1982-12

Refinements in a DCT based non-uniform embedding watermarking scheme

Giakoumakis, Michail D.

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/1112

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REFINEMENTS IN A DCT BASED NON-UNIFORM EMBEDDING WATERMARKING SCHEME

by

Michail D. Giakoumakis

March 2003

Co-Advisors:	Roberto Cristi Ron J. Pieper Craig Rasmussen

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Refinements in a DCT Based Non-Uniform Embedding Watermarking Scheme			**5. FUNDING NUMBERS**
6. AUTHOR(S) Giakoumakis, Michail D.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8.PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

Perceptual watermarking is a promising technique towards the goal of producing invisible watermarks. It involves the integration of formal perceptual models in the watermarking process, with the purpose of determining those portions of an image that can better tolerate the distortion imposed by the embedding and ensuring that the watermarking will inflict the least possible degradation on the original image . In a previous study the Discrete Cosine Transform was used, and the watermark embedding was done in a non-uniform manner with criteria based on both the host image and the watermark. The decoder model employed made use of apriori access to unmarked and marked images as well as to the watermark. A fair level of success was achieved in this effort. In our research we refine this scheme by integrating a perceptual model and by proposing a modification to the decoder model that makes possible the successful recovery of the watermark without apriori access to it. The proposed perceptual scheme improves the watermark's transparency while at the same time maintains sufficient robustness to quantization and cropping. The proposed semi-blind variation offers adequate transparency and robustness to quantization, but its performance against cropping is considerably degraded.

14. SUBJECT TERMS Digital Image Watermarking, JPEG Compression, Discrete Cosine Transform, Perceptual Models, Human Visual System			**15. NUMBER OF PAGES** 143
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

REFINEMENTS IN A DCT BASED NON-UNIFORM EMBEDDING WATERMARKING SCHEME

Michail D. Giakoumakis
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATH
and
MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2003

Author: Michail D. Giakoumakis

Approved by: Roberto Cristi
Co-Advisor

Ron Pieper
Co-Advisor

Craig Rasmussen
Co-Advisor

Clyde Scandrett
Chairman, Department of Mathematics

Dan C. Boger
Chairman, Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Perceptual watermarking is a promising technique towards the goal of producing invisible watermarks. It involves the integration of formal perceptual models in the watermarking process, with the purpose of determining those portions of an image that can better tolerate the distortion imposed by the embedding and ensuring that the watermarking will inflict the least possible degradation on the original image. In a previous study the Discrete Cosine Transform was used, and the watermark embedding was done in a non-uniform manner with criteria based on both the host image and the watermark. The decoder model employed made use of apriori access to unmarked and marked images as well as to the watermark. A fair level of success was achieved in this effort. In our research we refine this scheme by integrating a perceptual model and by proposing a modification to the decoder model that makes possible the successful recovery of the watermark without apriori access to it. The proposed perceptual scheme improves the watermark's transparency while at the same time maintains sufficient robustness to quantization and cropping. The proposed semi-blind variation offers adequate transparency and robustness to quantization, but its performance against cropping is considerably degraded.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to acknowledge

- Dr. Roberto Cristi, Dr. Ron J. Pieper and Dr. Craig Rasmussen for their expert knowledge and guidance.
- Professors Vasiliki Kanta and Xiroudakis Nikolaos for their patience and determination. Thank you for believing in me.
- Maria, my wife, for her moral support.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The basic ideas behind digital watermarking have emerged steadily over the last few decades. In order for the watermarking to be dependable it is imperative that it has certain characteristics. The most important of these are transparency of the watermark (it should be imperceptible to the Human Visual System) and robustness against common tampering with the image. Perceptual watermarking is a promising technique towards invisible watermarks. It involves the integration of formal perceptual models in the watermarking process with the purpose of determining those portions of an image that can better tolerate the distortion imposed by the embedding and thus make sure that the watermarking will inflict the least possible degradation on the original image.

In a recent study [1] the Discrete Cosine Transform was used and the watermark embedding was done in a non-uniform manner with criteria based on both the host image and the watermark. The decoder model employed made use of apriori access to unmarked and marked images as well as to the watermark. A fair level of success was achieved in this preliminary effort. With our work we provide a technique for the integration of a perceptual model in this proposed preliminary scheme and investigate the feasibility of employing a semi-blind structure.

The perceptual model we incorporated in the watermarking process is based on the VMSE, which stands for Visual Mean Square Error and corrects the inefficiency of the Mean Square Error by taking advantage of the unitary property of the DCT transform. The analytic form for the VMSE can be written as

$$VMSE = \frac{1}{64 \times \mathrm{K}} \sum_{k=1}^{K} \sum_{i=1}^{8} \sum_{j=1}^{8} W[i,j,k] \times (C_o[i,j.,k] - C_W[i,j,k])^2,$$

where i=1,2,..8 and j=1,2,…8 are indices for the elements of an 8x8 DCT block in the original image, k=1,2,…,K is index identification for the 8x8 blocks of the original image, C_o and C_w are the 8x8 block DCT representations of the original and watermarked images respectively. The weighting coefficients W[i,j,k] are given by

$$W[i,j,k] = \left(\frac{1}{s[i,j,k]}\right)^2,$$

where s[i,j,k] are the “slacks” calculated through the Corrected Watson Distance perceptual model. The Corrected Watson Distance perceptual model is the outcome of an experimental luminance correction we applied to the well-known Watson perceptual model. [2]

In the context of our scheme, the embedding takes place in the DCT domain, which is also used by the JPEG standard, and allows for the exploitation of the domain's particular characteristics for the attainment of watermark transparency and robustness. Both the watermark and the image are transformed using the DCT. To integrate the VMSE model we modified the method used in the preliminary scheme for rating the 8x8 blocks of the DCT of the image according to their Priority Coefficient (PC).

For each 8x8 block of the DCT coefficients of the image, we calculate the Complexity Factor (CF), a metric for measuring the capacity of each block to receive watermark coefficients. The CF is defined as the combination of the Total Visual Strength Factor (TVSF) and the Embedding Visual Correction Factor (EVCF) through the power relation

$$CF_k = TVSF^w \times EVCF^{1-w},$$

where $w \in [0,1]$ is a power weighting factor determined experimentally. The TVSF uses the VMSE model to provide a pooling of the ability of all block DCT coefficients to withstand distortions, while the EVCF aims to account for the fact that only some of the block DCT coefficients are used for embedding.

Additionally, following the rationale of the preliminary scheme, for each block of the cover image we calculate the Center of Interest Proximity Factor (CIPF), which is a measure of significance of each 8x8 block with respect to cropping resistance. We first determine the Euclidean distance r between the center of the block and the Center of Interest (CI). In our experiments we assumed that the CI is the center of the image. The Euclidean distance r, is then normalized over the diagonal (i.e. the maximum possible distance within the image) to produce a normalized value *rnorm*. This normalized distance is then processed by a transformer with characteristic function *f*, where

$$f(rnorm) = -\frac{1}{p} \cdot \tan^{-1}(k \cdot (rnorm - \frac{2}{3})) + \frac{1}{2}.$$

We call this index *f(rnorm)* as CIPF.

The CF of each 8x8 block is scaled by the CIPF to produce a Priority Coefficient (PC), which is attached to the block and contains all the information that is required for its rating. The blocks are now sorted by descending order of their PC.

The DCT coefficients of the watermark are sorted according to magnitude and divided into *m* groups of descending magnitude with equal number of elements. We then form embedding sets of coefficients. Each set contains *m* coefficients, one from each group. The sets are then embedded into *m* frequency coefficients of the 8x8 DCT image blocks. Embedding in the lowest frequencies allows for higher robustness of the watermark against JPEG compression, since these coefficients are the least affected by the quantization process. However, the lower frequencies are the most perceptible ones, but we manage to compensate for the latter by appropriately adjusting a weighting factor a.

The decoder works in reverse order and requires both the original image and the watermark. The level of detection is based on the correlation coefficient ?, which is given by

$$r = \frac{\sum_i \sum_j W(i, j) Wr(i, j)}{\sqrt{\sum_i \sum_j [W(i, j)]^2 \cdot \sum_i \sum_j [Wr(i, j)]^2}},$$

and is a measure of similarity between the watermark and the extracted pattern.

For the semi-blind variation the sorting of the DCT coefficients is omitted, allowing the decoder to extract the watermark without apriori access to it. The resulting scheme maintains adequate transparency and robustness to quantization, but its performance against cropping is considerably degraded.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Electronic distribution of multimedia content is an important byproduct of the confluence of recent technological advances. The growth of the Internet has made communication easier and more extensive than ever before. However, the rapid growth of the demand for quick and reliable dissemination of digitized media products has generated a challenging need for stronger protection of intellectual property (IP) rights for audio, video, images, and other digital documents. To satisfy this need, an integrated system design is necessary [3]. The basic IP protection system consists of three main components. First, the media is compressed and is stored in a cryptographic container before distribution. Second, a flexible licensing mechanism is implemented to determine the credibility of those seeking access to the content. Third, digital watermarks are embedded in the media so that it can be recognized if the system is breached. A secure system design integrates these three components [4]. Under this framework, the watermarking component of the IP protection system is expected to play a very important role in the protection of copyrights.

The basic ideas behind digital watermarking have emerged steadily over the last few decades. Recently, there has been a drastically increased interest from both academia and industry in this area, as witnessed by numerous patents filed for watermarking techniques for the protection of a broad array of multimedia products. Some international organizations are even considering combining watermarking techniques with existing standards [5]. Concerning image watermarking, most recent research focuses on invisible watermarks, those that are imperceptible under normal viewing conditions. The classification of different techniques that are being used for invisible image watermarks has been treated in detail in [1]. Other surveys regarding invisible watermarking classification can be found [7], [8] and [9]. In the present thesis, we focus our attention on a special class of invisible watermarking schemes known as perceptual or image adaptive watermarks, which are specifically designed to exploit perceptual information in the watermarking process through the use of formal perceptual models.

A. PURPOSE

In order for the watermarking to be dependable it is imperative that it has certain characteristics. The most important of these are transparency of the watermark (it should be imperceptible to the Human Visual System) and robustness against common tampering with the image. In a recent study [1] the Discrete Cosine Transform was used and the watermark embedding was done in a non-uniform manner with criteria based on both the host image and the watermark. The decoder model employed made use of apriori access to unmarked and marked images as well as to the watermark. A fair level of success was achieved in this preliminary effort. The purpose of this research is to refine the DCT based non uniform embedding watermarking scheme developed in [4] by investigating whether the integration of a perceptual model could further improve the transparency of the watermark and whether the decoder model may be modified to successfully recover the watermark without apriori access to it.

In the course of this research we studied several perceptual models and the methods for their integration in the watermarking process used in the relevant literature. This background work, along with the ideas and concepts put forward in [1], served as the basis for the development of a watermarking scheme that uses a formal perceptual model to improve the transparency of the watermark while maintaining a fair level of robustness. In addition, we examined the prospect of modifying the resulting scheme so that the decoder only requires the marked and unmarked images to successfully recover the watermark. Our efforts resulted in a Semi-blind variation of the developed watermarking scheme that maintains satisfactory robustness to quantization.

B. RESEARCH QUESTIONS

There are a number of research questions that we strive to answer in this thesis. First, we investigate how image quality can be analytically assessed and whether a generic perceptual model for use in watermarking schemes can be formulated. Towards that end we unravel some methods aiming at quantifying image quality, and recast them as methods for measuring the transparency of the watermark.

Furthermore, we look into techniques designated to integrate perceptual models in watermarking and propose a general approach for incorporating any frequency domain based perceptual model into the watermarking scheme put forward in [1].

Finally, we treat the question of how to implement a semi-blind structure in the resulting perceptual watermarking scheme.

C. THESIS OUTLINE

This thesis is organized as follows:

Chapter II develops the notion of perceptual watermarks. Perceptual watermarks are the result of the incorporation of formal perceptual models in the encoding process. In this chapter, therefore, we start by defining the concept of perceptual models and presenting the principles upon which they rely. We continue by providing a systematic analysis of several commonly used perceptual models and finally link perceptual models to perceptual watermarks. Among the perceptual models considered is the VMSE, which is used in our algorithm and may provide a generic approach towards perceptual watermarks.

Chapter III provides an overview of the original watermarking scheme developed in [1] as background material to the arguments used in this thesis.

In Chapter IV we develop a technique to incorporate the VMSE model in the original watermarking algorithm and present the obtained results. This chapter in essence consists of three sections. In the first section we introduce our proposed modifications of the original algorithm and explain the rationale behind them. Our train of thought gradually leads to a new algorithm that makes full use of the VMSE model to determine the portions of the host image that can better tolerate the distortion imposed by the embedding and thus make sure that the watermarking will inflict the least possible degradation on the original image. In the next section, we test the resulting algorithm and present the obtained results. The images and watermarks used for testing are the same as the ones used in [4], both for consistency and because we found that the pixel values distribution of the proposed test images and watermarks are quite representative of the range and distribution shapes that one might come across in practice. Finally, in the third section, we compare the new algorithm to the original in terms of watermark transparency and robustness.

In Chapter V we implement a straightforward variation to the algorithm developed in Chapter IV in order for the decoder to extract the watermark from the marked image without making use of the original watermark itself, present the obtained results and compare the variation's performance to that of the original and modified algorithms.

Finally, in Chapter VII, our work is briefly summarized and conclusions following the experimental results are made. In addition, we make suggestions for possible future work based on this material.

D. EXPECTED BENEFITS OF THE THESIS

One of the most challenging tasks that the watermarking community faces is balancing the mutually competitive requirements for watermark transparency and robustness. Perceptual models seem to be effective in the examination of the feasibility of this task. In the past, perceptual models have been employed for both for comparing different watermarking schemes and minimizing the perceptual impact of the watermarks. However, none of the developed algorithms has been able to provide a dependable and complete generic approach to the integration of perceptual models in watermarking schemes.

In our research we investigate an original approach for the development of perceptual watermarks. Our approach provides a general methodology for creating perceptual watermarks that permits us to quickly update the perceptual model used in the watermarking process. Moreover, the resulting algorithm allows the user to flexibly balance the transparency and robustness requirements and may as well be us ed for further research in the field. Finally, the elimination of the need for apriori access to the watermark in the decoding process enhances the security of the algorithm. With our research we hope have contributed towards the direction of developing a composite algorithm that addresses collectively the watermarking problem.

II. PERCEPTUAL WATERMARKS AND PERCEPTUAL MODELS

It is generally accepted that, in order for an invisible watermarking scheme to be considered effective, the watermark should be

- perceptually invisible within the host media,
- statistically invisible to thwart unauthorized removal,
- readily extracted by the image owner, and
- robust to accidental and intended signal distortions incurred by the host image, e.g. filtering, compression, re-sampling, re-touching, cropping, etc. [1], [9], [10]

These desired qualities for the invisible watermarking schemes are mutually competitive and cannot be clearly optimized at the same time. A reasonable compromise is always a necessity [9]. Earlier approaches addressed the issue by applying common sense rules for the choice of the host image's frequency bands or spatial elements used for embedding. Typically, these rules favoured the embedding of the watermark data into the least-significant bits of the host image's pixels (e.g., in [11]) or into the medium spatial frequency bands of the host image (e.g. in [12]). However, these approaches exhibit relatively low robustness or don't provide a clear indication as to exactly where the watermark should be hidden and as to what extend the host image can be modified to find the necessary compromise between the robustness and transparency requirements [13]. Watermarks produced by such schemes are generally referred to as image-independent watermarks. The requirement to address the issue gave rise to the development of watermarking schemes that incorporate the characteristics of the Human Visual System (HVS) in the watermark encoder design through the employment of perceptual models. Perceptual models aim to provide an analytic measure of the perceptual competence of the HVS and are used by the encoder to determine the location and maximum strength of the watermark signal that can be tolerated in every portion of the image without producing changes in the watermarked image's fidelity that are not tolerated by the HVS. The watermarks produced by such schemes are generally referred to as perceptual or image-adaptive watermarks. Perceptual watermarks take advantage of the research advances regarding perceptual models. Therefore, we begin our presentation by introducing the perceptual models and the principles upon which they rely.

A. PERCEPTUAL MODELS

The final judge of a watermarking scheme's achieved transparency is the HVS. In practice few, if any, watermarking systems produce watermarks that are perfectly imperceptible. However, a watermarking system can be considered perceptually successful if it manages to produce watermarks that are unnoticeable to the HVS. Consequently, there is the need to define a measure of the perceptibility of the HVS, by means of perceptual models, both for the purpose of improving and comparing different watermarking schemes.

Perceptual models can be based either on the space domain (e.g. [11]) or on a transform domain (e.g. [1], [5], [13]). Perceptual models based on the space domain are generally better in accounting for perceptual differences resulting from distortions related to slight geometric transformations (rotation, scaling, translation, skew etc), whereas those based on a transform domain are generally better in accounting for perceptual differences resulting from distortions related to the non-uniform response of the HVS. The choice of a particular visual model depends on the application it is intended to supplement and sometimes involves a trade off between accuracy and computational cost. A perceptual model that would account for the impact on perceptibility of all possible types of distortion would be very complicated. In practice, most models only account for the impact of a limited number of distortions and thus their accuracy may be limited to only a class of applications [2]. In that context, perceptual models must be considered as a means for obtaining an insight which is almost always accurate, but it may be misguiding in certain cases. As models are constantly updated and corrected, they may become a lot more accurate in the future. In the meantime, the researcher must make do with the limitations and shortcomings of currently available perceptual models and be conscious of the possibility that they might err in their estimations.

1. Definition and Purpose of Perceptual Model

Models that produce an analytic estimation of the perceptual difference between two images are generally referred to as Perceptual Models. In that sense the perceptual difference between the original and the watermarked version of an image provides a measure of the transparency of the watermark. In the watermarking framework perceptual models are applied in two different instances:

In comparing the original and watermarked picture: In this context we wish to measure the fidelity of the watermarked image to the original. This approach differs from that used in the context of image quality, which is an absolute measure of appeal and where perceptual difference criteria are used to evaluate the effects on the image quality of the applied image processing techniques. In the watermarking context, if a low quality image is watermarked then the watermarked image will also be of low quality. However, if the watermarked version of the image is indistinguishable from the original, the watermarking scheme is considered successful. In this case, perceptual models provide a measure of the similarity between the original and watermarked image.

In developing perceptually based watermarking algorithms: Such algorithms use perceptual models to determine the location and maximum strength of the watermark signal that can be tolerated in every portion of the image without producing changes in the watermarked image's fidelity that are not tolerated by the HVS. The goal is to use the perceptual models in order to ensure that the watermark will inflict as little perceptual changes in the original image as possible.

It should be pointed out that perceptual models are very subjective in nature and hence imperceptibility should not be viewed as a binary condition. A watermark having a higher or lower level of imperceptibility means that there is a lesser or greater likelihood that a given observer will perceive it. This likelihood cannot be given absolute values or direct practical meaning, since perceptual similarity is not transitive in nature. [2]

Perceptual models are based upon the notion of Just Noticeable Difference (JND). JND is a term related to psychometrics and refers to the minimum amount by which stimulus intensity must be changed in order to produce a noticeable variation in a sensory experience [14]. The implementation of the notion in watermarking lies in the fact that the process of embedding a watermark in any image can be regarded in the same way as adding noise to the image. This process leads to an alteration of the host original image.

Obviously altering a large number of pixel values arbitrarily will result in noticeable image distortions. These distortions depend on the amplitude of watermark image as well as the spectral properties of both the image and the watermark. Taking advantage of this relation, a given image can be distorted only to a certain limit without making the difference between the original and the altered one perceptible. This limit varies according to the image and watermark content and is referred to as JND.

The determination of JND needs to make inferences on the judgment of human beings. One way to do that is to conduct rigorous observations involving a large number of human subjects and analyze the results statistically. This approach, although it may produce very accurate information on image fidelity, is time-consuming, expensive and cannot easily be repeated. An alternative approach to overcome these difficulties is to use an algorithmic fidelity measure based on perceptual models. The goal of perceptual models is to predict an observer's response. In practice though, it is very difficult to predict the human judgment accurately. Consequently, visual perception models are designed to produce a measure of perceptual difference between watermarked and original images, without calibrating those distances in terms of expected results. Under this scope, perceptual models are functions that give a measure of the distance between the original and watermarked image, by taking into account the characteristics of the HVS. [13]

2. The HVS in Perceptual Models

To better understand the structure of a perceptual model, it is useful to appreciate the perception mechanisms of the HVS, which are not uniform in nature. The HVS response to spatial and temporal frequency, luminance and color varies significantly with the input and a perceptual model should account for that. When an image is processed by the HVS, the different parameters describing spatial and temporal sensitivity to chromatic and achromatic image components are used to form the visual perception of the image. In the case of still grayscale images the most important component of the HVS image processing is its response to the spatial frequency and the luminance of the image. This response is described through Contrast Sensitivity (CS). Contrast sensitivity may be defined as the reciprocal of the minimum contrast between a lighter and a darker spatial area that is required for a viewer to detect differences between the areas. The contrast of a given visual pattern is typically expressed as the ratio of the difference between the luminance of light and dark areas over the sum of their luminance [15]. Analytically,

$$C = \frac{(L - D)}{(L + D)}, \quad \langle 2.1 \rangle$$

$$CS = \frac{1}{C_{\min}}. \quad \langle 2.2 \rangle$$

where L and D are the luminances of light and dark spatial areas, respectively, C is the contrast between the lighter and darker areas, and C_{min} is the minimum contrast required to distinguish between lighter and darker areas. In a sensitive visual system, only a small luminance difference between light and dark spatial area (low contrast) is necessary for the human to detect the difference between the dark and light spatial areas. In a less sensitive visual system, a larger difference in luminance (high contrast) is necessary before the difference between the light and dark spatial areas is recognizable. Contrast sensitivity is greatly affected by the spatial frequency response of the HVS. With regard to the HVS, spatial frequency indicates the number of alternating bright and dark areas (cycles) per degree in the field of view, as shown in Figure 1[1]. Every image has spatial frequencies that present variations of brightness (alternating bright and dark areas) to the eye. The eye resolves the variations into a recognizable image. The effect of the spatial frequency in the Contrast Sensitivity is expressed by the Contrast Sensitivity Function (CSF) that plots the contrast sensitivity for all spatial frequencies. The shape of the CSF varies with many factors and is highly dependent upon a person's HVS abilities, the viewing distance and angle, the orientation and nominal

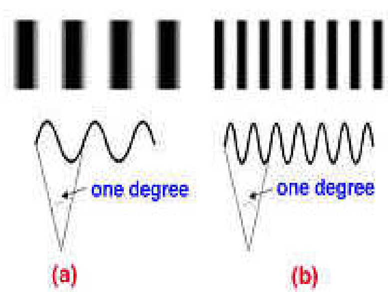


Figure 1. Spatial Frequency.

[1] Figure taken from DeValois R. L. and DeValois K. K*: "Spatial Vision"* Oxford 1998 - Oxford University Press.

luminance of the observed light and dark spatial areas, the light conditions of the observation as well as the characteristics of the display media (photo print, PC monitor etc.) [16]. The shape of the CSF determines the human perception of an image; therefore it is an important component of a perceptual model. The subjective nature of the CSF and its dependence upon so many factors make the development of a generalized perceptual model difficult. In practice, perceptual models assume standard viewing conditions and simulate the behavior of the CSF vis-à-vis spatial frequency and luminance to quantify the perceptual difference between images.

Figure 2[2] shows the shape of the CSF for a range of luminance differences between light and dark areas. For a fixed luminance, the human visual system is more sensitive in the intermediate spatial frequencies, less sensitive in the lower spatial frequencies and even less sensitive in the higher spatial frequencies. Moreover, the higher the nominal luminance of the observed spatial areas is, the more sensitive the HVS is. These characteristics of the CSF imply that higher spatial frequencies and low luminance areas of an image can withstand more distortion before the HVS detects a difference in the image. Under this scope, the CSF can be thought as a means of accounting for two non-uniform components of the HVS: its response to spatial frequency and its response to luminance.

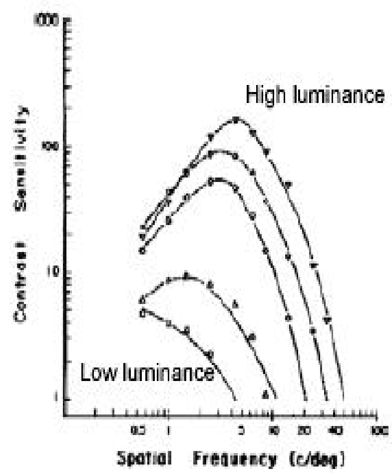


Figure 2. Contrast Sensitivity Function.

Besides the CFS there are two other phenomena that determine human perception of an image. These are Masking and Pooling. The CFS represents the human response to spatial frequency assuming that only one stimulus is present and only one characteristic

[2] Figure taken from DeValois R. L. and DeValois K. K*: "Spatial Vision"* Oxford 1998 - Oxford University Press.

of the stimulus (spatial frequency) is varied. However, an image practically consists of many spatial areas of variable dimensions, each with different characteristics and visual properties. Masking is a measure of the HVS response to one stimulus when a second stimulus is present. The presence of the second stimulus defines the context under which a stimulus is observed and affects perception. The perception of still grayscale images is influenced by two masking phenomena. These are Contrast Masking, where the presence of one spatial frequency masks the perception of another, and Luminance Masking, where local luminance masks contrast changes. The combined effect of Contrast and Luminance Masking is often called Spatial Masking [17]. Pooling is the effect on perception caused by the combined result of the multiple frequency and luminance variations that are present in an image. [2]

The effect of Luminance Masking is illustrated in Figure 3[3], which consists of a sequence of four images. For each of the four images, the spectral amplitudes are exactly the same, but their luminance is different, the highest being on the left and the lowest on the right. It is easily observed that the pattern is more distinguishable in the case of the brighter luminance. Hence, when the background luminance is higher, the same variation would be less visible than in a darker region. This phenomenon is referred to as luminance masking.

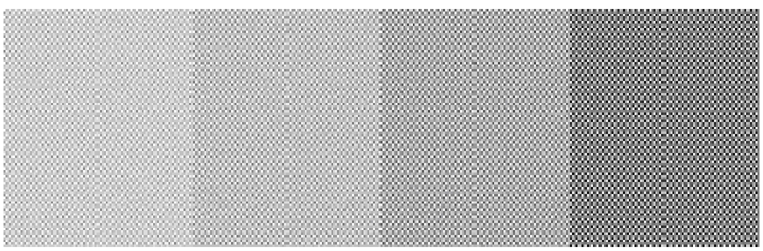


Figure 3. Luminance Masking Effect.

Contrast Masking refers to the fact that when an image component is observed in the presence of other image components with similar spatial frequency and orientation characteristics, that image component becomes less visible to the HVS. This effect is

[3] Figure taken from S. S. Henami*: "Perception of Extremely Low-Rate Images & Video: Psychophysical Evaluations and Analysis"*, Cornell University Visual Communications Lab 2001.

illustrated in Figure 4, in which uniform noise was added in the New York image. It is obvious that although the noise is uniform, its distorting effect is quite non-uniform and is highly dependent on the local image frequency structure. In the relatively flat region of the sky, the noise is clearly visible, whereas in the highly textured regions of the buildings, whose spatial frequency characteristics are closer to those of the uniform noise, the distortion is almost imperceptible by the HVS.

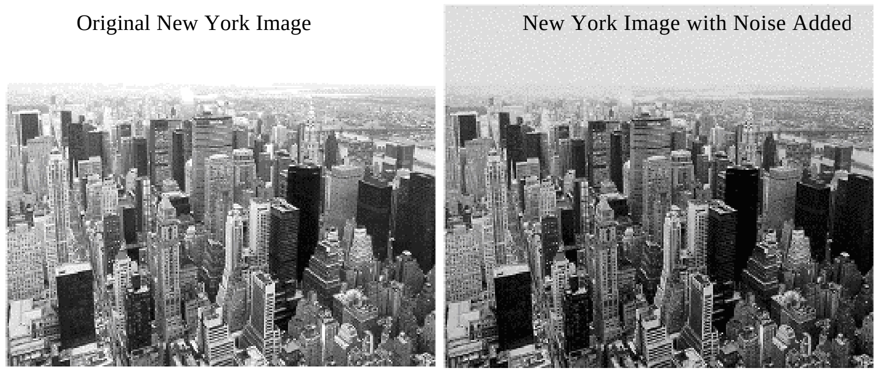


Figure 4. Contrast Masking Effect.

Finally, pooling aims to combine the perceptibilities of separate distortions, due to multiple frequency and luminance variations that are present in an image, into a single estimate of the overall change of the image. It is common to apply a formula of the form

$$D = \sqrt[p]{\sum_i (d[i])^p}, \langle 2.3 \rangle$$

where d[i] is an estimate of the difference between original and distorted image due to an individual parameter such as a spatial pixel or a Fourier or DCT frequency coefficient. [18]. Equation 2.3 is often referred to as a Minkowski summation or an L_p-norm.

The structure of visual models attempts to account for the HVS response to spatial frequency, luminance, masking and pooling in order to produce a measure of the perceptual difference between original and watermarked images [2]. To that end, visual models need to include two important components.

The Frequency and Luminance sensitivity thresholds: These represent the amount of the smallest magnitude of a spectral or luminance coefficient that is discernible in the absence of any masking and are determined experimentally through the CSF measurements.

The Masking thresholds: These correct the sensitivity thresholds to account for the effects of spatial masking. Most commonly, the masking thresholds are computed analytically through formulas derived from experimental procedures. In earlier years, the most common formula, which is known as Weber's Law, was

$$MaskingThreshold = k \times I, \langle 2.4 \rangle$$

where k is an experimentally determined constant and I is the masking stimuli (luminance or contrast). Weber's Law, more simply stated, says that the size of the masking threshold is a constant proportion of the masking stimulus value [15]. However, modern visual models use more complicated formulas to achieve more accuracy in the determination of the masking thresholds.

3. An Overview of the Existing Perceptual Models

One of the initial applications of Perceptual Models in the area of Image Processing involved their use in Source Coding and Compression, where they were used to generalize, revise and supplement simple, mathematically defined image quality measures such as the peak signal to noise ratio (PSNR) or the mean squared error (MSE). Originally, perceptual models viewed the HVS as a filter with characteristics determined by the CSF. Models based on this principle included a compressive nonlinearity to account for luminance masking, filtered the original image according to the CSF, and then calculated a difference metric between the original and compressed images. Among the first perceptual models of this kind was suggested as early as 1974 by Mannos and Sakrison in [19]. More recent implementations of this philosophy, like that proposed by Ahumada in [20], also incorporate estimates of image contrast in an attempt to capture some of the properties of contrast masking. The progress of the research on the HVS properties gave rise in the early 90s to a new set of perceptual models which attempt to model the HVS in as complete a manner as possible taking into account psychophysical and physiological evidence and thus provide a perceptually meaningful measure of image quality near the visual threshold. The most popular representative of the set of perceptual models is that of Watson introduced in [20]. The popularity of the model is due to the fact that it's based in the 8x8 block DCT domain which was incorporated in the Joint Photographic Experts Group (JPEG) standard, one the most widely used standards for lossy image compression. Recent trends involve the development of perceptual models based on the wavelet domain, like those described in [21] and [22], and the development of perceptual based on new philosophies regarding the function of the HVS, like the one introduced in [23].

Despite the difficulties experienced by perceptual models in fully reflecting the response of the HVS, they are usually reported to provide more consistent estimates of image quality than mathematically defined metrics when artefacts are near the visual threshold [24]. It was therefore logical for researchers of the watermarking community to investigate and integrate them in their watermarking schemes. At this point, we present in detail four perceptual models that have been widely used by the watermarking community and are important in the development of our modification of the original watermarking scheme presented in [4]. Most of these models are based in the 8x8 DCT block domain. The reason is that our scheme is intended to be applied in the context of the JPEG standard and thus by choosing a framework that matches the current compression standards, it's easier to design the algorithm to avoid embedding in the coefficients that are normally discarded or severely quantized during compression. In this way we can ensure robustness to this particular kind of compression [25].

a. The MSE as a Perceptual Model

Although the MSE cannot be regarded as a perceptual model in the strict sense ([24]), it is often used as a rough test of a watermarking system fidelity. If $c_o[i,j]$ and $c_w[i,j]$ are the spatial matrix representations of the original and watermarked images respectively, then the MSE between the original and watermarked image is defined as

$$D_{MSE}=\frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(c_o[i,j]-c_w[i,j]\right)^2, \quad \langle 2.5 \rangle$$

where N and M are the dimensions of the image.

The MSE as defined above is calculated in the space domain representation of the images. However, due to the fact that the DCT is unitary ([26]), the MSE can equivalently be calculated us ing the 8x8 DCT block transformation of the image; in both cases the resulting MSE is the same. If C_o [i,j] and C_w [i,j] are the 8x8 DCT matrix representations of the original and watermarked images respectively, the above property can be analytically writ ten as

$$D_{MSE} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(c_o [i,j] - c_w[i,j] \right)^2 = \frac{1}{N \times M} \sum_{k=1}^{K} \sum_{i=1}^{8} \sum_{j=1}^{8} \left(C_o [i,j] - C_w[i,j] \right)^2, \langle 2.6 \rangle$$

where K is the total number of 8x8 blocks of the image (i.e., K= $\frac{N \times M}{64}$).

Equation 2.5 clearly demonstrates that the MSE in fact treats changes in all frequency components of the images equally, without taking into consideration the non- uniform response of the HVS. Consequently, it is bound to err when used to measure perceptual differences between images [27]. In Figure 5 the original Lena image was used to create two distorted versions. One was created by adding a Low Pass filtered White Noise pattern, and the other by adding a High Pass filtered White Noise pattern. Given that the HVS is more sensible to low frequencies than it is to high frequencies, one expects that the Low Pass Filtered Noise will produce more distortion. This effect is visually demonstrated in Figure 5, where the Low Pass Filtered Noise version on the right produces a considerably worse visual effect than the High Pass Filtered Noise version on the left. However, the MSE between original and distorted images is practically the same for both versions, so the MSE in this case fails to capture the exact extent of the perceptual difference between the original and distorted versions of the images.

MSE=826.91 -D_{Wats}=35.99
DwatCorr=16.61- VMSE=0.16

MSE=827.04 -D_{Wats}= 84.12
DwatCorr=34.2-VMSE=0.66

Figure 5. MSE and Watson Distance Comparison.

b. The Watson Distance Perceptual Model

The Watson perceptual model estimates the perceptual impact of changes applied to the individual terms of an image's 8x8 block DCT representation. Following the general structure of perceptual models described earlier, the estimation of the perceptual distance is based on a sensitivity function, two masking components (one for luminance and one for contrast) and a pooling component.

The frequency sensitivity function is defined through an 8x8 frequency sensitivity table t, whose entries represent the amount of change in the corresponding DCT coefficient that produces one JND. The derivation of the frequency sensitivity table takes into account a number of parameters that describe the viewing conditions, the image resolution and the HVS response for an average individual [28]. For the purposes of the thesis, a set of the parameter values has been chosen for which the resulting frequency sensitivity table t, shown below, proved experimentally to produce good results [13].

$$
t[i,j] = \begin{vmatrix}
1.40 & 1.01 & 1.16 & 1.66 & 2.40 & 3.43 & 4.79 & 6.56 \\
1.01 & 1.45 & 1.32 & 1.52 & 2.00 & 2.71 & 3.67 & 4.93 \\
1.16 & 1.32 & 2.24 & 2.59 & 2.98 & 3.64 & 4.60 & 5.88 \\
1.66 & 1.52 & 2.59 & 3.77 & 4.55 & 5.30 & 6.28 & 7.60 \\
2.40 & 2.00 & 2.98 & 4.55 & 6.15 & 7.46 & 8.71 & 10.17 \\
3.43 & 2.71 & 3.64 & 5.30 & 7.46 & 9.62 & 11.58 & 13.51 \\
4.79 & 3.67 & 4.60 & 6.28 & 8.71 & 11.58 & 14.50 & 17.29 \\
6.56 & 4.93 & 5.88 & 7.60 & 10.17 & 13.51 & 17.29 & 21.15
\end{vmatrix}
$$

From a perceptual point of view, the higher an entry of t [i,j] is, the less sensitive the HVS is at the frequency represented by the entry. These thresholds account for the HVS non-uniform response to spatial frequency without any masking stimuli present.

To adjust the frequency sensitivity table for Luminance Masking the model defines the luminance masking threshold $t_L[i,j,k]$ for every element of the 8x8 DCT block DCT representation of the original image to be

$$t_L[i,j,k] = t[i,j] \times \left(\frac{C_o[1,1,k]}{\overline{C_{o,o}}} \right)^{a_T}, \langle 2.7 \rangle$$

where i=1,2,..8 – j=1,2,…8 are indices for the elements of an 8x8 DCT block in the original image, k=1,2,…,K are indices for the blocks of the original image, $C_o[1,1,k]$ is the DC term of the k[th] block, $\overline{C_{o,o}}$ is the average DC of the DC terms of all blocks and a_T is a constant with a suggested value of 0.649. From a perceptual point of view, $\overline{C_{o,o}}$ represents the overall luminance of the image and a_T controls the degree to which this masking occurs (if a_T is set to 0, the masking effect is suppressed).

These thresholds are then adjusted for the effects of contrast masking by correcting them through the formula

$$s[i,j,k] = \max\left\{ t_L[i,j,k], C_o[i,j,k]^{w[i,j]} \times t_L[i,j,k]^{1-w[i,j]} \right\}, \langle 2.8 \rangle$$

where w[i,j] is a factor between 0 and 1 which depends on the frequency coefficient of the blocks. Watson refers to the thresholds adjusted for luminance and contrast masking as slacks and suggests a constant value of w[i,j]=0.7, $\forall$ i,j. The formula was derived experimentally and is a generalization of Weber's Law. When w[i,j] is set to 0, the threshold is constant in log or percentage terms (for $c_{ijk} > t_{ijk}$) as the Weber Law suggests. When w[i,j] is set to 1, the contrast masking is suppressed.

Finally, to compare an original image C_o with a distorted version C_w the differences of the corresponding DCT coefficients are scaled by their respective slacks and the scaled differences are subsequently pooled in a single perceptual difference estimate represented by the L_4-norm of the scaled differences defined in 2.8

$$d[i, j, k] = \frac{C_W[i, j, k] - C_O[i, j, k]}{s[i, j, k]}, \langle 2.9 \rangle$$

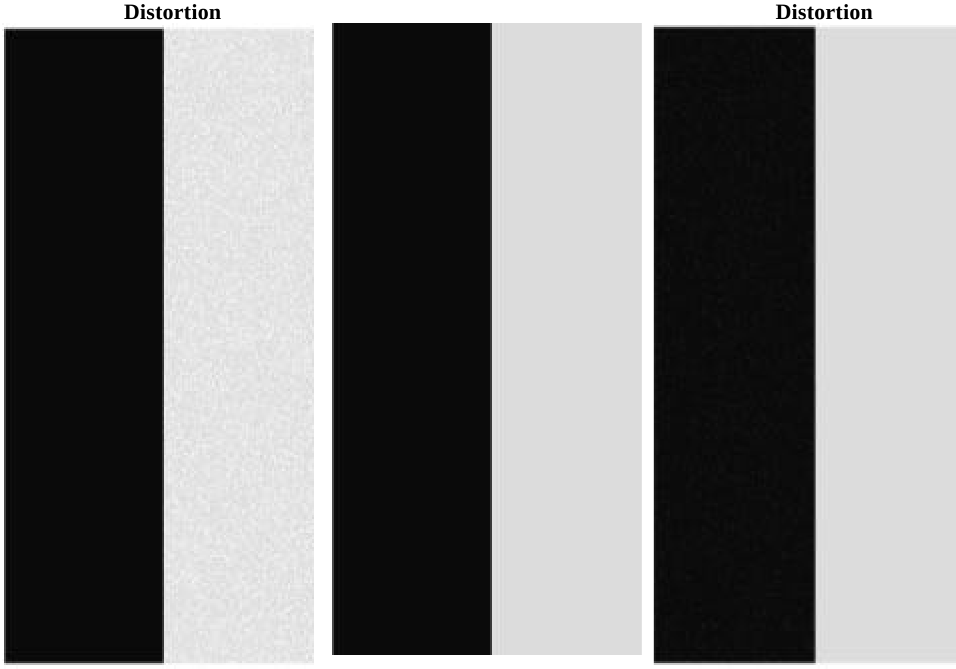
and $$D_{Wats} = \sqrt[4]{\sum_{k=1}^{K} \sum_{i=1}^{8} \sum_{j=1}^{8} \left| d[i, j, k] \right|^4} . \langle 2.10 \rangle$$

The resulting metric is a far better estimate of the perceptual difference between images than the MSE. In Figure 5, the Watson Distance for the Low Pass Filtered Noise distorted version of the original 'Lena' image on the right is 84.12 whereas the Watson Distance for the High Pass Filtered Noise distorted version on the left is 35.99. This implies that the High Pass Filtered Noise distorted version is perceptually closer to the original, which agrees with the visual observation. Therefore, the Watson Distance manages to capture the effects of the HVS response to the distortion of the original image, whereas the MSE as we explained earlier, failed.

c. The Corrected Watson Distance Perceptual Model

The Watson Distance as defined in the previous paragraph underestimates the ability of the low luminance blocks to withstand greater distortion without producing perceptual effects when no stimuli are present. This shortcoming is demonstrated in Figure 6, where a synthetic 512x512 pixel image consisting of two vertical regions was constructed. The region on the left has a low luminance, while the one on the right has a high luminance. Two distorted versions of the original synthetic image were subsequently created, by inflicting the same amount of distortion in each of the two regions. On the left distorted version, white noise was added in the high luminance region of the original image while the low luminance region was left intact. On the right distorted version, white noise with the same characteristics was added in the low luminance region of the original image while the high luminance region was left intact. From a perceptual point of view, the distortion on the high luminance region is more perceptible, since the HVS is more sensitive to high luminance. However, the Watson distance model erratically suggests that the low luminance distortion version has greater perceptual difference than the higher luminance.

High Luminance Distortion | **Original** | **Low Luminance Distortion**



D_{Wats}=54.39 - Dwatcorr=13.6
VMSE= 0.08

D_{Wats}=404.39 - Dwatcorr=9.1
VMSE=0.036

Figure 6. Erratic Estimation of the Perceptual Difference by the Watson Model.

To overcome this shortcoming, a correction on the Watson Distance definition was implemented by applying a suitable multiplicative correction to the Luminance thresholds defined in equation 2.6, before substituting them in equation 2.7. The multiplicative correction factor depends on the block DC coefficient and was determined experimentally.

To determine the correction factor a 512x512 image with constant luminance, determined by the common value of DC coefficient of the image's blocks, was created and the value of a number of pixels of the image was changed in small increments. When the change produced visible distortion, its value was recorded. The procedure was repeated for all possible values of the block DC coefficient and the recorded values were translated to a multiplicative correction factor. The resulting correction factor as a function of the DC block coefficient is depicted in Figure 7.

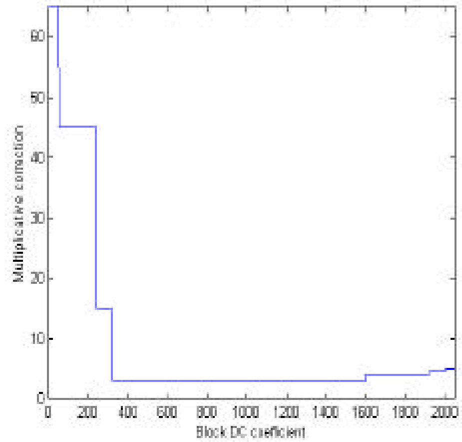


Figure 7. Correction to the Watson Model.

d. *The Weighted MSE (VMSE) Perceptual Model*

The Visual MSE (VMSE) model seeks to correct the inefficiency of the MSE, by taking advantage of the unitary property of the DCT transform. It is based upon the logic that since the MSE can equivalently be calculated in the 8x8 DCT block domain, one may account for the HVS response by appropriately weighing the squared differences of equation 2.5 [19]. The analytic form for the VMSE can be written:

$$VMSE = \frac{1}{64\times \mathrm{K}} \sum_{k=1}^{K}\sum_{i=1}^{8}\sum_{j=1}^{8} W[i,j,k]\times (C_o[i,j.,k]-C_W[i,j,k])^2, \langle 2.11\rangle$$

where i=1,2,..8 and j=1,2,…8 are indices for the elements of an 8x8 DCT block in the original image, k=1,2,…,K are indices for the blocks of the original image, C_o and C_w are the 8x8 block DCT representations of the original and watermarked images respectectively, and W[i,j,k] are the weights of the squared differences. A normalized VMSE version can also be used to produce an index of perceptual difference between original and watermarked images. In this context the Normalized VMSE is defined as

$$VMSE = \frac{1}{64\times \mathrm{K}} \frac{\sum_{k=1}^{K}\sum_{i=1}^{8}\sum_{j=1}^{8} W[i,j,k]\times (C_o[i,j.,k]-C_W[i,j,k])^2}{\sum_{k=1}^{K}\sum_{i=1}^{8}\sum_{j=1}^{8} W[i,j,k]\times \left(\max\left[C_o[i,j.,k],C_W[i,j,k]\right]\right)^2} . \langle 2.12\rangle$$

The VMSE has the advantage of being able to describe a variety of perceptual models in a coherent way. The squared difference summation of the VMSE can be thought as a pooling process based on L_2-norm and the weights as the adjusting functions of the HVS. Any model based on the frequency domain processes these characteristics of pooling and frequency adjustment. So, by appropriately choosing the weights of the VMSE, we can make it reflect the effects of any visual model. For example, if the weights are chosen to be:

$$W[i,j,k] = \left(\frac{1}{s[i,j,k]}\right)^2, \langle 2.13 \rangle$$

where s[i,j,k] are the slacks calculated through the Corrected Watson Distance perceptual model, then the VMSE reflects the Corrected Watson Distance perceptual model, in the sense that if a watermark embedding algorithm uses the VMSE metric to minimize the perceptual difference between an original and a watermarked image, it will also be minimizing the Corrected Watson Distance metric. This relationship between the Corrected Watson Distance metric and the VMSE is reflected in Figures 5 and 6, where the VMSE reflected the change observed in the Corrected Watson Distance metric. The fact that the MSE is equivalently calculated in the space domain and the 8x8 DCT block domain will also be true for any unitary transform. Hence, the VMSE can also be calculated using appropriate weights in any unitary transform domain. Consequently, watermark embedding algorithms based on VMSE minimization can be easily modified (just by adjusting the weights) to reflect perceptual models based on any unitary transform. From this point of view, the VMSE can provide a general approach for a perceptual watermarking scheme. That is why, for the purposes of the thesis, we will be using the VMSE as the basis of our visually adjusted embedding algorithm with weights as defined in equation 1.9 in order to ensure its functionality for a variety of perceptual models.

B. FROM PERCEPTUAL MODELS TO PERCEPTUAL WATERMARKS

The techniques for the incorporation of perceptual models in the watermarking schemes involve a frequency domain transform. No scheme based entirely on the space domain has so far been reported in the literature [25]. The most widely used frequency domain is by far the 8x8 block DCT domain, mainly because of the popularity of the JPEG standard and the extent of the existing supporting literature. However, the wavelet domain shows signs of growing attractiveness among researchers, especially after the announcement of the new JPEG 2000 standard, which incorporates it.

One popular embedding approach towards perceptual watermarks is that of the image adaptive DCT (IA-DCT). The embedding procedure for the IA-DCT scheme is described by the equation

$$C_w[i,j,k] = \begin{cases} Co[i,j,k]+J[i,j,k]xW[i,j,k], \text{ if } Co[i,j,k] > s[i,j,k] \\ Co[i,j,k], \text{ otherwise} \end{cases} \qquad \langle 2.14 \rangle$$

where: i=1,2,..8 and j=1,2,…8 are indices for the elements of an 8x8 DCT block in the original image, k=1,2,…,K are indices for the blocks of the original image, C_o and C_w are the 8x8 block DCT representations of the original and watermarked images respectectively, J[i,j,k] are the JND calculated through the Watson model and W[i,j,k] is the sequence of watermark values generated through a normal distribution. Similarly, an embedding procedure can be defined for an image adaptive Wavelet procedure [25]. Variations of this general approach have been considered in [29] and [30]. Our approach is somewhat different and is based presented in Chapter IV.

C. SUMMARY

In this chapter we established that perceptual watermarks come as a result of the incorporation of perceptual models in the watermarking schemes.

We analyzed the principles upon which perceptual models rely and presented in detail four commonly used perceptual models:

- The MSE, which is often used as a rough test of a watermarking system fidelity impact but which fails to reflect the HVS non-uniform response.
- The Watson distance model, which is a lot more accurate and flexible than the MSE but fails to capture the effect of the HVS's luminance sensitivity.
- The Corrected Watson Distance, which aims to correct the shortcoming of the Watson Distance regarding the effect of the HVS's luminance sensitivity.
- The VMSE which provides a general approach in the incorporation of perceptual models in perceptually based watermarking algorithms.

These models are representative of the general structure of perceptual models. Their implementation and interpretation must always be judged in the context of the subjectivity involved in the fidelity judgment.

Finally, we outlined a general approach towards the integration of perceptual models in watermarking algorithms. In the following chapters, we present two watermarking schemes that use the VMSE model to minimize the perceptual effects of the embedded watermark in the original image.

During the course of our research we also explored a fifth model, the Universal Quality Index (QI), proposed in [23]. Our expectation was to attain an independent verification of the watermark's transparency amelioration achieved by our algorithms. Unfortunately, the (QI) failed in producing conclusive results. The reader may find more details in Appendix A.

THIS PAGE INTENTIONALLY LEFT BLANK

III. AN OVERVIEW OF THE ORIGINAL ALGORITHM

Before presenting our ideas on how to integrate the VMSE perceptual model with the basic watermarking algorithm developed by I. Retsas in [1], we find it useful to briefly describe the rationale and concepts related to it. Since our id eas for the incorporation of the VMSE perceptual model involve modification of the theoretical concepts used in the development of the original scheme, the reader should be aware of its structure and philosophy.

A. PRESENTATION OF THEORETICAL CONCEPTS

In order to assure that the watermarking scheme sustains basic attacks of cropping and compression, while at the same time maintaining sufficient transparency, I. Retsas formulated the concepts of the Center of Interest Proximity Factor (CIPF), the Complexity Factor (CF), and the Priority Coefficient (PC), by introducing the notions of Embedding Size (es) and Embedding sets. These concepts or variations of them are also used in the modified algorithm we propose.

1. The Center of Interest Proximity Factor (CIPF)

The purpose of the CIPF is to increase the watermarking scheme's resistance to cropping. The rationale behind it is that, since the resistance of the image to cropping depends heavily on the spatial location of the image blocks that are selected for embedding the watermark coefficients, the scheme should use for embedding spatial portions of the image that are unlikely to be cropped. To that end, the CIPF is used to provide a ranking of the host image's 8x8 blocks according to the likelihood of not being removed as a result of a cropping attack.

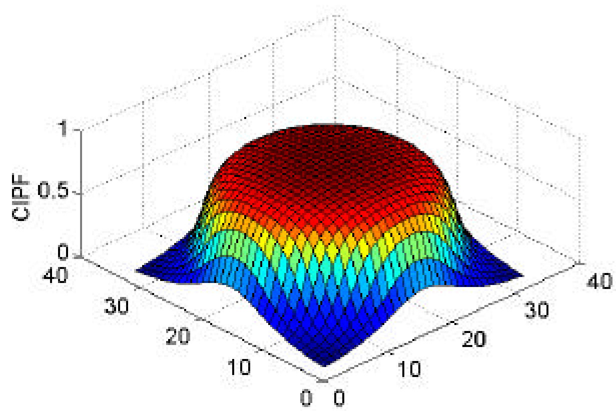


Figure 8. Typical CIPF Distribution for a 256×256 Image with k=15.

For the definition of the CIPF, it is asserted that in cases of commercially used images there is a Region of Interest (RI), where most of the image information is concentrated. For the purposes of the analysis, for each given image the RI was assumed to be a specific point, called Center of Interest (CI), coinciding with the center of the image ($M/2, N/2$ for an $M \times N$ image). Of course, the CI may be chosen to be any other point of the image. It is reasonable to assume that any cropping attack would be directed against some portion near the borders of the image, avoiding portions of the image situated around the CI. To define the CIPF, the following steps are followed:

- For each 8x8 block of the host, with center (m,n), we determine the Euclidean distance *r(m,n)*, from the CI.
- The calculated Euclidean distance is then normalized over the diagonal (i.e. the maximum possible distance within the image) to produce a normalized value *rnorm*, where $rnorm \in [0,1]$.
- This normalized distance is then processed through a transformer with characteristic function *f*, defined by

$$f(rnorm) = -\frac{1}{p} \cdot \tan^{-1}(k \cdot (rnorm - \frac{2}{3})) + \frac{1}{2} \langle 3.1 \rangle,$$

where k is a constant that can typically vary in the range [10,25].

The result is the Center of Interest Proximity Factor (CIPF=*f*(*rnorm*)). A typical distribution of the CIPF is depicted in Figure 8 above.

2. **The Complexity Factor (CF)**

The purpose of the CF is to provide a ranking of the host image's 8x8 blocks according to their capacity to withstand distortions without producing visible effects. To that end, the absolute values of each 8x8 block's DCT coefficients are weighted according to the part of the spectrum that they describe, and then added to produce the CF of the block. The DC coefficient is excluded from the calculations. The weights used in the summation are determined by the coefficients' position in the standard JPEG zigzag plan. Analytically,

$$CF_k = weight \cdot |D'_k| \langle 3.2 \rangle,$$

where *weight*=[1,2,…,63] is a vector (1x63), D_k is a vector (63x1) containing the DCT coefficients of the k^{th} block of the image according to the standard zigzag arrangement (excluding the DC coefficient), $(\cdot)$ is the matrix multiplication operation, and CF_k is the resulting Complexity Factor for that block.

It should be noted that the CF as defined can be thought of as a crude Visual Model with the following characteristics:

- The response of the HVS to the spatial frequency represented by a DCT coefficient of an 8x8 block is assumed to be proportional to the position of that coefficient in the standard JPEG zigzag arrangement.
- The response of the HVS to luminance and the effects of masking are not taken into account.
- Pooling is carried out using the L_1 norm.

The indirect implementation of this crude Visual Model permits a rough estimation of the portions of a host image that can better tolerate the distortion imposed by the embedding.

3. The Priority Coefficient (PC)

The purpose of the PC is to combine the CIPF and CF in order to produce a relative priority ranking of blocks for embedding. The PC of a block is defined as the product of the corresponding CIPF and PC:

$$PC_k = CIPF_k \cdot CF_k \langle 3.3 \rangle,$$

where PC_k, $CIPF_k$, and CF_k are the Priority Coefficient, the Center of Interest Proximity Factor, and the Complexity Factor of the host image's k^{th} block, respectively.

4. The Embedding Size (es) and Embedding Sets

In order to preserve transparency, only a relatively small number of the watermark's 8x8 block DCT coefficients can be embedded in each 8x8 block of the host image. The number of the watermark's 8x8 block DCT coefficients allowed for embedding in each in each 8x8 blocks of the host image is referred to as the embedding size. In practice, the embedding size is allowed to take the values of 2, 4, or 8 watermark coefficients per image block. The set of DCT coefficients that are embedded in a host image's block is referred to as an embedding set.

B. THE ENCODER

The structure of the encoder is depicted in Figure 9. In the operation of the Encoder, two components play an important role and need special explanation, which is provided in sections B.1 and B.2.

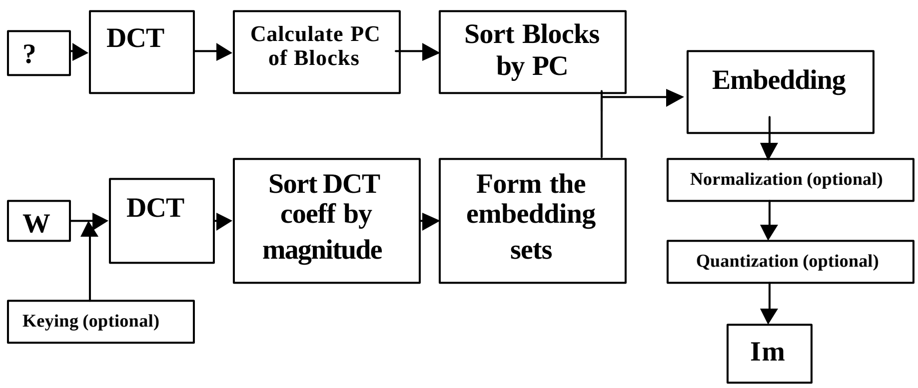


Figure 9. Encoder of the Original Scheme.

1. Formation of the Embedding Sets

The algorithm used to produce the embedding sets consists of the following steps:

- First, the 8x8 DCT block coefficients are sorted in descending order of absolute magnitude and divided into m number of groups, where m is the embedding size. Each group consists of L/m coefficients, where L is the total number of the watermark's coefficients ($L = M_w \times N_w$, where Mw and Nw are the dimensions of the watermark)
- Then the coefficients are regrouped to form the embedding sets. Each set contains m coefficients, one from each groups created in the previous step.

The procedure is depicted in Figure 10 for the case of an embedding size equal to 4.

The rationale behind the adoption of the above procedure for the formation of the embedding lies in ensuring that

- The watermark coefficients with higher magnitude are embedded in the higher-rated image blocks, so that they cause the least possible distortion and are better protected against cropping.
- Not too many of the higher magnitude watermark coefficients are embedded in one image block, so that they do not cause severe distortion.

- Higher magnitude watermark coefficients are embedded in image block coefficients that correspond to lower spatial frequencies, so that they are better protected against quantization.

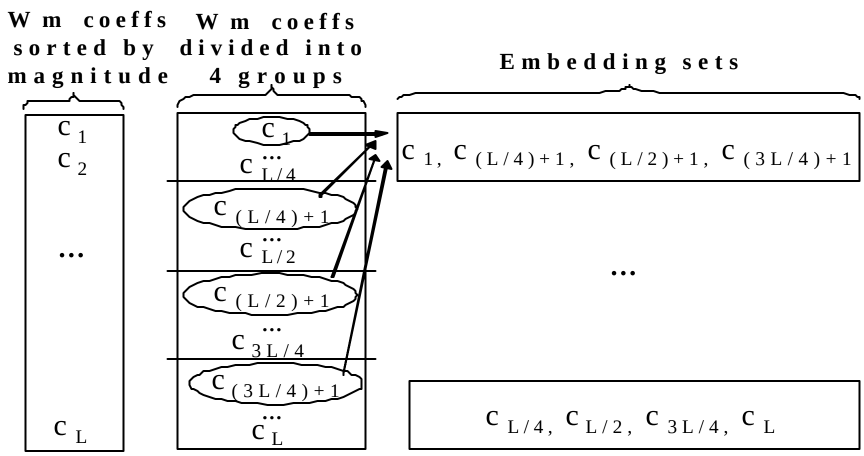


Figure 10. Formation of the Embedding Sets.

2. Embedding

For embedding, only the first N sorted by descending order of their PC host image blocks are used, where N is the total number of embedding sets. For the embedding procedure, the 8x8 DCT block coefficients of the host image used are rearranged according to the JPEG standard zigzag plan and each set is embedded into m coefficients of the corresponding image block following the formula,

$$\left.\begin{aligned} u'_{i(xstart)} &= u_{i(xstart)} + a \cdot C_{i1} \\ u'_{i(xstart+1)} &= u_{i(xstart+1)} + a \cdot C_{i2} \\ &\bullet\bullet\bullet \\ u'_{i(xstart+m-1)} &= u_{i(xstart+m-1)} + a \cdot C_{im} \end{aligned}\right\} \quad ,\langle 3.4\rangle$$

where a is a weighting factor that typically ranges around 0.1, C_{ij} is the j^{th} watermark coefficient of the i^{th} embedding set, u_{ij} is the j^{th} coefficient on the zigzag arrangement of the i^{th} block, u'_{ij} is the modified image coefficient u_{ij} after embedding, and xstart is the first coefficient in the standard JPEG zigzag plan used for embedding (user defined). Finally, the embedded DCT block coefficients of the host image are put back in their 8x8 arrangement. The procedure for the case of an embedding size m=4 is depicted in Figure 11.

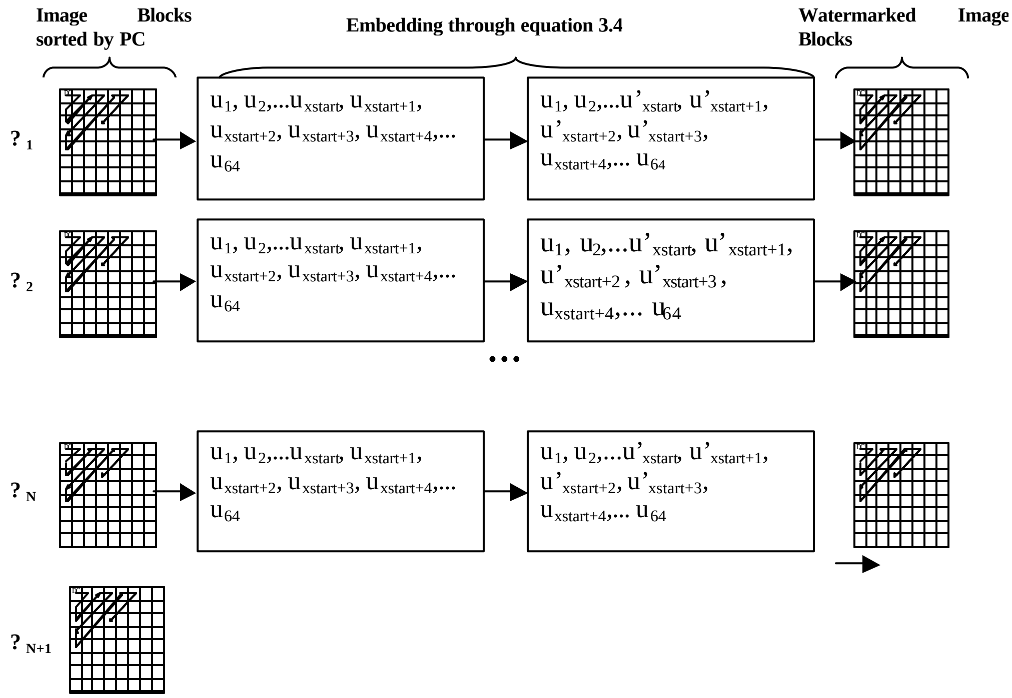


Figure 11. The Embedding Process.

C. DECODER AND DECISION MAKING

The decoder works in reverse order and requires both the original image and the watermark. The DCT coefficients of the test image are subtracted from the DCT coefficients of the original. The sorting information of the watermark coefficients is used to reassemble the potentially recovered watermark and the result is IDCT transformed to produce the recovered object.

The decision-making device compares the recovered object to the original watermark by calculation of the correlation coefficient r , which is defined by

$$r = \frac{\sum_i \sum_j W(i,j) Wr(i,j)}{\sqrt{\sum_i \sum_j [W(i,j)]^2 \cdot \sum_i \sum_j [Wr(i,j)]^2}}, \langle 3.5 \rangle$$

where *W(i,j)* is the *(i,j)* pixel of the original watermark, and *Wr(i,j)* is the *(i,j)* pixel of the recovered object. The decoder decides whether the recovered object corresponds to an actual watermark, based on a predetermined threshold *T*.

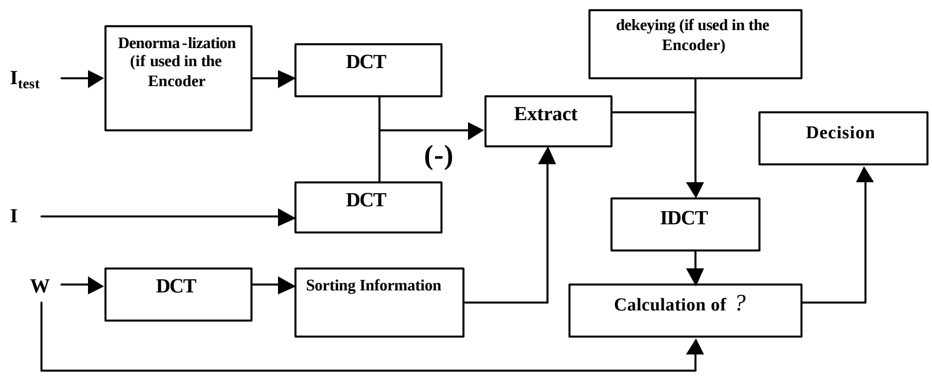


Figure 12. Decoder of the Original Scheme.

D. SUMMARY

The original algorithm, developed by I. Retsas, produces satisfactory results in terms of watermark transparency and resistance to cropping. It implicitly employs a crude visual model to select which image blocks to use for embedding and applies a simple embedding procedure. The decoder needs the original and watermarked images as well as the watermark, which classifies the algorithm as a private watermarking scheme.

We'll be attempting to integrate a improved visual model in the scheme and to explore the possibility of producing a semi-blind variation of the original algorithm. Our endeavour consists of adjusting the definition of the concepts introduced in the original algorithm to inc orporate the VMSE model and revising the structure of the encoder to ensure that the watermark sorting information is not needed in the decoder. The following two chapters present our proposed modifications and the obtained results.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. NEW ALGORITHM BASED ON THE VMSE PERCEPTUAL MODEL

Watermarking algorithms based on perceptual models are ideally suited to addressing the watermarking problem. Perceptual models can be used to determine the portions of an image that can better tolerate the distortion imposed by the embedding and thus make sure that the watermarking will inflict the least possible degradation of the original image.

In this chapter, we will propose a modification of the watermarking algorit hm developed by I. Retsas in [1], to incorporate the VMSE perceptual model in order to achieve higher transparency of the watermark. In addition, we will compare the new algorithm to the original both in terms of transparency and in terms of efficiency in retrieving the watermark. Finally, we will investigate the new algorithm's resistance to cropping.

A. PROPOSED MODIFICATION OF THE ORIGINAL ALGORITHM

The proposed encoder follows the same philosophy implemented in the original algorithm and is depicted in Figure 13. The modification lies in the definition of the Complexity Factor (CF) of the original image DCT blocks. The CF reflects the ability of the blocks to withstand distortions without producing visible effects. Under the assumption that the more complex a block is, the greater is its ability to withstand distortions, the CF aims to provide a ranking of the original image's blocks according to their complexity.

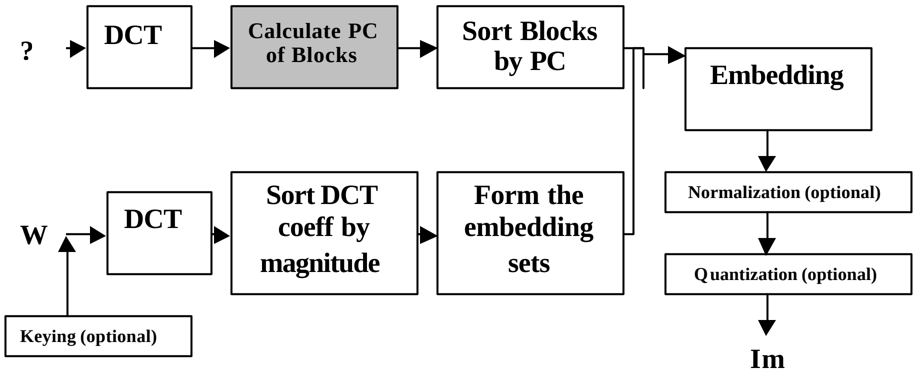


Figure 13. Encoder for the New Algorithm.

One measure of a block's complexity that has been proposed in the literature [31] is the variance of the image blocks in the space domain. This measure, though, does not provide accurate estimation of a block's visual complexity. In Figure 14, both 8x8 blocks have the same number of black and white pixels, hence the same variance (0.2540) in the space domain. However, it's obvious that changing any one pixel on the left block will be immediately detected by the human eye, whereas, the same alteration on the right block (which has a more complicated visual pattern) would require more thorough observation for detection. The shortcoming of the variance of the image blocks in the space domain as a measure of a block's visual complexity has been pointed out in the thesis of I. Retsas and in [32], where it was also demonstrated that the CF, as defined in Chapter III, was capable of correctly indicating that the block on the right is more complex. Using the CF method for the same example of Figure 14, we get a factor of 44.2044 for the left block against a factor 790.8275 for that on the right.

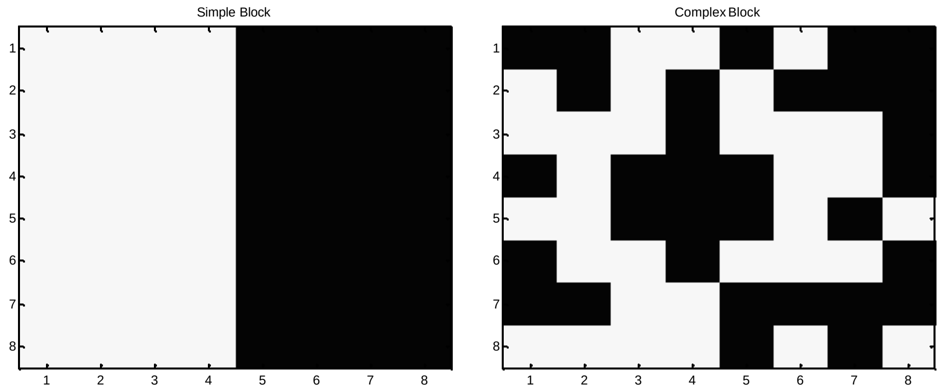


Figure 14. Failure of a Block's Variance in the Space Domain to Reflect its Visual Complexity.

As stated in the previous chapter, the original algorithm indirectly uses a crude visual model based on the JPEG standard zigzag path to define a measure of a block's visual complexity through calculation of the CF for the blocks. Our idea is use the VMSE model to refine the crude CF calculation implemented in the original algorithm in order to ensure higher transparency of the watermark. To calculate the CF for the k^{th} 8x8 block of the host original image we define two new concepts associated with the visual properties of a block. These are the Total Visual Strength Factor (TVSF) and the Embedding Visual Correction Factor (EVCF).

The definition of TVSF for the k^{th} 8x8 DCT block of the host image is based on the same principles as was the CF associated with the original algorithm, and is calculated through the formula

$$TVSF_k = \sum_{\substack{i=1 \\ (i,j)\,?\,(1,1)}}^{8} \sum_{j=1}^{8} \frac{1}{W[i,\, j,\, k]} \times \frac{|C_o[i, j, k]|}{|C_o[1,1,k]|}, \langle 4.1 \rangle$$

where i=1,2,..8–j=1,2,…8 are indices for the elements of an 8x8 DCT block of the original image, k=1,2,…,K are indices for the blocks of the original image, W[i,j,k] are the VMSE weights (equation 2.13) associated with the $(i,j)^{th}$ DCT coefficient of the k^{th} block, and C_o is the 8x8 block DCT representation of the original image. Since the reciprocal of the VMSE weights represents the ability of a DCT coefficient to withstand variations without producing visible effects, the TVSF can be thought as a pooling of the ability of all block DCT coefficients to withstand variations. The DC coefficient is not directly included in the summation of equation 4.1, although it is taken into account for the TVSF calculation, where the rest DCT coefficients of the blocks are weighted against the DC coefficient of the block. Although the DC coefficient of the block is not used for embedding and has been taken into account for the calculation of the VMSE weights, we find it necessary to include it in the TVSF calculation because its effect upon the block's ability to withstand variations is very significant. Figure 15 demonstrates this necessity. For Figure 15, two 8x8 blocks are considered, one with low luminance and one with high luminance. These blocks are depicted in the top part of the figure. If the DC coefficient is not taken into account, the TVSF for the block with high luminance is 0.16, while the TVSF for the block with low luminance is 0.0001, suggesting that the block with high luminance has far greater ability to withstand distortions than the block with low luminance. However, this is not really true. In the bottom part of Figure 15, six coefficients of DCT transform of the blocks were altered by the same amount. This variation resulted in visibly more distortion for the high luminance block contrary to what the TVSF implies. The situation is remedied if the DC coefficient is taken into account using formula 4.1. In that case, the TVSF for the high and low luminance blocks is 0.000081 and 0.00008 respectively, allowing correct comparison of the blocks' ability to withstand distortions.

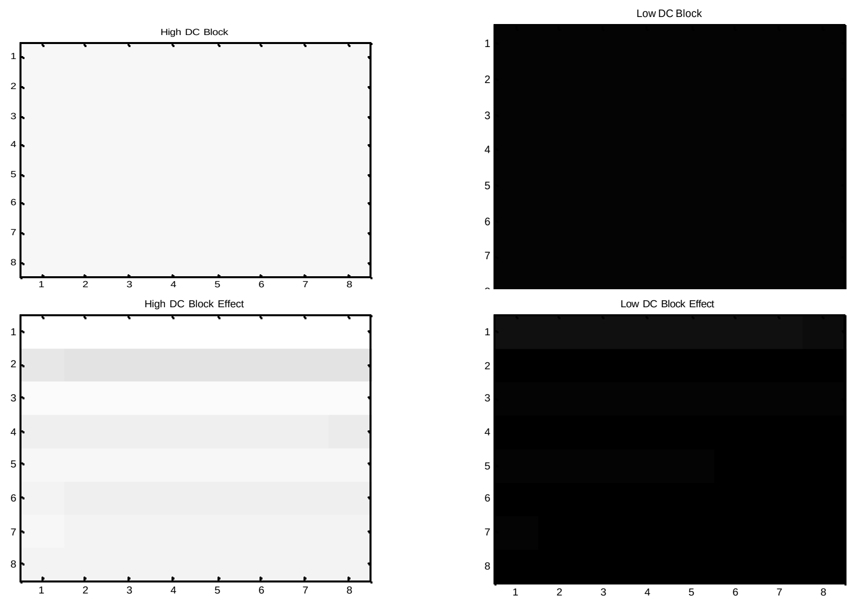


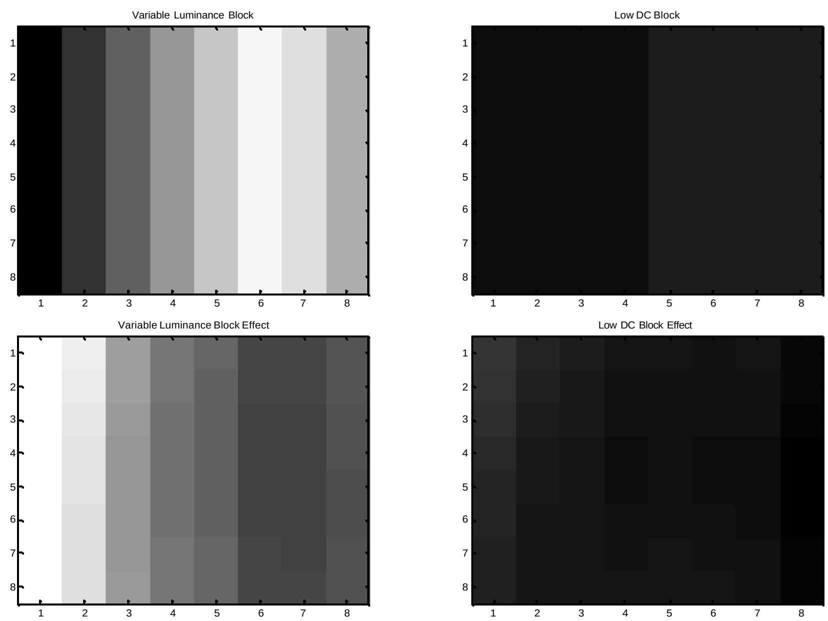
Figure 15. Necessity to Consider the DC Coefficient in the TVSF.

In the context of our watermarking scheme we embed the watermark coefficients according to the JPEG standard zigzag path, using a number of the original image bloc ks' DCT coefficients starting from a specified coefficient. The TVSF provides the same estimate for a block's ability to withstand distortion regardless of the coefficients that are actually used for embedding. Consequently, the TVSF is bound to err if used as the sole criterion for determining the CF of a block. This is illustrated in Figure 16, for which two blocks were considered: one with variable luminance and one with low luminance. These blocks are depicted in the top part of the figure. The block with variable luminance is actually an 8x8 version of the stripes watermark, while the block with low luminance consists of two vertical regions, one being slightly darker than the other. The TVSF for the variable luminance block is 0.0003 while that of the low luminance block is 0.0001, suggesting that the variable luminance block can withstand greater amount of distortion that the low luminance block. However, if the 2nd through 8th standard JPEG path DCT coefficients of both blocks are altered by the same amount, the effect on the variable luminance block is more visible, as it can be seen in the bottom part of Figure 16. To overcome this shortcoming, we introduce the EVCF, which is defined by

$$EVCF_k = \frac{\sum_{i=1}^{8}\sum_{j=1}^{8}\frac{1}{W[i,\,j,k]}\times\left|C_o[i,j,k]\right|\times d(i,j,k)}{\sum_{i=1}^{8}\sum_{j=1}^{8}\frac{1}{W[i,\,j,k]}\times d(i,\,j,k)}, \langle 4.2\rangle$$

where: i=1,2,..8 – j=1,2,…8 are indices for the elements of an 8x8 DCT block of the original image, k=1,2,…,K are indices for the blocks of the original image, W[i,j,k] are the VMSE weights associated with the $(i,j)^{th}$ DCT coefficient of the k^{th} block, C_o is the 8x8 block DCT representation of the original image, and

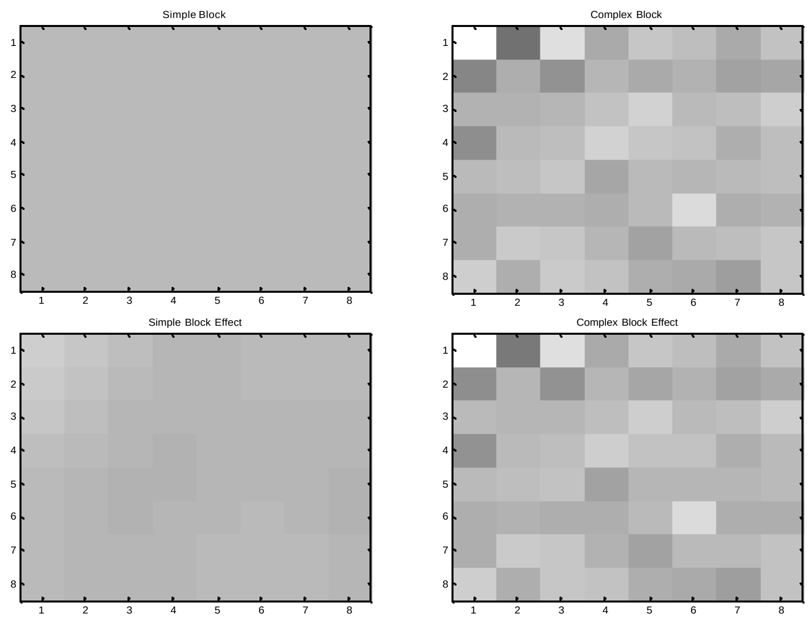
$$d(i,j,k) = \begin{cases} 1, & \text{if the } (i,j)^{th} \text{ coefficient of the } k^{th} \text{ block is used for embedding a watermark coefficient} \\ 0, & \text{otherwise} \end{cases}$$



TVSF=0.0003 – CF=0.2 TVSF=0.0001 – CF=0.12

Figure 16. Necessity to Correct the TVSF.

Thus EVCF is the weighted absolute mean of the block coefficients that are actually used for embedding and can be thought of as a variant TVSF that takes into account only those coefficients used for embedding. It should be pointed out that the EVCF alone cannot be used as an estimate of a block's ability to withstand variations without producing noticeable visible distortion. A trivial example is illustrated in Figure 17, where two blocks, one with complex spatial representation and one with simple spatial representation are depicted. As in the case of the blocks of Figure 16, the 2nd through 8th standard JPEG path DCT coefficients were altered by the same amount. Although the two blocks share the same EVCF, the effect on the simple block is a lot more visible. This indicates that the EVCF alone is not a good measure of a block's complexity.



EVCF=0.29 – CF=0.01 EVCF=0.29 – CF=0.04

Figure 17. Necessity to Consider the TVS.

It is obvious that in order to calculate the CF one needs to suitably combine the TVSF and the EVCF. Our idea is to combine them through the power relation

$$CF_k = TVSF^{w} \times EVCF^{1-w}, \langle 4.3 \rangle$$

where $w \in [0,1]$ is a power factor to be determined experimentally.

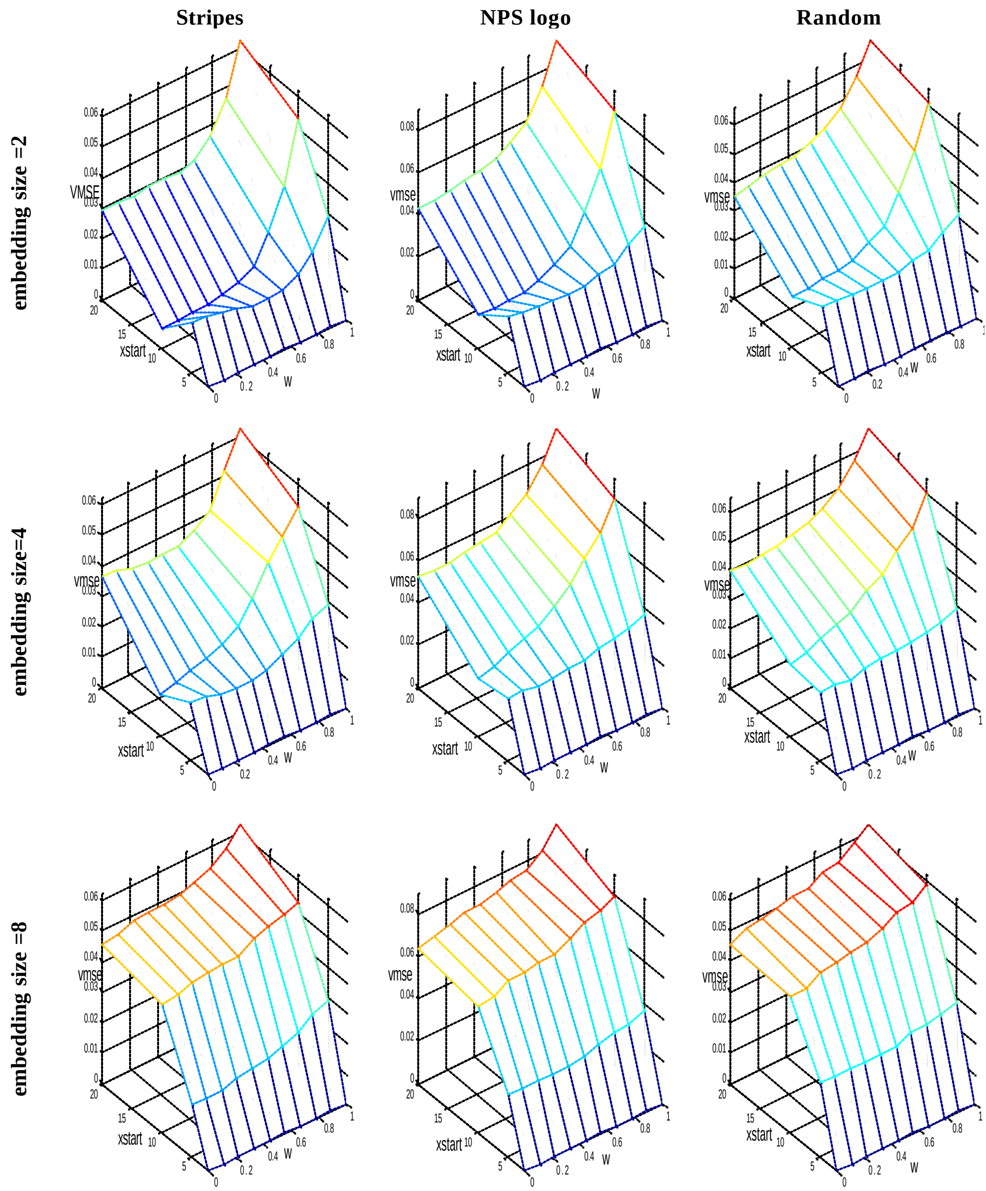


Figure 18. Determination of w.

For the experimental determination of w, we employed the embedding algorithm to mark our regular images using different watermarks, embedding sizes and xstart values, keeping the embedding factor a constant equal to the proposed value of 0.1 of the original algorithm. For each embedding, we plotted the VMSE between original and marked image for different values of w and chose the value of w that minimizes the VMSE. Having no a priori knowledge of the new scheme's behavior with regards to the embedding factor a one may argue that the above choice might not be suitable. However, since the embedding procedure of the new algorithm follows the same philosophy as the original algorithm, and since the choice of embedding factor doesn't influence the CF, choosing the proposed value a =0.1 seems reasonable. In fact, as the proposed modification only affects the choice of blocks used for embedding and not the embedding procedure itself, we expect our decoder to exhibit roughly the same properties as the original, hence the proposed optimal value of a =0.1 of the original algorithm should be close to the one of the modified algorithm. The results can be seen in Figure 18, which depicts the mean VMSE of all regular images as a function of w and xstart for different embedding sizes. The behavior of the VMSE with regards to w varies considerably in nature and depends on the host image, the watermark and the choice of xstart. The embedding size effect is also important, especially when the embedding size is equal to 8. However, the results on Figure 18 indicate that the general tendency is for the mean VMSE to become minimum at $w \approx 0.4$ for values of xstart less than 5 and at $w \approx 0$ for values of xstart greater than 5. The latter implies that when higher frequency bands are used for embedding, only the DCT coefficients of the block that are actually used for embedding are important in the determination of the CF. We decided to use this set of values for w in our encoder. Figures 16 and 17 seem to confirm this choice. In both cases the CF version we propose provides correct comparison between the blocks' ability to withstand distortions. Moreover, the modified CF, just like the original, also captures the relative complexity of the blocks in Figure 14, where the block on the left has a CF equal to 0.008, while block on the left has a CF equal to 1.168.

Having calculated the CF of the original image blocks, the PC of the blocks is determined through multiplication by the CIPF, just as in the original algorithm. As expected, the blocks chosen by the two algorithms are different. Figure 19 depicts the blocks chosen by the algorithms for the ‘Fishing Boat’ image.

es3
New Algorithm

Original Algorithm

Figure 19. Choice of Blocks for the ‘Fishing Boat’ Image.

B. PERFORMANCE OF THE DECODER

The decoder and the decision-making device used for our watermarking scheme are the same as in the original algorithm (Figure 12) and require both the original image and the watermark.

To test the performance of the decoder we calculated the resulting r before and after quantization using different values for a , xstart and embedding sizes, for all possible combinations of our test images and watermarks. Figures 22 and 23 depict the mean r achieved by the regular and synthetic images respectively before and after quantization for various values of a , xstart and embedding sizes. The results of our experiments can be summarized as follows:

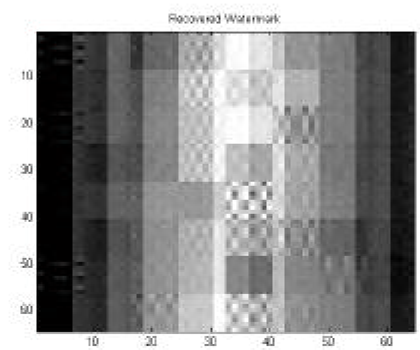


Figure 20. Recovery of Stripes Watermark for ‘New York’ (a =0.1-q=0 -xstart=5-es=8) r =0.98.

1. Performance Before Quantization

If no quantization is performed, would logically expect the recovery of the watermark to be perfect (r =1). However, this is not the case, due to rounding errors introduced during the decoding process. The introduction of rounding errors emanates from two sources. One is the inverse DCT transform procedure and the other is the conversion of the inverse DCT results to integer values for the visual presentation of the recovered watermark. Because of these rounding errors, the correlation coefficient r is in most cases lower than 1, even when the marked image has not been tampered with. The behavior of r before quantization can be described as follows:

- For a fixed value of xstart, up to a certain value of a , we have a dramatic performance improvement. This value is close to 0.1. As a still increases, the amount of improvement is reduced and the performance becomes essentially unchanged. When a reaches a value of approximately 0.3, the performance starts to deteriorate slightly.
- For a fixed value of a , up to a certain value of xstart the performance slightly decreases. When xstart reaches a certain value, which depends on the image the watermark and the embedding size, the performance starts to somewhat improve.
- The effects of the embedding size and the watermark type in the performance of the decoder before the quantization are minimal. Although smaller embedding sizes tend to produce slightly larger r , the predominant factors that dictate the performance of the decoder remain a and xstart.

2. Performance Under Quantization

As expected, the performance of the decoder under quantization varies significantly with the frequency band that is selected for the embedding of the watermark coefficients and the value of the embedding factor a . The behavior of r under quantization can be described as follows:

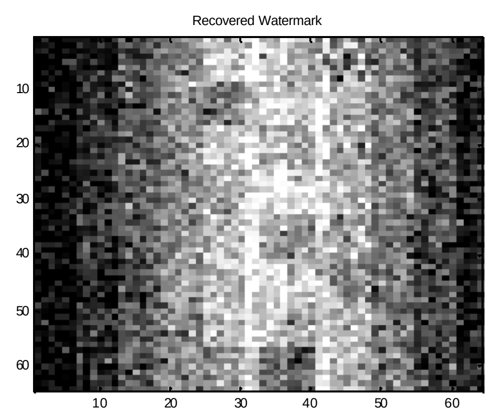


Figure 21. Recovery of Stripes Watermark for 'New York' (a =0.1- q=50% xstart=5-es=8) r =0.83.

- For a fixed value of a , as we embed in higher frequency coefficients (larger *xstart*), there is a general tendency for the performance to deteriorate (smaller r). This is due to the fact that under the JP EG scheme, the higher frequency coefficients are severely quantized and so after quantization the embedded watermark information is essentially lost

Before Quantization **After Quantization**

embedding size=2 NPS Watermark

embedding size=2 Random Watermark

embedding size=8 Stripes Watermark

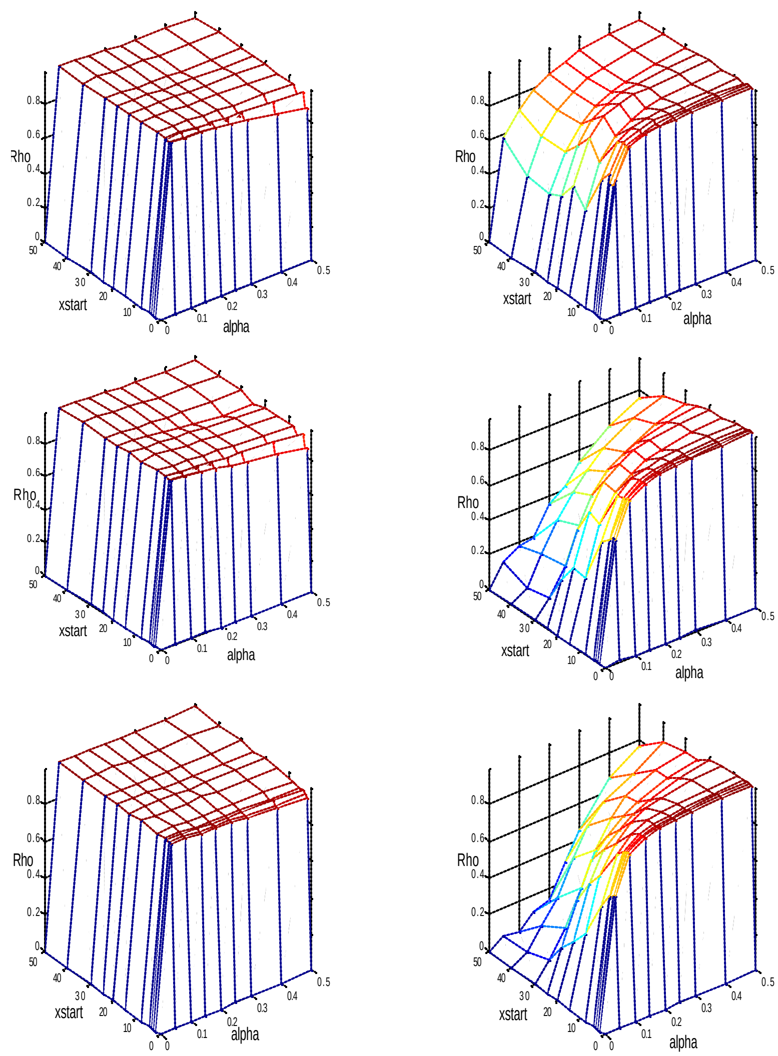


Figure 22. Mean r Achieved by Regular Images for Various Embedding Sizes and Watermarks.

- For a fixed value of xstart, as a is increased, the performance improves. This is logical, since higher values of a imply more prominent embedding of the watermark coefficients, hence better protection against quantization.
- The effects of the embedding size in the performance of the decoder after quantization are more pronounced than in the case where no quantization was performed. For fixed a and xstart, the smaller the embedding size is, the better the performance of the decoder becomes. This is due to the fact that when the embedding is small, the algorithm uses lower frequency bands for embedding, hence the effects of the quantization are less severe.
- The effects of the watermark type in the performance of the decoder after quantization are also more pronounced than in the cas e where no quantization was performed. It was observed that the more random elements are present in the watermark's structure, the worse the performance of the decoder becomes. This can be easily explained if we consider that a simple perceptual pattern like the one of the 'stripes' watermark, has some large coefficients in the lower frequencies and most of its remaining higher DCT coefficients are close to zero, where as a pattern with random elements has a large number of non zeros higher frequency DCT coefficients. Since quantization greatly affects higher order coefficients, watermarks with many non-zero higher order DCT coefficients are bound to perform worse.

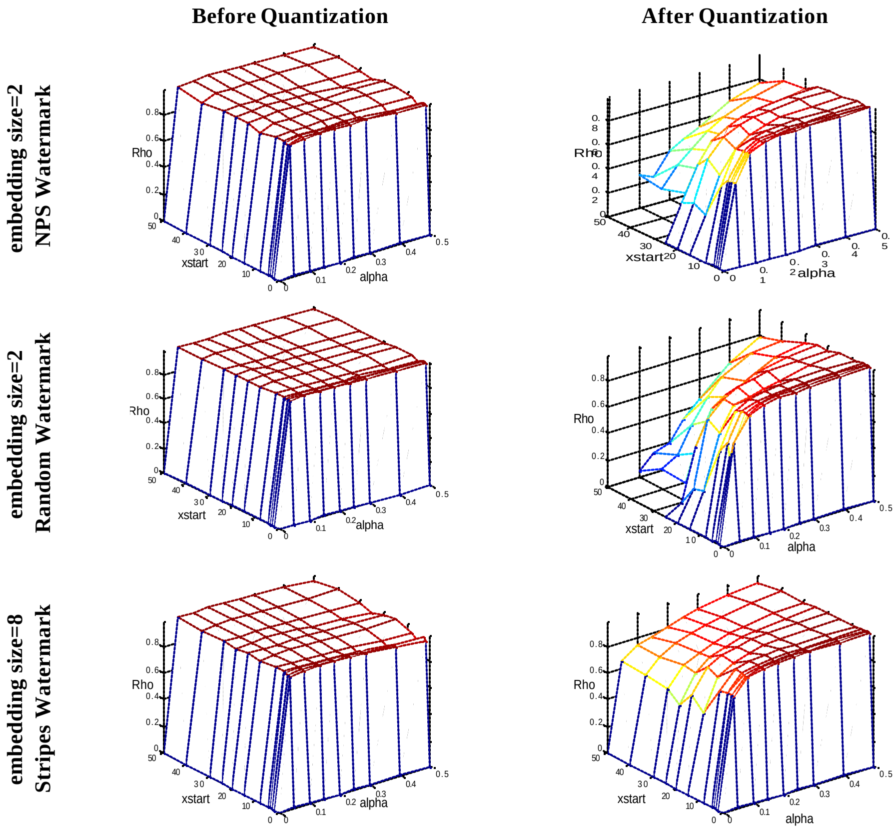
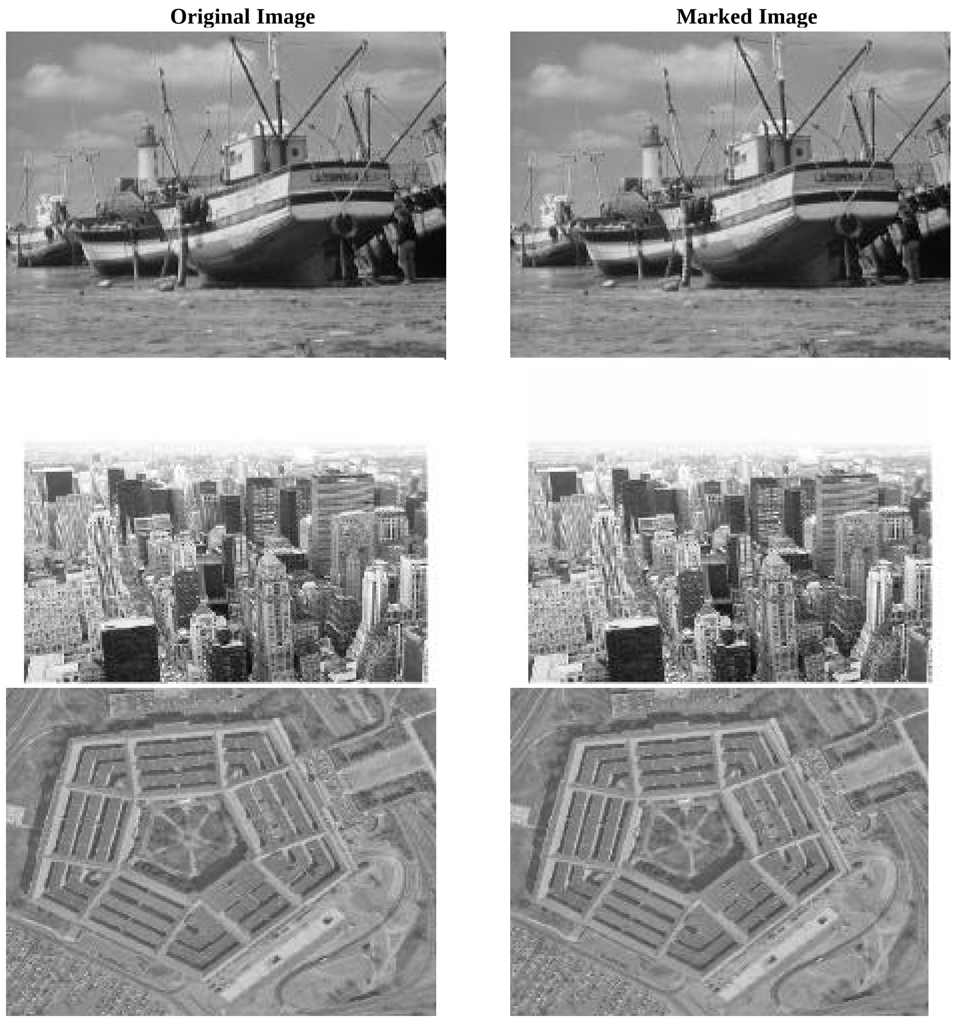


Figure 23. Mean ρ Achieved by Synthetic Images for Various Embedding Sizes and Watermarks.

C. TRANSPARENCY

Our experiments showed that by using the modified algorithm with no normalization and images of type uint8 we may obtain very satisfactory performance in terms of the transparency of the watermark. Figure 24 depicts the NPS logo marked image next to the original using the parameter setting α =0.1, x*start*=4 and *es*=8. The observed visual effects of the parameters in the watermark transparency can be summarized as follows:



Parameters used: *a* =0.1, xs*tart*=4, embedding size =8. All images were marked using NPS logo

Figure 24. Original and Marked Images Fishing Boat, New York and Pentagon.

- For a fixed value of xstart, as a increases the watermark becomes less transparent, i.e. more visible to the HVS. This is expected since higher values of a imply more prominent embedding of the watermark coefficients, hence greater inflicted distortion of the block used for embedding.

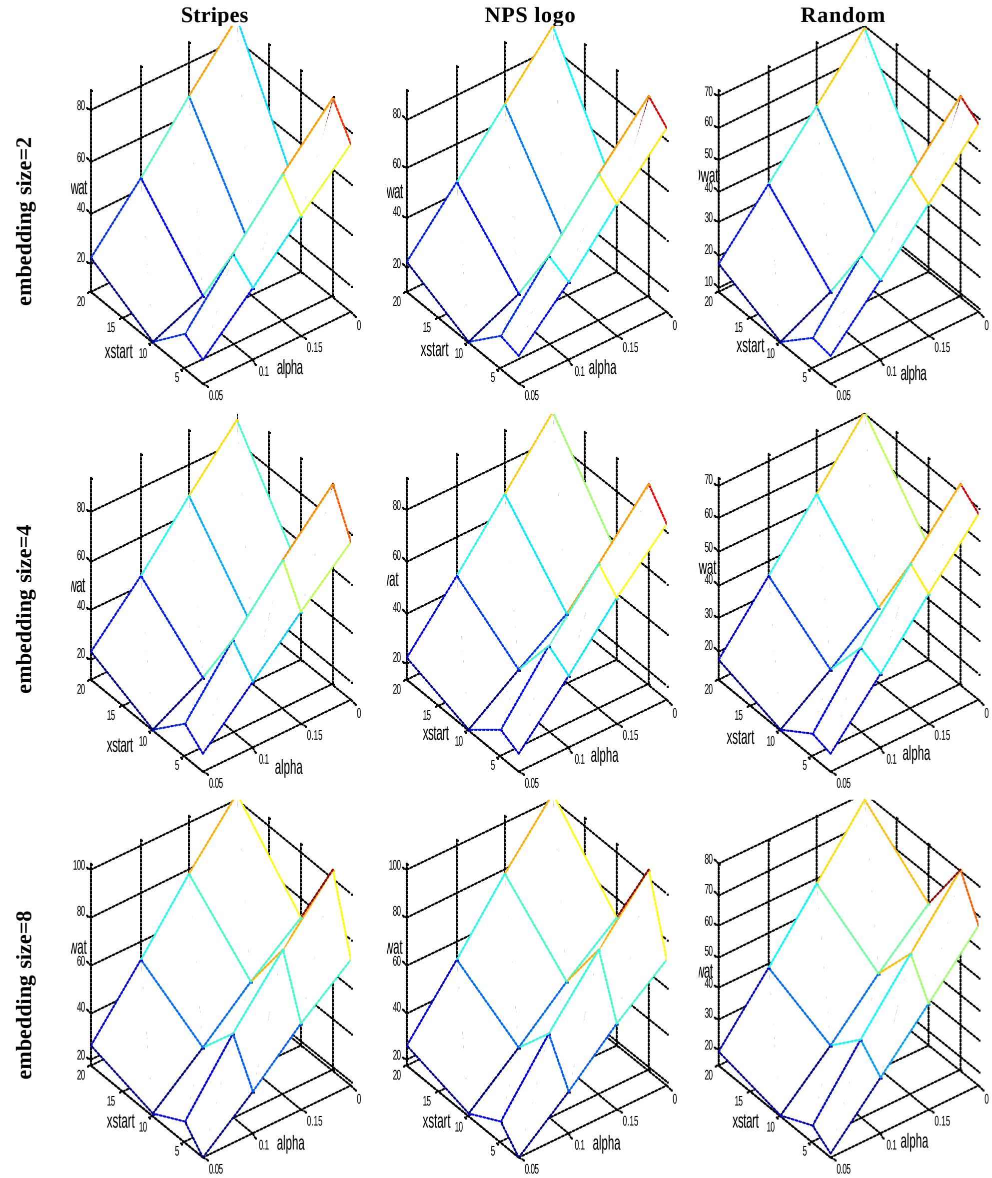


Figure 25. Perceptibility of the Watermark through the Corrected Watson Distance.

- For a fixed value of a , one would expect that using higher values of xstart would make the watermark more transparent, since the HVS is more sensitive to low frequency distortions. Although the above expectation is visually verified when one compares the transparency achieved by using values of xstart that differ significantly (more than 20 positions), it is not always true when one compares the transparency achieved by using relatively close small values of xstart. For example, it was visually observed that the watermark is generally more transparent when xstart=5 than when xstart=20. This phenomenon can be justified by the fact that, as explained in Chapter I, for a fixed luminance, HVS is more sensitive in the intermediate spatial frequencies, less sensitive in the lower spatial frequencies and even less sensitive in the higher spatial frequencies.
- For fixed a and xstart, as the embedding increases, the watermark becomes less transparent. The effect is more pronounced when the embedding size is equal to 8. This is due to the fact that smaller embedding sizes inflict to each block used for embedding considerably smaller distortion than bigger embedding sizes.
- The effects of the choice of the watermark in the achieved transparency are minimal.

Since our proposed modification uses the VMSE model, whose structure is based on the Corrected Watson Distance model, a potential failure of the Corrected Watson Distance model to reflect the above visually observed results would inevitably result in failure of our algorithm. Therefore, it is important at this point to investigate whether the Corrected Watson Distance model manages to capture the above visually observed results. Figure 25 depicts the mean Corrected Watson Distance between original and marked images of all test images as a function of a and xstart for different embedding sizes and types of watermark. It can be seen that the shape of the surfaces reflects the visual observations on the perceived watermark transparency presented earlier by attaining lower values for parameter settings that correspond to low HVS sensitivity and higher values for parameter settings that correspond to high HVS sensitivity. Moreover, the shape and values of the surfaces are independent of the watermark used for embedding. Thus, it is verified that the Corrected Watson Distance model provides accurate estimation of the watermark transparency.

D. ROBUSTNESS TO CROPPING

The performance of our algorithm when the image is subjected to cropping varies with the images marked. The inclusion of the CIPF in the calculation of the PC for the blocks offered a significant improvement to the algorithm's robustness to cropping. The results are shown in tables 1 and 2. For Table 1 the images were severely cropped, reducing their total number of pixels to 64% of their initial quantity. Despite the severe cropping, in most cases, the algorithm managed to achieve a value of r close to 0.3. As expected, the performance improves as the embedding size increases.

parameter setting: a =0.1, *xstart*=4 using NPS Logo					
Image	**Maintained pixels after cropping (initially 512´ 512)**	**Embedding Size**	r **without CIPF**	r **with CIPF**	**Improvement (%)**
Lena	(50:460,50:460)	2	0.3167	0.4174	31,79665
		4	0.2678	0.5586	108,5885
New York		2	0.0827	0.2721	229,0206
		4	0.033	0.3269	890,6061
fishing boat		2	0.2053	0.2856	39,11349
		4	0.1743	0.2919	67,46988
peppers		2	0.0144	0.0734	409,7222
		4	0.012	0.088	633,3333

Table 1. Effect of the CIPF.

parameter setting: a =0.1, *xstart*=4, es=2 using NPS Logo				
Maintained pixels after cropping (initially 512´ 512)	**Achieved** r			
	Lena	**Peppers**	**fishing boat**	**New York**
(11:502, 11:502)	0.7526	0.7787	0.6753	0.4931
(31:482, 31:482)	0.5030	0.112	0.3804	0.2905
(51:462,51:462)	0.3747	0.098	0.2518	0.2112
(71:442, 71:442)	0.3075	0.063	0.1242	0.1278

Table 2. Performance Against Cropping.

E. SELECTION OF THE PARAMETERS

The system's performance is affected by the values of three parameters: the weighting factor a, the frequency band used for embedding defined by xstart and the embedding size. In order to choose values for these parameters, one needs to carefully balance two contradicting demands: the need for a correct assessment whether the recovered object corresponds to an actual watermark by the decision making device and the need for high transparency.

Since the ultimate requirement is for the system to produce an accurate judgment whether the recovered object corresponds to an actual watermark, the principal factor to be considered is the effectiveness of the decision making device. In order to ensure that the decision-maker produces a correct evaluation, the watermark needs to be recovered with the highest r possible, and in any case higher than the decision threshold T used by the decision maker. The threshold should be calculated to match the desired required probability of detection, P_D, and probability of false alarm, P_{FA}. The determination of the threshold as a function of P_D and P_{FA} is outside the scope of the present thesis. Until such a calculation is carried out, we may assume that the user seeks to maximize the achieved r before and after quantization, under the sole restriction that the watermark inflicts the least possible distortion.

Examining the behaviour of r before under quantization (Figure 22) and the perceptibility of the watermark (Figure 25), we can say that the best results are obtained when $a \approx 0.1$, xstart ≈ 10 and embedding size equal to 4.

F. COMPARISON WITH THE ORIGINAL ALGORITHM

The proposed modification aims to produce a new algorithm with better visual qualities. However, in order to make a judgment on whether the proposed modification should be accepted, one needs to consider the overall system performance under the modification. To that end, the performance of the decoder and the resistance to cropping should be considered in addition to the watermark's transparency.

1. Performance of the Decoder

Since practically the only modification of the original algorithm lies in the choice of the blocks used for embedding, one expects that the modification would not inflict major changes in the performance of the decoder. Our experiments verified this expectation. Figure 22 indicates that the behavior of the decoder both before and after quantization is almost identical to the one of the original algorithm, as described by I. Retsas in [1].

2. Transparency of the Watermark

Figures 26 and 27 depict the mean difference between the VMSE and the Corrected Watson Distance of original and marked regular images achieved by the modified and original algorithms. A negative value suggests that the modified algorithm achieved higher transparency, whereas a positive value suggests the opposite. Figures 26 and 27 indicate that the modified algorithm achieves lower VMSE and Corrected Watson distance in all cases except when the embedding size is equal to 8 and xstart=20. Moreover, they indicate that the logic upon which the modification is based is valid.

Indeed, the modified algorithm serves its purpose by reducing the VMSE and reducing the VMSE results in lower Corrected Watson Distance. Despite the indications of Figures 26 and 27, however, the decision on whether the modification increases the transparency of the watermark should be based on subjective judgment by independent observers. Both the VMSE (directly) and the Corrected Watson Distance (indirectly) models have been integrated in the embedding procedure, so a judgment based solely on these models is bound to be biased.

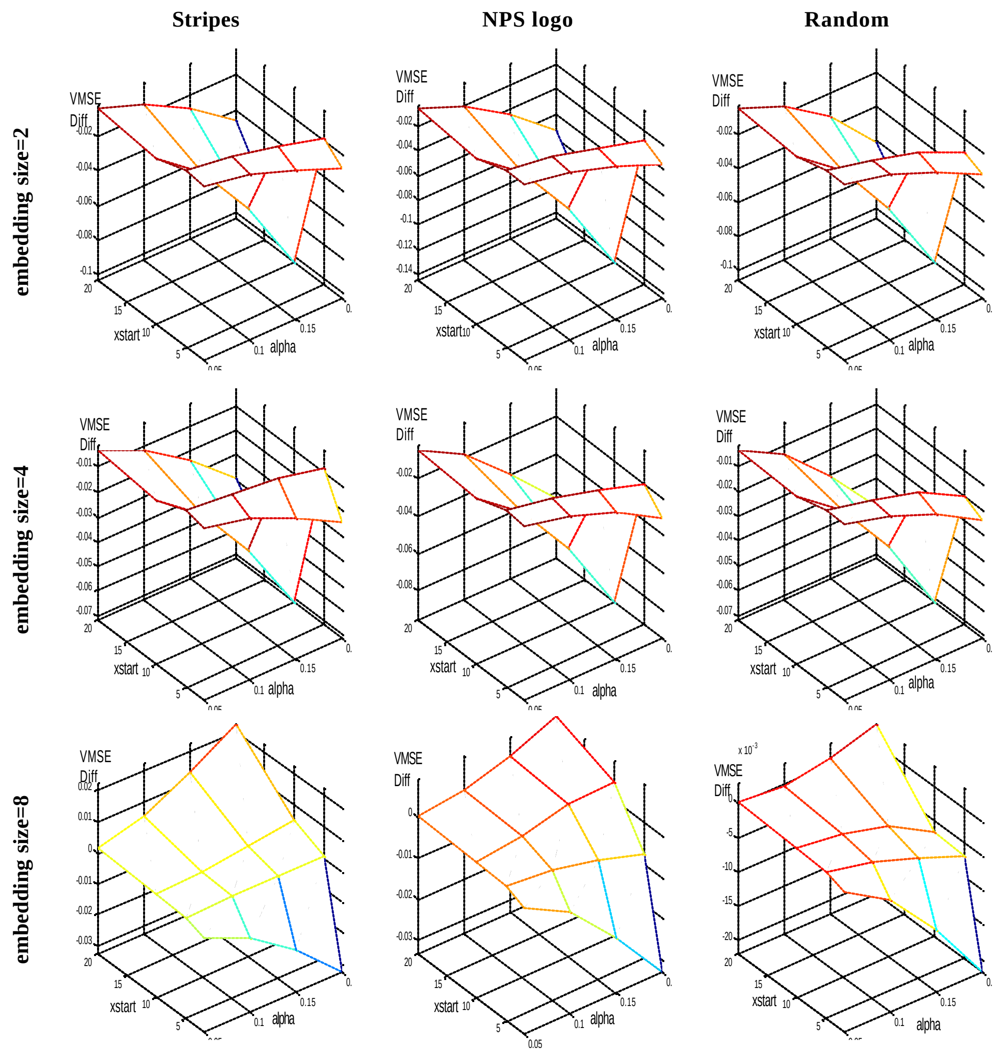


Figure 26. Mean VMSE Difference Between Original and Modified Algorithms.

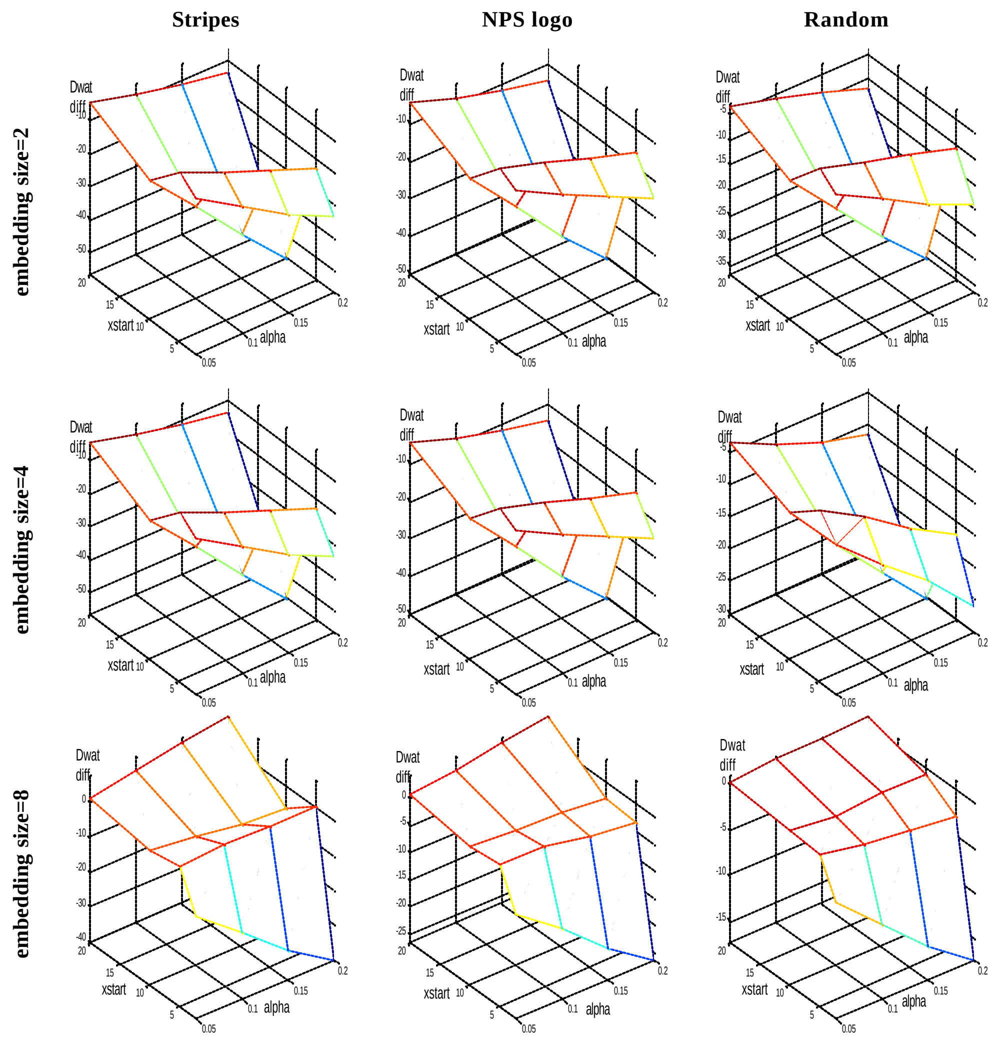


Figure 27. Mean Corrected Watson Distance Difference Between Original and Modified Algorithms.

Original Algorithm **Modified Algorithm**

Parameters used: a **=0.15, xs*tart*=2, *es*=8. All images were marked using Stripes**

Figure 28. Comparison of the Watermark Transparency Between Original and Modified Algorithms.

In our opinion, the modified algorithm achieves higher transparency of the watermark and in some cases, especially for small values of xstart, the improvement is considerable. Figure 28 depicts the stripes marked image by the modified algorithm next to the corresponding marked image by the original using the parameter setting a =0.15, xs*tart*=2 and embedding size equal to 8.

3. Robustness to Cropping

Our experiments demonstrate that the modified algorithm is less robust to cropping than the original. Table 3 shows the difference between achieved r by the two algorithms after cropping. The original algorithm behaves a lot better under cropping and manages to achieve an average of 15% higher r than the modified. The modified algorithm, although it behaves better than the original without the use of the CIPF, still falls short when compared to the original.

parameter setting: a =0.1, *xstart*=4, es=2 using NPS Logo				
Maintained pixels after cropping (initially 512´ 512)	**Difference between achieved r**			
	Lena	**Peppers**	**fishing boat**	**New York**
(11:502,11:502)	0.16	0.14	0.17	0.14
(31:482,31:482)	0.25	0.18	0.18	0.19
(51:462,51:462)	0.18	0.11	0.1	0.13
(71:442,71:442)	0.06	0.04	0.11	0.09

Table 3. Comparison of the Algorithms' Robustness to Cropping.

G. SUMMARY

In this chapter we proposed a modification of the watermarking algorithm developed by I. Retsas, to incorporate the VMSE perceptual model in order to achieve higher transparency of the watermark. The modification involved the redefinition of the CF of the image blocks through TVSF and the EVCF. The purpose of the modification was to ensure that the algorithm chooses the most distortion resistant blocks of the image for embedding, so that the watermark inflicts the least possible degradation of the host image.

The algorithm succeeded in lowering both the VMSE and the Corrected Watson Distance between original and marked images. However, since it incorporates both of them in the embedding process, the judgment of independent observers is necessary to verify that the algorithm achieves higher transparency than the original. Our personal observations indicate that the visual amelioration of the embedding process can be considerable.

The robustness of the modified algorithm to quantization is satisfactory and comparable to that of the original. However, its robustness to cropping is lower than that of the original. The decision whether to accept the modification lies in the ju dgment of the user. To make the judgment the user needs to evaluate the relative importance of robustness to cropping and transparency.

The encoder of the modified algorithm still needs the original and watermarked images as well as the watermark, which classifies it as a private watermarking scheme, like the original. It turns out that a simple modification may be applied to make it possible for the decoder to require only the original and watermarked images. In the next chapter we will present this modification and examine the performance of the resulting semi-blind variation of the modified algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

V. A SEMI-BLIND VARIATION OF THE MODIFIED ALGORITHM

In the previous chapter we introduced a modification incorporating the VMSE perceptual model to the original algorithm developed by I. Retsas, and presented the obtained results. The proposed modification involved the definition of the PC for the host image's blocks and not the embedding process itself. Consequently, the decoder needs the watermark, in addition to the original and marked images, in order to function. In this chapter we present a variation of the modified algorithm in which the decoder only needs the original and marked images.

A. THE PROPOSED VARIATION

A simple observation of the decoder's function, both for the original and the modified algorithms, reveals that the only information regarding the watermark needed is the sorting of its 8x8 block DCT coefficients. The sorting information is used to rearrange the extracted embedding sets in their original order so that the recovered object could be reconstructed. Under the light of this observation, a straightforward variation that does not need the original watermark in the decoding process would be to omit sorting the watermark before the embedding process in the encoder.

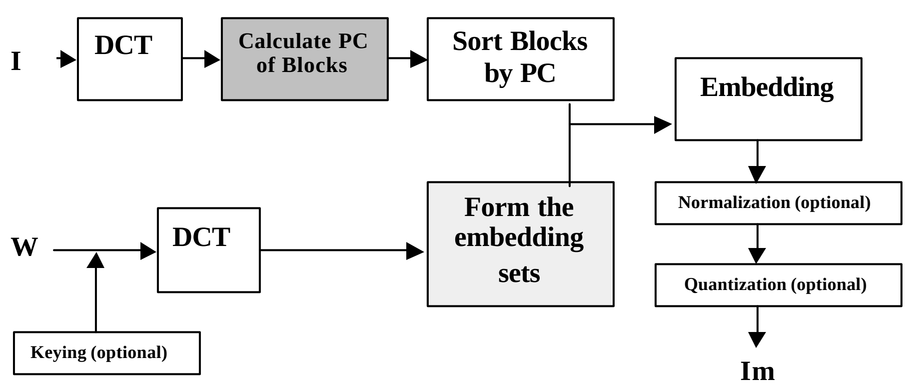


Figure 29. Encoder for the Semi Blind Variation.

The encoder in such a setting becomes simpler and is depicted in Figure 29. An additional simplification would be the modification of the embedding sets' formation. The procedure described in Chapter II aimed to take advantage of the fact that the 8x8 block DCT coefficients of the watermark had been sorted according to descending order of magnitude in order to enhance the watermark's transparency and its recovery under quantization. If the watermark coefficients are not sorted, there is no point in implementing the above procedure. Therefore, in the encoding process of the semi-blind variation, we adopt a procedure for the formation of the embedding sets that consists of a simple rearrangement of the watermark 8x8 block DCT coefficients in groups of size equal to the desired embedding size. The procedure follows two steps:

- The watermark DCT coefficients are reshaped to form a MwxNw vector, where Mw and Nw are the dimensions of the watermark
- The resulting vector is subsequently split to form the embedding sets.

This is depicted in Figure 30 for the case of a 16x16 watermark and an embedding size equal to 4.

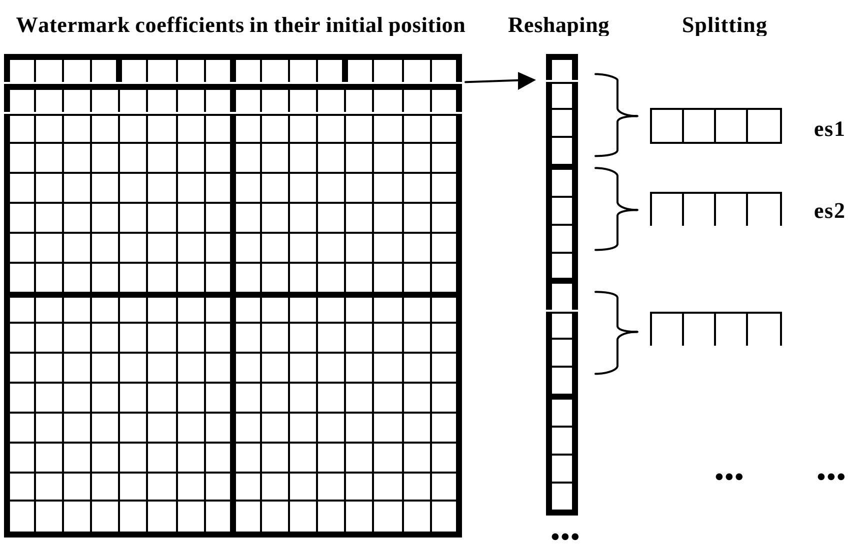


Figure 30. Formation of the Embedding sets in the Semi Blind Case.

When the above embedding procedure is used, the decoder is able to recover the watermark by following in reverse order the steps used to recover the watermark, without requiring the watermark as input, since in such a setting the decoder needs no additional information regarding the watermark's DCT coefficients' relative position. The resulting decoder for the semi-blind variation is depicted in Figure 31. It can be seen that the watermark is only needed at the decision making device.

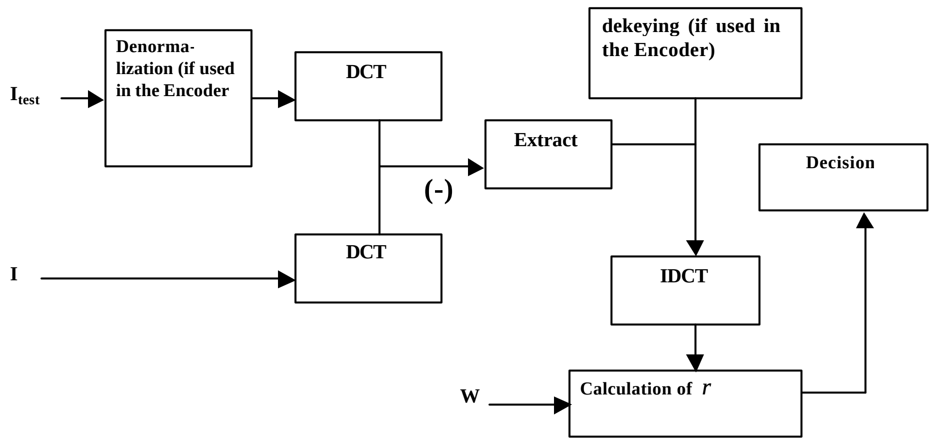


Figure 31. Decoder for the Semi-Blind Variation.

B. PERFORMANCE OF THE DECODER – COMPARISON WITH THE ORIGINAL AND MODIFIED ALGORITHMS

To test the performance of the decoder we calculated the resulting r before and under quantization using different values for a, xstart and embedding size for all possible combinations of our test images and watermarks, just like we did for the case of our modified algorithm. It was observed that the behavior of r, both before and after quantization, follows the general trends described in the previous chapter. However, there exist differences, mainly in the critical values that determine r's behavior with regards to a and xstart, which can be described as follows:

- The maximum value of r before quantization for fixed xstart is achieved at $a \approx 0.08$, compared to $a \approx 0.1$ for the original and modified algorithms.

- The effects of the watermark type and the embedding size in the performance under quantization are less pronounced than in the case of the original and modified algorithms.

If we were to make a direct comparison of the achieved r between the semi-blind variation and the original and modified algorithms we would conclude that although, as expected, the semi-blind variation performs more poorly in general, the difference between the algorithms is not significant. Figure 32 depicts the mean difference of achieved r by regular images for various embedding sizes and watermarks between the modified algorithm and its Semi-blind variation. It can be seen that the maximum difference does not exceed 0.1; moreover, when lower frequency bands and values of a are used, the difference becomes almost negligible. However, when comparing the performance of the decoder for semi-blind variation to the one of either the original or the modified algorithms one needs to take into account the fact that the semi-blind variation requires lower a values to achieve satisfactory transparency, especially for low values of xstart. In light of this, the semi-blind variation performs worse than Figure 32 suggests especially under quantization, because, as we stated in Chapter IV, lower a values result in lower achieved r when xstart is fixed.

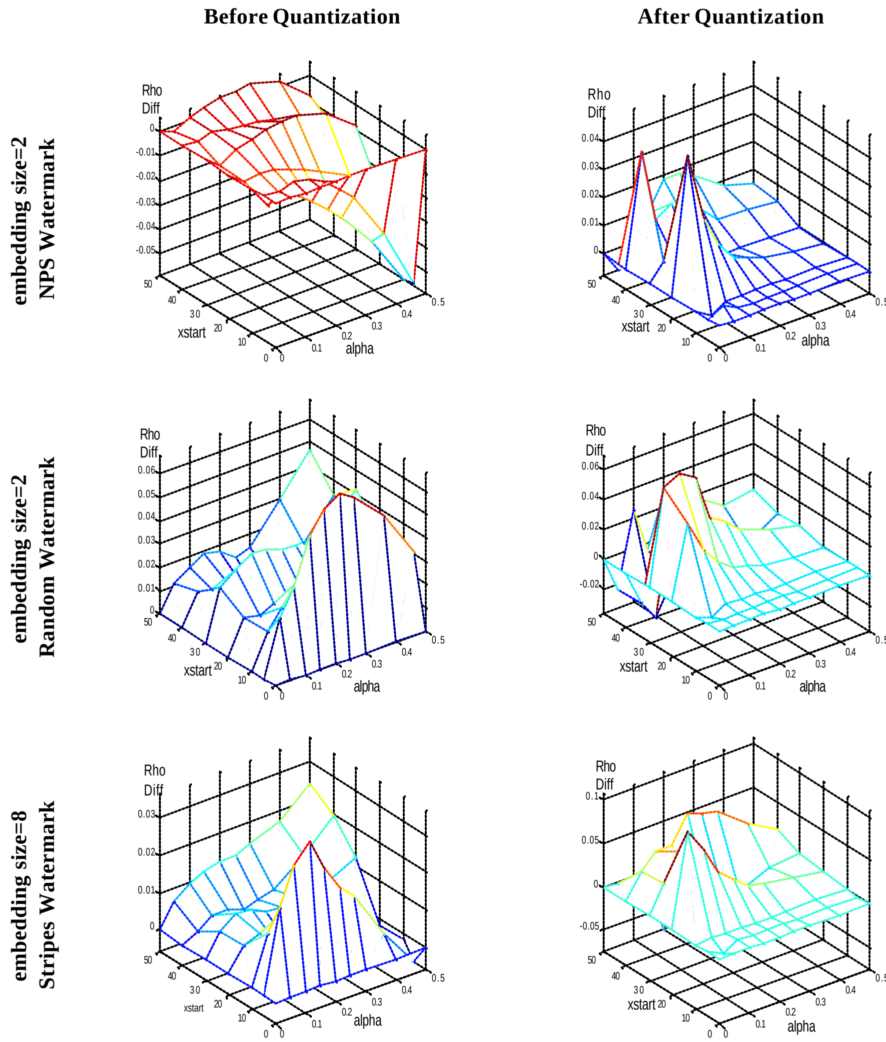
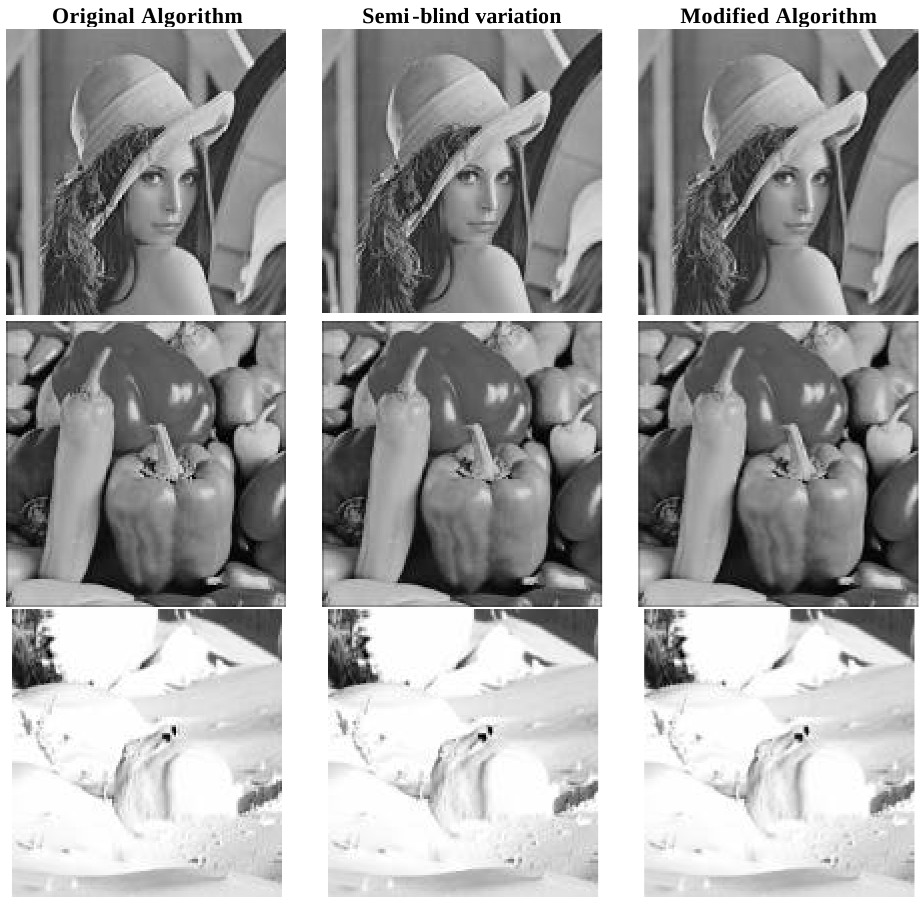


Figure 32. Mean Difference of Achieved PY Regular Images for Various Embedding Sizes and Watermarks Between the Modified Algorithm and Its Semi-Blind Variation.

C. TRANSPARENCY - COMPARISON WITH THE ORIGINAL AND MODIFIED ALGORITHMS

Our experiments showed that by using the semi blind variation with no normalization and images of unsigned 8bit integer type we obtain satisfactory performance in terms of the transparency of the watermark for values of a <0.08 and for values of xstart>8. Figure 33 depicts the NPS logo marked image next to the corresponding marked image obtained by the original and modified algorithms using the parameter setting α=0.1, xs*tart*=4 and embedding size equal to 8. As expected, the transparency achieved is worse than that of both the original and modified algorithms. For values of xstart>10 though, the difference in transparency is not that pronounced.



Parameters used: α =0.1, xs*tart*=4, *es*=8. All images are of type were marked using NPS Logo

Figure 33. Comparison of the Watermark Transparency for the Semi-Blind Case.

The proceeding visual observation is verified also by the Corrected Watson Distance model. In Figure 34 we plotted the mean difference between the Corrected Watson Distance of original and marked regular images achieved by the modified algorithm and the semi-blind variation. A negative value suggests that the modified algorithm achieved higher transparency, whereas a positive value suggests the opposite.

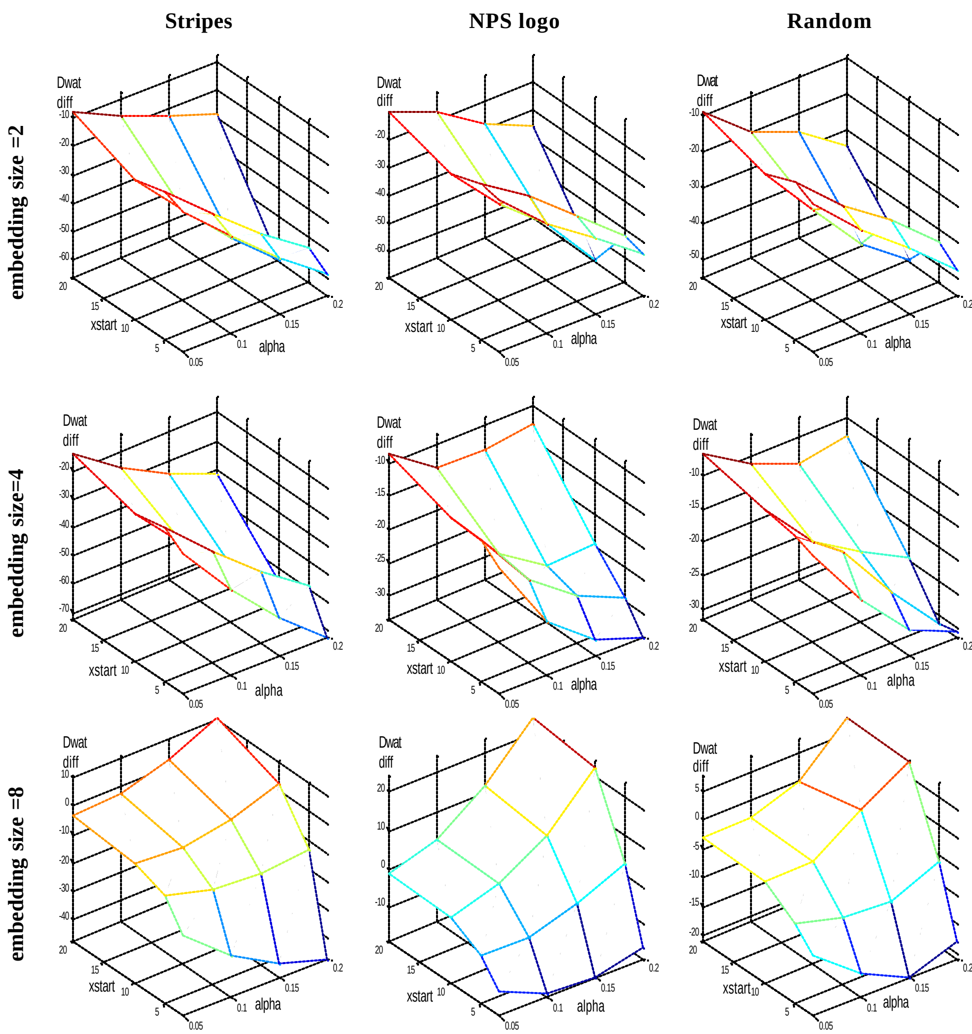


Figure 34. Mean Corrected Watson Distance Difference Between the Modified Algorithm and the Semi-Blind Variation.

D. ROBUSTNESS TO CROPPING - COMPARISON WITH THE ORIGINAL AND MODIFIED ALGORITHMS

The performance of the semi-blind variation when subjected to cropping follows the same trends observed in the case of the modified algorithm. How ever, our experiments demonstrated that the semi-blind variation is less robust to cropping than both the original and the modified algorithms. Tables 4 and 5 show the obtained results. It can be seen that the semi-blind variation achieves 16% lower r than the modified algorithm, which implies a 30% lower r that the original [1].

parameter setting: a =0.1, *xstart*=4, embedding size=2 using NPS Logo				
Maintained pixels after cropping (initially 512´ 512)	**Achieved r**			
	Lena	**Peppers**	**fishing boat**	**New York**
(11:502, 11:502)	0.5086	0.3549	0.5232	0.2726
(31:482, 31:482)	0.3361	0.2083	0.2505	0.1703
(51:462,51:462)	0.2884	0.1581	0.1934	0.1483
(71:442, 71:442)	0.2423	0.1083	0.1058	0.1031

Table 4. Performance Against Cropping of the Semi-Blind Variation.

parameter setting: a =0.1, *xstart*=4, embedding size=2 using NPS Logo				
Maintained pixels after cropping (initially 512´ 512)	**Difference between achieved r**			
	Lena	**Peppers**	**fishing boat**	**New York**
(11:502,11:502)	0.244	0,4238	0.1521	0.2205
(31:482,31:482)	0.1669	-0.0963	0.1299	0.1202
(51:462,51:462)	0.0863	-0.061	0.0584	0.0629
(71:442,71:442)	0.0652	-0.0453	0.0184	0.0247

Table 5. Cropping Robustness Performance Comparison Between the Semi-Blind Variation and the Modified Algorithm.

E. SELECTION OF THE PARAMETERS

Following the same rationale as in the case of the modified algorithm and taking into account the rather poor performance of the semi-blind variation against cropping, we may say that the best results are obtained when $a \approx 0.08$, xstart ≈ 10 and embedding size equal to 8.

F. SUMMARY

In this chapter, we implemented a straightforward variation in the modified algorithm that made it possible for the decoder to extract the wate rmark from the marked image without making use of the original watermark itself. The resulting semi-blind variation produces satisfactory results in terms of transparency. However, it achieves lower r under quantization and performs worse against cropping, compared to both the original and the modified algorithms. Up to a certain extent this is expected because of the smaller amount of information that is available throughout the watermarking process in the case of semi-blind algorithms. To decide whether the degradation of the scheme's performance under the semi-blind variation should be accepted, one needs to weigh two key parameters involving the overall system performance:

- The practical advantages of semi-blind watermarking schemes.
- The necessary minimum r that the decision making device should use as the decision threshold.

During the course of our research we also explored the possibility of producing a semi-blind variation with better performance that would only use the watermarked image and the watermark in the decoding process. However, we discovered that in such a setting, although the achieved watermark transparency is of the same quality as in the modified algorithm, the decoder performance is extremely poor. The reader may find more details in Appendix B.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

A. SUMMARY

In this thesis we explored the possibilities for refinement of the original DCT-based non-uniform embedding watermarking scheme developed in [1] by investigating the advantages of integrating a perceptual model and employing a semi-blind variation.

In Chapter II we introduced the notion of perceptual watermarking, presented the principles upon which perceptual models are based, analyzed some frequently used perceptual models, and linked them to perceptual watermarks. After going through a brief overview of the original watermarking scheme in Chapter III, we presented in Chapter IV a modification that incorporates the VMSE model and compared it to the original. In Chapter V we introduced a straightforward semi-blind variation of the resulting modified algorithm and presented the obtained results. In this chapter, we discuss other parts of our research.

B. SIGNIFICANT REMARKS

The application of perceptual models in digital watermarking provides a significant insight for embedding the coefficients in a way that will take full advantage of the Human Visual System. However, the inherent uncertainties involved in human judgment make the task of evaluating the performance of perceptual watermarking schemes quite challenging. In this context, the researcher must be conscious of the limitations and shortcomings of the perceptual model he employs and make efforts to complement his estimations with statistical observations involving independent observers.

The task of balancing the mutually competitive requirements of robustness and transparency is non trivial. A complete watermarking scheme should address the issue by proving flexibility to the user for applying his judgment through the selection of appropriate values for the scheme's parameters. Moreover, the scheme should offer a suggested default set of parameter values. This default set of parameter values should give priority to the accurate operation of the decision-making device by determining a reasonable decision threshold.

The performance of Discrete Cosine Transform - based perceptual watermarking still needs further investigation. The researcher should take into account the developments of perceptual modeling and the emerging trends towards the integration of perceptual models in the watermarking process. Moreover, the development of the JPEG 2000 standard and how this will affect the unquestionable current domination of the JPEG standard should be also considered for future research in the perceptual watermarking area.

In the proposed perceptual scheme we used a metric (CF) for measuring the relative capacity of each image block to receive watermark information without perceptual distortion of the overall image. In addition, another metric was used (CIPF) to provide protection cropping attacks. The combination of the two metrics is used to prioritize the image blocks and determine the watermark coefficients that will be embedded in each one of them. The achieved watermark transparency improvement appears to be satisfactory, but the evaluation is rather subjective, based only on observation. Moreover, the resulting scheme exhibits satisfactory robustness to quantization and cropping, although its robustness to cropping is slightly worse than that of the original scheme.

The semi-blind variation consisted in omitting the sorting of the watermark coefficients during the encoding process. The resulting scheme maintains satisfactory transparency and robustness to quantization. However, its robustness to cropping is considerably degraded.

C. FUTURE WORK

Our watermarking scheme is far from being complete. Considerable interesting future work may be carried out towards refining and complementing it. Some of the areas demanding special attention include the following:

1. Refinement of the Watson Model

The selection of the JND thresholds used for the Frequency Sensitivity Table in the Watson Model was not based on formal statistical analysis. Although a statistical model has been used in [28] to estimate the effects of the parameters involved in the JND threshold derivation, no study has been conducted to determine the average value of the parameters for the typical human individual. A refinement based on a large number of observations involving human subjects is necessary for obtaining more precise estimation of the JND thresholds. This field of research has also application in Image Quality Assessment and Human Factors Engineering.

2. Improvement of the Embedding Process

The embedding process of our scheme does not take advantage of the watermark spatial frequency distribution characteristics to improve transparency and robustness. The procedure for the formation of the embedding sets is identical for any watermark regardless of its characteristics. Researching possible computationally effective methods for allowing the watermark to interact with the host image during the watermarking process for the selection of the blocks used for embedding could considerably improve the transparency and robustness of the scheme.

3. Improvement of the Robustness to Cropping

As we mentioned, the modified algorithm's robustness to cropping is slightly worse than that of the original algorithm, while that of the semi-blind variation is considerably degraded. This implies that in these schemes the CIPF is less effective in protecting against cropping. In our opinion the reason lies in the fact that the dynamic range of the PC of our scheme is quite different from the one of the original algorithm. Possible remedies could be:

- Transforming the dynamic range of each 8x8 block's VMSE weight so that it takes values between 1 and 64. The simplest transformation would be

$$W'[i,j,k] = 1 + 63 \times \frac{W[i,j,k]}{\max\limits_{(i,j)}(W[i,j,k])}, \langle 6.1 \rangle$$

where i=1,2,..8 and j=1,2,…8 are indices for the elements of an 8x8 DCT block in the original image, k=1,2,…,K are indices for the blocks of the original image, W[i,j,k] are original VMSE weights of the k^{th} block and W'[i,j,k] are the transformed weights.

- Modifying the CIPF definition so that blocks likely to be cropped are not used for embedding. The simplest method to achieve that would be to redefine the characteristic function of the CIPF transformer using the equation

$$f(rnorm) = \begin{cases} -1, \text{ if } rnorm > r_{max} \\ -\frac{1}{p} \cdot \tan^{-1}(k \cdot (rnorm - \frac{2}{3})) + \frac{1}{2}, \text{ otherwise} \end{cases} , \langle 6.2 \rangle$$

where rnorm is the normalized Euclidean distance of the block from the Center of Interest of the host image and r_{max} is the maximum normalized Euclidean block distance allowed for embedding. Equation 6.2 makes sure that blocks away from the Center of Interest are not used for embedding by putting them at the bottom of the PC ranking that takes place in the Encoder.

The feasibility and effectiveness of these modifications along with the examination of other possible techniques to improve robustness to cropping should be considered in tandem to the watermarks transparency and constitute a challenging field for future research.

4. Determination of the Decision Threshold

The ultimate requirement is for the system to produce accurate judgment whether the recovered object corresponds to an actual watermark. This makes the effectiveness of the decision-making device a principal factor to be considered in the system's design. To ensure the functionality of the decision making device, the decision threshold should be calculated to match the desired probability of detection, P_D, and probability of false alarm, P_{FA}, requirements. Considerable research is necessary to determination of the threshold as a function of P_D and P_{FA}. In this effort watermark benchmarking tools like the one presented in [33] can be of great help. In addition, benchmarking tools can be used for comparison between algorithms.

5. Further Improvements

Our watermarking scheme currently treats only grayscale images. There is a need to complement it in order to address the issue of color image watermarking. Moreover, further investigation is required to employ techniques that would protect the resulting watermark from other possible attacks besides quantization and cropping. Finally, as the JPEG 2000 standard is expected to gradually substitute the JPEG standard in image processing applications, the feasibility of reformulating the model with the WT (wavelet transform) used in lieu of the DCT must be investigated in detail.

D. EPILOGUE

The watermarking community is still far from presenting a dominating watermarking scheme. Perceptual watermarks are currently the most promising technique for invisible watermarking. The research on perceptual watermarking has been following closely the advances of Image Quality Assessment and Psychometrics to respond to the need for an effective and dependable watermarking scheme. However, little effort has been dedicated for the development of a perceptual model specifically designed for watermarking needs. Such a perceptual model would directly address the particularities of the watermarking problem, promoting rapid advancements on the field. Maybe the watermarking community should invest in a more independent approach that would allow sufficient time for the research in the perceptual watermarking area to mature and produce results.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. THE UNIVERSAL QUALITY INDEX (QI)

A. THE NEED FOR AN ADDITIONAL PERCEPTUAL MODEL

The perceptual models described in Chapter II can be used for the evaluation of a watermarking scheme's achieved transparency or as a part of a watermark embedding algorithm seeking to minimize the perceptual difference between original and watermarked image. Applying the same model for embedding the watermark and for the evaluation of the results achieved would inevitably lead to a biased judgment. One way around that would be to use different metrics for embedding and evaluation, when it comes to incorporating a perceptual model in the algorithm. These metrics would have to be independent of each other, based on different philosophies and tested in the literature for their correctness and efficiency. The downside of this approach is that in certain (hopefully rare) cases these metrics being independent might disagree. However, if the visual impact in the watermarked image is imperceptible, it makes sense to accept the embedding result as satisfactory even in these cases.

Using another VMSE-derived metric as the second independent evaluation metric will not result in independence of judgment. The Universal Quality Index (QI) proposed in [23] uses the space domain representations co [i,j,k], cw [i,j,k] of the original and watermarked images to produce an index that reflects the quality of the watermarked image viewed as a reproduction of the original. This quality measurement approach does not depend on the viewing conditions or the individual observers and cannot be considered as a perceptual model in the strict sense, since it treats the HVS from a general qualitative point of view rather than quantifying its response to specific stimuli. However, it provides a comparative measurement of quality and hence it may be used as an evaluation tool for watermarking algorithms.

B. DEFINITION OF THE QI

The QI is based on the philosophy that the main function of the HVS is to extract structural information from the viewing field and that the HVS is highly adapted for this purpose. Therefore, a measurement of structural distortion should be a good approximation of the perceived image distortion caused by the process of embedding a watermark. For the calculation of the QI, both the original and watermarked versions of the images are divided into 8x8 blocks. For each of the K blocks of the images, a quality index Q_k is calculated through the formula

$$Q_k = \frac{4 \times s_{co,cw}[k] \times \hat{c}_o[k] \times \hat{c}_w[k]}{\left(s_{co}^2[k] + s_{cw}^2[k]\right) \times \left[\left(\hat{c}_o[k]\right)^2 + \left(\hat{c}_w[k]\right)^2\right]}$$

$$= \frac{s_{co,cw}[k]}{\left(s_{co}[k] \times s_{cw}[k]\right)} \times \frac{2 \times \hat{c}_o[k] \times \hat{c}_w[k]}{\left[\left(\hat{c}_o[k]\right)^2 + \left(\hat{c}_w[k]\right)^2\right]} \times \frac{2 \times s_{co}[k] \times s_{cw}[k]}{\left(s_{co}^2[k] + s_{cw}^2[k]\right)}, \langle \mathrm{A}.1 \rangle$$

where $\hat{c}_o[k] = \sum_{i=1}^{8} \sum_{j=1}^{8} c_o[i, j, k]$ is the mean of the k^{th} block values of the original image,

$\hat{c}_w[k] = \sum_{i=1}^{8} \sum_{j=1}^{8} c_w[i, j, k]$ is the mean of the k^{th} block values of the watermarked image,

$s_{co}^2[k] = \frac{1}{63} \sum_{i=1}^{8} \sum_{j=1}^{8} \left(c_o[i, j, k] - \hat{c}_o[k]\right)^2$ is the variance of the k^{th} block values of the original image,

$s_{cw}^2[k] = \frac{1}{63} \sum_{i=1}^{8} \sum_{j=1}^{8} \left(c_w[i, j, k] - \hat{c}_w[k]\right)^2$ is the variance of the k^{th} block values of the original image, and

$s_{co,cw}^2[k] = \frac{1}{64} \sum_{i=1}^{8} \sum_{j=1}^{8} \left(c_o[i, j, k] - \hat{c}_o[k]\right) \times \left(c_W[i, j, k] - \hat{c}_w[k]\right)$ is the covariance between the k^{th} blocks values of the original and watermarked images.

As equation A.1 indic ates, this index models any distortion between respective blocks as a combination of three different factors: loss of correlation, luminance distortion and contrast distortion. Each factor is represented by a term of equation A.1. The first term is the correlation coefficient between the blocks of the original and watermarked images and measures the degree of linear correlation between them. Its dynamic range is [-1,1] and the highest value is achieved if the blocks are linearly correlated. The second term is a measure of how close the mean luminance of the blocks is and has a dynamic range of [0,1]. Finally the third term is a measure of how similar the contrasts of the blocks are and has a dynamic range of [0,1]. Having calculated the quality indices Q_k for all the blocks, the total QI for the images is then the mean of the indices of the blocks, i.e.,

$$QI = \frac{1}{K}\sum_{k=1}^{K} Q_k . \langle \mathrm{A.2} \rangle$$

The resulting QI will also have a dynamic range of [-1,1]. Although the QI as defined does not directly provide a measure of perceptual distance between compared images, one might translate it to a distance metric by transforming it to a percentage scale through the formula

$$QI_{metric} = \frac{|1 - QI|}{2} \times 100 . \langle \mathrm{A.3} \rangle$$

The QI metric has a dynamic range of [0,100] reflecting the visual distance between compared images.

C. RESULTS

The fact that the QI is based on the space domain representation of the original and watermarked images while the VMSE is based on the DCT domain makes the QI suitable for an independent evaluation of a watermarking scheme that incorporates the VMSE as a visual model. To that end, we used the QI to compare the achieved watermark transparencies of the original and modified algorithms presented in Chapters III and IV respectively. The results revealed that the QI and the VMSE perception models generally disagreed as to which of the two algorithms produces better watermark transparency.

The disagreement between the two perceptual models puts forward the question as to which of the two is actually more accurate in its assessment. To answer that we relied on our own subjective judgment since it was not possible for the purposes of the thesis to conduct rigorous observations involving a large number of human subjects. We concluded that the VMSE model is more accurate in accounting for the perceptual differences caused by the watermarking schemes we investigated. The accuracy of our judgment is amplified by two additional facts:

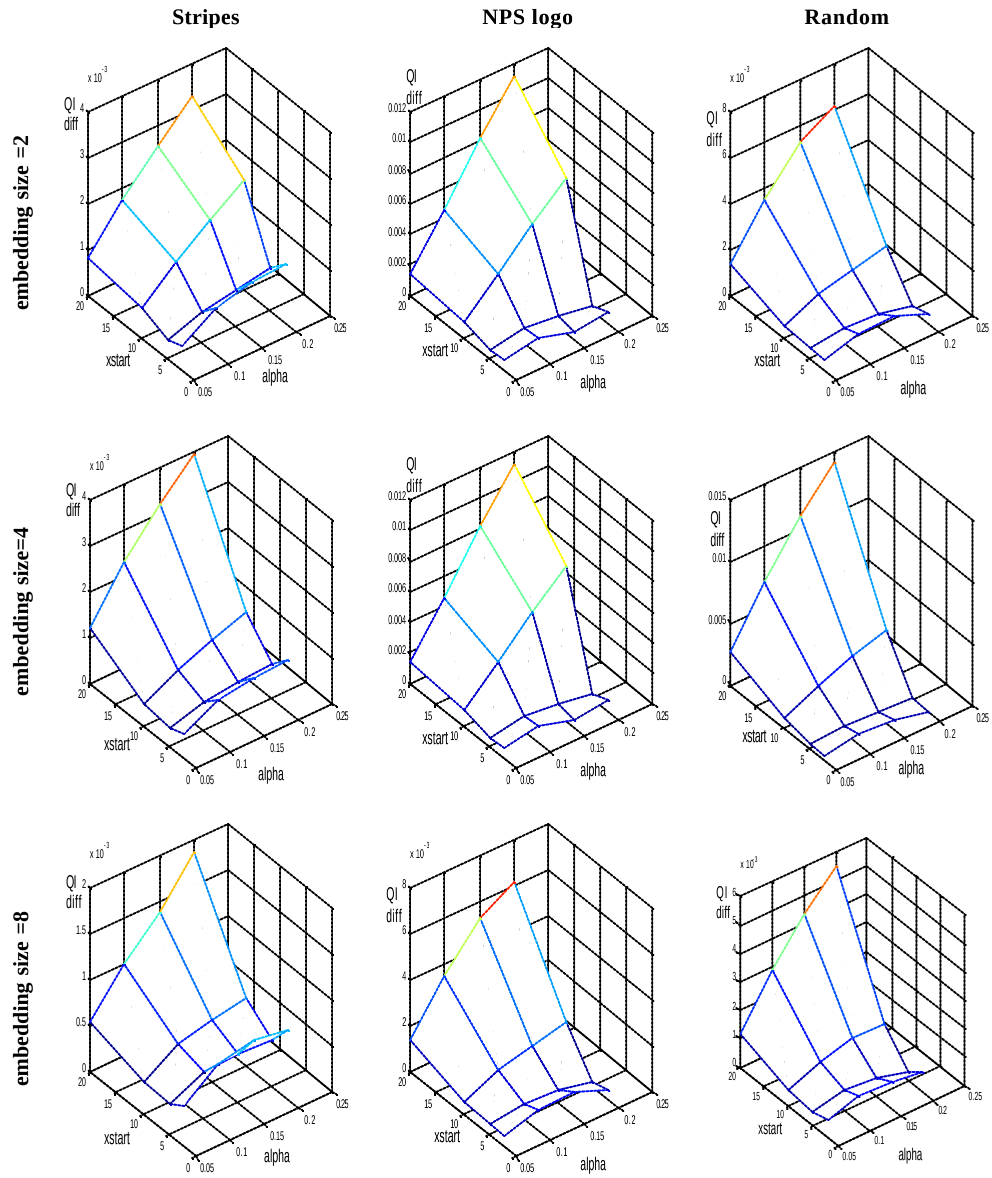


Figure 35. Mean QI Difference Between Original and Modified Algorithms.

- The fact that perceptual models based on the space domain are less precise in accounting for perceptual differences resulting from distortions related to the non-uniform response of the HVS [2].

- The fact that the QI indicates that the achieved transparency of the two algorithms is almost the same. Figure A.1 depicts the absolute mean QI difference of original and marked regular images achieved by the modified and original algorithms, which is of the order of 10^{-3} or of less than 1% in percentage terms. This small difference when compared to the 30% respective difference implied by the VMSE model signifies that the QI cannot produce conclusive results regarding the comparison of the original and modified algorithms.

D. SUMMARY

In the course of our research we explored the possibility of using the QI, a perceptual model based on space domain, to get an independent confirmation of the modified algorithm's competence to ameliorate the watermark transparency. The implementation of the QI model revealed that it generally disagreed with the VMSE model as to which of the two algorithms produces better watermark transparency. The disagreement of the two models required a judgment call as to which of the two is actually more accurate in its assessment. Based on our personal visual observations, the characteristics of our watermarking scheme and the obtained results, we concluded that the VMSE is more accurate in accounting for the effect of the distortions caused by the watermarking schemes we investigated.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. AN UNSUCCESSFUL SEMI-BLIND VARIATION

A. THE RATIONALE BEHIND THE SECOND SEMI-BLIND VARIATION

The shortcomings of the semi-blind variation presented in Chapter IV are due to the abolition of the watermark DCT coefficients' sorting procedure in the encoder. This resulted in lower achieved transparency and robustness to cropping, compared to the original and modified algorithms. Because of that, the formation of the embedding sets procedure implemented in the case of the original and modified algorithms cannot be applied. Consequently, the high magnitude watermark DCT coefficients, which are the most significant in its recovery, are not fully protected against cropping and quantization. Indeed, both the original and modified algorithms used the watermark DCT coefficients sorting procedure to suitably formulate the embedding sets in order to make sure that watermark coefficients of higher magnitude are embedded in blocks of higher PC and lower frequency image DCT coefficients, thus offering greater protection against cropping and quantization. A method to overcome this inherent deficiency of the proposed semi-blind variation would be to revise the decoder so that only the watermark and the marked image are needed.

B. THE DEVELOPMENT OF THE SECOND SEMI-BLIND VARIATION

In the context of the original and modified algorithms, the unmarked host image is subtracted from the marked image at the decoder to recover the watermark. If the original unmarked image is not available then the decoder must be endowed with the ability to acquire information on the exact location of the original image's DCT coefficients used for embedding. With that information in hand, the decoder would be able to use the marked image and the watermark to assemble the original image and then use it to recover the watermark. From our scheme's perspective, providing the decoder with the sorting information of the original image's 8x8 DCT blocks according to their PC would be sufficient. In such a case, the decoder would be able to simply utilize the values of a and xstart used for embedding and construct the original image. However, the only means that the decoder has at its disposal to acquire the necessary sorting information is the marked image. If the sorting of the marked image's 8x8 DCT blocks in descending order of their PC is very similar (ideally identical) to that of the original image, the encoder should have no trouble constructing the original image and recovering the watermark.

This train of thought gives rise to the development of a watermarking scheme that would use the same encoder as the modified algorithm and thus be able to achieve high watermark transparency and a decoder whose function is depicted in Figure 36.

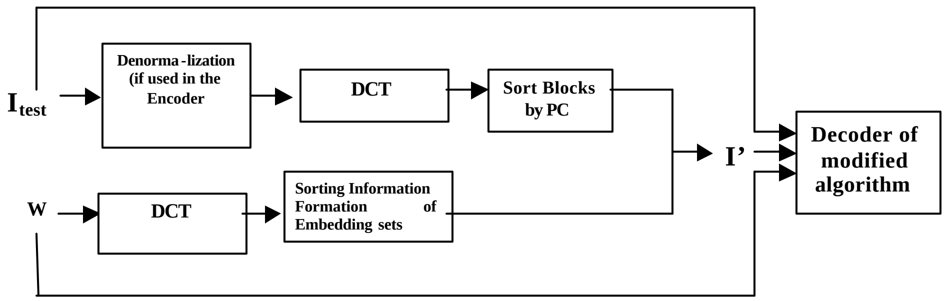
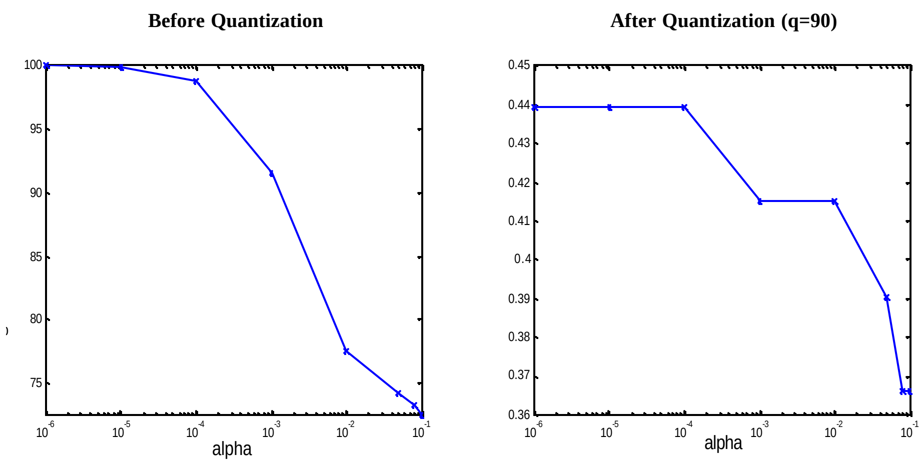


Figure 36. Decoder for the Second Semi-Blind Variation.

C. PERFORMANCE OF THE SCHEME

The performance of the scheme in terms of transparency is identical to that of the modified algorithm, since the both use the same encoder. However, the performance of the scheme's decoder is extremely poor. The decoder was unable to successfully recover the watermark with satisfactory r, even if no quantization takes place. In general the achieved r was very close to 0, even in the most favorable for recovery cases (a =0.1, xstart=2 and embedding size equal to 2).

The reason for this profound failure lies in the fact that mismatches between the sorting information provided by the marked image and the actual one of the original image generate tremendous effects in the decoding process. Moreover, for the practical ranges of the parameter settings used in the scheme, the PC ranking mismatch between original and marked images is substantial. In Figure 37 we see the percentage of blocks that preserved their ranking position in the PC sorting of the marked image for different values of a before and after quantization. In the case where no quantization is applied, for very small values of a the mismatch is not that pronounced. However, for values near 0.1, less than 75% of the marked image blocks retain the same PC ranking position they occupied in the corresponding ranking resulting from the original image. The effect is far more evident when quantization is applied. As a consequence of the ranking mismatch, the decoder is not using the correct blocks to reconstruct the original image, which results in the failure of the decoder.



Parameters used: xs*tart*=2, *embedding size equal to* 4 with NPS logo

Figure 37. Comparison of the PC Sorting of Original and Marked Images.

D. SUMMARY

In the course of our research we explored the possibility of developing a semi-blind variation of the modified algorithm that would eliminate the need to make available the original image to the decoder. The expected benefit from such a scheme was the amelioration of the performance of the semi-blind variation presented in Chapter III, both in terms of watermark transparency and robustness to cropping. The development of the scheme was based on the assertion that the PC ranking of the marked image's 8x8 DCT blocks would be very similar to that of the original image and that potential differences in the ranking would not inflict major consequences in the watermark recovery. Unfortunately, the above assertion was proven wrong and as a result the scheme's performance is inadequate.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. SOFTWARE

In this appendix we include the listings of the Matlab codes we created for the thesis. The appendix is divided in three parts. The first part consists of the codes related to the perceptual models we described in Chapter II, the second part consists of the codes related to the modified algorithm we analyzed in Chapter III and the third part consists of the codes related to the semi-blind variation we presented in Chapter IV.

PART 1: PERCEPTUAL MODELS

```matlab
function m = mse(C1,C2)

%************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: June 20, 2002
% FUNCTION: mse
% INPUT: Two 2D arrays corresponding to images.
% DESCRIPTION: Mean square error calculation. One image is the original,
%             the other is the distorted
% RETURNS: A double real corresponding to the MSE.
%************************************************************************

disp('MSE Processing...')
b=size(C1);
C1=double(C1);
C2=double(C2);
m=0;
for i=1:b(1)
  for j=1:b(2)
    m=m+ ( C1(i,j)-C2(i,j))^2;
  end
end
m=m/(b(1)*b(2));
return
```

```matlab
function Dwat= Watson_Dist(ORIG,DIST)
%**************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: June 20, 2002
% FUNCTION: Watson_Dist
% INPUT: Two images to calculate Watson distance
% DESCRIPTION: returns the Watson Distance of the images of input
%            ORDER CONVENTION: the first input is the original.
% RETURNS: A double real corresponding to the watson distance.
% Watson_dist calls Freq_Sensi_table, getfirst, Lum_Mask.
%**************************************************************************
disp('Watson Distance Processing...')
[M,N] = size(ORIG); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
   fprintf(1,'The dimensions of the selected image are not multiples of
         8\nand errors will occur;\nTHE PROGRAM IS TERMINATED\N');
   return
end
wij=0.7;
% Making the DCT
% It is assumed that the input images are in uint8 form [0 255]
ORIG = double(ORIG);
DIST = double(DIST);
TD = dctmtx(8);
dctORIG = blkproc(ORIG,[8 8],'P1*x*P2',TD,TD');
dctDIST = blkproc(DIST,[8 8],'P1*x*P2',TD,TD');
% Getting the frequency sensitivity table
T=Freq_Sensi_table;
% Getting the mean of the DC components
Coo=blkproc(dctORIG,[8 8],'getfirst');
Coo=mean2(Coo);
% Getting the Luminance Masking
TL=blkproc(dctORIG,[8 8],'Lum_Mask',T,Coo);
% Getting the slacks
s=zeros(M,N);
for i=1:M
  for j=1:N
    l= (abs(dctORIG(i,j))^wij)*(TL(i,j)^(1-wij));
    s(i,j)=max(TL(i,j),l);
  end
end
% Getting the Watson distance
d=zeros(M,N);
e=abs(dctORIG-dctDIST);
for i=1:M
  for j=1:N
    d(i,j)=(e(i,j)/s(i,j));
  end
end
d=reshape(d,M*N,1);
Dwat=norm(d,4);

% Reference:
% I. COX, M. MILLER, J. BLOOM, DIGITAL WATERMATKING Chapter7, pp215 -218
```

```
function DC= getfirst(mat)
%**************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: July 22, 2002
% FUNCTION: getfirst
% INPUT: A vector or a matrix
% DESCRIPTION: If the input is a block of the 8x8  DCT of images
%            getfirst returns the DC coefficient.
% RETURNS: The leftmost (first) element of the matrix or vector.
%**************************************************************************
DC=mat(1);


function [TL] = Lum_Mask(dctBlock,T,Coo)
%**************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: July 22, 2002
% FUNCTION: Lum_Mask
% INPUT: A block of a DCT transformed matrix dctBlock, a Frequency
%        Sensitivity table T and the mean Coo of the DC components of
%        all the DCt blocks
% DESCRIPTION: Uses Formula 7.3 of the reference
% RETURNS:  The Luminance masking table for an image.
%*************************************************************************
DCcomp= dctBlock(1,1);
% Getting the Luminance Masking table
TLl=T*((DCcomp/Coo)^0.649);
TL=TLl;

% Reference:
% I. COX, M. MILLER, J. BLOOM, DIGITAL WATERMARKING Chapter7, pp215-218


function [q]=Freq_Sensi_table
%**************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: June 20, 2002
% FUNCTION: Freq_Sensi_table
% INPUT: -
% DESCRIPTION: -
% RETURNS: The proposed DCT Frequency Sensitivity table
%**************************************************************************
q= [ 1.4   1.01  1.16  1.66  2.4    3.43   4.79   6.56;
    1.01  1.45  1.32  1.52  2      2.71   3.67   4.93;
    1.16  1.32  2.23  2.59  2.98   3.64   4.6    5.88;
    1.66  1.52  2.59  3.77  4.55   5.3    6.28   7.6;
    2.4   2     2.98  4.55  6.15   7.46   8.71  10.17;
    3.43  2.71  3.64  5.3   7.46   9.62  11.58  13.51;
    4.79  3.67  4.6   6.28  8.71  11.58  14.5   17.29;
    6.56  4.93  5.88  7.6  10.17  13.51  17.29  21.15  ];
```

```
% Reference:
% I. COX, M. MILLER, J. BLOOM, DIGITAL WATERMARKING Chapter7, pp215-218
function Dwat= Watson_Dist_Corr(ORIG,DIST)
%***********************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: September 10, 2002
% FUNCTION: Watson_Dist_Corr
% INPUT: Two images to calculate Corrected Watson distance
% DESCRIPTION: returns the Corrected Watson Distance of images of
%             the input.
%             ORDER CONVENTION: the first input is the original.
% RETURNS: A double real corresponding to the corrected Watson distance.
% Watson_dist_Corr calls Freq_Sensi_table, getfirst, Lum_Mask_Corr.
%***********************************************************************
disp('Watson Distance Processing...')
[M,N] = size(ORIG); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
   fprintf(1,'The dimensions of the selected image are not multiples of 8\nand errors will occur;\nTHE PROGRAM IS TERMINATED\N');
   return
 end
wij=0.7;
% Making the DCT
% It is assumed that the input images are in uint8 form [0 255]
ORIG = double(ORIG);
DIST = double(DIST); TD = dctmtx(8);
dctORIG = blkproc(ORIG,[8 8],'P1*x*P2',TD,TD');
dctDIST = blkproc(DIST,[8 8],'P1*x*P2',TD,TD');
% Getting the frequency sensitivity table
T=Freq_Sensi_table;
% getting the mean of the DC components
Coo=blkproc(dctORIG,[8 8],'getfirst');
Coo=mean2(Coo);
% Getting the Luminance Masking
TL=blkproc(dctORIG,[8 8],'Lum_Mask_Corr',T,Coo);
% Getting the slacks
s=zeros(M,N);
for i=1:M
  for j=1:N
    l= (abs(dctORIG(i,j))^wij)*(TL(i,j)^(1-wij));
    s(i,j)=max(TL(i,j),l);
  end
end
% Getting the Corrected Watson distance
d=zeros(M,N);
e=abs(dctORIG-dctDIST);
for i=1:M
  for j=1:N
    d(i,j)=(e(i,j)/s(i,j));
  end
end
d=reshape(d,M*N,1);
Dwat=norm(d,4);
```

```
% References:
% I. COX, M. MILLER, J. BLOOM, DIGITAL WATERMARKING Chapter7, pp215 -218
% Chapter II of the Thesis

function [TL] = Lum_Mask_Corr(dctBlock,T,Coo)
%*********************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: September 21, 2002
% FUNCTION: Lum_Mask_Corr
% INPUT: A block of a DCT transformed matrix dctBlock, a Frequency
%       Sensitivity table T and the mean Coo of the DC components  of
%       all the DCt blocks
% DESCRIPTION: Adjusts Formula 7.3 of the reference to correct for
%            the HVS response to luminances
% RETURNS:  The Luminance masking table for an image.
% Lum_Mask_Corr calls multcorWD
%*********************************************************************
DCcomp= dctBlock(1,1);
% Getting the Luminance Masking table
TLl=T*((DCcomp/Coo)^0.649);
TLl=multcorWD(DCcomp,TLl);
TL=TLl;

% References:
% I. COX, M. MILLER, J. BLOOM, DIGITAL WATERMARKING Chapter7, pp215 -218
% Chapter II of the Thesis
```

```
function [TLc] = multcorWD(DCo,TL)
%***************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: September 10, 2002
% FUNCTION: multcorWD
% INPUT: The DC coefficient of a DCT block and the luminance thresholds for
%        the bock as calculated by the Watson distance formulas
% DESCRIPTION: multcorWD corrects the luminance threshold for the block
%              by multiplying them according to the DC coeff of the
%              block
% RETURNS: A matrix corresponding to the corrected luminance threshold
%          of the block.
%***************************************************************************
if DCo<50
  TLc=65*TL;
  return
elseif DCo<240
  TLc=45*TL;
  return
elseif DCo<320
  TLc=15*TL;
  return
elseif DCo<1600
  TLc=3*TL;
   return
elseif DCo<1920
  TLc=4*TL;
  return
elseif DCo<2000
  TLc=4.5*TL;
  return
else
   TLc=5*TL;
end

% Reference:
% Chapter II of the Thesis
```

```
function vmse= VMSE(ORIG,DIST)
%**************************************************************************
% Giakoumakis Michail
% September 2002
% LAST MODIFICATION: September 29, 2002
% FUNCTION: WMSE
% INPUT: Two images to calculate VMSE
% DESCRIPTION: returns the visual MSE of the two images of the input
%            ORDER CONVENTION: the first input is the original.
% RETURNS: A double real
% vmse calls VMSE_Coeff
%**************************************************************************
disp('VMSE Processing...')
[M,N] = size(ORIG); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
   fprintf(1,'The dimensions of the selected image are not multiples of
          8\nand errors will occur;\nTHE PROGRAM IS TERMINATED\N');
   return
end
% Making the DCT
% It is assumed that the input images are in uint8 form [0 255]
ORIG = double(ORIG);
DIST = double(DIST);
TD = dctmtx(8);
dctORIG = blkproc(ORIG,[8 8],'P1*x*P2',TD,TD');
dctDIST = blkproc(DIST ,[8 8],'P1*x*P2',TD,TD');
coeff=VMSE_Coeff(dctORIG);
vmse=zeros(M,N);
vmse=(dctORIG-dctDIST).^2;
vmse=coeff.*vmse;
vmse=sum(sum(vmse));
vmse=(1/(M*N))*vmse;
return


% Reference:
% Chapter II of the Thesis
```

```
function coeff= VMSE_Coeff(dctORIG)
%************************************************************************
% Giakoumakis Michail
% June, 2002
% LAST MODIFICATION: September 10, 2002
% FUNCTION: VMSE_Coeff
% INPUT: The 8x8 DCT block transform of images to calculate the
%        coefficients of the VMSE
% DESCRIPTION: returns the weight coefficients for
%              ORDER CONVETION: the first input is the original.
% RETURNS: A double real matrix containing the slacks for the image.
% VMSE_Coeff calls Freq_Sensi_table, getfirst, Lum_Mask_Corr.
%************************************************************************
% In our case Just use the Corrected Watson Distance derived weights
[M,N] = size(dctORIG); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
   fprintf(1,'The dimensions of the selected image are not multiples of 8\nand errors will occur;\nTHE
PROGRAM IS TERMINATED\N');
   return
 end
wij=0.7;
% Getting the frequency sensitivity table
T=Freq_Sensi_table;
% getting the mean of the DC components
Coo=blkproc(dctORIG,[8 8],'getfirst');
Coo=mean2(Coo);
% Getting the Luminance Masking
TL=blkproc(dctORIG,[8 8],'Lum_Mask_Corr',T,Coo);
% Getting the slacks
s=zeros(M,N);
for i=1:M
  for j=1:N
    l= (abs(dctORIG(i,j))^wij)*(TL(i,j)^(1-wij));
    s(i,j)=max(TL(i,j),l);
  end
end
for i=1:M
  for j=1:N
    if s(i,j)==0
      s(i,j)=eps;
    end
  end
end
coeff=(1./s).^2;
return

% Reference:
% Chapter II of the Thesis
```

```
function qimet= QImet(image1,image2)
%**************************************************************************
% Giakoumakis Michail
% September 2002
% LAST MODIFICATION: September 29, 2002
% FUNCTION: WMSE
% INPUT: Two grayscale images to calculate the quality index metric
% DESCRIPTION: returns the visual quality index metric for input images
% RETURNS: A double real
% QImet calls img_qi, a function written by ZHOU WANG that can be
% downloaded from
% http://anchovy.ece.utexas.edu/~zwang/research/quality_index
%**************************************************************************
display('Quality Index processing...');
qi=img_qi(image1,image2);
qimet=(abs(1-qi))*50;
return



% References:
% Zhou Wang and Alan C. Bovik: A Universal Image Quality Index
% IEEE Signal Processing Letters, vol. 9, no. 3, March, 2002
% http://anchovy.ece.utexas.edu/~zwang/research/quality_index/demo.html
% Appendix A of the Thesis
```

PART 2: MODIFIED ALGORITHM

```
%**************************************************************************
% Giakoumakis Michail
% November 2002
% LAST MODIFICATION: January 10, 2003
% FILE NAME: Encoder
% DESCRIPTION: This is the main encoding file of our watermarking
%             framework.
% Encoder calls functions:
% (i) imageSelection, WmTypeC, SNR and qFunc that can be found in
%     I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
%     Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
%      March 2002.
% (ii) embedder.
%**************************************************************************
clear all
delete C:\MATLABR11\work\*.mat
disp('Processing...')
%*****************************SETUP SECTION*****************************
% _______________ Select Image to mark - check dimensions ______________
I = imageSelection; % selecting an image for processing from the gallery
save C:\MATLABR11\work\I I
[M,N] = size(I); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
    fprintf(1,'The dimensions of the selected image are not multiples of
            8\nand errors will occur;\nTHE PROGRAM IS TERMINATED\N');
    return
 end
% ______________Select Watermark to use - set dimensions________________
fprintf(1,'The watermark size is set by default to 64x64;\n');
% Any modification of the size should consider the dimensions of the
% image and the embedding size.
Mw = 64;
Nw = 64;
W = WmTypeC(Mw,Nw);
save C:\MATLABR11\work\W W
% ___________________Get parameters from user _________________________
%-------------------------WEIGHTING FACTOR------------------------------
alpha = input('Set the weighting factor alpha (recommended value
            0.1);\n');
disp('Processing...')
save C:\MATLABR11\work\alpha alpha
%-----------------------------XSTART ----------------------------------
start = input('Set the index of the coefficient (1 to 64) where
            the\nembedding would start in each block;\n');
while (start <= 0)|(start >= 64)|(start/fix(start)~=1)
    fprintf(1,'Your choice was either beyond the allowed range or was
             not an integer;\n');
    start = input('Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\start start
%----------------------------EMBEDDING SIZE----------------------------
fprintf(1,'Set the embedding size (number of watermark coefficients per
          block);\n');
length = input('Choose 2, 4 or 8;\n');
```

```
while (length ~= 2)&(length ~= 4)&(length ~= 8)
    start = input('Your choice should be 2, 4 or 8; Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\length length
% __________________________Output choices _____________________________
%-----------------------------CROPPING SETUP---------------------------
flagCrop = input('For cropping press 1; otherwise press 0;\n');
while (flagCrop ~= 0)&(flagCrop ~= 1)
    flagCrop = input('Your choice should be either 0 or 1; Try
                    again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\flagCrop flagCrop
if flagCrop == 1
    leftB = input('Enter the column that will be the new LEFT border of
                the Image;\n');
    disp('Processing...')
    rightB = input('Enter the column that will be the new RIGHT border
                of the Image;\n');
    disp('Processing...')
    upperB = input('Enter the row that will be the new UPPER  border of
                the Image;\n');
    disp('Processing...')
    lowerB = input('Enter the row that will be the new LOWER border of
                the Image;\n');
    disp('Processing...')
    cropParam = [leftB rightB upperB lowerB];
    save C:\MATLABR11\work\cropParam cropParam
end
%-----------------------------QUANTIZATION------------------------------
flagQ = input('For quantization press 1; otherwise press 0;\n');
while (flagQ ~= 0)&(flagQ ~= 1)
    flagQ = input('Your choice should  be either 0 or 1; Try again:\n');
end
disp('Processing...')
%--QUALITY FACTOR--
if flagQ == 1
    q_jpeg = input('Set the quality factor q_jpeg in the range
                [1,100];\n');
    while (q_jpeg < 1)|(q_jpeg > 100)|(q_jpeg/fix(q_jpeg)~=1)
        fprintf('Your choice was either beyond the allowed range or was
                not an integer;\n');
        q_jpeg = input('Try again:\n');
    end
    disp('Processing...')
end
save C:\MATLABR11\work\flagQ flagQ
%-------------------------MARKED IMAGE IN UINT8-------------------------
flag8 = input('For marked image in uint8 press 1; otherwise press
            0;\n');
while (flag8 ~= 0)&(flag8 ~= 1)
    flag8 = input('Your choice should be either 0 or 1; Try again:\n');
end
disp('Processing...')
```

```
save C:\MATLABR11\work\flag8 flag8
%-- IF MARKED IMAGE REAL --> NORMALIZATION
if flag8 == 0
    flagNorm = input('For normalization press 1; otherwise press 0;\n');
    while (flagNorm ~= 0)&(flagNorm ~= 1)
        flagNorm = input('Your choice should be either 0 or 1; Try
                          again:\n');
    end
    disp('Processing...')
    save C:\MATLABR11\work\flagNorm flagNorm
end
 %**************************END OF SETUP SECTION************************
%********************PROCESSING SECTION*****************************
%-----------------------DCT OF THE IMAGE -------------------------------
Id = double(I);
T = dctmtx(8);
dctI = blkproc(Id,[8 8],'P1*x*P2',T,T');
save C:\MATLABR11\work\dctI dctI
%--------------------DCT OF THE WATERMARK------------------------------
Wd = double(W);
dctW = blkproc(Wd,[8 8],'P1*x*P2',T,T');
save C:\MATLABR11\work\dctW dctW
clear Wd
%-------------------------EMBEDDING-----------------------------------
dctI = embedder(dctI,dctW,alpha,start,length);
clear dctW
%---------------IDCT OF MARKED IMAGE COEFFICIENTS----------------------
Im = blkproc(dctI,[8 8],'P1*x*P2',T',T); % scrambled marked image
clear dctI
%---------------------------UINT8 - SNR--------------------------------
if flag8 == 1
    Im = uint8(Im); % Im is the marked image in uint8
    SNR8 = SNR(Id,double(Im));
    fprintf(1,'SNR of uint8 image, SNR8(dB)=%1.4f\n',SNR8);
else
%------------------------NORMALIZATION - SNR---------------------------
    SNRr = SNR(Id,Im);
    fprintf(1,'SNR of real image, SNRr(dB)=%1.4f\n',SNRr);
    Im = Im/255; % reduce Image to range [0 1] plus some distortion
                  caused from the embedding
    if flagNorm == 1
       n = 3.5;  % selected optimal value
        save C:\MATLABR11\work\n n
        Im = 1/pi*atan(n*(Im-1/2))+1/2; % normalization
        SNRnorm = SNR(I,(255*Im));
        fprintf(1,'SNR of real, normalized image,
                  SNRnorm(dB)=%1.4f\n',SNRnorm);
    end
end
save C:\MATLABR11\work\Im Im
%----------------------QUANTIZATION----------------------------------
if flagQ == 1
    if flag8 == 0
        Im = 255*double(Im); % we multiply by 255 to return to the
                               correct scale
```

```
    end
    Imq = qFunc(Im,q_jpeg);
    if flag8 == 0
        SNRrmq = SNR(Id,Imq);
        fprintf(1,'SNR of real, marked and quantized image,
                SNRrmq(dB)=%1.4f\n',SNRrmq);
        Imq = Imq/255;
    else
        Imq = uint8(Imq);
        SNR8mq = SNR(Id,double(Imq));
        fprintf(1,'SNR of uint8, marked and quantized image,
                SNR8mq(dB)=%1.4f\n',SNR8mq);
    end
    save C:\MATLABR11\work\Imq Imq
end
clear Id
%*********************END OF PROCESSING SECTION*************************
%*********************DISPLAY SECTION*********************************
figure(1)
imshow(I)
title('Original Image')
figure(2)
imshow(Im)
title('Marked Image')
if flagQ == 1
    imhist(Imq,64)
end
if flagQ == 1
     figure
     imshow(Imq)
     title('Quantized Marked Image')
end
%*****************END OF DISPLAY SECTION*******************************

% References:
% I. Retsas, "A DCT -based Image Watermarking Algorithm Robust to
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter IV of the Thesis
```

```
function [dctIo] = embedder(dctI,dctW,alpha,start,length)
%*********************************************************************
% GIAKOUMAKIS MICHAIL
% October 2002
% LAST MODIFICATION: January 23, 2003
% FUNCTION: embedder
% INPUT: The matrix dctI which will be marked, the matrix dctW which
%       will mark dctI, the weighing factor alpha, the coefficient in
%       each 8x8 % block where the embedding starts, the number of
%       coefficients that are embedded in each 8x8 block.
% DESCRIPTION: As in Chapter IV of the thesis
% RETURNS: A matrix dctIo with the marked coefficients.
% CAUTION:IT IS REQUIRED THAT [LENGTH] DIVIDES EXACTLY (MW*NW) AND THAT %
(MW*NW/LENGTH)IS EQUAL OR SMALLER THAN THE NUMBER OF 8X8 IMAGE BLOCKS.
% embedder calls functions:
% (i)  zigzag and zzRvs that can be found in
%      I. Retsas, "A DCT -based Image Watermarking Algorithm Robust to
%      Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
%      March 2002.
% (ii) VMSE_Coeff,
%*********************************************************************
[M,N] = size(dctI);
[Mw,Nw] = size(dctW);
% _______________________Get the embedding sets_________________________
[x, index2(Mw*Nw:-1:1)] = sort(abs(dctW(:)));
dctW = dctW(index2); % this way we avoid changing the values to positive
                 after sorting by var
% group the dct coefficients of the watermark in [length] groups
for i = 1:length
    gr(:,i) = dctW((i-1)*Mw*Nw/length+1:i*Mw*Nw/length)';
    indexGr(:,i) = index2((i-1)*Mw*Nw/length+1:i*Mw*Nw/length)';
 end
save C:\MATLABR11\work\indexGr indexGr
K=(Mw*Nw)/length;
% _____________________Get the CIPF of the image blocks_________________
% we mark each 8x8 block with its Eucledean distance from the center
% r(x,y) is the distance of the center of block (x,y) from the center of
% the image
for m=1:8:M
    for n=1:8:N
        r(fix(m/8)+1,fix(n/8)+1) = (((m+3)-M/2)^2 + ((n+3)-N/2)^2)^(1/2);
    end
end
% we (row-wise) reshape the matrix r with the distances
r_line = reshape(r',size(r,1)*size(r,2),1);
% we calculate for each block the CIPF (Center of Interest Proximity
  Factor)
rmax = max(max(r));
CIPF =  -1/pi*atan(14*(r/rmax-2/3))+1/2;
lenCIPF=size(CIPF,1)*size(CIPF,2);
CIPF=reshape(CIPF,1,lenCIPF);
% ________________Get the slacks-Reshape slacks and image_______________
% get slacks
slk = 1./VMSE_Coeff(dctI);
slk=slk/max(max((slk)));
```

```
% Reshaping to get matrices containing the image 8x8 DCT blocks and
% slacks
k=1;
for i = 1:8:M
    S(:,:,k:k+N/8-1) = reshape(slk(i:i+7,:),8,8,N/8); % S is 8x8xK!!!
    k = k+N/8;
end
k = 1;
for i = 1:8:M
    B(:,:,k:k+N/8-1) = reshape(dctI(i:i+7,:),8,8,N/8); % B is 8x8xK!!!
    k = k+N/8;
end
% ___________________ Get the weightinf factor the blocks________________
if (start<5)
    w=0.4;
elseif (start<10)
    w=0.1;
else
    w=0;
end
% _______________________ Get the PC for the blocks______________________
% get the CF
for i = 1:size(B,3)
   V1=abs(zigzag(B(:,:,i)));
   if V1(1)==0
      V1(1)=eps;
   end
   W1=zigzag(S(:,:,i));
   k= (V1(2:64)/V1(1)).*S(2:64);
   F1(i)=sum(k); % The TVSF
   V2 = V1(start:start+length-1);
   W2 = W1(start:start+length-1);
   D=W2.*V2;
   F2(i)=sum(D)/sum(W2); % The EVCF
end
CF=(F1.^w).*(F2.^(1-w));
% Normalize CF
CF=(CF/max(CF));
% Get PC
PC=CIPF.*CF;
% _____________________ __ Perform the Embedding_________________________
% sort the blocks in descending order of PC
[varB(size(B,3):-1:1), index(size(B,3):-1:1)] = sort(PC);
B(:,:,:) = B(:,:,index); % B contains the 8x8 blocks sorted by
                        % descending order of PC
save C:\MATLABR11\work\index index
% embedding
for i = 1:size(gr,1)
    V = zigzag(B(:,:,i)); % V is a row vector that contains the elements
                         % of an 8x8 block aligned in zz fashion.
    V(start:start+length-1) = V(start:start+length-1) + alpha*gr(i,:);
    B(:,:,i) = zzRvs(V);
end
% desorting the 8x8 dct blocks of the image to get their original order
B(:,:,index) = B(:,:,:);
```

```
% ______________________ Return the marked image________________________
 k = 1;
for i = 1:8:M
    dctIo(i:i+7,:) = reshape(B(:,:,k:k+N/8 -1),8,N); % contains the
                                                     % marked dct coeffs
                                                     % of the image
    k = k+N/8;
 end

 return

% References:
% I. Retsas, "A DCT -based Image Watermarking Algorithm Robust to
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter IV of the Thesis
```

```
%***********************************************************************
```

```
% Giakoumakis Michail
% November 2002
% LAST MODIFICATION: February 12, 2003
% FILE NAME: Decoder
% DESCRIPTION: This file recovers the Watermark from a marked Image
% Decoder Call functions:
% (i) BER, BERmod, corCoef, that can be found in
%     I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
%     Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
%     March 2002.
% (ii) extract
%**********************************************************************
%______________________________ DATA LOADING______________________________
clear all
load C:\MATLABR11\work\indexGr
load C:\MATLABR11\work\index
load C:\MATLABR11\work\alpha
load C:\MATLABR11\work\start
load C:\MATLABR11\work\length
load C:\MATLABR11\work\flagCrop
load C:\MATLABR11\work\flagQ
if flagQ == 1
   select = input('Press 0 to process the marked image; press 1 to
            process the quantized, marked image \n');
   while (select ~= 0)&(select ~= 1)
      select = input('Your choice should be either 0 or 1; Try
               again:\n');
   end
end
disp('Processing...')
load C:\MATLABR11\work\flag8
if flag8 == 0
   load C:\MATLABR11\work\flagNorm
end
load C:\MATLABR11\work\I
load C:\MATLABR11\work\W
% load C:\MATLABR11\work\Key
[Mw,Nw] = size(W);
if flagQ == 1
   if select == 0
      load C:\MATLABR11\work\Im
      Itest = Im;
      clear Im
   elseif select == 1
      load C:\MATLABR11\work\Imq
      Itest = Imq;
      clear Imq
   end
else
   load C:\MATLABR11\work\Im
   Itest = Im;
   clear Im
end
Itest = double(Itest);
%__________________________CROPPING____________________________________
```

```
if flagCrop == 1
    load C:\MATLABR11\work\cropParam
    if flag8 == 0
        I1 = 0.5*ones(size(Itest,1),size(Itest,2));
    else
        I1 = 128*ones(size(Itest,1),size(Itest,2));
    end
    I1(cropParam(3):cropParam(4),cropParam(1):cropParam(2)) = ...
        Itest(cropParam(3):cropParam(4),cropParam(1):cropParam(2));
    title_array = strcat('Cropped Marked Image (alpha=', num2str(alpha),
                    ')')
    if flag8 == 0
        figure(5), imshow(I1), title(title_array)
    else
        figure(5), imshow(uint8(I1)), title(title_array)
    end
    Itest =I1;
    clear I1
 end
 [M,N] = size(Itest); % final dimensions after cropping
%___________________________PROCESSING__________________________________
% DENORMALIZATION
if flag8 == 0
    if flagNorm == 1
        load C:\MATLABR11\work\n
        Itest = 1/2 + tan(pi*(Itest-1/2))/n;
    end
    Itest = 255*Itest; % bring to range [0,255]
end
% ORIGINAL IMAGE DCT
load C:\MATLABR11\work\dctI
% TEST IMAGE DCT
T = dctmtx(8);
dctItest = blkproc(Itest,[8 8],'P1*x*P2',T,T');
clear Itest
% RECOVERED WATERMARK DCT (EXTRACTED)
dctWr = extract(dctI,dctItest,Mw,Nw,index,indexGr,alpha,start,length);
clear dctI
clear dctItest
% IDCT ON RECOVERED WATERMARK COEFFICIENTS
Wr = blkproc(dctWr,[8 8],'P1*x*P2',T',T); %recovered watermark
Wr = uint8(round(Wr));
clear dctWr
% BER
ber = BER(W,Wr);
fprintf(1,'BER(bits per pixel)=%1.4f\n',ber);
bermod = BERmod(W,Wr);
fprintf(1,'BERmod(bits per pixel with error)=%1.4f\n',bermod);
% rho
rho = corCoef(W,Wr);
fprintf(1,'rho=%1.4f\n',rho);
figure(5)
imagesc(Wr,[0 255]), colormap(gray), title('Recovered Watermark')
% References:
% I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
```

```
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter IV of the Thesis

function dctW=extract(dctI,dctIm,Mw,Nw,index,indexGr,alpha,start,length)
%*********************************************************************
% Giakoumakis Michail
% Novemebr 2003
% LAST MODIFICATION: January 17, 2003
% FUNCTION: extract
% INPUT: The matrix dctI, the marked matrix dctIm, the watermark
%        dimensions Mw and Nw, the weighting factor a, the embedding
%        size length, xstart and  the indexes for the embedding sets
%        and watermark sorting.
% DESCRIPTION: Uses the input information to recover the watermark.
% RETURNS: The dct coeffs of the retrieved wat ermark.
%*********************************************************************
[M,N] = size(dctI);
dctDif = dctIm - dctI;
k = 1;
for i = 1:8:M % reshape
   B(:,:,k:k+N/8-1) = reshape(dctDif(i:i+7,:),8,8,N/8); % B is 8x8x4!!!
   k = k+N/8;
end
B = B(:,:,index); % sorting using index (:,:,:)
embeddingSetsNumber = Mw*Nw/length;
for i = 1 : embeddingSetsNumber
   V = zigzag(B(:,:,i)); % V is a row vector that contains the elements
                    % of an 8x8 block aligned in zz fashion.
   dctWr(i,:) = V(start:start+length-1)/alpha;
   B(:,:,i) = zzRvs(V);
end
dctWr(indexGr(:)) = dctWr(:); % desorting the dctW coefficients using
                      % indexNew
dctW = reshape(dctWr,Mw,Nw);
```

PART 3: SEMI-BLIND VARIATION

```
%*************************************************************************
% Giakoumakis Michail
% December 2002
% LAST MODIFICATION: February 23, 2003
% FILE NAME: Encoderblind
% DESCRIPTION: This is the main encoding file of the Semi-blind
%             watermarking framework.
% Encoder calls functions:
% (i) imageSelection, WmTypeC, SNR and qFunc that can be found in
%     I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
%     Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
%     March 2002.
% (ii)embedderblind.
%*************************************************************************
clear all
delete C:\MATLABR11\work\*.mat
disp('Processing...')
%******************************SETUP SECTION****************************
% _____________ Select Image to mark - check dimensions ________________
I = imageSelection; % selecting an image for processing from the
                 gallery
save C:\MATLABR11\work\I I
[M,N] = size(I); % M,N are the image dimensions
if ((M/8)/fix(M/8) ~= 1) | ((N/8)/fix(N/8) ~= 1)
   fprintf(1,'The dimensions of the selected image are not multiples of
           8\nand errors will occur;\nTHE PROGRAM IS TERMINATED\N');
   return
end
% _____________ Select Watermark to use - set dimensions_________________
fprintf(1,'The watermark size is set by default to 64x64;\n');
% Any modification of the size should consider the dimensions of the
% image and the embedding size.
Mw = 64;
Nw = 64;
W = WmTypeC(Mw,Nw);
save C:\MATLABR11\work\W W %Needed for the decision making device
save C:\MATLABR11\work\Mw Mw
save C:\MATLABR11\work\Nw Nw
% ______________________Get parameters from user _______________________
%-------------------------WEIGHTING FACTOR------------------------------
alpha = input('Set the weighting factor alpha (recommended value 0.08);\n');
disp('Processing...')
save C:\MATLABR11\work\alpha alpha
%-------------------------XSTART ---------------------------------------
start = input('Set the index of the coefficient (2 to 56) where the\nembedding would start in each block;\n');
while (start <= 1)|(start >= 57)|(start/fix(start)~=1)
   fprintf(1,'Your choice was either beyond the allowed range or was not an integer;\n');
   start = input('Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\start start
%-------------------------EMBEDDING SIZE--------------------------------
fprintf(1,'Set the embedding size (number of watermark coefficients per block);\n');
length = input('Choose 2, 4 or 8;\n');
while (length ~= 2)&(length ~= 4)&(length ~= 8)
```

```
    start = input('Your choice should be 2, 4 or 8; Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\length length
% _________________________Output choices _________________________ ____
%---------------------------CROPPING SETUP----------------------------
flagCrop = input('For cropping press 1; otherwise press 0;\n');
while (flagCrop ~= 0)&(flagCrop ~= 1)
    flagCrop = input('Your choice should be either 0 or 1; Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\flagCrop flagCrop
if flagCrop == 1
    leftB = input('Enter the column that will be the new LEFT border of the Image; \n');
    disp('Processing...')
    rightB = input('Enter the column that will be the new RIGHT border of the Image;\n');
    disp('Processing...')
    upperB = input('Enter the row that will be the new UPPER border of the Image; \n');
    disp('Processing...')
    lowerB = input('Enter the row that will be the new LOWER border of the Image; \n');
    disp('Processing...')
    cropParam = [leftB rightB upperB lowerB];
    save C:\MATLABR11\work\cropParam cropParam
end
%----------------------------QUANTIZATION-----------------------------
flagQ = input('For quantization press 1; otherwise press 0;\n');
while (flagQ ~= 0)&(flagQ ~= 1)
    flagQ = input('Your choice should be either 0 or 1; Try again:\n');
end
disp('Processing...')
%--QUALITY FACTOR--
if flagQ == 1
    q_jpeg = input('Set the quality factor q_jpeg in the range [1,100]; \n');
    while (q_jpeg < 1)|(q_jpeg > 100)|(q_jpeg/fix(q_jpeg)~=1)
        fprintf('Your choice was either beyond the allowed range or was not an integer; \n');
        q_jpeg = input('Try again:\n');
    end
    disp('Processing...')
end
save C:\MATLABR11\work\flagQ flagQ
%------------------------MARKED IMAGE IN UINT8-------------------------
flag8 = input('For marked image in uint8 press 1; otherwise press 0;\n');
while (flag8 ~= 0)&(flag8 ~= 1)
    flag8 = input('Your choice should be either 0 or 1; Try again:\n');
end
disp('Processing...')
save C:\MATLABR11\work\flag8 flag8
%-- IF MARKED IMAGE REAL --> NORMALIZATION
if flag8 == 0
    flagNorm = input('For normalization press 1; otherwise press 0;\n');
    while (flagNorm ~= 0)&(flagNorm ~= 1)
        flagNorm = input('Your choice should be either 0 or 1; Try again:\n');
    end
    disp('Processing...')
    save C:\MATLABR11\work\flagNorm flagNorm
```

```
end
%***************************END OF SETUP SECTION***********************
%************************PROCESSING SECTION*****************************
%----------------------DCT OF THE IMAGE--------------------------------
Id = double(I); % It is assumed that the image from is in uint8 form [0 255]
T = dctmtx(8);
dctI = blkproc(Id,[8 8],'P1*x*P2',T,T');
save C:\MATLABR11\work\dctI dctI
%--------------------DCT OF THE WATERMARK-------------------------------
Wd = double(W);
dctW = blkproc(Wd,[8 8],'P1*x*P2',T,T');
save C:\MATLABR11\work\dctW dctW
clear Wd
%------------------------EMBEDDING-----------------------------------
dctI = embedderblind(dctI,dctW,alpha,start,length);
clear dctW
%-------------IDCT OF MARKED IMAGE COEFFICIENTS-----------------------
Im = blkproc(dctI,[8 8],'P1*x*P2',T',T); % scrambled marked image
clear dctI
%------------------------UINT8 - SNR----------------------------------
if flag8 == 1
   Im = uint8(Im); % Im is the marked image in uint8
   SNR8 = SNR(Id,double(Im));
   fprintf(1,'SNR of uint8 image, SNR8(dB)=%1.4f\n',SNR8);
else
%-----------------------NORMALIZATION - SNR---------------------------
   SNRr = SNR(Id,Im);
   fprintf(1,'SNR of real image, SNRr(dB)=%1.4f\n',SNRr);
   Im = Im/255; % reduce Image to range [0 1] plus some distortion
             % caused from the embedding
   if flagNorm == 1
      n = 3.5;  % selected optimal value
      save C:\MATLABR11\work\n n
      Im = 1/pi*atan(n*(Im-1/2))+1/2; % normalization
      SNRnorm = SNR(I,(255*Im));                    fprintf(1,'SNR of real, normalized image, SNRnorm(dB)=%1.4f\n',SNRnorm);
   end
end
save C:\MATLABR11\work\Im Im
%-----------------------QUANTIZATION----------------------------------
if flagQ == 1
   if flag8 == 0
      Im = 255*double(Im); % we multiply by 255 to return to the
                        % correct scale
   end
   Imq = qFunc(Im,q_jpeg);
   if flag8 == 0
       SNRrmq = SNR(Id,Imq);
      fprintf(1,'SNR of real, marked and quantized image, SNRrmq(dB)=%1.4f\n',SNRrmq);
      Imq = Imq/255;
   else
      Imq = uint8(Imq);
      SNR8mq = SNR(Id,double(Imq));
      fprintf(1,'SNR of uint8, marked and quantized image, SNR8mq(dB)=%1.4f\n',SNR8mq);
   end
```

```
    save C:\MATLABR11\work\Imq Imq
end
clear Id
%************************END OF PROCESSING SECTION*********************
%**********************DISPLAY SECTION********************************
figure(1)
imshow(I)
title('Original Image')
figure(2)
imshow(Im)
title('Marked Image')
if flagQ == 1
    imhist(Imq,64)
end
if flagQ == 1
     figure
     imshow(Imq)
     title('Quantized Marked Image')
end
%********************END OF DISPLAY SECTION****************************

% References:
% I. Retsas, "A DCT -based Image Watermarking Algorithm Robust to
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter V of the Thesis
```

```
function [dctIo] = embedderblind(dctI,dctW,alpha,start,length)
%**************************************************************************
% GIAKOUMAKIS MICHAIL
% October 2002
% LAST MODIFICATION: January 23, 2003
% FUNCTION: embedderblind
% INPUT: The matrix dctI which will be marked, the matrix dctW which
%        will mark dctI, the weighing factor alpha, the coefficient in
%        each 8x8 block where the embedding starts, the number of
%        coefficients that are embedded in each 8x8 block.
% DESCRIPTION: As in Chapter V of the thesis
% RETURNS: A matrix dctIo with the marked coefficients.
% CAUTION:IT IS REQUIRED THAT [LENGTH] DIVIDES EXACTLY (MW*NW) AND THAT %
(MW*NW/LENGTH)IS EQUAL OR SMALLER THAN THE NUMBER OF 8X8 IMAGE BLOCKS.
% is equal or smaller than the number of 8x8 image blocks.
```

```
% embedder calls functions
% (i) zigzag and zzRvs that can be found in
%     I. Retsas, “A DCT-based Image Watermarking Algorithm Robust to
%     Cropping and Compression”. Master's Thesis, NPS, Monterey, CA,
%     March 2002.
% (ii)VMSE_Coeff,
%*********************************************************************
[M,N] = size(dctI);
[Mw,Nw] = size(dctW);
% Simple reshaping of the watermark
dctWvec=reshape(dctW,1,Mw*Nw);
gr=reshape(dctWvec,Mw*Nw/length,length);
K=(Mw*Nw)/length;
% _____________________Get the CIPF of the image blocks__________________
% we mark each 8x8 block with its Eucledean distance from the center
% r(x,y) is the distance of the center of block (x,y) from the center of
% the image
for m=1:8:M
    for n=1:8:N
        r(fix(m/8)+1,fix(n/8)+1) = (((m+3)-M/2)^2 + ((n+3)-N/2)^2)^(1/2);
    end
end
% we (row-wise) reshape the matrix r with the distances
r_line = reshape(r',size(r,1)*size(r,2),1);
% we calculate for each block the CIPF (Center of Interest Proximity
% Factor)
rmax = max(max(r));
CIPF =  -1/pi*atan(14*(r/rmax-2/3))+1/2;
lenCIPF=size(CIPF,1)*size(CIPF,2);
CIPF=reshape(CIPF,1,lenCIPF);
% ________________Get the slacks-Reshape slacks and image________________
% get slacks
slk = 1./VMSE_Coeff(dctI);
slk=slk/max(max((slk)));
% Reshaping to get a martix containing the image 8x8 DCT blocks
k=1;
for i = 1:8:M
    S(:,:,k:k+N/8-1) = reshape(slk(i:i+7,:),8,8,N/8); % S is 8x8xK!!!
    k = k+N/8;
end
k = 1;
for i = 1:8:M
    B(:,:,k:k+N/8-1) = reshape(dctI(i:i+7,:),8,8,N/8); % B is 8x8xK!!!
    k = k+N/8;
 end
 % __________________ Get the weighting factor _________________________
 if (start<5)
    w=0.4;
 elseif (start<10)
    w=0.1;
 else
    w=0;
 end
% _____________________ Get the PC for the blocks__________ _______________
% get the CF
```

```
for i = 1:size(B,3)
  V1=abs(zigzag(B(:,:,i)));
  if V1(1)==0
    V1(1)=1;
  end
  W1=zigzag(S(:,:,i));
  k= (V1(2:64)/V1(1)).*S(2:64);
  F1(i)=sum(k); %TVSF
  V2 = V1(start:start+length-1);
  W2 = W1(start:start+length-1);
  D=W2.*V2;
  F2(i)=sum(D)/sum(W2); %EVCF
end
CF=(F1.^w).*(F2.^(1-w));
% Normalize CF
CF=(CF/max(CF));
% Get PC
PC=CIPF.*CF;
% _______________________ Perform the Embedding__________________________
% sort the blocks in descending order of PC
[varB(size(B,3):-1:1), index(size(B,3):-1:1)] = sort(PC);
B(:,:,:) = B(:,:,index); % B contains the 8x8 blocks sorted by
                    % descending order of PC
save C:\MATLABR11\work\index index
% embedding
for i = 1:size(gr,1)
   V = zigzag(B(:,:,i)); % V is a row vector that contains the elements
                    % of an 8x8 block aligned in zz fashion.
   V(start:start+length-1) = V(start:start+length-1) + alpha*gr(i,:);
   B(:,:,i) = zzRvs(V);
end
% desorting the 8x8 dct blocks of the image to get their original order
B(:,:,index) = B(:,:,:);

% _____________________ Return the marked image_________________________
k = 1;
for i = 1:8:M
   dctIo(i:i+7,:) = reshape(B(:,:,k:k+N/8-1),8,N); % contains the
                                        % marked dct coefs
                                        % of the image
   k = k+N/8;
end

return

% References:
% I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter IV of the Thesis
```

```
%***********************************************************************
% Giakoumakis Michail
% November 2002
% LAST MODIFICATION: February 12, 2003
% FILE NAME: Decoderblind
% DESCRIPTION: This file recovers the Watermark from a marked Image with
%             no apriori access to the watermark.
% Decoderblind calls functions:
% (i) BER, BERmod, corCoef, that can be found in
%     I. Retsas, "A DCT-based Image Watermarking Algorithm Robust to
%     Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
%     March 2002.
% (ii) extractblind
%***********************************************************************
%______________________________DATA LOADING____________________________
```

```
clear all
load C:\MATLABR11\work\index
load C:\MATLABR11\work\alpha
load C:\MATLABR11\work\start
load C:\MATLABR11\work\length
load C:\MATLABR11\work\flagCrop
load C:\MATLABR11\work\flagQ
load C:\MATLABR11\work\Mw
load C:\MATLABR11\work\Nw
if flagQ == 1
   select = input('Press 0 to process the marked image; press 1 to
                process the quantized, marked image \n');
   while (select ~= 0)&(select ~= 1)
      select = input('Your choice should be either 0 or 1; Try
                   again:\n');
   end
end
disp('Processing...')
load C:\MATLABR11\work\flag8
if flag8 == 0
   load C:\MATLABR11\work\flagNorm
end
load C:\MATLABR11\work\I %the original image needed
% load C:\MATLABR11\work\Key %-> load in case keying was used
if flagQ == 1
   if select == 0
      load C:\MATLABR11\work\Im
      Itest = Im;
      clear Im
   elseif select == 1
      load C:\MATLABR11\work\Imq
      Itest = Imq;
      clear Imq
   end
else
   load C:\MATLABR11\work\Im
   Itest = Im;
   clear Im
end
Itest = double(Itest);
%*********************************************************************
%***************************CROPPING**********************************
if flagCrop == 1
   load C:\MATLABR11\work\cropParam
   if flag8 == 0
      I1 = 0.5*ones(size(Itest,1),size(Itest,2));
   else
      I1 = 128*ones(size(Itest,1),size(Itest,2));
   end
   I1(cropParam(3):cropParam(4),cropParam(1):cropParam(2)) = ...
      Itest(cropParam(3):cropParam(4),cropParam(1):cropParam(2));
   title_array = strcat('Cropped Marked Image (alpha=', num2str(alpha), ')')
   if flag8 == 0
      figure(5), imshow(I1), title(title_array)
   else
```

```
        figure(5), imshow(uint8(I1)), title(title_array)
    end
    Itest =I1;
    clear I1
 end
 [M,N] = size(Itest); % final dimensions after cropping
%*******************************************************************
%******************************PROCESSING**************************
%--------------------------DENORMALIZING-----------------------------
if flag8 == 0
    if flagNorm == 1
        load C:\MATLABR11\work\n
        Itest = 1/2 + tan(pi*(Itest-1/2))/n;
    end
    Itest = 255*Itest; % bring to range [0,255]
end
%---------------------ORIGINAL IMAGE DCT -------------------------------
load C:\MATLABR11\work\dctI
%---------------------------TEST IMAGE DCT ---------------------------------
T = dctmtx(8);
dctItest = blkproc(Itest,[8 8],'P1*x*P2',T,T');
clear Itest
%---------------------RECOVERED WATERMARK DCT (EXTRACTED)--------------
dctWr = extractblind(dctI,dctItest,Mw,Nw,index,alpha,start,length);
clear dctI
clear dctItest
%------------IDCT ON RECOVERED WATERMARK COEFFICIENTS-----------------
Wr = blkproc(dctWr,[8 8],'P1*x*P2',T',T); %recovered watermark
Wr = uint8(round(Wr));
clear dctWr
load C:\MATLABR11\work\W %the original watermark. Not needed for the
                    % decoder.
                    % We use it to calculate BER and rho
%------------------------------BER------------------------------------
ber = BER(W,Wr);
fprintf(1,'BER(bits per pixel)=%1.4f\n',ber);
bermod = BERmod(W,Wr);
fprintf(1,'BERmod(bits per pixel with error)=%1.4f\n',bermod);
%------------------------------rho------------------------------------
rho = corCoef(W,Wr);
fprintf(1,'rho=%1.4f\n',rho);

figure(11)
imagesc(Wr,[0 255]), colormap(gray), title('Recovered Watermark')

% References:
% I. Retsas, "A DCT -based Image Watermarking Algorithm Robust to
% Cropping and Compression". Master's Thesis, NPS, Monterey, CA,
% March 2002.
% Chapter IV of the Thesis
```

```
function dctWr=extractblind(dctI,dctIm,Mw,Nw,index,alpha,start,length)

%*************************************************************** ****
% Giakoumakis Michail
% December 2002
% LAST MODIFICATION: March 2, 2003
% FUNCTION: extractblind
% INPUT: The matrix dctI, the marked matrix dctIm, the watermark
%       dimensions Mw & Nw, the weighting factor a, the embedding
%       size lengt h and xstart.
% DESCRIPTION: Uses the input information to recover the watermark.
% RETURNS: The dct coeffs of the retrieved watermark.
%******************************************************************* [M,N]          =
size(dctI);
dctDif = dctIm - dctI;
k = 1;
for i = 1:8:M % reshape
   B(:,:,k:k+N/8-1) = reshape(dctDif(i:i+7,:),8,8,N/8); % B is 8x8x4!!!
```

```
    k = k+N/8;
end
B = B(:,:,index); % sorting using index (:,:,:)
embeddingSetsNumber = Mw*Nw/length;
for i = 1 : embeddingSetsNumber
    V = zigzag(B(:,:,i)); % V is a row vector that contains the elements
                            of an 8x8 block aligned in zz fashion.
    dctWr(i,:) = V(start:start+length-1)/alpha;
    B(:,:,i) = zzRvs(V);
end
dctWr = reshape(dctWr,Mw,Nw);
```

LIST OF REFERENCES

[1] I. Retsas, "A DCT-Based Image Watermarking Algorithm Robust to Cropping and Compression", Master's Thesis, Naval Postgraduate School, Monterey, California, March 2002.

[2] I. J. Cox, M. L. Miller, and J. A. Bloom, "Digital Watermarking", Morgan Kaufmann Publishers, 2002.

[3] J. Lacy, D. P. Maher, and J. H. Snyder, "Media on the Internet and the Intellectual Property Protection Problem", Proc. International Symposium on Industrial Electronics, Guimaraes, Portugal, July 1997.

[4] J. Lacy et. al., "Watermarking and Intellectual Property Protection", Submitted to the Second International Information Hiding Workshop, Portland, Oregon, 1998.

[5] Jian-Guo Cao, James E. Fowler, and Nicholas H. Younan, "An Image-Adaptive Watermark Based on a Redundant Wavelet Transform", Proc. IEEE International Conference on Image Processing, Thessaloniki, Greece, October 2001.

[6] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques", Proc. IEEE Special Issue on Protection of Multimedia, Vol. 87, July 1999.

[7] "Information Hiding: A Survey", Proc. IEEE Special Issue on Protection of Multimedia, Vol. 87, July 1999.

[8] M. B. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", Proc. IEEE Vol. 86, June 1998.

[9] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual Watermarks for Digital Images and Video", Proc. IEEE, Vol. 87, No. 7, July 1999.

[10] J. K. Su, J. J. Eggers, and B. Girod, "Optimum Attack on Digital Watermarks and its Defence", Proc. 34th Asilomar Conference on Signals, Systems and Computers Asilomar, California, November 2000.

[11] R. G. Van Schyndell, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark," in *Proc. ICIP* Vol. 2, Austin, Texas, November 1994.

[12] E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labelling", Nonlinear Signal and Image Processing Workshop, Greece, 1995.

[13] C. S. Lu, S. K. Huang, C. J. Sze, and H. Y. M. Liao, "Cocktail Watermarking for Digital Image Protection", IEEE Transactions on Multimedia, Vol. 2, December 2000.

[14] C. R. Carlson and R. Cohen, "A Simple Psychophysical Model for Predicting the Visibility of Displayed Information," Proc. Society for Information Display, Vol. 21, 1980.

[15] Wickens, C., Gordon, S. and Liu, Y., *"An Introduction to Human Factors Engineering"*, New York – Addison Wesley Longman, 1998.

[16] Sekuler R. and Blake R., *"Perception"*, 4th Edition, New York - McGraw Hill, 2002.

[17] S. S. Henami, "Perception of Extremely Low -Rate Images & Video: Psychophysical Evaluations and Analysis", Cornell University Visual Communications Lab, 2001.

[18] Z. Wang, A. C. Bovic, "Why is Image Quality Assessment So Difficult", IEEE International Conference on Acoustics, Speech, & Signal Processing, Vol. 4, May 2002.

[19] J. Mannos and D. Sakrison, "The Effects of Visual Fidelity Criterion on the Encoding of Images", IEEE Trans. Information Theory, Vol. 20-Issue 4, 1974.

[20] A. B. Watson, "DCTune: A Technique for Visual Optimization of DCT Quantization Matrices for Individual Images, SID Digest of Technical Pa pers XXIV, 1993.

[21] T. P. O'Rourke and R. L. Stevenson, "Human Visual System Based Wavelet Decomposition for Image Compression", Journal of Visual Communication and Image Representation, Vol. 6 Issue 2, 1995.

[22] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of Wavelet Quantisation Noise", IEEE Trans. Image Processing, Vol. 6, August 1997.

[23] Z. Wang and A. C. Bovic, "A Universal Image Quality Index", IEEE Signal Processing Letters, Vol. 9 – March 2002.

[24] M. P. Eckert and A. P. Brandley, "Perceptual Quality Metrics Applied to Still Image Compression", Signal Processing, Vol. 70, Issue 3, November 1998.

[25] R. B. Wolfgang, C. I. Podilchuk and E. J. Delp, "Perceptual Watermarks for Digital Images and Video", Proceedings of the IEEE, Vol. 87, No. 7, pp. 1108-1126, July 1999.

[25] M. Ramkumar, A. N. Akansu and A. A. Alatan, "On the Choice of Transforms for Data Hiding in Compressed Video", *IEEE International Conference on Acoustics, Speech, and Signal Processing 1999, Proceeding s*, Vol. 6, 1999.

[26] R. C. Gonzales and Paul Wintz, "Digital Image Processing", Addison-Wesley Publishing Company, Inc., 1987.

[27] B. Girod, *"What's Wrong with the MSE"*, Digital Images and Human Vision-MIT Press 1993.

[28] A. B. Watson, *"DCT Quantitization Matrices Optimized for Individual Images"*, Human Vision, Visual Processing and Digital Display, SPIE 2697,1993.

[29] S. Suthaharan, S. W. Kim, H. K. and Lee, S., Sathananthan, "Perceptually Tuned Robust Watermarking Scheme for Digital Images", Pattern Recognition Letters 21, 2000.

[30] C. I. Podilchuk and W. Zeng, "Perceptual Watermarking of Still Images", *IEEE First Workshop on Multimedia Signal Processing*, 1997.

[31] C.-T. Hsu, J.-L. Wu, "Hidden Digital Watermarks in Images"**,** *IEEE Transactions on Image Processing*, Vol. 8, No. 1, pp. 58-68, January 1999.

[32] I. Retsas, R. Pieper and R. Cristi, "Watermark Recovery with a DCT-Based Scheme Employing Nonuniform Imbedding", *34th Southeastern Symposium on System Theory (SSST-02)*, Alabama, March 18-19, 2002.

[33] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet and T. Pun, "Second Generation Benchmarking and Application Oriented Evaluation", [http://vision.unige.ch/], January 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman
   Department of Electrical and Computer Engineering
   Monterey, California

4. Chairman
   Information Warfare Academic Group
   Monterey, California

5. Prof. Roberto Cristi
   Department of Electrical and Computer Engineering
   Monterey, California

6. Prof. Ron Pieper
   Department of Electrical and Computer Engineering
   Monterey, California

7. Prof. Craig Rasmussen
   Department of Mathematics
   Monterey, California

8. Hellenic Navy General Staff
   Holargos, 15500, Athens
   Greece

9. Lt Giakoumakis Michail HN
   Kallithea, Athens, 17671
   Greece