone-mapped foreground by the ratio image. Figure 1 illustrates the flowchart of the proposed watermark embedding framework for high dynamic range images.

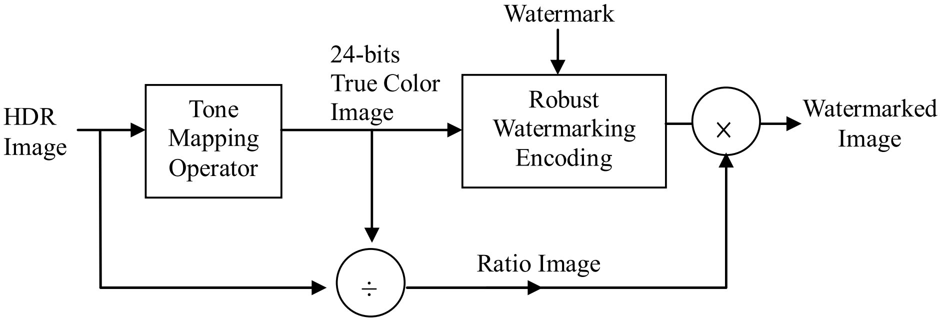


Fig. 1. Illustration of the proposed watermark embedding process for high dynamic range images.

A tone-mapped LDR image is first generated by a tone-mapping operator from the original high dynamic range image. A ratio image is then obtained by dividing the HDR image by the tone-mapped LDR image. Any conventional watermarking technique for LDR images can be applied to embed the watermark bits into the tone-mapped LDR image. Finally, by multiplying the watermarked image by the ratio image, the HDR image with watermark is then obtained. Due to the watermark is embedded in the tone mapped LDR image, the proposed HDR watermarking scheme is robust against the tone mapping attacks.

Discrete cosine transform (DCT) is widely used in signal and image processing for lossy data compression. To demonstrate the powerfulness of the proposed HDR image watermarking scheme, a simple DCT-based watermarking method is adopted to embed the watermark into the tone-mapped LDR image. We transformed the LDR image into the frequency domain and embedded the watermark into the lower middle-frequency blocks, for example the coefficient $DCT_{3,3}$ of each 8×8 image block. We used a neighboring difference-based method to embed the watermark as shown in Table 1, where X denotes a block $DCT_{3,3}$, Y denotes its neighboring block $DCT_{2,4}$, and X' denotes the X block after the bit is embedded. If the watermark bit is 1, X' is set to be larger than or equal to Y . If the bit is 0, X' is set to be smaller than Y . After embedding the watermark, the watermarked LDR image is obtained by inversely transforming from the modified DCT image. It is obvious that the modification is invisible and the watermark is robust to various common attacks, such as blurring and noising and cropping. Finally multiplying the watermarked LDR image by the ratio image produces the watermarked high dynamic range image.

To improve the security to the proposed watermarking technique, the index table of the watermark is disturbed by a 1-D binary pseudo-random sequence using the private key as seed. The encrypted watermark is then embedded into the tone mapped LDR image and yields the watermarked HDR image using the procedure shown in Fig. 1.

W = 0		W = 1	
if $X \ge Y$	$X' = Y - diff$	if $X \ge Y$	$X' = X + diff$
if $X < Y$	$X' = X - diff$	if $X < Y$	$X' = Y + diff$

Table 1. DCT-based watermark embedding algorithm.

2.2 Watermark extraction

Figure 2 shows the flowchart for extracting the watermark. We first use the tone mapping operator to convert the corrupted watermarked HDR image to the watermarked LDR image. The watermark detection method for conventional LDR image is then used to blindly extract the watermark bits without the original HDR image. The watermark is decrypted by the secrete private key.

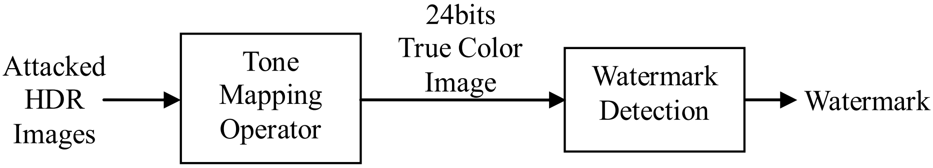


Fig. 2. The flowchart of the HDR image watermark retrieving procedure.

3. Experimental results

Several HDR images are used in the experiments to verify the proposed HDR image watermarking algorithm. The watermark used in this chapter is the 80×60 logo watermark as shown in Fig. 3. We applied several common signal processing attacks to the watermarked HDR images to evaluate the proposed watermarking scheme.

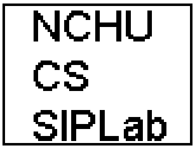


Fig. 3. The logo watermark used in the experiments.

To quantify the robustness of the proposed algorithm, we computed the value of the normalized correlation (NC) coefficient between the original watermark and the extracted one to measure the quality of the retrieved watermark bits, the formula is shown below:

$$\rho(W,W') = \frac{\sum_{m=1}^{N_w} w_m w'_m}{\sqrt{\sum_{m=1}^{N_w} w_m^2 \sum_{m} w'^2_m}} , \qquad (1)$$

where W and W' denote the embedded watermark and the extracted watermark respectively and N_W is the length of the watermark. The coefficient is bounded by $-1 \le \rho(W, W') \le 1$. Since the watermark is a binary sequence of ± 1, we have

$$\sum_{m=1}^{N_w} w_m^2 = \sum_{m=1}^{N_w} (w'_m)^2 = N_w \tag{2}$$

The normalized correlation coefficient can also be written as

$$\rho(W, W') = \frac{\sum_m w_m w'_m}{N_w} \tag{3}$$

A larger ρ indicates a better retrieval performance.

Figure 4 shows a high dynamic range image –hot spring, which is recovered from six LDR photographs with different exposure time, the image size is 640×480. It is used for experiments to prove the robustness of the proposed method against common signal processing attacks on the watermarked HDR image. We first converted the original HDR image to its corresponding LDR image using a tone mapping operation and then computed the ratio image. The dynamic range compression algorithm based on fast bilateral filtering (Durand & Dorsey, 2002) is used as the tone mapping operator in the watermark embedding procedure. A simple DCT- based watermarking method is then used to embed the logo watermark shown in Fig. 3 into the tone mapped LDR image.

Figure 4 shows a high dynamic range image –hot spring, which is recovered from six LDR photographs with different exposure time, the image size is 640×480. It is used for experiments to prove the robustness of the proposed method against common signal processing attacks on the watermarked HDR image. We first converted the original HDR image to its corresponding LDR image using a tone mapping operation and then computed the ratio image. The dynamic range compression algorithm based on fast bilateral filtering [10] is used as the tone mapping operator in the watermark embedding procedure. A simple DCT- based watermarking method is then used to embed the logo watermark shown in Fig. 3 into the tone mapped LDR image.

Figure 5(a) depicts the tone mapped LDR image, and the watermarked LDR image is shown in Fig. 5(b). This figure shows that the two images are visually indistinguishable and the peak signal-to-noise ratio (PSNR) value between them is 41.48 dB. Finally the watermarked HDR image is produced by multiplying the watermarked LDR image by the ratio image.

Table 2 shows the result of the watermarked high dynamic range image corrupted by cropping, blurring and noising attacks. The normalized correlation coefficient between the extracted and original watermark are all higher than 0.3, and the extracted watermarks are distinguishable. It shows the effectiveness of the proposed HDR watermarking algorithm.

Mean square error (MSE) is not a good performance index to measure the difference between the original high dynamic range image and the watermarked one, it is because that the intensity range of the high dynamic range radiance map recovered by different approaches or programs are quite varying. In order to provide a fair measurement to the

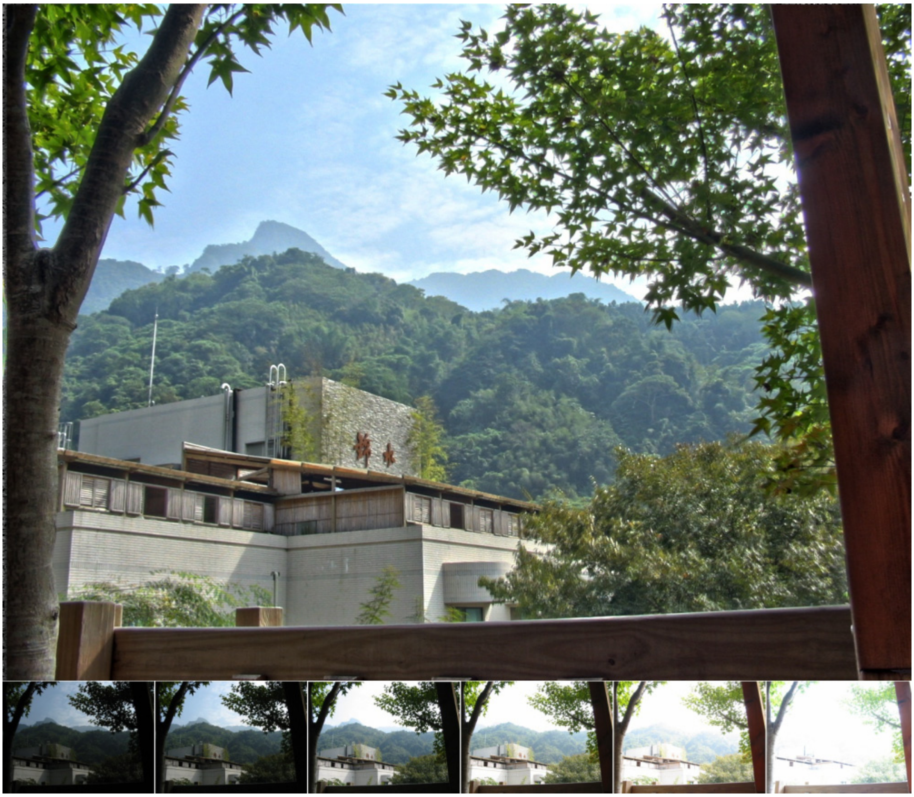
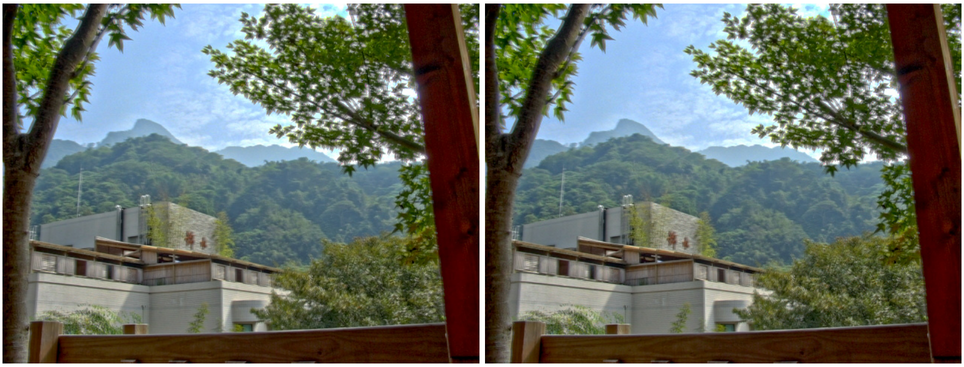


Fig. 4. A high dynamic range image, hot spring, reconstructed from six differently exposed photographs.



(a) (b)

Fig. 5. Tone mapped LDR image and its watermarked version. (a) The tone mapped LDR image using the bilateral-filter based algorithm. (b) The watermarked LDR image by the DCT-based watermarking method.

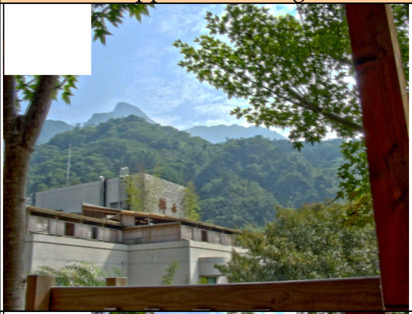
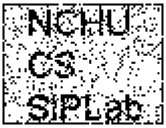
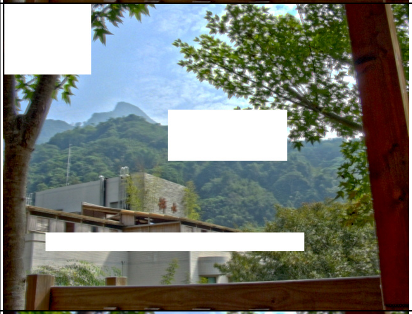
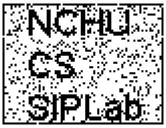
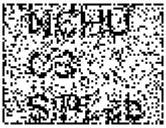
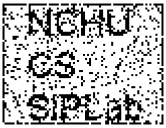
Distortion attack	Tone-mapped LDR image	Extracted watermark
Cropping 1		 ρ= 0.78
Cropping 2		 ρ= 0.75
Gaussian blur (radius=0.8)		 ρ= 0.55
Addaptive Gaussian noise (variance=9)		 ρ= 0.71

Table 2. The extracted watermark and the correlation coefficient ρ upon common signal processing attacks – cropping, blurring and noising.

quality of the watermarked high dynamic range image, we propose to normalize the original high dynamic range image to the range $[0, 255]$ in the experiments, the peak signal-to-noise ratio (PSNR) is then calculated to measure the distortion between the original HDR image and the watermarked one, the formula is shown in Eqs. (4) and (5).

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right) \text{ dB} \tag{4}$$

and

$$MSE = \frac{1}{3WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{c\in\{R,G,B\}}\left(O_c(x,y) - I_c(x,y)\right)^2 \tag{5}$$

where W and H are the total number of pixels in the horizontal and the vertical dimensions of the images; $O_c(x,y)$ and $I_c(x,y)$ denote the pixels of the original and watermarked image respectively. According to our experience, the distortion for the high dynamic range images is invisible if the PSNR of the normalized HDR image is higher than 55 dB.

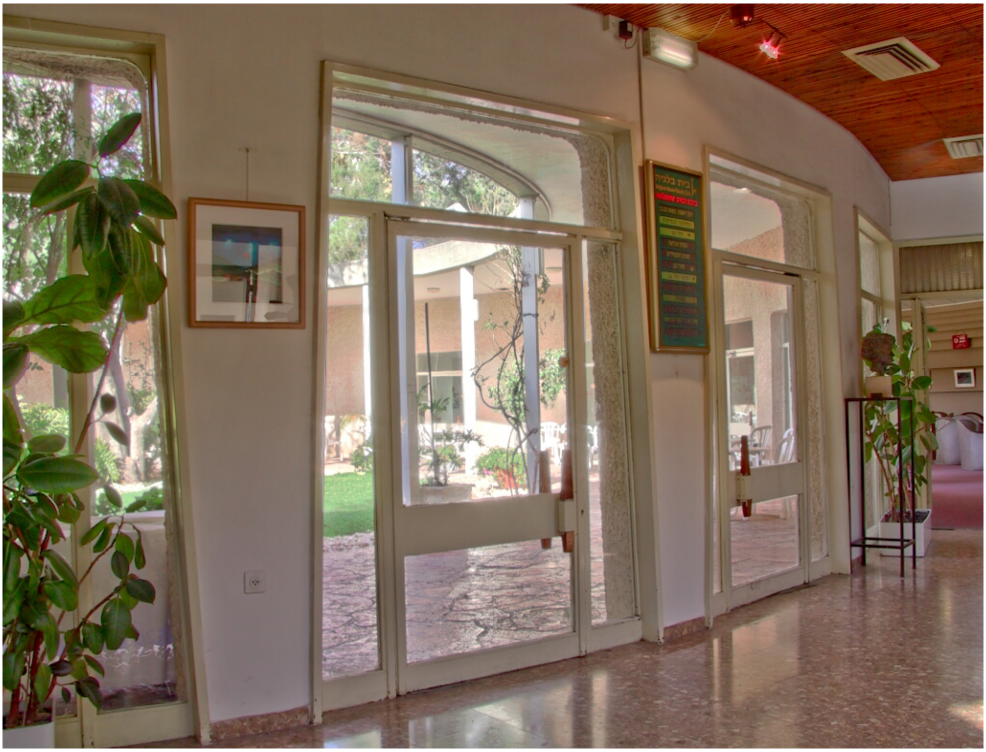


Fig. 6. A high dynamic range image- Belgium house (Fattal, *et al.*, 2002), the image size is 1024×760 .

The algorithms that preparing HDR images for display on LDR devices are called tone mapping operators or tone reproduction. Three famous and popular tone mapping algorithms include the fast bilateral filter approach proposed by Durand and Dorsey [10], photographic method by Reihard. (Reihard. *et al.*, 2002), and the gradient domain compression (GDC) algorithm by Fattal. (Fattal *et al.*, 2002). Just as described in previous section, tone mapping is the most often used attack for high dynamic range images. To evaluate of the robustness of the proposed method against the tone mapping attacks, all three approaches mentioned above are used to test the proposed watermarking algorithm. In this experiment, a high dynamic range image - Belgium house as shown in Fig. 6 is used , which is obtained from the work of Fattal (Fattal *et al.*, 2002), and its size is 1024 by 760.

The PSNR of the watermarked HDR image using the bilateral filtering method [10], photographic method (Reihard. *et al.*, 2002), and the gradient domain compression (GDC) algorithm in the watermark embedding procedure are 77.58, 77.36 and 73.00 respectively. We observed that the watermarked high dynamic range image by using the GDC as the tone mapping operator in the watermark embedding step produced higher distortion compared two other tome mapping approaches. However, the visual quality is still satisfied and the PSNR is much higher than 55dB. Table 3 shows the performance comparison of the different tone mapping operators are used in the watermark embedding and retrieving procedures. It is worthy to notice that the watermarked HDR image using GDC in the watermark embedding procedure perform best against the various tone mapping attacks.

Table 4 shows the robustness comparison of the watermarked HDR image when different tone mapping methods are used in the watermark embedding procedure. Two common signal processing attacks- blurring and cropping are corrupted on the watermarked HDR image. It shows the watermarking method used GDC method achieves the highest normalized correlation coefficient. In the following experiments, we adopt GDC as the tone mapping operator in the proposed HDR watermark embedding procedure.

Finally, two another high dynamic range images obtained from Debevec's work (Debevec & Malik, 1997) are used to verify the proposed watermarking method, as shown in Fig. 7. Table 5 and 6 show the results, they demosntrate the efffectivenss and robust ness of the proposed method.

4. Conclusion

Researching the watermarking scheme for high dynamic range images is an important task in image processing and computational photography fields. This chapter presents a new blind watermarking algorithm for HDR images. It achieves the robustness by embedding the watermark bits into a tone mapped version of the original HDR image. Experimental results show that the proposed algorithm is robust against various tone mapping operations, which is an inherent problem in watermarking HDR images. A simple DCT-based watermarking method for the derived tone-mapped LDR image is used in the watermark embedding procedure, it protects the watermarked HDR image from several common signal processing attacks, including noising, blurring and cropping. In the future work, the geometric attack invariant features will be put into analysis to enhance the robustness. Meanwhile, the capacity and fidelity are also taken into account.

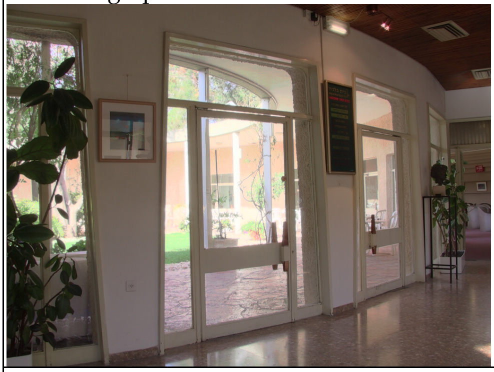
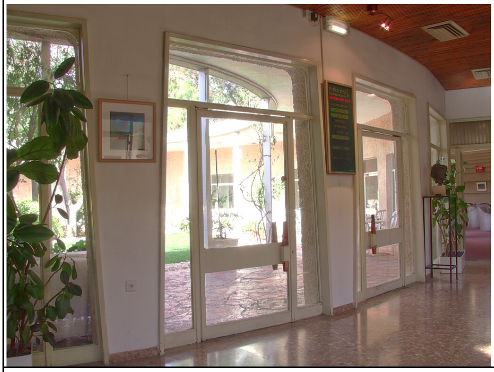
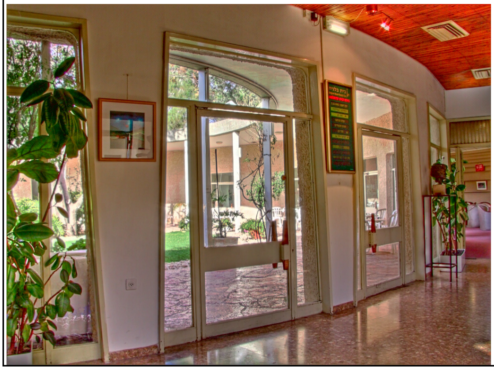
Embedding Retrieving	Photographic	Bilateral	GDC
Photographic 	0.87	0.82	0.90
Bilateral 	0.83	0.83	0.87
GDC 	0.88	0.87	0.92

Table 3. The comparison of the different tone mapping operators are used in the HDR watermark embedding and retrieving procedures.

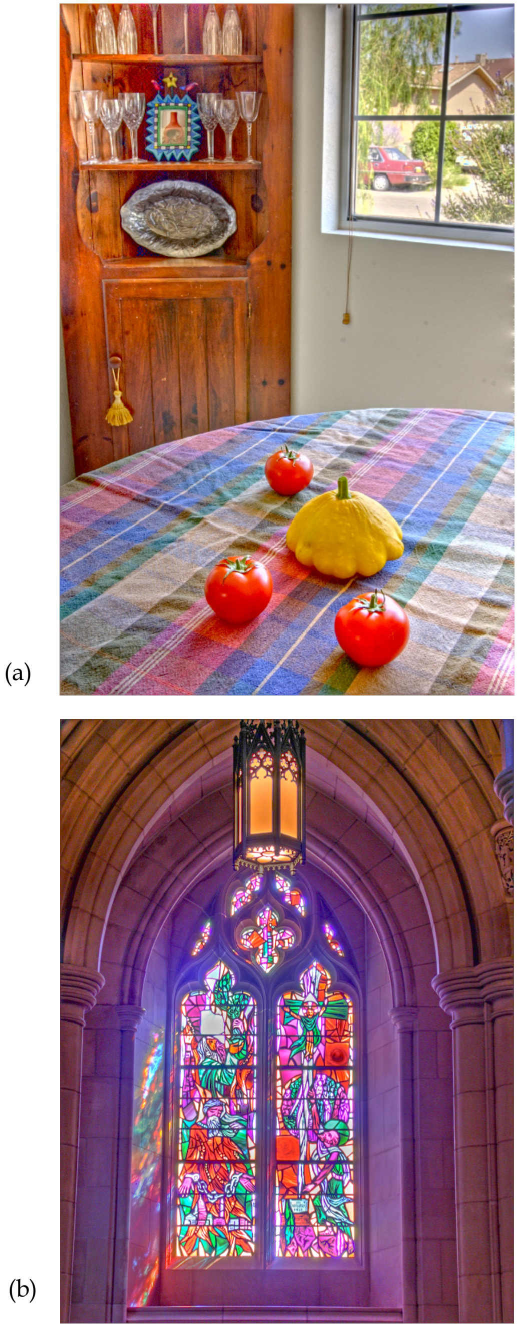


Fig. 7. Two high dynamic range images (Debevec & Malik, 1997). (a) indoor; (b) church window.

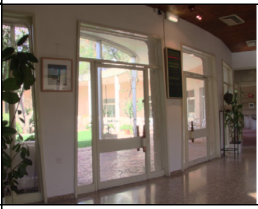
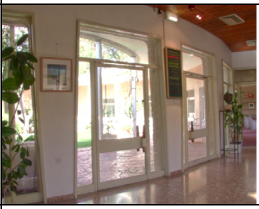
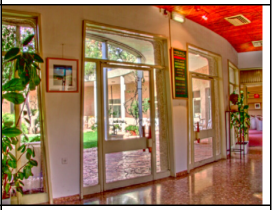
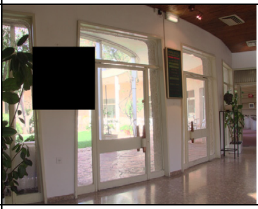
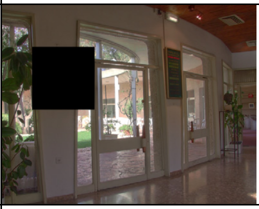
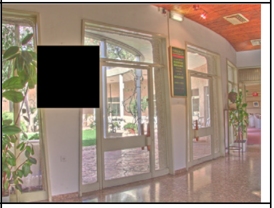
Embedding Distortion		PhotoGraphic	Bilateral	GDC
Bluring	Tone mapped corrupted image			
	PSNR of distorted HDR image	40.18	40.18	40.18
	NC of the retrieved watermark	ρ =0.72	ρ =0.68	ρ =0.78
Local cropping	Tone mapped corrupted image			
	PSNR of distorted HDR image	35.46	35.46	35.46
	NC of the retrieved watermark	ρ =0.72	ρ =0.69	ρ =0.75

Table 4. The comparison of the robustness of the watermakre HDR image when different tone mapping methods are used in the watermark embedding procedure under bluring and cropping attacks.

Distortion	PSNR of the corrupted HDR image (dB)	Correlation coefficient ρ of the extracted watermark
Tone mapping by Bilateral approach	61.51	0.86
Tone mapping by photographic method	59.78	0.83
Gaussian blurring (radius=0.8)	36.19	0.71
Blurring + cropping	54.40	0.80

Table 5. The watermarking result using the HDR image-indoor.

Distortion	PSNR of the corrupted HDR image (dB)	Correlation coefficient ρ of the extracted watermark
Tone mapping by photographic method	43.34	0.94
Gaussian blurring (radius=0.8)	28.31	0.74
Cropping	31.33	0.74
Blurring + cropping	26.61	0.55

Table 6. The watermarking result using the HDR image-church window.

5. References

Debevec, P. & Malik, J. (1997). Recovering High Dynamic Range Radiance Maps From Photographs, *ACM Transactions on Graphics*, pp. 369-378.

Reinhard, E.; Ward, G. & Pattanaik, S. (2005) High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting with CDROM, Morgan Kaufmann Publishers ISBN 978-0-12-585263-0.

Reinhard,E.; Kunkel, T.; Marion, Y.; Brouillat, J.; Cozot, R. & Bouatouch, K. (2007) Image Display algorithms for High and Low Dynamic Range Display Devices, *Journal of the Society for Information Display*, Vol. 15, No. 2, pp. 997-1014..

Tsang, K. F. & Au, O. C. (2001) A review on attacks, problems and weakness of digital watermarking and the pixel reallocation attack, *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents III*, Vol. 4314, pp. 385-393.

Piva,A.; Barni, M.; F. Bartolini & Cappellini, V. (1997) DCT-Based Watermark Recovering without Resorting to The Uncorrupted Original Image, *Proceedings of International Conference on Image Processing*, Vol. 1, pp. 520-523.

Barni, M.; Bartolini, F.; Cappellini, V.; Lippi, A. & Piva, A. (1999) DWT-Based Technique for Spatio-Frequency Masking of Digital Signatures, *Proceedings of SPIE, Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 31-39.

Wang, Y.; Doherty, J. F. & Van Dyck, R. E. (2002) A Wavelet-Based Watermarking Algorithm for Ownership Verification of Digital Images, *IEEE Transactions on Image Processing*, Vol. 11, No. 2, pp. 77-88.

Suhail, M. A. & Obaidat, M. S. (2003) Digital Watermarking-Based DCT and JPEG Model, *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No. 5, pp. 1640-1647.

Ward, G. & Simmons, M. (2004) Subband Encoding of High Dynamic Range Imagery, *Proceeding Fist Symposium Applied Perception in Graphics and Visualization (APGV)*, ACM Press, pp.83-90.

Durand, F. & Dorsey, J. (2002) Fast Bilateral Filtering for the Display of High Dynamic Range Image, *ACM Transactions on Graphics*, Vol. 21, No.3, pp. 257-265.

Reihard,E.; Stark, P.; Shirley, M. & Ferwerda, J. (2002) Photographic Tone Reproduction for Digital Images, *ACM Transactions on Graphics*, Vol. 21, No.3, pp.267-276.

Fattal, R.; Lischinski, D. & Werman, M. (2002) Gradient Domain High Dynamic Range Compression, *ACM Transactions on Graphics*, Vol. 21, No.3, pp.249-256.

11

Improve Steganalysis by MWM Feature Selection

B. B. Xia, X. F. Zhao and D. G. Feng
Institute of Software Chinese Academy of Sciences
China

1. Introduction

Steganography is the art of invisible communication. It is derived from the ancient Greece thousands of years ago (Johnson & Jajodia, 1998; Kahn, 1996), and grow rapidly in the past few years along with the development of digital technology and the internet. Modern steganography has been widely used in various scenarios such as secret communication, digital rights management, data temper detection and recovery, etc (Provos & Honeyman, 2003). The main goal of modern steganography is to hide some secret messages into the so-called cover data, e.g. images, videos, audios, documents…, which produced the so-called stego data, and make the existence of the hidden messages unnoticeable to everyone expect the prospective receiver (Provos, 2001; Fridrich & Goljan, 2002; Fridrich, 2005). Usually an embedding key is also involved in the steganography scheme to provide security. The malicious can never tamper, remove, nor obtain the secret messages in the stego data, as long as the embedding key is kept unknown.

In contrast to steganography, steganalysis is developed to detect the presence of the secret messages, and furthermore estimate the length or even extract the content of the embedded messages. Although the secret message in stego data is always imperceptible to human's visual, the embedding process changes some statistics of the cover medium nevertheless. This can be utilized by steganalysis methods to distinguish stego mediums containing secret messages from the clean cover mediums (Lyu & Farid, 2002; Fridrich et al., 2002; Fridrich, 2005; Harmsen & Pearlman, 2003; Ker, 2005; Tzschoppe & Aauml, 2003; Xuan et al., 2005).

Steganalysis methods can be roughly divided into two categories: the targeted (or specific) steganalysis which detects a particular known steganography method, and the blind (or universal) steganalysis which can deal with a wide variety of steganography methods. Though the targeted methods often have slightly better accuracy and efficiency than the blind ones, it is quit reasonable to assume that the nature of the cover data and the embedding method used for steganography is unknown to the analysers beforehand. Therefore, blind steganalysis based on learning and classifying are more valuable from a practical point of view (Fridrich & Goljan, 2002; Provos & Honeyman, 2003; Tzschoppe & Aauml, 2003).

The typical framework of blind steganalysis is a procedure of two-class classification, which consists of training and classifying. First, a set of statistics called steganalysis features is extracted from a pair of training set which contains cover and stego mediums respectively. A classifier is then trained by these extracted features. Then, given the medium under test, the steganalysis features is extracted similarly and input to the classifier to decide whether it contains hidden messages.

The choice of steganalysis features is crucial to the classification accuracy. As mentioned earlier, embedding messages in cover medium will change some of the statistics. It is obvious that choosing statistics which are sensitive to the steganography embedding process will provide better steganalysis accuracy. The statistic moments and transition probabilities have been proved to be more efficient than other choice for a wide variety of steganography methods, and so are used frequently in modern steganalysis (Davidson & Jalan, 2010; Fridrich et al., 2002; Lyu & Farid, 2002; Pevný et al., 2010a; Pevný & Fridrich, 2007).

To improve the steganography security, some embedding methods attempt to maintain the statistics of cover medium by means of minimizing the embedding distortion. Encoding methods such as matrix encoding or wet paper encoding are implemented to steganography process to reduce the embedding distortion (Fridrich et al., 2004; Westfeld, 2001). LSB(Least Significant Bit) matching method solves the imbalance problem introduced to the sample value histogram by the original LSB replacement method, and thus provide good security against steganalysis based on 1*st* order statistic features (Mielikainen, 2006). An adaptive steganography called HUGO (Highly Undetectable Steganography) is proposed recently. Before embedding secret messages into a cover image, HUGO determines a distortion measure for each pixel by calculating a weighted sum of difference between the features derived from cover and stego images. 1*st* and 2*nd* order transition probabilities of SPAM steganalysis features are chose as the features (Pevný et al., 2010a), which makes HUGO undetectable using steganalysis methods based on 1*st* and 2*nd* order statistic features. The details of this algorithm can be found in (Pevný et al., 2010b).

As steganography methods try to reduce embedding distortion and preserve representation of covers approximately for low-order statistic features, it is natural that steganalysis takes one more step in the same direction. That is to say, higher-order statistic features should be used as steganalysis features for better classification accuracy. However, this leads to a catastrophic growth of the amount of feature dimensions. Recently, Gul & Kurugollu propose a 1237 dimension feature set constructed by k-variate probability distribution function (PDF) estimates (Gul & Kurugollu, 2011). Fridrich et al. suggest a final HOLMES feature set that consists of 33,963 features to obtain better accuracy against HUGO (Fridrich et al., 2011).

The increasing dimensions of features bring new challenges to steganalysis. Training classifiers in high dimensions requires relatively large number of samples. With the significant growth of the feature dimensions, it becomes harder or even impossible to obtain sufficient samples. Furthermore, the computational complexity of training the classifier on a large-scale training set in high-dimensional spaces also becomes prohibitive.

Feature selection is a typical method to deal with the excessive feature dimensions. Feature selection not only reduces the number of dimensions, but also removes the inefficient or redundant features, leaves the efficient ones to the classifier for better training and classification. The theoretical ideal way of feature selection is exhaustive searching all the possible combination of feature dimensions, which can not be achieved in practical scenario.

Miche et al. (Miche et al., 2006) uses a fast classifier called K-Nearest-Neighbors combined with a forward selection method to achieve feature selection, which is still limited to deal with the relative low dimension feature sets. Dong et al. (Dong et al., 2008) make use of the Boosting Feature Selection (BFS) algorithm as the fusion tool to select a subset of original statistic features. Each dimension of the original features is treated as a weak classifier, and the output of BFS classification is calculated for each of them as an evaluation indicator. The final classifier is then constructed by every weak classifier with the evaluation indicator as their weight. Gul & Kurugollu (Gul & Kurugollu, 2011) also establish an evaluation indicator for each single dimension of the original features by means of calculating the co-variance between features and the embedding rates. After that, the original features are sorted corresponding to their co-variance in the decreasing order. The best K features are then determined to form the final classifier, by adding all the features one by one to the classifier and test the performance. Fridrich et al. (Fridrich et al., 2011) propose another method of ensemble classifier to reduce the dimension of steganalysis classifier. The ensemble classifier consist of weak base learners which constructed by randomly choose subsets of the original features. The dimensionality of each base learner is significantly smaller than the full dimensionality of the original feature set. The final decision is obtained by fuse the result of all base learners together under certain voting rule.

In this chapter, we present a novel methodology called MMD-weighted-MI (MMD, Maximum-Mean-Discrepancy; MI, Mutual-Information) feature selection to deal with high-dimensional steganalysis features. Before training the classifier, a MMD-weighted-MI (MWM) value is calculated and assigned to each dimension of the original features by evaluating the distribution of the extracted features using the MI and MMD indicators. The MI and MMD are both efficient measurements used exclusively in steganography benchmarking, but focus on different aspects of the feature distribution (Pevný & Fridrich, 2008). The MMD gives an overall view of a subset of the features, evaluates the difference of feature distribution between cover and stego training sets by means of generate a set of functions from the kernel ones and then calculate the maximum mean discrepancy. On the other hand, the MI, which is calculated only for single feature dimension due to its unacceptable complexity introduced by estimation of high-dimensional distribution, gives more details about how each dimension contributes to the classifier in steganalysis. When combined together as MWM values, these two indicators can give us a more comprehensive impression about difference between features extracted from the cover and stego training sets. After the MWM values are assigned, feature selection is simply implemented by choosing feature dimensions with high value.

The organization of this chapter is as follows: Section 2 introduces some basic concepts of MI and MMD, as well as elaborates their different effect in feature selection briefly. Section 3 describes the proposed approach of MWM feature selection. Experimental results are presented in Section 4. The chapter is finally concluded in Section 5.

2. Basic concepts of MI and MMD

In this section, we explain the basic concepts of MI and MMD. These two measurements have been used in steganography benchmarking due to their characteristic of evaluating the difference between two distributions, which makes them natural candidates for steganalysis feature selection.

Without loss of generality, images are chose as the cover medium in the following discussion. The same result holds for other form of mediums such as videos, audios and documents.

2.1 Mutual Information (MI)

Denote X the whole set of images corresponding to the steganalysis system. X can be divided into two non-overlap subsets, namely cover set C and stego set S respectively. Denote P and Q the distribution of the cover and stego set, with p and q as the probability distribution function (pdf) respectively. Then the difficulty of distinguishing stego images from cover ones can be measured using statistic called Kullback-Leibler divergence (Cachin, 1998)

$$KL(P\,||\,Q)=\int_x p(x)\log\frac{p(x)}{q(x)}dx \tag{1}$$

where x denotes the sample medium drawn from the whole set of image X .

The KL divergence is a fundamental quantity for steganography benchmarking, which provides good estimate to the difference between cover and stego sets for certain features (Cover & Thomas, 2001). However, the asymmetry in calculating the KL divergence becomes a main drawback. From (1), it is obvious that

$$KL(P\,||\,Q)\neq KL(Q\,||\,P)\,. \tag{2}$$

This computing asymmetry, without carefully treatment, could cause inconsistent in the quantitative evaluation for feature dimensions, and thus leads to inconveniency and ambiguity in feature selection. To overcome this difficulty, we use Mutual Information (MI) to substitute KL divergence

$$I(P,Q)=\sum_i\sum_j\phi(x_i,y_j)\log\frac{\phi(x_i,y_j)}{p(x_i)q(y_j)} \tag{3}$$

where x_i and y_i denote the steganalysis features extracted from images in cover and stego set respectively, $\phi(x_i,y_i)$ denote the joint probability distribution function, $p(x_i)$ and $q(y_i)$ denote the marginal probability distribution functions respectively. It is obvious that the definition of MI is symmetric

$$I(P,Q)=I(Q,P)\,. \tag{4}$$

The MI can be represented as an expectation of KL divergence as below

$$I(X:Y)=E_Y\Big(KL\big(p(x\,|\,y)\,||\,p(x)\big)\Big) \tag{5}$$

where $p(x\,|\,y)$ denotes the conditional probability of image x drawn from the cover set, given the image y from the stego set, and $E_Y(\cdot)$ denotes the expectation for the random variable y . The relationship between the MI and the KL divergence in (5) suggests that MI maintains the characteristics of KL divergence in steganography benchmarking, provides a fundamental quantity that estimates the difference between the distributions of the features obtained from the cover and stego set. In this way, the MI establishes a measurement of how much the features contribute to the final classifier. These properties and the computing symmetry shown in (4) make the MI an appropriate choice in evaluating the value of feature dimensions in steganalysis feature selection.

The calculation of the MI relies on the estimation of the distributions of P and Q, which is quite difficult or even impossible to achieve for high dimensional features in a practical point of view. Thus, we treat each dimension of the original features as a single feature and calculate MI separately. Histogram estimates are applied to each single feature to provide estimation of their distributions. The details can be found in Section 3.

2.2 Maximum Mean Discrepancy (MMD)

Given two distributions P and Q defined on the whole set of images X, the disparity of P and Q can be evaluated by a statistic called Maximum Mean Discrepancy (MMD) (Gretton et al., 2007). The main idea behind MMD is based on the statement that P and Q are the same distribution if and only if their probability distribution functions (pdf) p and q satisfy that

$$E_{x\sim p}\left(f\left(x\right)\right)=E_{x\sim q}\left(f\left(x\right)\right),\forall f\in C(X) \tag{6}$$

where $C(X)$ denotes the set of all continuous bounded functions on X, $E_{x\sim p}(\cdot)$ and $E_{x\sim q}(\cdot)$ denotes the expectation for the random variable x with p and q as the pdf respectively.

The number of functions in $C(X)$ is infinite, but only part of the functions in $C(X)$ can be utilized because of the finite number of samples in the training sets in practical steganalysis scenarios. Denote Γ a subset of $C(X)$, then the difference between distributions P and Q is evaluated by MMD values corresponding to Γ as

$$MMD\left[\Gamma,X_D,Y_D\right]=\sup_{f\in\Gamma}\left(\frac{1}{D}\sum_{i=1}^{D}f\left(x_i\right)-\frac{1}{D}\sum_{i=1}^{D}f\left(y_i\right)\right) \tag{7}$$

where $X_D=\{x_1,\ldots,x_D\}$ and $Y_D=\{y_1,\ldots,y_D\}$ are observations of the cover and stego distributions P and Q respectively.

The choice of Γ affects the performance of MMD significantly. It has to be rich enough to make p and q distinguishable, while still under the restriction of the finite number of images in cover and stego training sets. A typical construction of Γ is the Reproducing Kernel Hilbert Spaces (RKHS) built from the so-called *kernel* function. The kernel function is a symmetric, positive definite function used to generate the RKHS. Gaussian kernel has been proved to be a valuable choice (Pevný & Fridrich, 2008) as

$$k\left(x,y\right)=\exp\left(-\gamma\left\|x-y\right\|_2^2\right),\gamma>0\ . \tag{8}$$

In this case, the MMD values corresponding to Γ are obtained by an unbiased estimate based on U-statistics as

$$MMD_u[\Gamma, X_D, Y_D] = \left[\frac{1}{D(D-1)}\sum_{i\neq j}\left(k\left(x_i, x_j\right) + k\left(y_i, y_j\right) - k\left(x_i, y_j\right) - k\left(x_j, y_i\right)\right)\right]^{\frac{1}{2}}. \quad (9)$$

Fig. 1 shows the effectiveness of MMD values as steganography benchmarking. Training sets generated by various embedding methods (Fridrich et al., 2004; Westfeld, 2001; Mielikainen, 2006) and different embedding rates are applied to calculate MMD values. The false rates of the classifier corresponding to each training sets are also obtained, and normalized to be comparable to the MMD values. Note that the original MMD values are replaced by $-\log_{10}(MMD)$ for better visual. The result shows that MMD values are good estimations of the performance of classifiers in steganalysis.

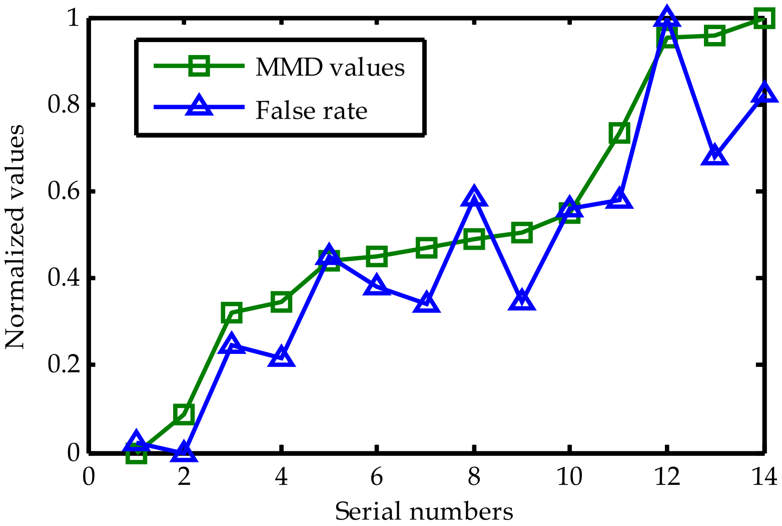


Fig. 1. Comparison between the MMD values (square marked) and the false rate of the classifiers (triangle marked)

The computational complexity of MMD with Gaussian kernel is $O\left(D^2\right)$, where D is the number of sample images. It is far more efficient in comparison to Support Vector Machines (SVM), which is a commonly used classifier in modern steganalysis. Further more, the MMD converges with error $1/\sqrt{D}$, yet almost independently on feature dimensions, which means that a sample set with roughly 10^3 images is sufficient for MMD to provide accurate estimations despite the feature dimensions. These advantages make MMD a natural choice to achieve feature selection for high-dimensional steganalysis features.

3. MMD-weighted-MI feature selection

The MI and MMD are both efficient measurements of evaluating the difficulty of distinguishing stego images from cover ones. Therefore, we apply them to steganalysis feature selection methods. As they focus on different aspects of the feature distributions, we combine these two indicators into MMD-Weighted-MI for more comprehensive feature selection.

3.1 MI values for single feature dimension

As shown in Section 2.1, MI is a fundamental quantity for evaluating the value of feature dimensions in steganalysis feature selection. However, the calculation of MI relies on accurate estimation of high-dimensional distribution of the original features, which is difficult or even impossible to achieve from a practical point of view. To solve this problem, we calculate MI for each single dimension of the original features separately instead of treating them as high-dimensional features. Denote $x=\left(x^1,x^2,\cdots,x^d\right)$ the original feature extracted from the images in X, d is the total number of dimensions. Denote P^k and Q^k the marginal distribution of the k-th dimension in original features extracted from the cover and stego set respectively, the MI value for the k-th dimension of the original features is defined as

$$MI^k = I(P^k,Q^k) = \sum_i \sum_j \phi_k(x_i^k,y_j^k)\log\frac{\phi_k(x_i^k,y_j^k)}{p_k(x_i^k)q_k(y_j^k)}, k=1,2,\cdots d \tag{10}$$

where $\phi_k(x_i^k,y_j^k)$ denote the joint probability distribution function of the k-th dimension of the cover and stego set, $p_k(x_i^k)$ and $q_k(y_j^k)$ denote the marginal distributions respectively. Since $p_k(x_i^k)$ and $q_k(y_j^k)$ are both distributions of single random variables, it is simple to estimate their pdf by histogram estimation as

$$\tilde{p}_k(x) = \frac{n_j}{nh} \tag{11}$$

where n_j is the frequency fell into the j-th category, and h denotes the interval of categories. $\phi_k(x_i^k,y_j^k)$ is estimated by the joint histogram similarly. The choice of h affects the discrepancy and variance of the histogram estimation. Higher value of h leads to larger discrepancy and smaller variance, or vice versa. To achieve balance between discrepancy and variance, we set intervals dynamically corresponding to the dynamic range of each feature dimension in the proposed algorithm in Section 3.3.

Fig. 2 gives an example of MI calculated for each single dimension. The training set consists of cover images in jpg format and corresponding stego images generated by F5 steganography algorithm (Westfeld, 2001) with embedding rate at 0.05 bpac[1]. The Merging Markov and DCT features (Pevný & Fridrich, 2007) are chose as original steganalysis features. Fig. 2 shows that MI value for each single dimension varies significantly, and the feature dimensions with higher MI values contribute more than others in steganalysis classifier.

3.2 MMD-Weighted-MI (MWM)

The calculation of MI values of single feature dimensions treats each dimension as an independent feature. However, the correlation of different dimensions also plays an important

[1] bpac, bit per AC coefficients

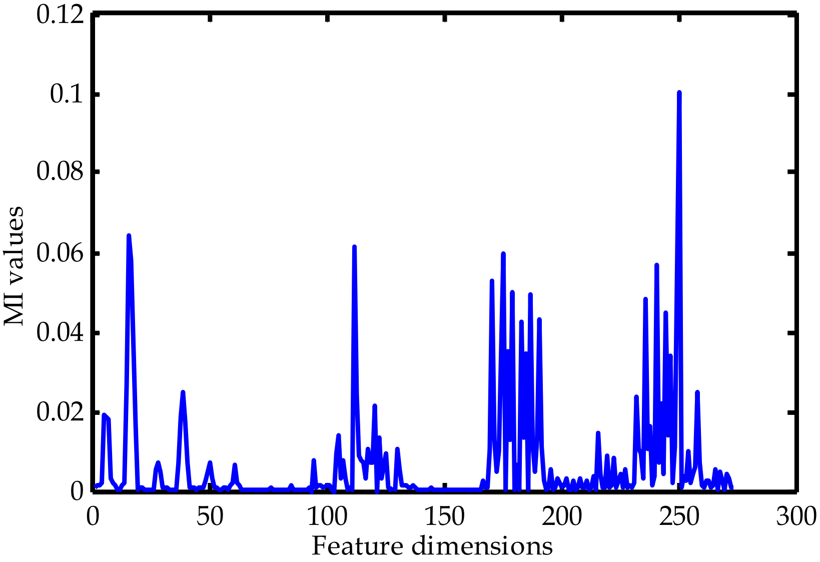


Fig. 2. MI values of Merging Markov and DCT features

role in training and classification. The absence of the information of feature correlation will bring troubles to the feature selection. Furthermore, the MI values of features drawn from different original feature sets have different dynamic range, which leads to an unfair comparison between feature sets if only MI values are used for feature selection. Fig. 3 shows the comparison of two feature sets as an example, where the MI values of most feature dimensions from the Partially Ordered Markov Model features sets (Davidson & Jalan, 2010) are relatively low than the Merging Markov and DCT features. The MI values of Y-Axis are limited to 0.04 for the purpose of clear observation though.

The raw MI values of single feature dimension are defective for feature selection due to the lack of correlation information. To overcome this difficulty, we introduce MMD as well for evaluating the feature dimensions. The MMD is numerically stable even in high-dimensional spaces, which makes it an excellent choice for providing information about correlation between feature dimensions. The computational complexity is also relatively low so that calculating MMD values for high-dimensional feature sets is feasible.

The combination of MI and MMD values provides a more comprehensive impression about the difference between the distribution of features extracted from the cover and stego training sets, which results in a new indicator called MMD-Weighted-MI (WMW) for better feature selection. Denote $MI(i,j)$ the MI value of the j-th dimension in the i-th feature set, and $MMD(i)$ the MMD value of the i-th feature set. Then a WMW value is assigned to each feature dimension as

$$WMW(i,j) = \frac{MI(i,j)}{MMD(i)} \tag{12}$$

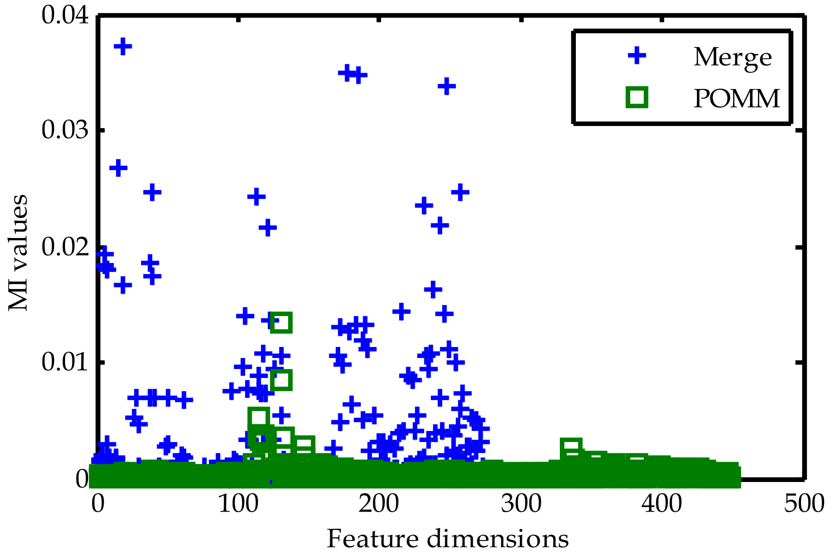


Fig. 3. Comparison of MI values between the Merging Markov and DCT features ('Merge', cross marked ones) and the Partially Ordered Markov Model features ('POMM', square marked ones)

Fig. 4 shows the MWM values derived from the two feature sets, namely the Merging Markov and DCT features and the Partially Ordered Markov Model features which is the

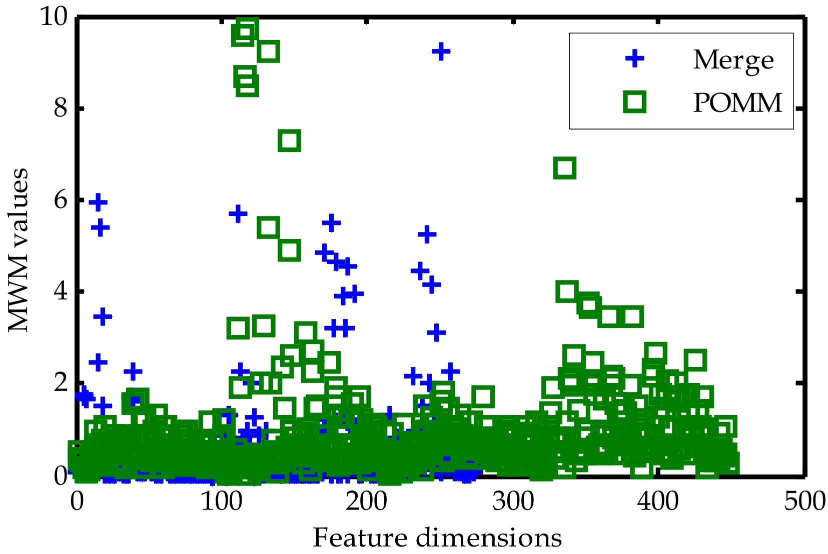


Fig. 4. Comparison of MI values between the Merging Markov and DCT features ('Merge', cross marked ones) and the Partially Ordered Markov Model ('POMM', square marked ones)

same as used in Fig. 3. Compared to the result of the raw MI values, MWM values achieve a fair comparison between different feature sets; make the dynamic range of the two feature sets comparable. This leads to a better feature selection for steganalysis and thus better accuracy of the classifier, which is supported by experiment results in Section 4.

3.3 Feature selection approach

The feature selection is implemented based on the MWM values and new steganalysis classifiers are constructed by the selected features. Fig. 5 shows the overview of our approach, and the details are presented as follows.

- Step 1. We choose several different steganalysis methods and extract the corresponding feature sets from the training set.
- Step 2. MI values are calculated and assigned to each feature dimension, and MMD values are calculated for each feature set as high-dimensional features. The MWM values are then generated based on MI and MMD values.
- Step 3. Feature dimensions with their MWM values larger than a given threshold are selected to assemble the new fused features. The serial numbers of the selected features are recorded as well. A classifier is trained with the fused features for steganalysis.

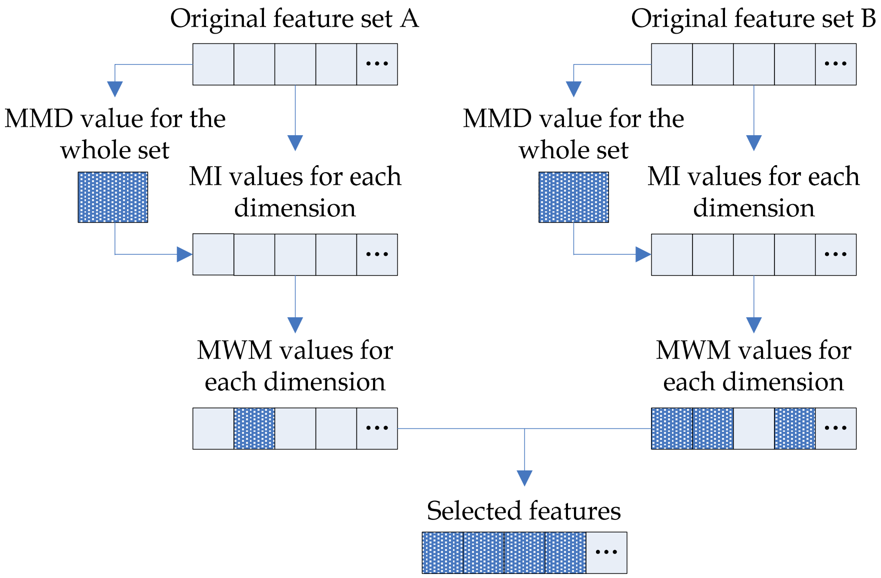

Original feature set A
Original feature set B
...
MMD value for the whole set
MI values for each dimension
MWM values for each dimension
Selected features

Fig. 5. Overview of the MWM feature selection

4. Experiment results

In this section, we experimentally investigate the performance of our WMW feature selection. We choose images of JPEG format as the cover images without loss of generality. The extensively used BOSSbase image database (Bas et al., 2010) is used as the source of cover images. Because the original images from BOSSbase are in the RAW format, we converse them into JPEG format with a 98 JPEG quality factor. The stego images are generated using the following two typical steganography algorithms:

a. F5 steganography by Westfeld (Westfeld, 2001)
b. Perturbed Quantization (PQ) steganography by Fridrich et al. (Fridrich et al., 2004)

The embedding rates vary from 0.05 bpac to 0.15 bpac. We randomly choose 1000 pair of cover/stego images as the training sets and 300 other pairs as the testing sets.

Three typical steganalysis feature sets are chose to provide original features for WMW feature selection:

a. Markov Transition Probability features by Shi et al. (Shi et al., 2007), with 900 dimensions.
b. Merging Markov and DCT features by Pevný & Fridrich. (Pevný & Fridrich, 2007), with 274 dimensions.
c. Partially Ordered Markov Model features by Davidson & Jalan (Davidson & Jalan, 2010), with 448 dimensions.

The total number of dimensions involved in our experiments is 900+274+448=1622. We gradually increase the number of the chosen dimensions by WMW feature selection and test the performance of the corresponding classifiers by TR [2]. The intervals of the feature numbers are set differently because of the uneven density of the distribution of the MWM values. For the purpose of a clear view, we set interval to 5 dimensions within the first 150 features and 100 dimensions for the rest of them. The TR of the classifiers are tested and shown in Fig. 6, 7 and 8 for different embedding rate (0.05bpac, 0.1bpac and 0.15bpac) respectively.

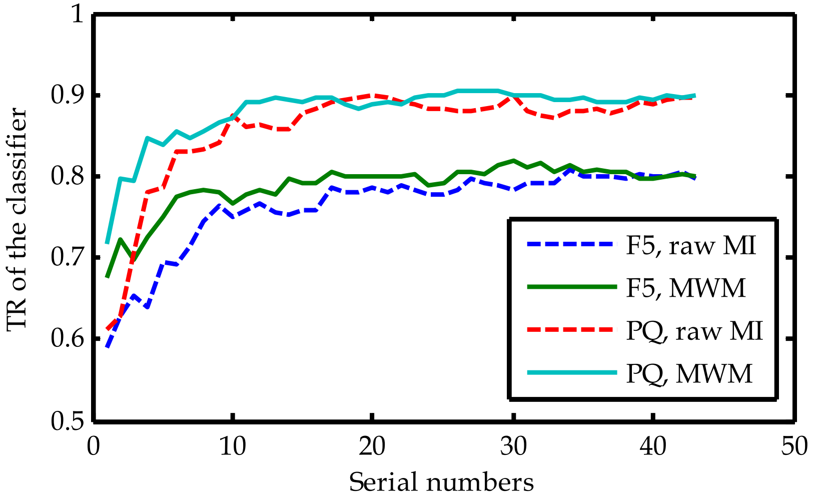


Fig. 6. Comparison of the performance between the classifier using WMW values (solid lines) and the raw MI values (dotted lines) for feature selection, with embedding rate 0.05bpac.

[2] TR, True Rate, the average of the True Positive rate (TP) and True Negative rate (TN)

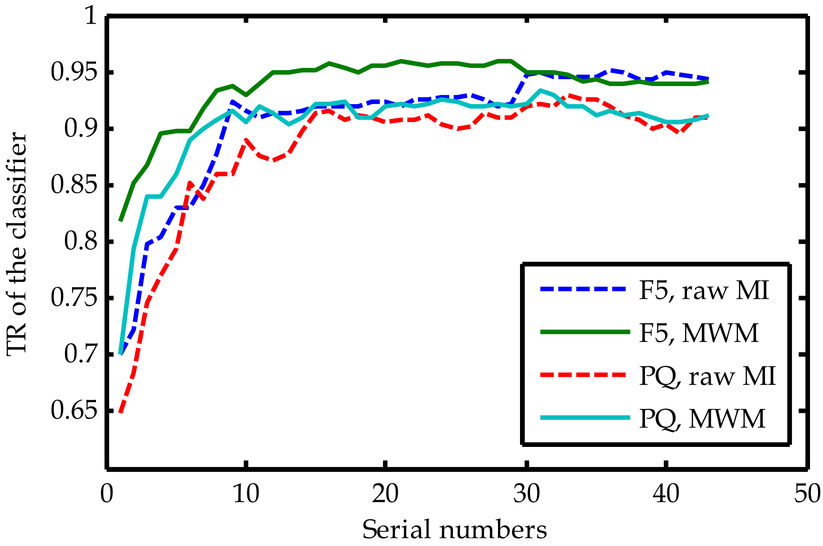


Fig. 7. Comparison of the performance between the classifier using WMW values (solid lines) and the raw MI values (dotted lines) for feature selection, with embedding rate 0.1bpac.

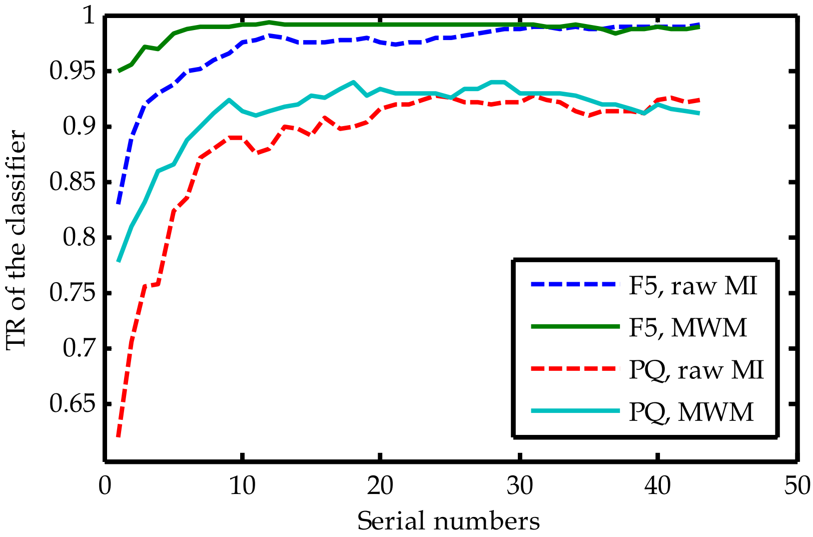


Fig. 8. Comparison of the performance between the classifier using WMW values (solid lines) and the raw MI values (dotted lines) for feature selection, with embedding rate 0.15bpac.

From Fig. 6, 7 and 8, we can observe that WMW feature selection provides higher TR of the classifier than feature selections using only the raw MI values. The reason of this has been discussed in Section 3.2. Note that the last TR value of each curve represents the performance of the classifier consist of all features without selection approach. It is then obvious that we can always achieve better accuracy of steganalysis using WMW feature selections than the original feature sets, whereas feature selection with raw MI values fail in some cases, e.g. F5 steganography with embedding rate 0.1 bpac in Fig. 7, and Perturbed Quantization steganography with embedding rate 0.15 bpac in Fig. 8. Table 1 shows the optimal accuracy of each case, and the TR of classifiers consist of the original feature set are also listed for comparison.

Embedding Cases	MPB	Merge	POMM	Total	Raw	MWM
F5, 0.05bpac	60.5%	80.7%	67.6%	79.7%	80.4%	**81.9**%
F5, 0.10bpac	73.1%	93.9%	85.5%	94.4%	95.2%	**96.0**%
F5, 0.15bpac	84.2%	98.3%	94.0%	98.9%	**99.2**%	**99.2**%
PQ, 0.05bpac	68.8%	88.4%	73.3%	89.9%	89.9%	**90.5**%
PQ, 0.10bpac	72.6%	89.2%	76.2%	91.2%	92.9%	**93.4**%
PQ, 0.15bpac	71.4%	90.9%	78.0%	92.4%	92.7%	**93.9**%

Table 1. Optimal accuracy of steganalysis for different embedding cases using different feature sets: Markov Transition Probability features ('MPB'), Merging Markov and DCT features ('Merge'), Partially Ordered Markov Model features ('POMM'), fused features contain all feature dimensions without feature selection ('Total'), feature selection using only raw MI values ('Raw'), and MWM feature selection ('MWM').

The best accuracy for each embedding case is marked in bold in Table 1, and from that we can assert that the MWM feature selection is always the better choice for steganalysis comparing to other methods.

5. Conclusion

In this chapter, we present a new approach of feature selection in steganalysis involving MI and MMD, which are both efficient indicators for evaluating the difference between cover and stego sets. Although the MI values are well understood theoretically, the computational difficulty of estimating the distribution of high-dimensional features makes it inconvenient in steganalysis feature selection. Thus, we treat each dimension as a single feature and calculate MI values separately.

This approach, however, abandons the correlation between feature dimensions, which makes raw MI values defective for feature selection. To solve this problem, MMD values are introduced in our approach as well. The MMD values are numerically stable even in high-dimensional spaces, and the computational complexity is relatively low. These advantages make MMD a natural complementary to MI values, and thus leads to our proposed approach of feature selection based on MMD-weighted-MI values.

To test the performance of the MWM feature selection, we apply our method to three typical steganalysis feature sets to generate new classifiers, and estimate the accuracy of these fused classifiers against two widely used steganography algorithms. Experimental results shows that the MWM feature selection approach outperforms the feature selections with raw MI values, and guarantees better accuracy comparing to the original feature sets.

6. Acknowledgment

This work was supported by the NSF of China under 61170281, the NSF of Beijing under 4112063, the Strategic and Pilot Project of Chinese Academy of Sciences (CAS) under XDA06030600, and the Project of Institute of Information Engineering, CAS, under Y1Z0051101 and Y1Z0041101.

7. References

Bas, P.; Filler, T. & Pevný, T. (2010). BOSS. http://boss.gipsa-lab.grenobleinp.fr/BOSSRank/, July 2010

Cachin, C. (1998). An information-theoretic Model for Steganography. *Information Hiding 1998, Lecture Notes in Computer Science*, Vol. 1525, (1998), pp. 306–318

Cover, T. M. & Thomas, J. A. (2001). *Frontmatter and Index, in Elements of Information Theory*. John Wiley & Sons, Inc., ISBN 9780471062592, New York

Davidson, J. & Jalan, J. (2010). Steganalysis Using Partially Ordered Markov Models. Proc. *Information Hiding 2010, Lecture Notes in Computer Science*, Vol. 6387, (2010), pp. 118-132

Dong, J.; Chen, X.; Guo, L. & Tan, T. (2008). Fusion Based Blind Image Steganalysis by Boosting Feature Selection. *Digital Watermarking, Lecture Notes in Computer Science*, Vol. 5041, (2008), pp. 87-98

Fridrich, J. (2005). Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes. *Information Hiding, Lecture Notes in Computer Science*, Vol. 3200, (2005), pp. 67-81

Fridrich, J. & Goljan, M. (2002). Practical Steganalysis of Digital Images: State of the Art. In: *Security and Watermarking of Multimedia Contents*, Vol. SPIE-4675, pp. 1–13, 2002

Fridrich, J.; Goljan, M. & Hogea, D. (2002). Steganalysis of JPEG Images: Breaking the F5 Algorithm. *5th International Workshop on Information Hiding*, Vol. 2578, pp. 310-323, Oct 07-09, 2002

Fridrich, J.; Goljan, M. & Soukal, D. (2004). Perturbed Quantization Steganography with Wet Paper Codes. *Proc. ACM Multimedia Workshop'04*, pp. 4-15, 2004

Fridrich, J.; Kodovský, J.; Holub, V. & Goljan, M. (2011) Steganalysis of Content-adaptive Steganography in Spatial Domain. *Information Hiding, Lecture Notes in Computer Science*, 2011

Gul, G. & Kurugollu, F. (2011). A New Methodology in Steganalysis: Breaking Highly Undetectable Steganography (HUGO). *Information Hiding, Lecture Notes in Computer Science*, Vol. 6958, pp. 71-84, 2011

Gretton, A.; Borgwardt, K.; Rasch, M.; Scholkopf, B. & Smola, A. (2007). A Kernel Method for the Two-sample-problem. *Advances in Neural Information Processing Systems*, Vol. 19, (2007), pp. 513–520, MIT Press, Cambridge

Harmsen, J. J. & Pearlman, W.A. (2003). Steganalysis of Additive Noise Modelable Information Hiding. In: *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents VI*, (2003), pp. 131–142

Johnson, N. F. & Jajodia, S. (1998). Exploring Steganography: Seeing the Unseen. In: *IEEE Computer Society*, Vol. 31, pp. 26–34, 1998

Kahn, D. (1996). The History of Steganography. *Information Hiding, Lecture Notes in Computer Science*, Vol. 1174, (1996), pp. 1-5

Ker, A. D. (2005). Steganalysis of LSB Matching in Grayscale Images. Signal Processing Letters, IEEE, Vol. 12, No. 6, (June 2005), pp. 441-444, ISSN: 1070-9908

Lyu S. & Farid, H. (2002). Detecting Hidden Messages Using Higher-order Statistics and Support Vector Machines. In: *5th International Workshop on Information Hiding,* pp. 340-354, 2002

Miche, Y.; Roue, B.; Lendasse, A. & Bas, B. (2006). A Feature Selection Methodology for Steganalysis. *Multimedia Content Representation, Classification and Security Lecture Notes in Computer Science*, Vol. 4105, (2006), pp. 49-56

Mielikainen, J. (2006). LSB matching revisited. *Signal Processing Letters, IEEE*, Vol. 13, No. 5, (May 2006), pp. 285-287, ISSN: 1070-9908.

Pevný, T.; Bas, P. & Fridrich, J. (2010). Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transaction on Information Forensics and Security*, Vol. 5, No. 2, (June 2010), pp. 215-224, ISSN: 1556-6013

Pevný, T.; Filler, T. & Bas, P. (2010). Using High-dimensional Image Models to Perform Highly Undetectable Steganography. *Information Hiding, 12th International Workshop, Lecture Notes in Computer Science*, Calgary, Canada, June 28–30, 2010

Pevný, T. & Fridrich, J. (2007). Merging Markov and DCT features for Multi-class JPEG steganalysis. Proc. *SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505, pp. 3-1 – 3-14, 2007

Pevný, T. & Fridrich, J. (2008). Benchmarking for Steganography. *Information Hiding, Lecture Notes in Computer Science*, Vol. 5284, (2008), pp. 251-267

Provos, N. (2001) Defending Against Statistical Steganalysis. In: *Proceedings of the 10th USENIX Security Symposium*, pp. 323–336, 2001

Provos, N. & Honeyman, P. (2003). Hide and seek: an introduction to steganography. *Security & Privacy, IEEE*, Vol. 1(3), (May-June 2003), pp. 32-44, ISSN 1540-7993

Shi, Y. Q.; Chen, C. H. & Chen, W. (2007). A Markov Process Based Approach to Effective Attacking JPEG Steganography. Proc. *Information Hiding '07*, vol.4437, (2007), pp. 249-264

Tzschoppe, R. & Aauml, R.B. (2003). Steganographic System based on Higher-order Statistics. In: *Proceedings of SPIE, Security and Watermarking of Multimedia Contents V*, (2003), Vol. 5020

Westfeld A. (2001). F5 - A Steganographic Algorithm High Capacity Despite Better Steganalysis. *4th International Workshop on Information Hiding*, Vol. 2137, (April 25-27, 2001), pp. 289-302

Xuan, G. R.; Shi, Y. Q. & Gao, J. J. (2005). Steganalysis based on Multiple Features formed by Statistical Moments of Wavelet Characteristic Functions. *Proc. Information Hiding 2005*, Vol. 3727, p.p. 262-277, 2005

The Digital Watermarking Techniques Applied to Smart Grid Security

Xin Yan and Yang Wu
Department of Computer Science, Wuhan University of Technology
P. R. China

1. Introduction

Power supply in the 21st century is facing more and more challenges, e.g., environment protection, energy shortage etc, such that the techniques related to power supply imminently need to be promoted. Complying with the development of green and low-carbon economy, the concept of Smart Grid has been proposed. By and large, Smart Grid may be regarded as an important application of the techniques of wireless sensor networks and Internet of Things etc in power grid. Smart Grid should be a complicated integrated system with high security, which involves many aspects, e.g., the security of power generation equipments, the security of power transmission equipments, the security of data communications, and so on. Nonetheless, Smart Grid is an open and inclusive system, which makes it unsafe inevitably.

The traditional security methods use cryptography to encrypt data for transmissions, for instance, data encryption, data integrity protection, and two-way authentication etc. The data communication networks employed by Smart Grid involve cable and wireless communication networks. Here wireless communication networks usually refer to wireless sensor networks (Akyildiz et al., 2002). Due to the limited resources at sensor nodes, cryptography methods will seriously abate the life time of sensor nodes. The reason is that encryption algorithms usually need to consume more energy, time, and memory space to compute and store data (Kleider et al., 2004; Zia & Zomaya, 2006). Anyway, the traditional encryption methods are not suitable for handling the security issue of data communications in Smart Grid. Thus, this chapter will investigate how to apply a digital watermarking technique to solve the security problem of data communications for wireless sensor networks in Smart Grid.

2. Smart Grid and wireless sensor networks

Smart Grid is an intelligent network built in some integrated, high-speed, two-way communication networks. Its objective is to implement the power reliability, security, and efficiency, as well as clean energy supply by using advanced sensor technology, measurement technology and advanced decision support systems. Smart Grid transmits a wide variety of data, including the key equipment operation parameters, the power facility information, the power distribution and scheduling information, the electricity usage state, early warning information, and so on (Divan & Johal, 2006). By using the rich information, Smart Grid can efficiently control the power generation, transmission, distribution, scheduling, and sub-time pricing, as well as timely error check etc (Amin & Wollenberg, 2005). The hierarchical model of information flows in Smart Grid is shown in Fig. 1.

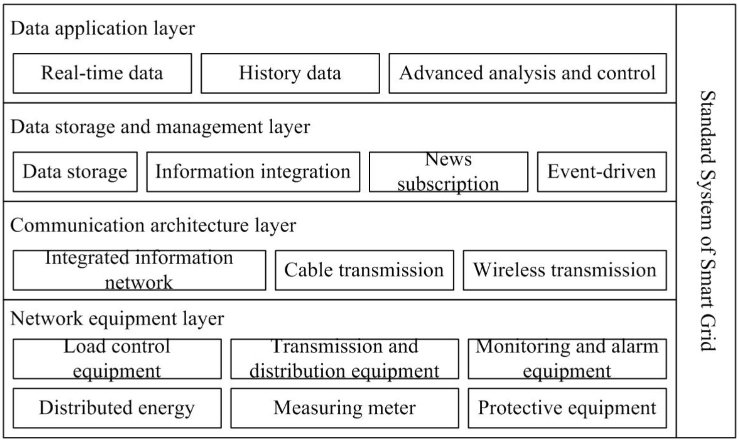


Fig. 1. The hierarchical model of information flows in Smart Grid

The communication networks related to Smart Grid consist of cable networks and wireless networks. The wireless networks mainly refer to wireless sensor networks that are usually used in some places where cable networks are not applicable to deploy or wireless sensor networks is more suitable. Smart Grid has a remarkable feature that its networks must be safer than other networks for general purposes. That is to say, Smart Grid must withstand the physical destructions and malicious network attacks without blackouts or a high cost of recovery (Perrig et al., 2004). Smart Grid security involves many aspects, where the data transmission security is one of the most important issues. Since the security mechanisms and techniques in cable networks are already quite rich and mature, we focus on trying to improve the security of the data in wireless sensor networks for Smart Grid.

Wireless sensor networks are a multi-hop self-organized network system, which contains a large number of miniaturized sensor nodes. These sensor nodes are distributed in a monitored area, and communicate in a multi-hop ad hoc way. They collaborate with each other to collect the sensitive information of monitored objects, and send them to a decision support center. The functions of wireless sensor networks consist of data collection, data transmission, and data analysis and processing. A sensor node, the smallest logical unit of wireless sensor networks, is a micro-system, which is integrated by sensor modules, data processing modules and communication modules. Sensor nodes build up wireless links to form a self-organized and distributed network architecture, depending on a certain network routing protocol that can fuse and aggregate the collected data and transmit them to the information processing centre (Chen et al., 2009). A network architecture of wireless sensor networks is shown in Fig. 2.

Smart Grid involves a large number of wireless sensor networks, so the data transmission security is an important issue in Smart Grid. However, due to wireless sensor networks with the large magnitude of energy-constrained sensor nodes and the high network dynamics caused by the node mobility or node failure, there still exist a lot of potential threats to the security of wireless sensor networks (Wang et al., 2006), e.g.:

1. The unauthorized interception of information. A sensor node transmits information to others by broadcasting, so any of communication devices within its RF radius may receive and intercept the information.
2. Sensor nodes are vulnerable to be captured easily. We must take into account what measurements should be taken to fight against, while a sensor node is captured and used as a pseudo terminal to launch malicious attacks.
3. In the practical environments, we must also consider which routing schemes should be adopted, in the case that some of sensor nodes do not work because of failures or attacks.
4. Tampering with information is usually regarded as the most dangerous attack. The tampered information can be spread throughout networks like normal messages, which can attack or even control the whole networks.

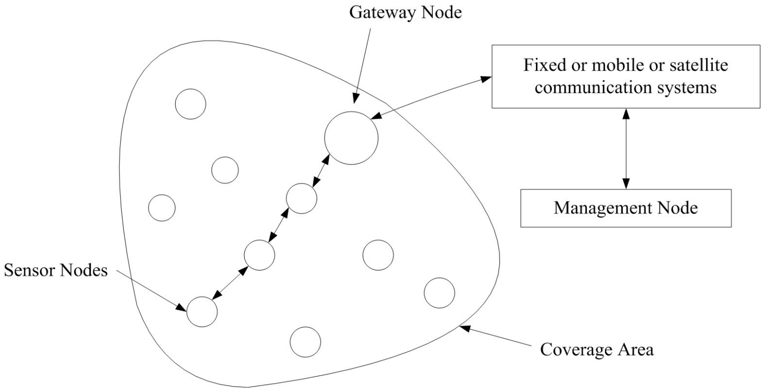


Fig. 2. A network architecture of wireless sensor networks

These potential threats to wireless sensor networks cause unsafe data communications in Smart Grid. To obtain safe communication services from Smart Grid, we must solve the security issues about wireless sensor networks. However, because of the differences between wireless sensor networks and traditional networks, the security policies for wireless sensor networks should not be borrowed directly from the existing mature security solutions for traditional networks. The security policies should be more suitable for wireless sensor networks. Data encryption methods are widely used in traditional networks, where the information needed to be protected is generated to cipher-text information without readability or obvious correlation. Nevertheless, the resources of computation and storage at sensor nodes are scarce and limited. The traditional data encryption methods will seriously consume the expensive resources at sensor nodes, because they require more power and memory space to accomplish the data encryption procedure. Therefore, we need to use digital watermarking methods to implement the security policies in wireless sensor networks, because digital watermarking needs much less resources at sensor nodes than traditional data encryption (Xiao et al., 2008).

3. Digital watermarking

Digital watermarking is a special kind of information hiding techniques, which is used to detect piracies or illegal copies. The watermark is transmitted with the information embedded identity in a digital form. Digital watermarking technique is suitable for the data-centric wireless sensor networks. Reasonable watermarking algorithms can ensure the data security at a low cost of operation, and tolerate effectively the impacts from data processing. Using digital watermarking techniques to solve the security issues in the wireless sensor networks for Smart Grid is a practical and effective solution (Xiao et al., 2007).

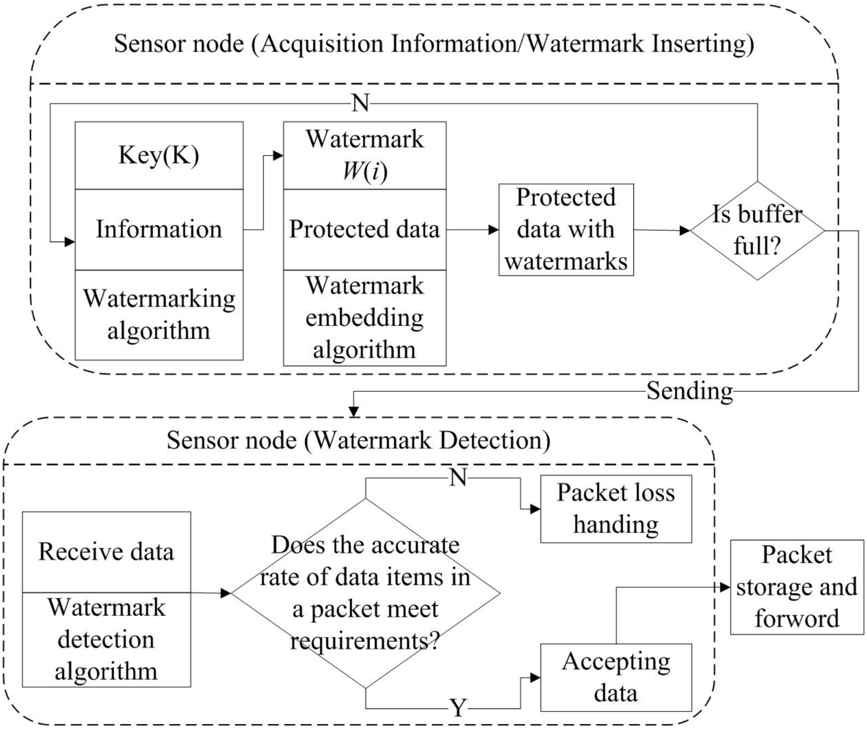

Sensor node (Acquisition Information/Watermark Inserting)
N
Key(K)
Information
Watermarking algorithm
Watermark W(i)
Protected data
Watermark embedding algorithm
Protected data with watermarks
Is buffer full?
Sending
Sensor node (Watermark Detection)
Receive data
Watermark detection algorithm
Does the accurate rate of data items in a packet meet requirements?
N
Packet loss handing
Y
Accepting data
Packet storage and forword

Fig. 3. The operation procedures of watermarking

Digital watermarking algorithms consist of three basic procedures: watermark generation, watermark embedding, and watermark extraction or detection. The main idea of watermarking algorithms is that watermarks are generated by watermark generation algorithms, and then are embedded into the data collected by sensor nodes. The watermark information is stored in the memory at a node before the data in this node is transmitted. The destination nodes operate the watermark detection in terms of the designated keys and parameters. Only the data with correct watermarks can be considered reliable, meanwhile, it must be eligible for storing and forwarding. Otherwise, it is considered counterfeit or damaged, and discarded directly (Feng & Potkonjak, 2003). The detailed operation procedures about watermarking are shown in Fig. 3.

In this chapter, based on alternating electric current and time window respectively, we propose two digital watermarking algorithms in wireless sensor networks for the data transmission security of Smart Grid.

4. Digital watermarking algorithm based on alternating electric current

4.1 Algorithmic process

4.1.1 Watermark generation

The electric current on electric transmission line is alternating, which means its current value and direction change periodically. In addition, the electric current is a monotonic function of time, a sine trigonometric function. That is to say, both current intensity and orientation are a unique value at any given time within a cycle. These features of alternating electric current are ideal for watermark generation (McDaniel & Mclaughlin, 2009). We use the alphabet *I* to represent the electric current. Physically, it is a vector that contains the information of both its value and direction. As electric current is periodic, the value may be equal although the current direction is different at different times. In order to generate diverse watermark information, we make some special changes to the reverse electric current before watermark generation (Cox et al., 2007).

Suppose the format of a sent packet is Packet = (Head, Send_Data), where Head is the packet's head including routing information, data type, and packet length etc. Send_Data is the data which the sensor node sends at a time. It is also the buffer content at the sensor node when its buffer is full. Send_Data contains a variety of collected data items, and the current at the moment of the data item acquisition. Send_Data = (Data[1], Data[2], Data[3], …, Data[*n*]), where Data[*i*] (*i* = 1, 2, …, *n*) represents one of the data items collected by the sensor node. Its data type definition is shown in Fig. 4.

```
Typedef struct Data_info {
    I;
/*The current at the moment of the data item acquisition, a vector*/
    Kernal_data;
/*Kernel data, which is the protected data*/
    Flag;
/*Boolean value, which identifies whether this data item has
watermark*/
} Data[i]
```

Fig. 4. The data type definition of Send_Data

Suppose that a packet consists of Head and *m* data items in a collection cycle. The watermark generation algorithm is described as follows:

1. Taking out each data item from Send_Data.
2. Using single hash function to compute its hash value, according to the key and electric current *I* at the moment of data collection. This step can be described as a program statement hsh[*i*] = Hash(Key, Data[*i*].*I*).

3. Getting the most significant bit in hsh[*i*]. The corresponding statement is MSB(hsh[*i*]).
4. Taking Num binary bits from MSB(hsh[*i*]), then XOR them. The result *W*[*i*] is the watermark of data item Data[*i*].

The detailed steps in the watermark generation algorithm are shown in Fig. 5.

```
Generate_W (Data[i], Key, Num)  {
 If (Data[i].I > 0) then
     I' = Data[i].I
 Else
     I' = Translate(Data[i].I)
/*Making some changes to the reverse current in order to generate a
variety of watermark*/
 End if
 hsh[i] = Hash(Key, I' );
 W[i] = Produce_W(MSB(hsh[i]), Num)
/*MSB(hsh[i]) means obtaining the most significant bit, and the function
Produce_W means XOR to generate watermark of Data[i].*/
 Data[i].Flag = 0
/*Initializing the value of the flag bit before watermark embedding*/
}
```

Fig. 5. The meta-code of the watermark generation algorithm

4.1.2 Watermark embedding

To minimize the varying range of data, only the watermark at the least significant bit of data item is embedded. Considering the fact that the energy at sensor nodes is limited, the watermark algorithm should be designed concisely, so we take the following two measures:

1. Selecting some items randomly from the data items (i.e., Data[1], Data[2], Data[3], …, Data[*m*]) to embed watermark, which can reduce the computational complexity.
2. Deriving the least significant bit of data item Data[*i*], which will be embedded watermark; and selecting some fixed binary bits of the least significant bit, which are the watermark embedding positions. That can simplify the watermark extraction.

The embedding algorithm uses the same key as the generation algorithm. The scaling parameter u is selected in terms of the requirements to security, which is used to control the percentage of data items needed to be embedded watermark. We only insert watermark into the data items whose random numbers can divided by u. Macroscopically the value of u reflects the dense degree of data items embedded watermark in a packet. Larger the value of u is, and smaller the probability of related data items inserted watermark is. After determining which data item should be inserted into watermark, we can get the watermark information by using the algorithm in Fig. 5. Next we insert it into the fixed position of the data item's LSB (the least significant bit). The detailed steps of the watermark embedding algorithm are shown in Fig. 6.

```
Embed_W (Send_Data , Key, Num, $u$) {
 For $i$ =1 to $m$
   Generate_W (Data[$i$], Key, Num)    /*Generating watermarks*/
   MSB_Data = MSB(Data[$i$].Kernal_data)
   rd = random (Key, Data[$i$].$I$, MSB_Data)
   If (rd mod $u$ = 0)  then
/*Determining which data item will be embedded watermark*/
       Select_Bits (LSB(Data[i].Kernal_data))
/*Selecting some fixed binary bits from LSB as the embedded
positions*/
        Embed watermark WM[$i$] in the fixed bits
        Flag = 1
   End if
 End for
}
```

Fig. 6. The meta-code of the watermark embedding algorithm

4.1.3 Watermark detection algorithm

The structure of received packet is the same as that of sent data. In order to illustrate it clearly, we describe a received packet as Packet_R = (Head, Receive_Data), where Receive_Data is the content of received packet with watermark information. The watermark detection process is as follows:

1. The node reads each data item in a received packet.
2. Retrieving the state of each data item's flag bit. If the flag is 1, the function Get_LSB (Data[i]) obtains the data item's watermark W', and compare W' to W = Generate_W (Receive_Data[i], Key, Num). If they are same, it means the data item Data[i] is safe.

However, the security of a data item does not ensure the packet is safe. In order to measure the security of a packet, we introduce a threshold parameter P. It represents the correct watermark rate of all data items in a packet, which shows the authentic level of all data items in a packet. If the watermark detection rate of all data items in a packet is larger than P, we say that the credibility of this packet's contents is fully consistent with the requirements. The packet is correct and acceptable; conversely, it should be dropped by the corresponding node. The meta-code of the watermark detection algorithm is shown in Fig. 7.

4.2 Performance analysis

We employ Matlab7.0 as our experimental network environment. The coordinate area of simulation configuration is 40m * 100m, and a total of 50 sensor nodes are distributed uniformly. We draw out 300 packets to analyze, and initialize each node's energy to 2 joules. In order to facilitate and simplify the simulation, the electric current value is measured as follows. The watermarking algorithm makes use of its value directly if the current value is positive. When it is negative, we multiply the current value by a constant, then use the transformed values to generate watermark.

```
Detect_W (Receive_Data, Key, Num, $P$) {
 Right_count = W_count = 0
/*Right_count is the total number of data items that can be correctly
detected the watermark in a packet. W_count is the total number of data
items that contains watermark in a packet*/
 For $i$ = 1 to $m$
   If (Receive_Data[$i$].Flag = 1) then
      $W'$ = Get_LSB (Receive_Data[$i$])
      W_count = W_count + 1
      $W$ = Generate_W (Receive_Data[$i$], Key, Num)
   End if
   If ( $W'$ = $W$ ) then   /*The watermark information is correct*/
       Right_count = Right_count +1
    End if
   End for
   If (Right_count/W_count > $P$) then
      Receive this reliable packet and forward it
   Else
      Drop this packet
   End if
}
```

Fig. 7. The meta-code of the watermark detection algorithm

Before evaluating the performance of this watermarking algorithm, it is necessary to verify that it is reasonable and viable for data security by experiments. Here we can reach the experimental goal by comparing the received watermarked message from a certain data packet to its original watermark message, as shown in Table 1 (three data packets are selected, i.e., Packet 1, Packet 2, and Packet 3). Next, we analyzed the algorithm's performance from three aspects: the security of algorithm, the network throughput, and the node energy consumption.

4.2.1 The security of algorithm

We evaluate this algorithm's security according to the statistics of its probability of handling the data correctly. For this purpose, we introduce the formula of algorithm detection rate.

$$P\_Dective = \frac{Dective\_Sum}{\mathrm{Re}ceive\_Sum} \tag{1}$$

Wherein *P_Dective* is the detection rate of packets, and *Dective_Sum* is the number of packets whose watermark are correctly detected. *Receive_Sum* is the total number of received packets. In this experiment, we add 7 attacking nodes, 3 camouflage nodes, and assign different values to the embedding parameter *u* at the same time. The results are shown in Fig. 8. Different values of parameter *u* have different impacts on the detection rate to some extent. Larger the value of parameter is, and less the detection rate is. The reason is that the larger value of *u*, the smaller probability of embedding watermark in data items, correspondingly, the less amount of watermark information the packet contains. But when *u* takes a smaller value, the detection rate is still quite large (nearly above 95%) regardless of the number of packets increasing. From the experimental results, we are able to anticipate that this algorithm can efficiently operate with a high security when the proper value of *u* is chosen.

Watermarks in Packet 1		Watermarks in Packet 2		Watermarks in Packet 3	
Original	Received	Original	Received	Original	Received
1010	1010	1110	1110	1011	1011
1101	1101	1001	1000	1001	1001
1000	1001	1111	1011	0000	0000
0110	0110	0001	0001	1110	0110
1011	1011	0011	0011	1010	1010
1010	1010	1011	1011	0010	1010
1011	1000	1001	1001	1000	1000
1001	1001	1000	1000	1101	1101
0110	0110	1110	0111	0111	0110
0111	0011	0000	0101	1111	1111

Table 1. The comparison received watermarks to original watermarks in 3 data packets

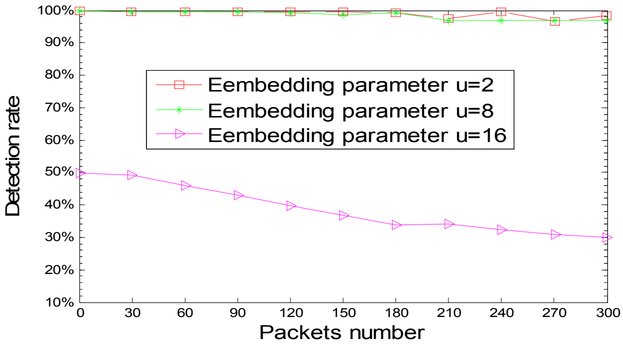


Fig. 8. The comparison of the securities

4.2.2 Network throughput

An important advantage of digital watermarking is that it does not increase the burden of network transmission. In this algorithm, we replace the most important part of the carrier with the watermark through the least significant bit (LSB) method, which does not import an additional data for the original data. Therefore, digital watermarking technique can maintain the throughput of the original network well, as shown in Fig. 9.

In general, the throughput of the networks without watermark information is slightly larger than that of ones with embedded watermark. At the beginning, the number of the nodes forwarding packets is smaller, such that the network is unimpeded and faster. Thus, the network throughput increases rapidly. But as more and more nodes begin to transmit packets, the number of sent packets increases, which leads to a slight decline in the throughput of the networks with watermarks because of the watermark embedding and the data operation frequently. At last, with the end of data collection, forwarding, transportation, and processing etc, the network throughput becomes less and less. From the experimental results, the digital watermarking technique can effectively protect the packet transmission. Moreover, it does not increase the burden of network throughput.

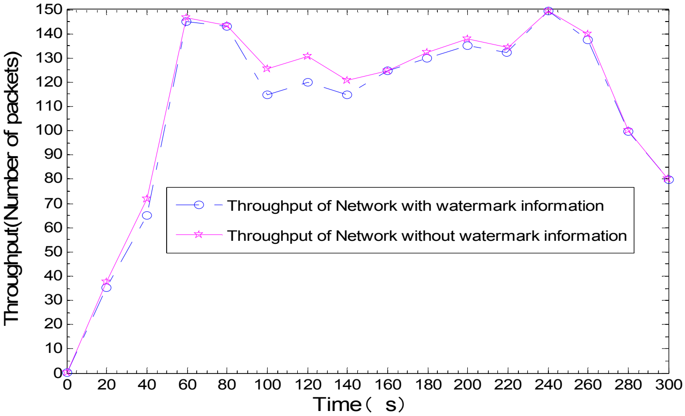


Fig. 9. The comparison of the network throughputs

4.2.3 The node energy consumption

Since the complexity of this watermarking algorithm is $O(m)$, it does not increase the energy cost at sensor nodes, when processing the watermark information. In addition, the watermark is directly embedded into the data item, which does not take up additional storage space at the node. So the node energy is mainly consumed on data transmission process. Therefore, the digital watermarking technique can well meet the requirement that the energy at sensor nodes is limited in wireless sensor networks. Table 2 is the energy consumption statistics of some nodes.

Node number	Node energy consumption (with watermark)	Node energy consumption (without watermark)
1	90	85
3	35	31
6	88	86
7	20	20
9	80	65
12	78	75
16	105	96
23	43	39
25	57	50
33	113	100

Table 2. The energy consumption at a part of nodes (unit: micro joule)

In the experiment, the energy consumption at the nodes that communicate with lots of neighbours is higher. On the contrary, the energy consumption at the nodes with less traffic is lower. Overall, the differences of energy consumption are quite slight in spite of the node with watermark or not.

5. Digital watermarking algorithm based on time window

5.1 Algorithmic process

According to the characteristics of time zone storage format of packets in wireless sensor networks and the digital watermarking, we proposed another new digital watermarking algorithm based on time window. At first, we defined the format of packets with encapsulated format, and divide the packet into eleven parts. The contents of each part are described in Table 3.

1	2	3	4	5	6	7	8	9	10	11

1: Beginning mark of message	7: Group ID
2: ACK	8: Length of data
3: Destination address	9: Content of data
4: Source address	10: CRC
5: Packet type	11: End mark of message
6: Time of sending packet	

Table 3. The definition of packet format

We use one byte to store the sending time of the packet, and set it to the float type. If we operate the lowest bit of this byte with 0 or 1, its value will be certainly changed. However, this change is quite slight, only between -0.3% and 0.3%. The higher accuracy we use, the smaller impact it will be. The offset value is always at the range of the sensor's tolerance deviation. Time information is recorded in the 6th fixed field of the packet format, so it is the lowest bit of the value. In this case, the sensor for different usages will not be affected even if the lowest bit of the time information is changed.

And then, we introduce the watermark embedding and detection algorithms based on time storage format. At first, we deal with the data area (i.e., the 9th field) of a packet by the MD5 algorithm, and get a unique mapping. After that, let the value which is generated by this mapping XOR the watermark. At last, embed the result into the hidden bit. The mapping is one-way and irreversible, such that the watermark adding to this mapping can ensure the better reliability of the transmission (Katzenbeisser & Petitcolas, 2000).

5.1.1 The watermark embedding

The watermark embedding procedure consists of the following four steps (see also Table 4 in detail):

1. Firstly, we select the 4th field of a packet as the original information M, and then operate the original information M with the key K and the watermark generation algorithm G. Then we get the watermark W.

```
Input: Information packet, and the key K
Output: The document embedded information, packet′

1. W ← G (M,K)
2. If (the data buffer is not full) then
3.     Continue collecting to fill the data buffer
4. Else
5.     X′ ← message_hash (X,K)
6. End if
7. For  i = 0 to 8
8.   If (the i′s value is less than the size of the data buffer) then
9.       $X_i'' = X_i' \oplus W_i$
10.      $i = i + 1$
11.  Else
12.      Go to loop 15
13.  End if
14. End for
15. $T_{lsb} = Em(X'', K)$
16. Using CRC algorithm to calculate the designated data in the packet
17. Output packet′
```

Table 4. The process of the embedding watermark based on time window

2. Secondly, calculate the hashing value of the data items of the packet with the MD5 algorithm, and then get a hashing value hsh which is mapping with the data items of the packet.
3. Third, let hsh XOR *W*, and embed the results into the lowest bit of the time information (i.e., the 6th field of the packet).
4. Finally, use CRC algorithm to check from the 3rd to the 9th field in the packet, and put the results into the 10th field in it.

5.1.2 The watermark extraction and detection

After transmitting through the relay nodes, packets will reach the base station. We will extract and detect the watermark.

As shown in Table 5, we use CRC algorithm to check from the 3rd to the 9th field data in the packets, and compare the results with the content in its 10th field. If they are not same, the packet should be discarded. Otherwise, we get the embedded data from the lowest bit in the time information, and then extract the watermark *W′* with the watermark extraction algorithm. At last, we compare *W′* with W''. If they are equal, the packet is accepted; if not, it will be discarded.

5.2 Performance analysis

We investigate the efficiency of the algorithm and its network performance by simulation experiments. The experimental configuration in Matlab7.0 is described as follows. The coordinates area is 40m * 100m, and a total of 50 sensor nodes are distributed. There are 300

```
Input: The document embedded information packet', and the key $K$
Output: The information packet

1. Using CRC algorithm to calculate the designated data in packet'
2. Compare its value with the content in its CRC field
3. If (they are same) then
4.    Go to loop 8
5. Else
6.    The packet loss is marked
7. End if
8. $W' \leftarrow get\_data(T_{lsb})$
9. $W \leftarrow G(M,K)$
10. $X' \leftarrow$ message_hash $(X,K)$
11. $W'' = X' \oplus W$
12. If $W'' = W'$ then
13.    The watermark is right, and this packet is accepted
14. Else
15.    The packet loss is marked
16. End if
17. Output packet
```

Table 5. The process of watermark extraction and detection based on time window

packets are drawn out for analysis, and the size of each packet is set to 128 bit. In addition, each node's energy is initialized to 2 joules. We take the embedded value as the source node's ID, and regard the collecting time of data as its sending time approximately. When the parameter configuration is ready, we start to embed watermark and to transmit data. During the transmission, the energy consumption, the processing speed, the time consumption, and all received data at the base station are recorded in different document files.

Similarly, by comparing the received watermark from a received data packet to its original watermark, as shown in Table 6, it can be seen that this watermark algorithm is also reasonable and viable for data security, because it is able to identify those malicious packets. At the end, we probe its performance from four aspects: the security of algorithm, the network throughput, the network delay, and the node energy consumption.

5.2.1 The security of algorithm

The main objective of this algorithm is to find the counterfeit or damaged data and discarded them directly when there are malicious node attacks during network transmissions. Fig. 10 is the comparison of packet loss between the transmissions with embedded watermark and ones without watermarking in a simulation network environment. In this algorithm, the packet loss in the base station consists of two parts: one is the packet loss in the network communication, and the other is the received packets that are malicious and discarded directly. Seen from Fig. 10, the number of packet loss with embedding watermark is more than that without digital watermarking. We should also note

Original watermark	Received watermark
1	1
0	0
0	0
1	1
1	1
1	0
0	0
1	1
0	0
1	1
0	1
0	0
1	1
1	0
1	1

Table 6. The original watermarks and received watermarks in 15 data packets

that the number of packet loss without watermarking algorithm only is the number of the lost packets during the network communication. However, the number of packet loss in wireless sensor networks with watermarking algorithm not only contains the lost packets during the network communication, but only includes the packet loss during the data processing at the base station. In short, from the experiments we can conclude that this watermarking algorithm for wireless sensor networks can implement the function of identifying and discarding the malicious packets.

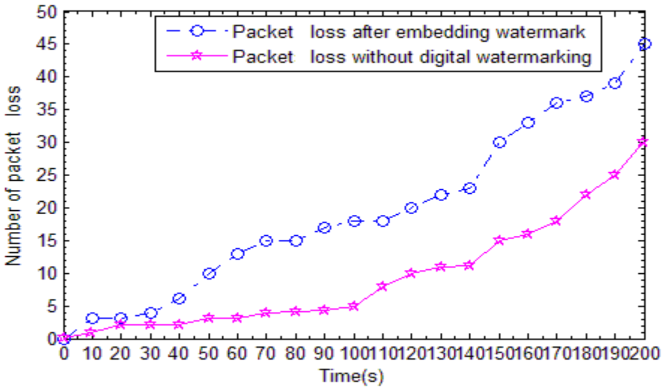


Fig. 10. The comparison of packet loss

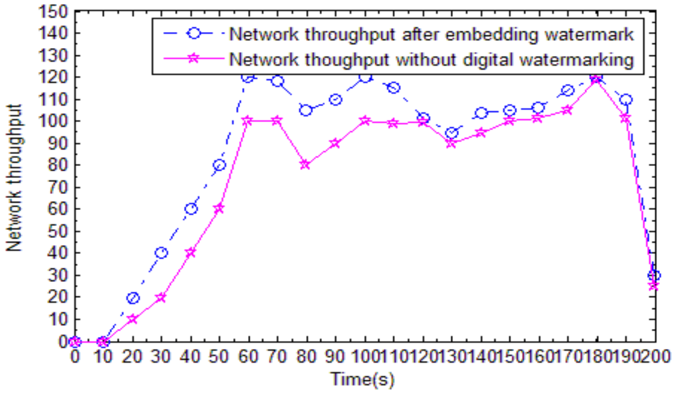


Fig. 11. The comparison of network throughput

5.2.2 Network throughput

It is shown in Fig. 11 that the comparison of network throughput between the wireless sensor networks with embedded watermark and that without watermarking in a simulation network environment. The packets which the whole network can send are changed as the simulation time increases. And the network throughput without watermarking is slightly higher than that containing watermarking, with the maximum throughput difference 20. The reason is that the nodes that transmit and forward packets are less at the beginning, and the network is smooth, fast, and less delay such that the data throughput increases slowly. However, as more nodes join into the transmission of packets, the whole network can send more and more packets. Due to embedding the watermark frequently, the throughput with watermarking algorithm will be slightly slower than that without watermarking. Finally, due to the end of data collection, the number of nodes joining transmission and forwarding gradually become less such that the whole network send less and less packets. And the network becomes smooth with less delay and unaffected data throughput.

5.2.3 Network delay

Generally speaking, the network delay is the interval between the sending time and the receiving time of packets in end-to-end network communication, which consists of the propagation delay, the transmission delay, the queuing delay, and the routing execution delay etc. Fig. 12 is the comparison of network delay between the wireless sensor networks with embedded watermark and that without watermarking. As the simulation time increases, the number of packets which the whole network can send is increasing. At this time, the network delay with watermarking is slightly more than that without watermarking. This reason is that the nodes that transmit and forward packets are less at the beginning, and the network is smooth and fast with less delay. The more nodes join into the transmission, the more packets the base station receives. Due to embedding the watermark frequently in wireless sensor networks with digital watermarking, the network throughput decreases, on the contrary, the network delay increases. Finally, since the transmission comes to a close, the network recovers with less delay.

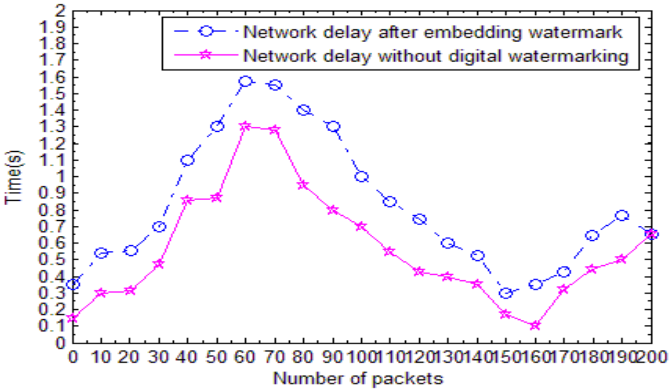


Fig. 12. The comparison of network delay

5.2.4 The node energy consumption

It is shown in Fig. 13 that the comparison of node energy consumption between wireless sensor networks with embedded watermark and that without watermarking in a simulation network environment. The nodes sending data can select their neighbour nodes according to the routing and calculated hop-count to transmit and forward packets. This figure shows that the nodes that frequently use the same path will consume more energy. When a sensor node is failure, the node will automatically select the other neighbor nodes to transmit. However, it will prolong the survival of the entire network. From the figure, we can find that the node energy consumption in wireless sensor networks with watermarking algorithm does not differ much from that without watermarking. Therefore, the digital watermarking based on time window can well meet the requirement that the energy consumption at sensor nodes is limited in wireless sensor networks.

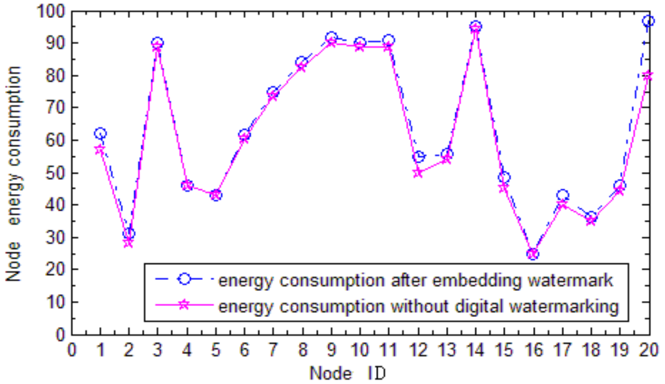


Fig. 13. The comparison of node energy consumption

6. Conclusion

This chapter begins with a general introduction to Smart Grid, wireless sensor networks, and their security issues. Next it is followed up by the basic principle of digital watermarking applied to Smart Grid. The chapter focus on two digital watermarking schemes based on alternating electric current and time window, respectively. Both of them consist of watermark generation, watermark embedding, and watermark extraction or detection algorithms. Afterward, we evaluate the two watermarking schemes from their security, network throughput and energy consumption etc by lots of simulation experiments. The results show that it is reasonable and beneficial to apply digital watermarking to handle the data security in Smart Grid. The watermarking schemes we propose fully take into account the characteristics of both Smart Grid and wireless sensor networks. With the development of wireless sensor networks and digital watermarking techniques, we believe that digital watermarking would play a more and more important role in Smart Grid.

The data communication security in Smart Grid is a comprehensive and complicated research topic. Although some research fruits are obtained in this chapter, there still remain some problems needed to solve. The digital watermarking schemes proposed in this chapter could bring some distortions for data when considering the interference from communication noise. In addition, the robustness of watermark is not explored yet so far. How to design a robust watermarking scheme without distortion is our future work.

7. Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61171075, the PhD Program Foundation of Ministry of Education of China under Grant No. 200804971030, and the Visiting Scholarship of State Key Laboratory of Power Transmission Equipment & System Security and New Technology (Chongqing University, China) under Grant No. 2007DA10512710407.

8. References

Akyildiz, I. F.; Su, W. & Sankar, Y. (2002). Wireless Sensor Networks: a Survey. *The International Journal of Computer and Telecommunications Networking*, Vol.38, No.4, (2002), pp. 393-442, ISSN 1389-1286

Amin, S. M. & Wollenberg, B. F. (2005). Toward a Smart Grid: Power Delivery for the 21st Century. *IEEE Power and Energy Magazine*, Vol.3, No.5, (Sept.-Oct. 2005), pp. 34-41, ISSN 1540-7977

Cox, I.; Matthew, M. & Bloom, J. (2007). *Digital Watermarking and Steganography (2nd)*, USA: Morgan Kaufman Publishers, ISBN 978-0-12-372585-1, San Francisco, CA

Divan, D. & Johal, H. (2006). A Smarter Grid for Improving System Reliability and Asset Utilization, *Proceedings of Power Electronics and Motion Control Conference*, ISBN 1-4244-0448-7, Shanghai, Aug. 2006

Feng, J. & Potkonjak, M. (2003). Real-Time Watermarking Techniques for Sensor Networks, *Proceedings of IEEE Int. Conf. on Security and Watermarking of Multimedia Contents*, Santa Clara, CA, USA, Jan 2003

Kleider, J. E.; Gifford, S. & Chuprun, S. (2004). Radio Frequency Watermarking for OFDM Wireless Networks, *Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, ISBN 0-7803-8484-9, USA, May 2004

Katzenbeisser, S. & Petitcolas F. A. P. (2000). *Information Hiding Techniques for Stegonagraphy and Digital Watermarking*, Artech Print on Demand, ISBN 1-58053-035-4, London

McDaniel, P. & McLaughlin, S. (2009). Security and Privacy Challenges in the Smart Grid. *IEEE Security & Privacy*, Vol. 7, No. 3, (May-Jun 2009), pp. 75-77, ISSN 1540-7993

Perrig, A.; Stankovic, J. & Wagner, D. (2004). Security in Wireless Sensor Networks. *The ACM Communications*, Vol.47, No.6, (June 2004), pp. 53-57

Wang, Y.; Attebury, G. & Ramamurthy, B. (2006). A Survey of Security Issues in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, Vol.8, No.2, (Feb. 2006), pp. 2-23, ISSN 1553-877X

Xiao, R.; Sun, X. & Yang, Y. (2008). Copyright Protection in Wireless Sensor Networks by Watermarking, *Proceedings of IEEE International Conference*, ISBN 978-0-7695-3278-3, New Zealand, Aug 2008

Xiao, X.; Sun, X.; Yang, L. & Chen, M. (2007). Secure Data Transmission of Wireless Sensor Network Based on Information Hiding, *Proceedings of 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, ISBN 978-1-4244-1024-8, Philadelphia, PA, USA, Aug 2007

Chen, Y. C.; Chuang, C. C.; Chang, R. I.; Lin, J. S. & Wang, T. C. (2009). Integrated Wireless Access Point Architecture for Wireless Sensor Networks, *Proceedings of ICACT 2009*, ISBN 978-89-5519-138-7, South Korea, Feb. 2009

Zia, P. & Zomaya, A. (2006). Security Issues in Wireless Sensor Networks, *Proceedings of the Int. Conf. on System and Network Communications*, ISBN 0-7695-2699-3, French, Oct. 2006