Αποδοτικοί Αλγόριθμοι Υδατοσήμανσης Εικόνων με Χρήση Ιδιοτήτων Μεταθέσεων

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνθεσης

του Τμήματος Μηχανικών Η/Υ & Πληροφορικής

Εξεταστική Επιτροπή

από τον

Άγγελο Φυλάκη

ως μέρος των Υποχρεώσεων για τη λήψη του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ

ΣΤΗΝ ΘΕΩΡΙΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ιούλιος 2013

Efficient Algorithms for Image Watermarking using Permutation Properties

MSc Thesis

Department of Computer Science & Engineering

University of Ioannina
GREECE

Angelos Fylakis

July 2013

Dedication

Η εργασία είναι αφιερωμένη...

...στον επιβλέποντα καθηγητή μου κύριο Σταύρο Δ. Νικολόπουλο...

...και στους γονείς μου, Περικλή και Ιωάννα.

Acknowledgements

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Σταύρο Δ. Νικολόπουλο, γιατί η αμέριστη συμπαράσταση και η πολύτιμη καθοδήγησή του ήταν καθοριστικές στην ολοκλήρωση της Μεταπτυχιακής Εργασίας. Τον ευχαριστώ για την εμπιστοσύνη που μου έδειξε αναλαμβάνοντας την επίβλεψή μου αλλά και για την συνεργασία που είχαμε για την παραγωγή ερευνητικού έργου. Θα ήθελα επίσης να ευχαριστήσω για την καθοριστική συνεργασία που είχαμε την υποψήφια διδάκτορα Μαρία Χρόνη. Τέλος, ευχαριστώ πολύ τα μέλη της τριμελούς επιτροπής, τον επίκουρο καθηγητή κύριο Χρήστο Νομικό και τον επίκουρο καθηγητή κύριο Χριστόφορο Νίκου για την συμμετοχή τους στην επιτροπή εξέτασης αυτής της εργασίας.

Table of Contents

List of Figures

List of Tables

Glossary

w: Watermark's Numerical Value

P: Host Digital Object

P_w: Watermarked Digital Object

I: Digital Image

I_w: Watermarked Digital Image

π: Permutation

π^*: Self-inverting Permutation

c: Additive Value for Robust Marks

c_{opt}: Optimal Additive Value (returned by function f)

$PSNR$: Peak Signal to Noise Ratio (Image Quality Metric Measured in dB)

$SSIM$: Structural Similarity Metric (Image Quality Metric)

Q: JPEG Compression Quality

Abstract

Angelos Fylakis
Department of Computer Science and Engineering, University of Ioannina, Greece
June 2013
Efficient Algorithms for Image Watermarking Using Permutation Properties
Thesis Supervisor: Stavros D. Nikolopoulos

This Master thesis proposes efficient codec algorithms based on graph properties for watermarking images that are intended for uploading on the web under intellectual property protection.

Before the description of the watermarking algorithms the thesis includes after an introduction that mentions briefly the theory behind intellectual property, the theory and the definitions about watermarking in general and digital image watermarking. There is also a part dedicated to the classification of the image watermarking techniques.

Specifically image watermarking techniques fall in two main categories. Those that embed information in the spatial domain of the images and those that embed information in the frequency domain of the images and of course there are also their subcategories as well which will be further analyzed. As for the described techniques there will be two approaches, one for each category.

Then there is a reference on previous work on digital image watermarking algorithms for each category. This part includes from each one of the previously described categories some of the most important algorithms that have been implemented over the time as well as some information about each implementation.

Now, headed to this thesis's scope direction, first there is our main idea which is going to be thoroughly described. This idea presents a way in which an integer number w that is in a numerical form, is being transformed into a self-inverting permutation. Having this transformation into a permutation, it can be later easily represented as a two dimensional (2D) object and thus, since images are 2D structures, we propose algorithms that embed marks into images using the 2D representation of w.

Based on the mentioned idea, first there is a report on the initial illustration using it, which embeds marks in the spatial domain of the images. Afterwards we describe the enhanced technique exploiting the same idea by now embedding marks in the frequency domain. To do that we use the discrete fourier transform for the regions we wish to mark.

Those modifications are made on the magnitude of specific frequencies and they are the least possible additive value ensuring robustness and imperceptiveness.

We have experimentally evaluated our algorithms using various images of different characteristics under JPEG compression. There is also a reference on some properties of the permutations that offer robustness against certain geometric attacks. Concerning the frequency domain algorithm, The experimental results show an improvement in comparison to the previously obtained results and they also depict the validity of our proposed codec algorithms.

Also there is an image quality assessment that includes results concerning the latest algorithm and its behavior against JPEG image compression with different compression ratios ($Q = 60$, $Q = 75$ and $Q = 90$) for multiple images of different characteristics grouped in three different size classes (200×200, 500×500 and 1024×1024). Following after the image quality assessment are other experimental outcomes.

Last and closing with this master thesis, there will be the conclusions for the proposed technique as well as suggestions and extensions for the future.

Εκτεταμενη Περιληψη στα Ελληνικα

Άγγελος Φυλάκης
Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Ελλάδα
Ιούνιος 2013
Αποδοτικοί Αλγόριθμοι Υδατοσήμανσης Εικόνων με Χρήση Ιδιοτήτων Μεταθέσεων
Επιβλέπων Καθηγητής: Σταύρος Δ. Νικολόπουλος

Η παρούσα μεταπτυχιακή εργασία προτείνει αποδοτικούς αλγορίθμους βασισμένους σε ιδιότητες γραφημάτων για υδατοσήμανση εικόνων με σκοπό την κοινοποίηση στο διαδίκτυο με προστασία των πνευματικών τους ιδιοτήτων.

Πριν προχωρήσουμε με τους αλγόριθμους υδατοσήμανσης η εργασία περιλαμβάνει έπειτα από μία εισαγωγή η οποία αναφέρει σύντομα την θεωρία πίσω από την πνευματική ιδιοκτησία, την θεωρία και ορισμούς για την υδατοσήμανση γενικά αλλά και συγκεκριμένα για τις τεχνικές υδατοσήμανσης εικόνων.

Συγκεκριμένα όσον αφορά τις τεχνικές υδατοσήμανσης εικόνων, συναντώνται σε δύο βασικές κατηγορίες. Αυτές όπου η πληροφορία εισάγεται στην χωρικό τομέα των εικόνων και αυτές όπου η πληροφορία εισάγεται στον τομέα των συχνοτήτων των εικόνων και φυσικά υπάρχουν και οι υποκατηγορίες για τις οποίες θα δοθούν περισότερες λεπτομέρειες στην εργασία. Όσων αφορά τους αλγορίθμους που επρόκειτο να περιγραφούν, θα υπάρξουν δύο προσεγγίσεις, μία για κάθε κατηγορία.

Στην συνέχεια της εργασίας θα γίνει αναφορά στους αλγορίθμους κάθε κατηγορίας που έχουν εξελιχθεί μέχρι σήμερα. Συγκεκριμένα, το σχετικό κομμάτι της εργασίας συμπεριλαμβάνει για κάθε μία από τις προηγουμένως αναλυθείσες κατηγορίες μερικούς από τους πιο σημαντικούς αλγορίθμους που έχουν υλοποιηθεί καθώς και μερικές πληροφορίες για κάθε υλοποίηση.

Τώρα, φτάνοντας στον κύριο σκοπό της εργασίας, πρώτα θα γίνει μία εκτενής αναφορά στην κεντρική μας ιδέα, σύμφωνα με την οποία ένας ακέραιος αριθμός w από αριθμητική μορφή μετατρέπεται σε μία αυτο-αναστρεφόμενη μετάθεση. Αυτή η μορφή της μετάθεσης, μπορεί αργότερα να αναπαρασταθεί σαν ένα δυσδιάστατο αντικείμενο και έτσι από την στιγμή που οι εικόνες είναι δυσδιάστατες δομές προτείνουμε αλγορίθμους που εισάγουν σημεία (marks) στις εικόνες εξιοποιώντας την δυσδιάστατη αναπαράσταση του w.

Με βάση την παραπάνω ιδέα, αρχικά γίνεται αναφορά στην πρωταρχική υλοποίηση σύμφωνα με την οποία εισάγονται σημεία (marks) στον χωρικό τομέα των εικόνων. Έπειτα

περιγράφουμε την βελτιωμένη τεχνική, η οποία και πάλι αξιοποιεί την ίδια ιδέα εισάγοντας αυτήν την φορά σημεία (marks) στον χώρο των συχνοτήτων και συγκεκριμένα κάτι τέτοιο γίνεται χρησιμοποιώντας τον διακριτό μετασχηματισμό fourier για τις περιοχές που θέλουμε να μαρκάρουμε. Αυτές οι τροποποιήσεις γίνονται στο μέτρο συγκεκριμένων συχνοτήτων και έχουν της μορφή της ελάχιστης επιπρόσθετης πληροφορίας στις εικόνες η οποία εξασφαλίζει ανθεκτικότητα και συγχρόνως δεν γίνεται αντιληπτή.

Έχουμε αξιολογήσει πειραματικά τους αλγορίθμους με χρήση πολλών διαφορετικών εικόνων, με διαφορετικά χαρακτηριστικά και χρήση JPEG συμπίεσης. Γίνεται επίσης αναφορά, σε κάποιες ιδιότητες των μεταθέσεων που προσφέρουν ανθεκτικότητα έναντι σε συγκεκριμένες γεωμετρικές επιθέσεις. Όσων αφορά την μέθοδο στον τομέα των συχνοτήτων, τα πειραματικά αποτελέσματα παρουσιάζουν βελτίωση σε σχέση με τα προηγούμενα. Έτσι, προσδίδεται εγκυρότητα στον νέο αλγόριθμο.

Ακόμη θα δούμε μία αξιολόγηση της ποιότητας των υδατοσημασμένων εικόνων με αποτελέσματα που αφορούν τον τελευταίο αλγόριθμο και την συμπεριφορά του έναντι σε JPEG συμπίεση με διαφορετικούς βαθμούς συμπίεσης (για $Q = 60$, $Q = 75$ και $Q = 90$) για πολλές εικόνες διαφορετικών χαρακτηριστικών ομαδοποιημένες σε τρείς ομάδες διαστάσεων (200×200, 500×500 και 1024×1024). Στην συνέχεια γίνεται αναφορά και σε λοιπά πειραματικά αποτελέσματα.

Τέλος και κλείνοντας με την μεταπτυχιακή εργασία, θα γίνει αναφορά στα συμπεράσματα που βγήκαν από την προτεινόμενη τεχνική όπως και προτάσεις για επεκτάσεις στο μέλλον.

Chapter 1

Introduction

1.1 Thesis's Scope

Internet technology, in modern communities, becomes day by day an indispensable tool for everyday life since most people use it on a regular basis and do many daily activities online [29]. This frequent use of the internet means that measures taken for internet security are indispensable since the web is not risk-free [16, 22]. One of those risks is the fact that the web is an environment where intellectual property is under threat since a huge amount of public personal data is continuously transferred, and thus such data may end up on a user who falsely claims ownership.

It is without any doubt that images, apart from text, are the most frequent type of data that can be found on the internet. As digital images are a characteristic kind of intellectual material, people hesitate to upload and transfer them via the internet because of the ease of intercepting, copying and redistributing in their exact original form [64]. Encryption is not the problem's solution in most cases, as most people that upload images in a website want them to be visible by everyone, but safe and theft protected as well. Watermarks are a solution to this problem as they enable someone to claim an image's ownership if he previously embedded one in it. Image watermarks can be visible or not,

but if we do not want any cosmetic changes in an image then an invisible watermark should be used.

That is what this thesis suggests, techniques according to which invisible watermarks are embedded into images using features of the image's spatial or frequency domain and graph theory as well.

The purpose of this work can be found on the fact that intellectual property protection and proper use are some of the greatest concerns over internet users today. That is because the domination of the internet after 1995 allowed users to transfer digital material easy, fast and efficiently. This also resulted in the DCMA (Digital Millennium Copyright Act) which was voted by the congress in 1998. The DCMA criminalizes production and dissemination of technology, devices, or services intended to circumvent measures, commonly known as digital rights management or DRM, that control access to copyrighted works. It also criminalizes the act of circumventing an access control, whether or not there is actual infringement of copyright itself. In addition, the DMCA heightens the penalties for copyright infringement on the Internet [70].

The thing is though, that an individual needs to validate somehow his copyrighted material, i.e. his intellectual property, in order to be able to claim it and thus enabled to protect it. And this is where Watermarking and in our case Image Watermarking comes to place. That is because Image Watermarking is a technique that serves this purpose ideally as in contrast with other techniques it allows images to be available to third internet users but simultaneously carry an "identity" that is actually the proof of ownership with them. This way image watermarking achieves its target of deterring copy and usage without permission from the owner.

There are already various techniques that can achieve that (See Chapter 2), but every single method requires attention because we can not discriminate a specific one as the best. Every case has its ideal solution and the same rule applies for image watermarking.

What this thesis's technique suggests, is an efficient and robust to specific transformations such as JPEG compression image watermarking technique for watermarking images based on graph properties. The important fact for this idea, is that it suggests a way in which an integer number w can be represented with a two dimensional representation (or, for short, 2D representation) via an interim representation of a self inverting permutation graph. Thus, since images are two dimensional objects that representation can be efficiently marked on them resulting the watermarked images. In a similar way, such a 2D representation can be extracted from a watermarked image and converted back to the integer w.

Concerning the mentioned interim state, this work suggests an efficient algorithm for encoding a self-inverting permutation π^* into an image I by first mapping the elements of π^* into an $n^* \times n^*$ matrix A^* and then using the information stored in A^* to mark specific areas of image I resulting the watermarked image I_w. This is done by placing an imaginary grid on the image and marking specific cells of this grid. That was initially archived by embedding marks in the image's spatial domain and later the method was

enhanced by now embedding them in the frequency domain of the image.

Vice versa, this work also suggests an efficient algorithm for extracting the embedded self-inverting permutation π^* from the watermarked image I_w by locating the positions of the marks in I_w. That enables us to reconstruct the 2D representation of the self-inverting permutation π^* and thus extract the embedded watermark w.

As mentioned in a previous paragraph, The suggested watermarking technique has properties that make it robust to multiple transformations which are further analyzed at Chapter 4 dedicated to the evaluation of our watermarking method. What is more and even new about the marking process in comparison to the other known watermarking techniques is the fact that it marks partial parts of images, also referred as grid-cells hereafter, and not the image overall, either on the spatial or the frequency domain.

This was done by first transforming a watermark from a numerical form into a two dimensional (2D) representation and, since images are 2D structures, it is possible to efficiently embed the 2D representation of the watermark by marking either in the spatial domain or in the the high frequency bands of specific areas of an image's frequency domain. The key idea behind our extracting method is that it does not actually extract the embedded information instead it locates the marked areas reconstructing the permutation π^* and thus the watermark's numerical value w (See, Figure 1.1).

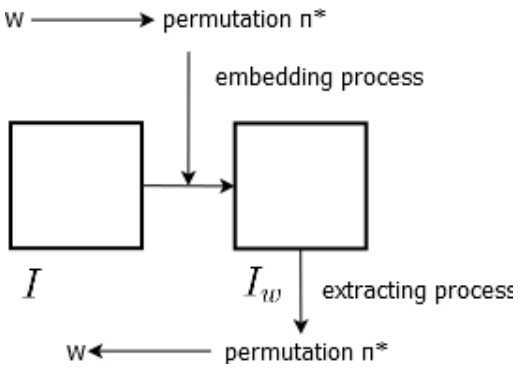


Figure 1.1: An illustration of the main idea behind the watermarking technique.

Closing the scope of this thesis, it should be also pointed out, that marking in the spatial domain was the initial illustration of the idea used in our research [14] that led to this thesis, and to the following works and publications [12, 11] where it was enhanced by using the same background concept in the frequency domain in order to get a more robust watermarking technique of higher fidelity.

In Chapter 3 there are references, and step-by-step descriptions of the initial method of marking in the spatial and of the enhanced method using marks in the frequency domain.

Following, in the next sections of this chapter there will be a description of what is intellectual property and what exactly is watermarking in both the physical and the digital world. Further, there is a reference of how it all started by presenting a brief history of watermarking. And the last section of this chapter includes a Road Map showing the layout of this thesis.

1.2 Intellectual Property

Before we continue with the notion of watermarking, what is intellectual property? The term intellectual property (IP) refers to a creation of a mind for which a set of exclusive rights are recognized [73]. That creation may have any form possible; for example, it may be a work of art, an invention, literary or artistic work, a discovery or even a phrase. More precisely, IP can be divided into two categories: industrial property , which includes inventions (patents), trademarks, industrial designs, and geographic indications of source; and copyright, which includes literary and artistic works such as novels, poems, plays, films, musical works, drawings, paintings, photographs, sculptures, and architectural designs.

The objective of recognizing intellectual property is to encourage innovation. This is because people won't have the incentive to create if they are not legally protected in order to get the social value that they deserve from their creations [52]. Of course the world's evolution and economic growth depend on creations and inventions and that makes intellectual property such an important and vital aspect [40].

There used to be laws protecting intellectual material, but it was not until the 19th century that the term "Intellectual Property" was used for the first time. The first modern usage of the term, goes back to 1867 when the North German Confederation granted legislative power over the protection of intellectual property. In 1893, the United International Bureaux for the Protection of Intellectual Property was established in Berne. Later, in 1960 this organization was relocated to Geneva, until 1967 when it was succeeded with the establishment of the World Intellectual Property Organization as an agency of the United Nations. That was also the time when the term became also popular in the United States [52].

Over the last years the internet has been expanding very rapidly and, thus, information is now spread freely, easily and cost-efficiently and that gives a greater importance to intellectual property. Because of the internet, the distribution of intellectual material went out of control. Just the fact that nearly every intellectual material that is produced today is published in digital form or can be transformed into digital form means that it can be easily transmitted free via the internet, without any permission from the creator.

The cyberspace is chaotic nowadays and that makes it difficult to have any kind of control over it. The figures talk by themselves; according to IFPI (International Federation of the Phonographic Industry) 95% of music downloads are pirated. What is more, a survey from Digital Life America showed as that things aren't any better for the movies. So, all that urged the adoption of new laws and the development of systems for the protection of intellectual property and that's where watermarking techniques also come to place.

1.3 The Notion of Watermarking

Nowadays, we can find Watermarks in nearly every official document. For example if you hold a bank note you may notice watermarks which are in fact various figures designed on the bank note's surface. Their purpose is to carry information about the object in which they are hidden and they are designed in such a way so that to be as difficult as possible to be reproduced by counterfeiting methods [21].

You may also notice that some watermarks may be directly visible with the human eye, while others are hidden from view during normal use and only become visible under special viewing processes such as specific viewing angle or specific lighting conditions i.e. perceptible and imperceptible watermarks.

Note that when most researchers refer to watermarks, and specifically digital watermarks, they consider the imperceptibility as a defining characteristic. Others of course do not take it as a defining characteristic considering watermarks either perceptible or imperceptible. More details about imperceptiveness are about to follow in Chapter 2.

The watermarks from the example with the bank notes are obviously in the second category as they are hidden from view during normal use and become visible as a result of special viewing processes. This is also the kind of digital watermarks that this work is dealing with. So hereafter the term watermarking refers to imperceptible alterations.

Of course, when we refer to watermarks it does not necessarily mean that they are only found on papers. In fact, they can be of any imaginable form and can be found on any object, physical or not such as, fabrics, garment labels and packages using special inks or music photographs or video using electronic signals respectively [75].

So generally we may define watermarks as symbols which are placed into physical objects such as documents, photos, etc. and their purpose is to carry information about objects' authenticity. More precisely watermarking can be described as the problem of embedding a watermark w into an object P and, thus, producing a new object P_w, such that w can be reliably located and extracted from P_w even after P_w has been subjected to transformations [17] (See, Figure 1.2).

In a similar manner we can tell that the generic watermarking system consists of the two processes.

- The embedding process.
- The extraction process.

Briefly, the two processes can be described as follows:

The embedding process takes as an input the cover work and the value of the watermark w and returns as an output the watermarked work.

Whereas the extracting process takes as an input the watermarked work and returns the value of the detected watermark w as an output.

Note that a technique similar to watermarking is steganography. Do not confuse these two terms as there is a slight but important difference between them. In steganography

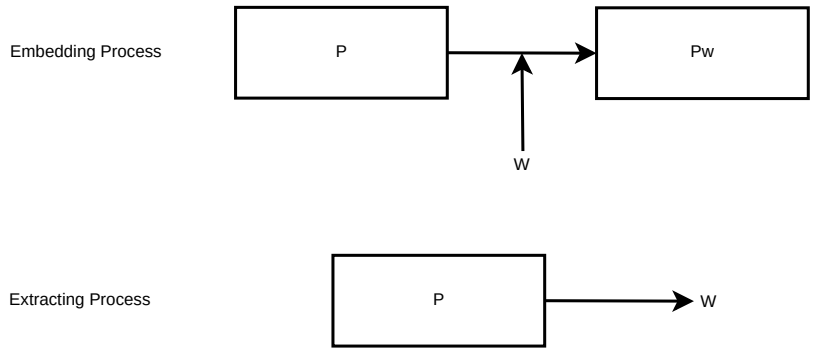


Figure 1.2: The watermarking scheme.

the hidden information has nothing to do with the cover work, as its target is to pass a hidden message through a cover work. While in watermarking, the hidden information describes the cover work itself or it is the cover work's identity [21].

More formerly, watermarking and steganography are defined as follows:

- We define as watermarking the practice of imperceptively altering a work to embed a message about that work.
- We define steganography as the practice of undetectably altering a work to embed a secret message.

In this thesis, we are only interested in watermarking where the cover work is a digital image. This is called digital image watermarking. Digital image watermarking is a branch of digital watermarking which can be briefly described as the process of hiding digital information in a carrier signal.

Similar to traditional watermarking, digital watermarks can only be perceptible under specific conditions such as, after using special extracting algorithms [81]. If a digital watermark distorts the carrier signal in a way that it becomes perceivable, it is of no use. In digital watermarking, the signal may be audio, picture, video, text, 3D models etc. A signal can carry several different watermarks at the same time. Unlike metadata which is added to the carrier signal, a digital watermark does not change the size of the carrier signal meaning that the digital watermark do not add additional payload to the object.

In the image watermarking process, the digital information, i.e., the watermark, is hidden in image data. The watermark is embedded into image's data through the introduction of errors not detectable by human perception [20]; note that, if the image is copied or transferred through the internet then the watermark is also carried with the copy into the image's new location.

Digital and Image Watermarking will be further analyzed at the following chapter, where there will be also references to its properties and definitions as well as information on the research that has taken place so far.

1.4 How it Started

Paper was invented in China over a thousand years ago. Nevertheless, watermarks were not used until 1282 when in Bologna, Italy Cartiere Miliani Mill used them for the first time on a paper. The first watermark was a small cross with tiny circles on its four edges.

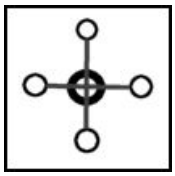


Figure 1.3: The First Watermark.

Initially watermarks were called papermarks, from the Dutch word papiermerken, while the French used the word filigrane which refers to the shaped or bent wire. The English began to use the name, watermark about at the beginning of the Eighteenth Century and that was the word that dominated at the end.

There is not a reference on the reason which boosted people of that era to use watermarks, but most likely watermarks were originally been utilized to place the trademark of the respective paper mill which produced the specific paper. This should have helped to resolve disputes in the event of paper makers accusing one another for theft. Of course there are are also chances that the first watermarks were only used for decoration.

The first reported watermarking method in history was the following: Using a piece of wire, shapes or logos were designed on the paper's surface during its production which in fact was the paper's watermark. The paper would be slightly thinner where the wire was pushed, and hence more transparent.

In 1286 John Marshall invented the dandy roll watermarking technique which quickly became very popular and revolutionized paper watermarking. The idea was that the paper was pressed by a cylindrical metal surface the so called dandy roller. The dandy roller was a cylinder cover made by a material similar to window screen that was embossed with a pattern. Specifically, the watermark's faint lines were designed by laid wires that run parallel to the axis of the dandy roll and the bold lines by chain wires that run around the circumference to secure the laid wires to the roll from the outside. Because the chain wires are located on the outside of the laid wires, they have a greater influence on the impression in the pulp, hence their bolder appearance than the laid wire lines [59] (See, Figure 1.4).

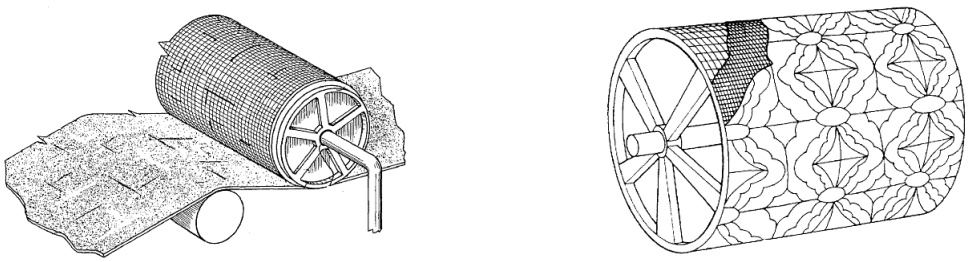
After 1400 watermarks were very commonly used and from now on it was rare to come across official documents which did not carry watermarks [85].

At about 1700 watermarks were also widespread in other places of the world except Europe, such as America and Japan. They were mostly used as trademarks to stamp the date of manufacturing and the size of the original sheets. It was also the era where watermarks began to be used for another purpose which was a measure for anticounterfeiting concerning bank notes and various other documents.

The first reference for someone using a method of counterfeiting was in 1979. As reported by the Gentleman's magazine, a man called John Mathison was caught and hang for discovering a method of counterfeiting watermarks of bank notes.

Of course, counterfeiting boosted research in watermarking technology. What is more, William Congreve, invented color watermarking. That was done by inserting dyed material into paper during papermaking. This technique was difficult in practice though and that's why even the Bank of England declined to use them.

In 1848 the cylinder mould watermarking technique was invented by William Henry Smith. Thanks to this technique it was now possible to design fading watermarks using tonal depth. In this case, instead of using wires the cylinder has specific areas of congestion on its surface. When the paper dries out it can once again pass from the cylinder to get shady watermarks which look like grayscale images. This time instead of wires the cylinders make use of regions of decongestion on their surface. Using this technique it became possible to design watermarks with more detail and generally clearer that those created from the dandy roll technique. Even today, this technique is still used on bank notes [59] (See, Figure 1.4).



(a) (b)

Figure 1.4: (a) The dandy roll technique. (b) The cylinder mould technique.

Such methods, having as a purpose the counterfeiting prevention were used from then until today on stamps, bank notes and governments' documents as an anti-counterfeiting measure. Specifically from the 18_{th} century and beyond watermarks were used on various documents as they were something usual and used worldwide.

The first example of a watermark that was similar to a digital watermark was used for the first time in 1995 by Emil Hembrooke of Muzak Corporation who managed to hide a watermark in a music disk. This was achieved using a narrow notch filter centered at $1KHz$ to embed an identification code that was actually a morse signal in a frequency that is not perceptible under normal conditions. To extract the message a similar filter was utilized [35].

It was in 1988 when the term "Digital Watermarking" was used for the first time by Komatsu and Tominaga [48]. But, it was not until the 90s though, when digital watermarking started getting popular.

From 1995 and beyond the internet's domination and the "cheap" solutions for storing and transferring data, and the acceding day by day internet speeds in wireless and land

line networks resulted in an efficient and low cost creation, copy, and transfer of digital information. As a result piracy started to flourish.

This also meant that interest in digital watermarking flourished as well. The number of papers on digital watermarking was acceding year by year by an astonishing rate [21].

Figure 1.5 demonstrates that rate, by showing the annual number of papers published on digital watermarking or steganography. You can notice as mentioned that after 1995 there was a stable incensement on the papers published until the year 2004 after which there were about 900 papers annually being published [10].

The first Information Hiding Workshop including digital watermarking as one of its primary topics took place in 1996. Some years later and specifically in 1999 SPIE held a conference specifically devoted to multimedia watermarking [2, 93, 94].

At about the same era, several organizations began using numerous watermarking technologies for inclusion in different standards. For example, the Copy Protection Technical Working Group tested watermarking systems to protect video on DVD disks [21]. The Secure Digital Music Initiative used watermarking as a central component of music protecting system. Two projects sponsored by the European Union, VIVA and Talisman tested watermarking for broadcast monitoring [23, 34] and the International Organization of Standardization (ISO) took an interest in the technology in the contest of designing advanced MPEG standards.

Last, in the late 1990s several companies begun marketing watermarking products. In the area of image watermarking, the Digimarc company gave its watermark embedders and detectors with Adobe Photoshop.

This list goes on until today. So, research towards digital watermarking is uncountably indispensable since its results are used by many applications. I hope that what follows can be considered as one more contribution to digital watermarking.

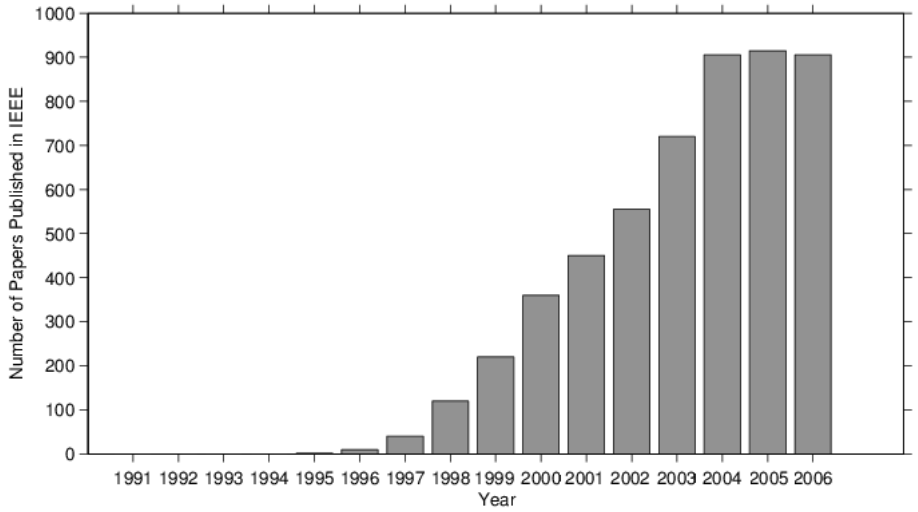


Figure 1.5: Annual number of papers published on watermarking and steganography.

1.5 Roadmap

This thesis is strictured as follows.

In Chapter 2 the thesis includes the theory behind image watermarking and it makes reference to the respective definitions and properties. After that, it analyzes how the watermarking techniques are categorized and for each category it describes its most important techniques developed so far.

In Chapter 3 the thesis describes the main idea behind the proposed image watermarking algorithms both in the spatial and the frequency domain by exploiting self inverting permutations using an efficient transformation of a watermark from an integer form to a two dimensional (2D) representation. It also includes the contribution of this work.

In Chapter 4 the thesis shows properties of the proposed image watermarking techniques and evaluates their performance.

Last Chapter 5 concludes the paper and discusses possible future extensions.

Chapter 2

Image Watermarking

2.1 Background

In general, watermarks are symbols placed into physical objects such as documents, photos, etc. and their purpose is to carry information about objects' authenticity [21].

A digital watermark is a kind of marker embedded in a digital object such as image, audio, video, or software and, like a typical watermark, it is used to identify ownership of the copyright of such an object. Digital watermarking (or, hereafter, watermarking) is a technique for protecting the intellectual property of a digital object; the idea is simple: a unique marker, which is called watermark, is embedded into a digital object which may be used to verify its authenticity or the identity of its owners [33, 17].

More precisely, watermarking can be described as the problem of embedding a watermark w into an object P and, thus, producing a new object P_w, such that w can be reliably located and extracted from P_w even after P_w has been subjected to transformations. In our case, which is digital image watermarking, the digital object is a digital image. In a similar manner the digital image watermarking problem, is the problem of embedding a watermark w into a digital image I so that w can be located and extracted from from I_w even if it has been subject to image transformations [17]; for example, compression, scaling or rotation.

In the image watermarking process the digital information, i.e., the watermark, is hidden in image data. The watermark is embedded into image's data through the introduction of errors not detectable by human perception [20]; note that, if the image is copied or transferred through the internet then the watermark is also carried with the copy into the image's new location.

2.1.1 Preliminaries

We have seen the description of the watermarking problem but the general definition of watermarking is the following [33, 17]:

Definition: We define watermarking as the practice of altering an object in order to embed a message about it.

Of course this message from the definition, is in fact information about the holder of the object's intellectual property, so that to enable identifying ownership.

Formally, a general watermarking system or more specifically a digital Image watermarking system is consisted of the following modules/parts:

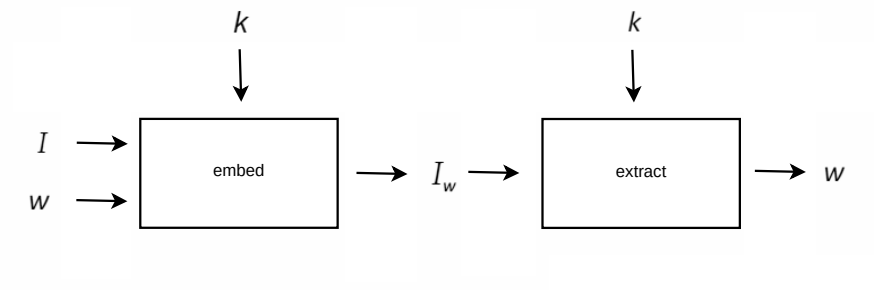


Figure 2.1: The structure of a watermarking system.

- **Encoder:** The encoder is responsible for the embedding process of a watermark. Input at the embedding process is an image I, a watermark w and optionally a key k. As for the output of the embedding process it is the watermarked image I_w.
- **Decoder:** The decoder on the other hand is responsible for the extracting process of a watermark. Input at the extracting process is a watermarked image I_w, and optionally a key k. As for the output of the extracting process it is now the value of the watermark w.

In computer science, watermarking is related with information hiding. Information hiding is a term encompassing a wide range of problems beyond that of embedding messages in content. The term hiding can refer to either making the information imperceptible (as in watermarking) or keeping the existence of the information secret [21].

In digital watermarking, the watermark is information which is embedded in an object which is in this case in the form of a digital signal. Thanks to digital watermarking, digital identities can be extracted from the carrier signal enabling owner identification. This means, that in cases of dispute, the owner of the digital object can use this ownership identification technique to claim his property. Among these, there are also multiple other applications of digital watermarks and they are going to be discussed in subsection 2.1.2. Like traditional watermarks, digital watermarks are only perceptible under certain conditions, i.e. after using special algorithms, elsewhere they remain imperceptible. If a

digital watermark distorts the carrier signal in a way that it becomes perceivable, it is of no use [81]. Since a digital copy of data is the same as the original, digital watermarking is a passive protection tool. It just marks data, but does not degrade it nor controls access to the data.

A watermarking system is usually divided into three distinct steps, embedding, attack, and extracting. In embedding, an algorithm accepts the host signal and the watermark, and produces a watermarked signal. Then, the transmitted/stored watermarked signal might undergo modifications, this is called an attack. While the modification may not be malicious, the term attack arises from copyright protection application, where third parties may attempt to remove the digital watermark through modification. There are many possible modifications, for example, lossy compression of the data, cropping an image or video, or intentionally adding noise. Extracting is an algorithm which is applied to the attacked signal to attempt to extract the watermark from it. If the signal was unmodified, then the watermark still is present and it may be extracted. In robust digital watermarking applications, the extracting algorithm should be able to produce the watermark correctly, even if the modifications were severe. In fragile digital watermarking, the extraction algorithm should fail if any change is made to the signal.

Digital Watermarking can be either image watermarking, video watermarking, audio watermarking, text watermarking, graphic watermarking etc. based on where watermarks are added. For example, image watermarking is adding a watermark in an image, video watermarking is adding a watermark in a video stream, text watermarking is adding a watermark in a pdf file and graphic watermarking is adding watermarks in 2-D or 3-D computer generated graphics [44]. In our case, as mentioned before the digital object where watermarks are embedded are the digital images.

What is more, it should also be mentioned that a watermark can be embedded either in the spatial or the frequency representation of a digital signal. In a few words each point in a frequency representation represents a particular frequency contained in the Spatial Representation [3, 31] . In the technique that is about to be presented here there was an initial approach where marks were hidden in the spatial domain and then this was enchased by embedding them in the frequency representation of the image or more specifically the discrete fourier representation. Of course more details are about to follow in the next chapters.

When referring to watermarks, we should always mention certain characteristics which are the key features defining each watermarking technique. Those characteristics describing a technique have mostly to do with the imperceptiveness and the resistance of the watermarks embedded using it. Furthermore, each watermarking technique also comes with an evaluation. That is measuring its behavior presenting certain characteristics. So, we have also certain evaluating techniques. Last we should also mention that digital image watermarking, mainly falls into two categories in accordance to where marks are hidden in an image. All those terms will be further analyzed in the following subsections.

2.1.2 Characteristics of Watermarking Systems

- **Effectiveness:** We consider a watermark as effective when the extracting algorithm is able to successfully extract it. So, we define as embedding effectiveness the probability that the extracting algorithm successfully extracts the embedded watermark without loosing any information Although 100% effectiveness is always desirable at a watermarking technique, it is not always achieved as it comes at a very high cost with respect to the other properties of the technique. For example, a 100% effective technique might result to a quite perceptible watermark [21].

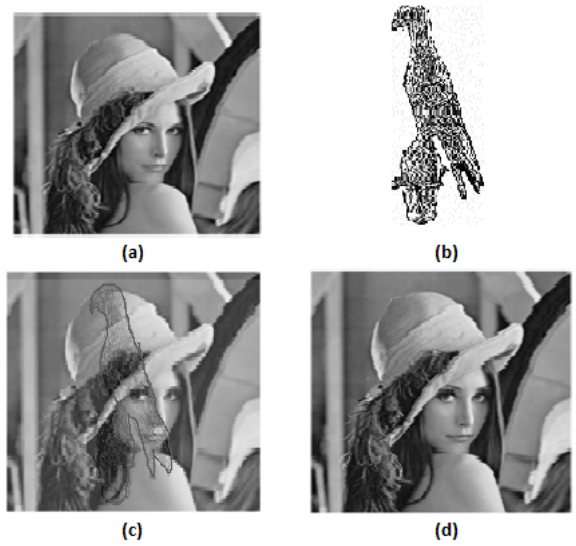


Figure 2.2: (a) A non-watermarked image. (b) A logo. (c) Visible watermarked image. (d) Invisible watermarked image.

- **Invisibility:** Invisible watermarks, are considered those that are hidden under normal use and can only appear when extracted from an authorized user using a special software [60].

  - An invisible watermark should not be noticeable nor should it degrade the image.
  - In order to be robust it should be resistant to signal distortions and tamperings.
  - Retrieval of the watermark should unambiguously identify the owner.
  - Its decoder should better be scalable with each computer generation.
  - When watermarking images the amount of pixel modifications should be minimum.

  Most researchers that refer to digital watermarks, consider the invisibility as granted. Others of course do not do that as we have already seen before.

While invisible watermarks are those that can not be visible after being embedded in an object such as logos putted on images in the web or a more striking example from the physical world, watermarks that we find on bank notes (See, Figure 2.2), we also find invisible watermarks which are those that are visible under normal use. Such a watermark could be a company's logo placed on the bottom of its mails. Or it can be a photographer's semitransparent logo, putted over his images.

So concerning visible watermarks, we have the following requirements:

- A visible watermark should be obvious in both color and grayscale images.
- It should spread in a large or important area of the image to prevent its deletion.
- It should be visible without obscuring image details.
- Removing it should be more costly than buying the image itself.

In our case, the technique is embedding invisible watermarks.

- **Robust Watermarks:** The target in most cases is to design a robust image watermarking technique. Piracy attacks or image processing should not affect the embedded watermark, visible or not. Robustness is measured according to how the watermarked image withstands attacks and transformations such as, geometric distortions such as scaling, rotation and translation, spatial filtering, additive noise, etc. A robust watermark also results to the destruction of the hosting image when a malicious user attempts removal or modifications. In a few words a robust watermark means that the cost of removing a watermark is greater that the cost of the image itself. Either due to the fact that the image is destroyed due to the removal or due to the fact that the time needed to remove it is to long. Here, we shall also point out the fact that not all watermarking applications require robustness towards all signal processing operations, but only those operations that are likely to occur between the time of embedding and the time of detection. For example, in television and radio broadcast monitoring, the watermark needs only to survive the transmission process [45] (See, Figure 2.4).

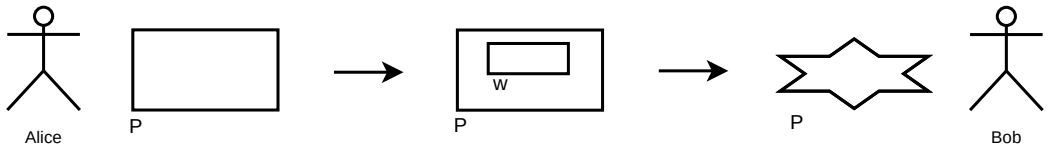


Figure 2.3: Alice embeds a watermark w in object P. The malicious user Bob is unable to remove w unless the object's destruction.

- **Fragile Watermarks:** Fragile watermarks, are also known as tamper-proof watermarks. This kind of watermarks unlike robust watermarks is destroyed by data manipulation. They are designed in such a way in order to be destroyed by any form of copying or encoding other that bit-by-bit digital copy. Their absence indicates that a copy of the digital object has been made [45] as slight errors occurred during copy destroyed the watermark. For example a fragile watermark designed for authentication purposes declares by its absence that the object is not original anymore as it has been through processing applications.

- **Fidelity:** The fidelity of a watermarking system refers to the similarity between the original and the watermarked image or digital object in general. Concerning images, their similarity can be measured using the so called quality metrics. For example, such metrics could be the PSNR or the SSIM and these are what we used in our case. Generally we consider as watermarked images with high fidelity those that have PSNR values over 40 and SSIM values over 0.9. But, we can also measure fidelity with how similar the two images look by just presenting them to a human. More details will be discussed at Chapter 4.

Figure 2.4: (a) Watermarked image of high fidelity. (b) Watermarked image of very low fidelity.

- **Data payload:** Data payload refers to the number of bits a watermark encodes within a unit of time or space when the object is a digital image. Specifically, concerning an image, the data payload would refer to the number of bits encoded within the image within a unit of space where space is measured by pixels. While for audio for example, data payload refers to the number of embedded bits per second that are transmitted. In the watermarking research literature many systems have been proposed in which there is only one possible watermark and the extracting process determines whether or not that watermark is present. Those are referred as one-bit watermarks because there are 2^1 possible outputs: Watermark is present and Watermark is not present [21].

- **Blind or Informed Detection:** There are applications where the original, undetermined work is also required as an input at the extracting process. In that case we are dealing with informed detection. Whether at the applications where detection is performed without the need of the non-watermarked version of the cover work, we are dealing with blind detection techniques. In our case we are going to present a technique based on the blind approach. For more information, in the watermarking literature, systems that use informed detection are called private watermarking systems, whereas those that do not are called public watermarking systems [21].

- **False Positive Rate:** A false positive is the detection of a watermark in a digital object that does not actually contain one. Concerning false positives we expect them to occur in a given number of runs of the extracting procedure. The false positive propagability is the propagability that given a fixed work and randomly selected watermarks, the detector will report that a watermark is in that Work. Alternatively the false positive probability is also defined as the probability that given a fixed watermark and randomly selected works the detector will detect that watermark in that work [21].

- **Security:** A watermark should be secret and thus be undetectable by an unauthorized users and only detectable by authorized ones. This requirement is regarded as a the security factor. Generally speaking, the security of a watermark refers to its ability to resist hostile attacks. Those attacks fall in three main categories [21].

  - **Unauthorized removal**
  - **Unauthorized embedding**
  - **Unauthorized detection**

  Unauthorized removal refers to the attacks that prevent a digital object's watermark from being detected. There are two types of attacks that can achieve that. First it is elimination and second it is masking attacks. Concerning the first type, eliminating a watermark means putting an attacked watermarked object into an extractor has similar results to putting an object that has not been watermarked. Meaning that it is not possible to detect it even with the most sophisticated decoding techniques. While masking a watermark means that the attacked work can still contain the watermark, but marks are undetectable by existing detectors. In this case developing more sophisticated detectors might enable us to detect and successfully extract the watermark. If we consider images as an example, an attacker might mask the watermark by just rotating the image. But developing an extracting algorithm that is able to detect watermark even though the image has been subject to rotations can solve that problem.

  Unauthorized embedding refers to the action of embedding illegitimate watermarks into digital objects that should not contain them. The target of the malicious user

here is not to alter a watermark of an authenticated work in order to disable its identification from the decoding techniques but to perform unauthorized embedding so that to cause the detector to identify an invalid work.

Last, unauthorized detection refers to the case when a malicious user manages to decode the watermark without having access to the extracting algorithm. In less severe cases of this situation the malory may have extracted the basic structure of the watermark but did not manage to acquire its numerical value and thus not being able to decipher what the marks mean.

- **Watermark keys:** It should also be mentioned that there is a specific category of watermarks that make use of keys. That means that input at the encoder are not just the watermark and the digital object but also a key which is also needed at the decoder as an input in order to successfully extract the watermark. Our algorithm does not make use of watermark keys.

- **Intended Applications:** In digital watermarking there are certain applications where a watermark can be of use. Some basic ones, are the following [21]:

  - **Broadcast Monitoring** Using a passive systems the broadcasting time of a T.V. or radio program can be measured by matching known watermarks from a database. There are several types of organizations and individuals interested in broadcast monitoring. For example advertisers want to ensure that they receive all of the air time they have purchased.

  - **Owner Identification** This is our case, where the purpose of the embedded watermark is to identify the ownership of a user. If users have access to watermark embedders and detectors, they are able to prove their ownership in cases of disputes.

  - **Transaction Tracking** In this application of watermarking, the watermark records a transaction that has taken place in the history of the copy of a digital object in which it is embedded.

  - **Content Authentication** Using fragile watermarks it is easy to detect which one of two objects has been altered and it is not the authentic one as it does not contain the watermark any more.

  - **Copy Control** We can also tell that our technique belongs to this category as well. Warning users of the presence of a watermark in a digital object can deter them from copying it without permission from the copyright holder.

2.1.3 Evaluating a Watermarking Technique

Before we continue with how to evaluate a watermarking system, let's first point out what makes a watermarking system better than another, or what level of performance would be best. The truth is, that there is not a watermarking technique which we may consider as the "best". But if we need a watermarking system for a specific application, then our evaluation criteria should fully depend on that specific application.

Note also, that criteria for a watermarking technique many times contradict one another such as for example imperceptiveness and robustness. In cases like those what we need is to find a "good" tradeoff between them.

For example, if we need a watermarking system for video copy control we might need to test it, in rotations or scaling. But if we just need it for normal broadcasting then no such tests are needed as video is only broadcasted as it is in terms of angle and size [21]. On the other hand, in such a case attention should be made on noise that the signal might acquire due to the broadcasting.

In our case, where images are intended for being uploaded to the internet, image compression was one of the most important criteria in designing the watermarking technique. So, we tested our technique in order to withstand various compression ratios using the JPEG protocol. We have also presented some error correction techniques in case where the image is rotated or some part of the image is missing, both of which are very possible scenarios in digital image usage. More are about to follow in Chapter 4. What is more, for our case we needed a watermarking technique that pursuits an ideal "tradeoff" between image fidelity and robustness.

Of course, when we are dealing with invisible image watermarking we always need as we have already mentioned before, a watermarking technique that between other criteria distorts the original image in the least possible degree. That means that the watermarking technique in question embeds the least possible information in a digital image that is then able to enable successful extracting under the required standards. Such standards might be the compression ratio that we would need the watermark to withstand. Of course, in that case, the greater the compression ratio the more the additive information needed.

To evaluate how small this distortion is, we need to compare somehow the watermarked image with the original one. This image quality assessment is performed with the quality metrics [89]. There are various quality metrics that compare two images and return a value depicting their similarity. In our case, concerning our technique we used the PSNR and the SSIM metrics [37]. The results that we gathered are about to be presented in this thesis in the evaluation section. So, concerning evaluation and image quality assessment about our watermarking system more details are about to follow in Chapter 4.

2.2 Image Watermarking Algorithms

Image Watermarking algorithms are found in categories depending on the point view from which we see them. As we have already seen, there are robust and fragile watermarking techniques according to how the watermark is removed, visible and invisible according to if they are visible by human perception, blind and informed and the list goes on.

But looking technically into digital image watermarking, we should mention here that image watermarking techniques fall in two main categories according on how do we exploit images so that to embed watermarks in them.

So, image watermarking algorithms fall in two main categories according to whether the watermark is embedded intom the spatial or frequency domain of the host digital image. More details about this classification are about to follow in subsection 2.2.1.

2.2.1 Marking in the Spatial and Frequency Domain

- **Spatial Domain:**

  The term spatial domain refers to the image plain itself and watermarking in the spatial domain is based in direct manipulation of pixels in an image. [31] Images in the spatial domain are 2D functions $f(x, y)$ in spatial coordinates (x, y) in an image plane.

  Each function describes how the brightness varies in space. Note that if we are using color images then we have three similar functions, one for each color channel (i.e. Red, Green, Blue).

  So, what is in fact a digital image in the spatial domain? It is the result of sampling a natural continuous image in a 2D array $f(x, y)$ containing N rows and M columns where (x, y) are the discrete coordinates.

  For notational clarity and efficiency we use integer values for these discrete coordinates:
  $x = 1, 2, \ldots, N$ and $y = 1, 2, \ldots, M$.

  In general the value at each point (x, y) of the image is denoted by $f(x, y)$ where $0 \leq f(x, y) \leq 255$. Where 0 is the least possible brightness (black color) and 255 is the most possible brightness (white color).

  Of course that is in case we are using 8-bit digital images. In case where the image is also a color image, the only difference is that we have 8-bit information for each one of the three color channels (i.e. Red, Green, Blue).

  In image watermarking we can find certain subcategories of spatial domain techniques, some basic ones are the following:

– **Spread Spectrum Techniques:**

Spread spectrum techniques are methods in which energy generated at one or more discrete frequencies is deliberately spread or distributed in time/space. This is used for a variety of reasons, including the establishment of secure communications, increasing resistance to natural interference and jamming, and to prevent detection.

When applied to the context of image watermarking, Spread Spectrum based watermarking algorithms embed information by linearly combining the host image with a small pseudo noise signal that is modulated by the embedded watermark [45].

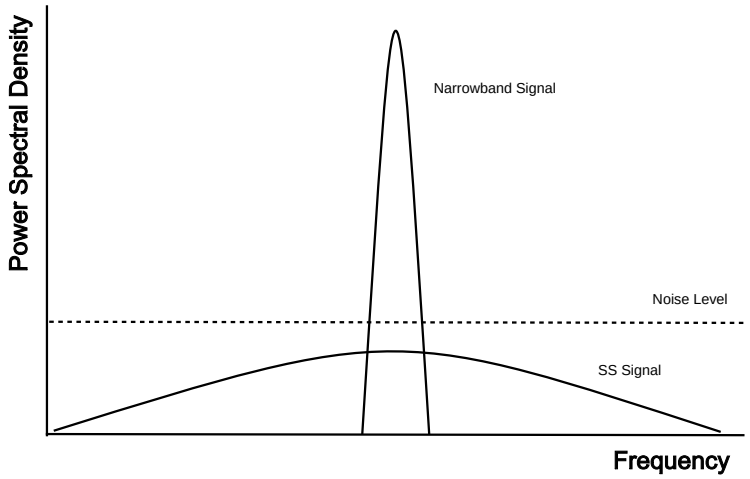


Figure 2.5: The narrow and the spread spectrum.

– **Least Significant Bit Techniques:**

Those are the earliest works of digital image watermarking. They embed watermarks in the Least Significant Bit (LSB) of image pixels. Given an image with pixels, and each pixel being represented by an $8-bit$ sequence, the watermarks are embedded in the last and least significant bit of certain pixels of the image. Note, that the least significant bit determines whether the specific numerical value is odd or even. This method is easy to implement and does not generate serious distortion to the image; however, it delivers poor robustness to attacks and other distortions (image quantization, geometric distortions etc.) For instance, very simply an attacker could simply randomize all LSBs, which of course eliminates all the embedded information [44, 45].

1	0	0	1	0	1	0	1

Figure 2.6: The binary representation of decimal 149, with LSB highlighted. MSB is an 8-bit binary number representing 128 decimal. MSB represents the value 1.

- **Frequency Domain:**

  Compared to spatial domain methods, frequency domain methods are more widely applied. In this case the idea is that we embed the watermarks in the spectral coefficients of the image. The most commonly used transforms to represent an image in the frequency domain are the Discrete Fourier Transform (DFT), the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT).

  The reason for watermarking in the frequency domain is that the characteristics of the human visual system (HVS) are better captured by the spectral coefficients. For example, the HVS is more sensitive to low frequency coefficients, and less sensitive to high frequency coefficients. In other words, low frequency coefficients are perceptually significant, which means alterations to those components might cause distortion to the original image. On the other hand, high frequency coefficients are considered insignificant; thus, processing techniques, such as compression, tend to remove high frequency coefficients aggressively. To obtain a balance between imperceptibility and robustness, most algorithms embed watermarks in the midrange frequencies.

  Following, there is more information about the most important transforms in the frequency domain:

  - **Discrete Fourier Transform (DFT):**

    The Fourier transform, named after Joseph Fourier, is a mathematical transform with many applications in physics and engineering. Very commonly it transforms a mathematical function of time, $f(t)$ or space $f(x, y)$, into a new function, sometimes denoted by F, whose argument is frequency with units of cycles (hertz) or radians per second. The new function is then known as the Fourier transform and/or the frequency spectrum of the function f. The Fourier transform is also a reversible operation. Thus, given the function F, one can determine the original function, f [3, 31].

    The Discrete Fourier Transform (DFT) is an important tool used in image processing and what it does is to decompose an image into its sine and cosine. components. The output of the transformation represents the image in the frequency domain, while the input image is the spatial domain equivalent. In the image's fourier representation, each point represents a particular frequency contained in the image's spatial domain.

    In image watermarking, it is the method that is more frequently used as it provides watermarking techniques that are also robust against various attacks. Although semi-blind and blind watermarking schemes based on Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) are robust to a number of attacks, they fail in the presence of geometric attacks such as rotation, scaling, and translation. The Discrete Fourier Transform (DFT) of a real image is conjugate symmetric, resulting in a symmetric DFT spectrum. Because

of this property, the popularity of DFT-based watermarking has increased in the last few years [27].

More formally, if $f(x, y)$ is an image of size $N \times M$ represented in the spatial domain we use the following formula for the Discrete Fourier Transform:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi(\frac{ux}{N} + \frac{vy}{M})} \tag{2.1}$$

for values of the discrete variables u and v in the ranges $u = 0, 1, \ldots, N-1$ and $v = 0, 1, \ldots, M-1$.

In a similar manner, if we have the transform $F(u, v)$ i.e the image's fourier representation we can use the Inverse Fourier Transform to get back the image $f(x, y)$ using the following formula:

$$f(x, y) = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{j2\pi(\frac{ux}{N} + \frac{vy}{M})} \tag{2.2}$$

for $x = 0, 1, \ldots, N-1$ and $y = 0, 1, \ldots, M-1$.

To get a better picture, following is an illustration of an image's DFT by showing initially its spatial representation and then its Fourier representation's magnitude.

(a) (b)

Figure 2.7: (a) The spatial representation of Lena. (b) The fourier representation of Lena.

Note that in the DFT representation the central point is the DC(0,0) and the further we get from it the higher the amplitude of the frequency.

As mentioned, in our method we use the Discrete Fourier Transform and we are interested in the magnitudes of DFT coefficients. The magnitude $|F(u, v)|$ of the Fourier transform at a point is how much frequency content there is and is calculated by Equation (2.1) [31].

– **Discrete Cosine Transform (DCT):**

The Discrete Cosine Transform was introduced by Ahmed, Natarajan and Rao [1], it is an expression of a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. Among image watermarking, it is important for numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG), to spectral methods for the numerical solution of partial differential equations.

The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient, whereas for differential equations the cosines express a particular choice of boundary conditions. It is also a fact that DCT corresponds more to the way humans perceive light, so that to enable spotting what can not be perceived and identified by human vision and throw it away.

As for Watermarking, DCT based techniques are as mentioned more robust compared to spatial domain techniques. Algorithms making use of the DCT are robust against simple image processing operations like low pass filtering, brightness and contrast adjustment, blurring etc. However, they are difficult to implement and are computationally more expensive. At the same time they are weak against geometric attacks like rotation, scaling, cropping etc. DCT domain watermarking can be classified into Global DCT watermarking and Block based DCT watermarking. Embedding in the perceptually significant portion of the image has its own advantages because most compression schemes remove the perceptually insignificant portion of the image [61].

Getting to a more formal Definition, it should be mentioned that there are variants of modified definitions, but the most commonly used one is the DCT-II. According to which the $(p+q)^{th}$ order DCT coeeficient for an image represented in the spatial domain with the intensity function $f(x, y)$ of size $N \times M$ is described by:

$$\mathrm{C}(p,q) = \frac{1}{\rho(p)\rho(q)} \sum_{x=0}^{N-1} D_p(x) D_q(y) f(x,y), \tag{2.3}$$

where, $0 \leq x, p \leq N-1$, $0 \leq y, q \leq M-1$,

$D_n(t) = cos(\frac{(2t+1)n\pi}{2N})$

and

$$\rho(n) = \begin{cases} N, & \text{if } n = 0 \\ N/2, & \text{otherwise} \end{cases}.$$

– **Discrete Wavelet Transform (DWT):**

The Discrete Wavelet Transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over fourier transforms is temporal resolution: it captures both frequency and location information. The first DWT was invented by the Hungarian mathematician Alfre'd Haar. For an input represented by a list of 2^n numbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale: finally resulting in $2^n - 1$ differences and one final sum [45].

Figure 2.8: The DWT coefficients of Lena.

The Discrete Wavelet Transform (DWT) is currently used in a wide variety of signal processing applications, such as in audio and video compression, removal of noise in audio, and the simulation of wireless antenna distribution. Wavelets have their energy concentrated in time/space and are well suited for the analysis of transient, time-varying signals. Since most of the real life signals encountered are time varying in nature, the Wavelet Transform suits many applications very well [9].

One of the main challenges of the watermarking problem is to achieve a better tradeoff between robustness and perceptivity. Robustness can be achieved by increasing the strength of the embedded watermark, but the visible distortion would be increased as well [49]. However, DWT is much preferred because it provides both a simultaneous spatial localization and a frequency spread of the watermark within the host image [8]. The basic idea of discrete wavelet transform in image processing is to multi-differentiated decompose the image into sub-image of different spatial domain and independent frequencies [21].

2.2.2 Previous Work

In the digital watermarking field, most of the attention was given on images by the research community. And this can be noticed by the fact that most of the research work and the publications if the research community concern image watermarking rather than audio or video for example. There are some very good reasons for that though. Firstly, because of the availability of numerous test images, secondly because it carries enough redundant information to provide an opportunity to embed watermarks easily, and last, it may be assumed that any successful image watermarking algorithm may be upgraded for videos as well.

As we have already seen at the previous subsection, image watermarking falls in two basic approaches. Watermarking in the spatial domain and watermarking in the frequency domain. So, next we shall see some of the most important algorithms from each category [79].

- **Algorithms in the spatial domain:**

  Starting with algorithms based on the LSB approach, Macq and Quisquater [58] in 1995 discussed the issue of image watermarking by making a general survey on cryptography and digital television. Specifically the proposed a technique for inserting a watermark into the least significant bit (LSB) of pixels located in the vicinity of image contours. But due to the fact that the watermark's information is at the LSB makes them really fragile. This method is restricted to images as it is relied on inserting the watermark into image regions that lie on the edge of contours.

  In the same year Rhoads [74] described a method that either adds or subtracts minor random quantities from each pixel of an image. This addition or subtraction is determined by a comparison of a binary mask of bits with the LSB of each pixel. The idea is that if the LSB is equal to the corresponding mask bit, then the referred random quantity is added, otherwise it is subtracted. The watermark is extracted by first computing the difference between the original and the watermarked image and then by examining the sign of the difference, pixel by pixel, to determine if it corresponds to the original sequence of additions and subtractions.

  Furthermore as mentioned at the previous subsection we also have the spread spectrum approach. There, we can employ code division multiple access (CDMA) spread spectrum schemes to scatter each of the bits randomly throughout the cover image. This way, robustness against cropping attacks is also achieved. In this case, the watermark has the form of a string rather than a 2D image representation. For each value of the watermark, a Pseudo Noise (PN) sequence is generated using an independent seed generated through PN methods. The summation of all of these PN sequences represents the watermark, which is then scaled and added to the cover image [41, 51]. To detect the watermark, each seed is used to generate its PN sequence which is then correlated with the entire image. If the correlation is

high, that bit in the watermark is set to "1", otherwise to "0". The process is then repeated for all the values of the watermark. CDMA improves on the robustness of the watermark significantly but it is of greater computational complexity.

Moving to other not sub-categorized methods for watermarking images in the spatial domain, Voyatzis and Pitas [88] in 1998 proposed a method that embeds a binary watermark in an image's spatial domain. A spatial transform maps each pixel of a watermark image to a pixel of the host image. Here the target is to achieve a Chaotic spread of the watermark image pixels in the host image by exploiting "toral automorphisms". For watermark embedding, the intensity of the selected pixels is modified by an appropriate function that takes into account neighborhood information in order to achieve robustness regarding the modifications. As for the extracting process, a suitable function is applied on each of the watermarked pixels to determine the binary digit that has been embedded. The inverse spatial transform is then used to reconstruct the binary watermark image.

In the same year Pitas [68] proposed a method where the image is split into two random subsets A and B and the intensity of pixels in A is increased by a constant embedding factor k. Watermark detection is performed by evaluating the difference of the mean values of the pixels in subsets A and B. This difference is expected to be equal to k for a watermarked image and equal to zero for an image that is not watermarked. Hypothesis testing can be used to decide for the existence of the watermark. The above algorithm is vulnerable to lowpass operations. Extensions to above algorithm were proposed proposed by Nikolaidis and Pitas in [62]. There, it was proposed that the robustness of the method can be increased by grouping pixels so that to form blocks of certain dimensions to enhance the low pass characteristics of the watermark signal. Alternatively, one can take advantage of the fact that different embedding factor can be used for each pixel, to shape appropriately the watermark signal. An optimization procedure that calculates the appropriate embedding value for each pixel so that the energy of the watermark signal is concentrated at low frequencies is proposed. Constraints that ensure that the watermark signal is invisible can be incorporated in the optimization procedure.

In the same year, Kalker et. al. [46] derived analytical expressions for the probabilities $P-$, $P+$ of false negative and false positive watermark detection. Their model assumes an additive watermark and a correlator-based detection stage. Both, the white watermarks and watermarks having low pass characteristics, are considered. The host image is treated as noise, assuming a first order separable autocorrelation function. The probabilities P-, P+ are expressed in terms of the watermark to image power ratio. The authors conclude that detection error rates are higher for watermarks with low pass characteristics.

- **Algorithms in the frequency domain:**

  Starting with algorithms that embed watermarks in the frequency domain, there are few algorithms that modify the DFT magnitude and phase coefficients in order to embed watermarks [79, 69].

  Ruanaidh et al. [76] in 1996 proposed a DFT watermarking scheme in which the watermark is embedded by modifying the phase information within the DFT. It has been shown that phase based watermarking is robust against image contrast operations [92].

  In 1998 Ruanaidh and Pun [78] showed how the Fourier Mellin transform could be used for digital watermarking. The Fourier Mellin transform is similar to applying the Fourier Transform to the log-polar coordinate system for an image This system, was robust against geometrical attacks.

  De Rosa et al. [24] in 1999 proposed a scheme to insert watermark by directly modifying the mid frequency bands of the DFT magnitude component.

  In 1999 Ramkumar et al. [71] also present a data hiding scheme based on DFT, where they modify the magnitude component of the DFT coefficients. Their simulations suggest that DFT magnitude survives practical compression, and this can be attributed to the fact that most practical compression schemes try to maximize the PSNR. Hence using magnitude DFT is a way to exploit the hole in most practical compression schemes. The proposed technique is shown to be resistant to JPEG and SPIHT compression.

  In 2001 Lin et al. presented an algorithm resilient to geometric attacks. In their algorithm, the watermark is embedded in the magnitude coefficients of the Fourier transform re-sampled by log-polar mapping. The scheme is, however, not robust against cropping and shows weak robustness against JPEG compression (Q = 70) [57].

  In 2001 Solachidis and Pitas [83] presented a novel watermarking scheme. They embed a circularly symmetric watermark in the magnitude of the DFT domain. Since the watermark is circular in shape with its center at image center, it is robust against geometric rotation attacks. The watermark is centered around the mid frequency region of the DFT magnitude. Neighborhood pixel variance masking is employed to reduce any visible artifacts. The scheme is computationally not expensive to recover from rotation. Robustness against cropping, scaling, JPEG compression, filtering, noise addition and histogram equalization is demonstrated.

  In 2004, a semi-blind watermarking scheme has been proposed by Ganic and Eskicioglu [28]. They embed circular watermarks with one in the lower frequency while the other is in the higher frequency. Their work is inspired by [72]. They follow the same argument as that endorsed by embedding watermarks in the low frequency component, which is robust against one set of attacks, while embedding in the high

frequency components is robust to another set of attacks.

In 2000 Pereira and Pun [66] propose robust watermarking algorithm resistant to affine transformations. They introduce the concept of Template. A template is a structure which is embedded in the DFT domain to estimate the transformation factor. Once the image undergoes a transformation this template is searched to resynchronize the image, and then use the detector to extract the embedded spread spectrum watermark.

Moving to Discrete Cosine Transform (DCT) domain watermarking, it can be classified into Global DCT watermarking and Block based DCT watermarking. One of the first algorithms presented by Cox et al. [19] in 1997 used global DCT approach to embed a robust watermark in the perceptually significant portion of the Human Visual System (HVS). Embedding in the perceptually significant portion of the image has its own advantages because most compression schemes remove the perceptually insignificant portion of the image. In spatial domain it represents the LSB however in the frequency domain it represents the high frequency components [85].

DCT based algorithms, differ either in the block selection criteria or coefficient selection criteria. Based on the perceptual modeling strategy incorporated by the watermarking algorithms they could be classified as algorithms with no perceptual modeling [63, 25] or Implicit Perceptual Modeling.

Koch, Rindfrey, and Zhao [47] in 1995 proposed a method for watermarking images based on the DCT approach. In that method, they break up an image into 8x8 blocks and compute discrete cosine transform (DCT) of each of these blocks. A pseudorandom subset of the blocks is chosen and then in each such block, a triplet of frequencies is selected from one of 18 predetermined triplets and modified so that their relative strengths encode a "1" or "0" value. The 18 possible triplets are composed by selection of three out of eight predetermined frequencies within the 8x8 DCT block. The choice of the eight frequencies to be altered within the DCT block is based on a belief that the "middle frequencies have moderate variance" i.e. they have similar magnitude. This property is used to allow the relative strength of the frequency triplets to be altered without requiring a modification that would be perceptually noticeable.

Several other DCT based schemes are presented in [69]. Using the DCT, an image can easily be split up in pseudo frequency bands so that the watermark can conveniently be embedded in the most important middle band frequencies. Furthermore, the sensitivity of the HVS to the DCT based images has been extensively studied, which resulted in the recommended JPEG quantization Table [31]. These results can be used for predicting and minimizing the visual impact of the distortion caused by the watermark. Finally, the block-based DCT is widely used for image and video compression. By embedding a watermark in the same domain as the compression

scheme used to process the image (in this case in the DCT domain), we can anticipate lossy compression because we are able to anticipate which DCT coefficients will be discarded by the compression scheme.

In 1998 Barni et. al. [91] embedded a watermark signal domain by modifying a number of predefined DCT coefficients. They used a weighting factor to weight the watermark signal in the spatial domain according to HVS characteristics.

In the same year Watson proposed a method of embedding watermark data in DCT Difference (JND) as predicted domain in perceptually meaningful way and used the Just Noticeable by model reported in [91].

Further improvements for DCT-domain correlation-based watermarking systems' performance were achieved by using watermark detectors based on generalized Gaussian model instead of the widely used pure Gaussian assumption as suggested by Hernandez et. al [36] in 2000. By performing a theoretical analysis for DCT-domain watermarking methods for images, Hernandez et. al. provided analytical expressions which could be used to measure beforehand the performance expected for a certain image and to analyze the influence of the image characteristics and system parameters (e.g. watermark length) on the final performance. Furthermore, the result of this analysis may help in determining the proper detection threshold T to obtain a certain false positive rate. The authors in [36] claimed that by abandoning the pure Gaussian noise assumption, some substantial performance improvements could be obtained.

In their work, Zhu et. al. [99] presented in 2006 a new image adaptive watermarking scheme based on perceptually shaping watermark block wise. Instead of the global gain factor, a localized one is used for each block. Watson's DCT-based visual model is adopted to measure the distortion of each block introduced by watermark, rather than the whole image. With the given distortion constraint, the maximum output value of linear correlation detector is derived in one block, which demonstrated the reachable maximum robustness in a sense. Meanwhile, an Extended Perceptually Shaped Watermarking (EXPSW)is acquired through making detection value which approaches to upper limit. It is proved mathematically that EX-PSW outputs higher detection value than Perceptually Shaped Watermarking (PSW) with the same distortion constraint. Authors used this idea and also discussed the adjustment strategies of parameters in EX-PSW, which were helpful for improving the local image quality. Experimental results show that scheme provides very good results both in terms of image transparency and robustness.

In 2004 Li and Xue [53] proposed a new DCT domain watermarking expressly devised for RGB color images facing each color channel separately based on the diversity technique in communication systems. The watermark is hidden within the data in the same sequence by modifying a subset of the block DCT coefficients of each color channel.

In 2005 Wang et. al. [90] presented an image watermarking scheme based on 3-D DCT. A gray-level image is decomposed into a 3-D sub-image sequence by sub sample of zigzag scanning order that is transformed using block-based 3-D DCT. Simultaneously, they proved that the distribution of 3-D DCT AC coefficients follows the generalized Gaussian density function using the distribution relative entropy theory. To satisfy the balance between the robustness and the imperceptivity, a 3-D HVS model is improved to adjust the embedding strength. In watermark detecting, the optimum detector is used to implement the blind detection. It is shown in experiments that the scheme is strongly robust against various attacks.

In 2007 Huang et. al. [38] presented an improved invariant wavelet and designed a DCT based blind watermarking algorithm against Rotation, Scaling and Translation (RST) attacks by exploiting the affined invariance of the invariant wavelet. Surviving geometric attacks in image watermarking is considered to be of great importance. In the face of geometrical attacks, all shortcomings of almost all digital watermarking algorithms have been exposed. Therefore, this paper presents an improved invariant wavelet that is better than the bilinear interpolation and whose performance is close to of bi-cubic when scaling factor is very close to 1, and designs a novel blind image watermarking algorithm based on DCT in the (RST) Xiong's Invariant Wavelet, i.e. RSTXIW domain. The experiments show that this novel watermarking algorithm is robust against filter, noise and arbitrary RST geometrical attacks, however, sensitive to local crop attacks.

Moving to the Discrete Wavelet Transform (DWT) watermarking techniques, here you can exploit the characteristics of the Human Visual System (HVS), it is possible to hide watermarks with more energy in an image, which makes watermarks more robust. From this point of view, the DWT is a very attractive transform, because it can be used as a computationally efficient version of the frequency models for the HVS [4]. For instance, it appears that the human eye is less sensitive to noise in high resolution DWT bands and in the DWT bands having an orientation of 45 degrees (i.e., HH bands). Furthermore, DWT image and video coding, such as embedded zero-tree wavelet (EZW) coding, are included in the upcoming image and video compression standards, such as JPEG2000 [79]. Thus DWT decomposition can be exploited to make a real-time watermark application.

Many approaches apply the basic schemes described at the beginning of this section to the high resolution DWT bands, LH, HH, and HL [36, 39]. A large number of algorithms operating in the wavelet domain have been proposed till date.

Concerning blind DWT detection let us begin with by the paper of Lu et al. [56] in 1999 which presents a novel watermarking technique called as "Cocktail Watermarking". This technique embeds dual watermarks which compliment each other. This scheme is resistant to several attacks, and no matter what type of attack is applied, one of the watermarks can be detected. Furthermore, they enhance this technique

for image authentication and protection by using the wavelet based just noticeable distortion (JND) values. Hence this technique achieves copyright protection as well as content authentication simultaneously.

In the same year Zhu et al. [100] present a multi-resolution watermarking technique for watermarking video and images. The watermark is embedded in all the high pass bands in a nested manner at multiple resolutions. This technique does not consider the HVS aspect; however, Kaewkamnerd and Rao [42, 43] improve this technique by adding the HVS factor in account.

Voyatzis and Pitas [88] presented the "toral automorphism" concept, providing a technique to embed binary logo as a watermark, which can be detected using visual models as well as by statistical means. So in case the image is degraded too much and the logo is not visible, it can be detected statistically using correlation. Watermark embedding is based on a chaotic (mixing) system. Original image is not required for watermark detection. However, the watermark is embedded in spatial domain by modifying the pixel or luminance values. A similar approach is presented for the wavelet domain [96], where the authors propose a watermarking algorithm based on chaotic encryption.

In 2004 Zhao et al. [98] presented a dual domain watermarking technique for image authentication and image compression. They use the DCT domain for watermark generation and DWT domain for watermark insertion. A soft authentication watermark is used for tamper detection and authentication while a chrominance watermark is added to enhance compression. hey use the orthogonality of DCT-DWT domain for watermarking.

As for non blind DWT based algorithms, as mentioned they require the original image for detecting the watermark. Most of the techniques found in literature use a smaller image as a watermark and hence cannot use correlation based detectors for detecting the watermark; as a result they rely on the original image for informed detection. The size of the watermark image (normally a logo) normally is smaller compared to the host image.

In 1997 Xia et al. [95] presented a wavelet based non-blind watermarking technique for still images where watermarks are added to all bands except the approximation band.

In 2001 Lu et al. [55] presented another robust watermarking technique based on image fusion. They embed a grayscale and binary watermark which is modulated using the "toral automorphism" described in [88]. Watermark is embedded additively. The novelty of this technique lies in the use of secret image instead of host image for watermark extraction and use of image dependent and image independent permutations to de-correlate the watermark logos.

In 2003 Raval and Rege [72] presented a multiple watermarking technique. The authors argue that if the watermark is embedded in the low frequency components

it is robust against low pass filtering, lossy compression and geometric distortions. On the other hand, if the watermark is embedded in high frequency components, it is robust against contrast and brightness adjustment, gamma correction, histogram equalization and cropping and vice-versa. Thus to achieve overall robustness against a large number of attacks the authors propose to embed multiple watermarks in low frequency and high frequency bands of DWT.

In 1997 Kundur and Hatzinakos [50] presented an image fusion watermarking technique. They use salient features of the image to embed the watermark. They use a saliency measure to identify the watermark strength and later embed the watermark additively. Normalized correlation is used to evaluate the robustness of the extracted watermark. Later the authors propose another technique termed as FuseMark, which includes minimum variance fusion for watermark extraction. Here they propose to use a watermark image whose size is a factor of the host by $2xy$.

In 2004 Tao and Eskicioglu [86] presented an optimal wavelet based watermarking technique. They embed binary logo watermark in all the four bands. But they embed the watermarks with variable scaling factor in different bands. The scaling factor is high for the LL sub band but for the other three bands its lower. The quality of the extracted watermark is determined by Similarity Ratio measurement for objective calculation.

Chapter 3

The Method

3.1 Method's Components

First described in this section will be the discrete structures that we used, namely, permutations and the sub-class of them, the self-inverting permutations.

Then, this section discusses a codec system which encodes an integer number w (watermark in a numerical form) into a self-inverting permutation π.

And last, it presents the method of transforming a watermark from a numerical form to a 2D/2DM form (i.e., 2D representation) through the exploitation of self-inverting permutation properties and before closing there will be also some information about color images.

3.1.1 Permutations

Think of a permutation π of length n as a permutation over the elements contained at a set $N = \{1, 2, ..., n\}$. That permutation π is represented as a sequence as follows: $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$. Having seen how we represent a permutation, note also that represented with π_i is the element of the permutation on the position with index i and with pi_i^{-1} the index of the element i.

Let us take for example a permutation of six elements $\pi = 4, 3, 6, 1, 5, 2$. Concerning this permutation $\pi_1 = 4$, $\pi_2 = 3$, $\pi_3 = 6$ etc. Vise versa, pi_i^{-1} is the position holding element i so, returning to our example $\pi_4^{-1} = 1$, $\pi_3^{-1} = 2$ $\pi_6^{-1} = 3$ etc.

Each permutation equals to a permutation graph. But, what is a permutation graph? Permutation Graphs are a subclass of perfect graphs with very interesting applications. Thanks to their properties, they are capable of solving numerous problems in polynomial time and they are widely used in mathematics and graph theory. Last, permutation graphs are distinguished for their numerous properties and their various ways of being represented and of course that's what this thesis's techniques exploit. A form of representing them, and certain properties that will allow to reconstructs partially destroyed watermarks [30].

First, studying more superficially the way to build a permutation's graph we can regard permutation graphs as a class of intersection graphs in the following manner. Write the numbers $1, 2, \ldots, n$ horizontally from left to right; underneath them write the numbers $\pi_1, \pi_2, \ldots, \pi_n$ in sequence, again horizontally left to right; finally, draw n straight line segments joining the two 1's, the two 2's, etc. We call this the matching diagram of π (see Figure 3.1). Notice that the i^{th} segment intersects the j^{th} segment if and only if i and j appear in reversed order in n; this is the same criterion for the vertices i and j of the graph $G[\pi]$ to be adjacent. Therefore, the intersection graphs of the segments of matching diagrams are exactly the permutation graphs [30].

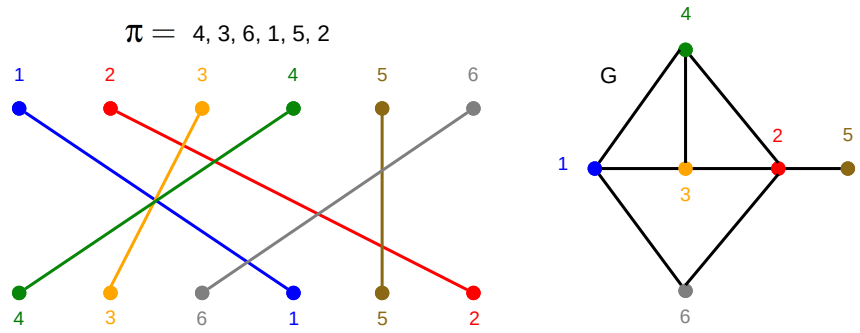


Figure 3.1: A permutation π's graph G through an intersection.

In order to construct an undirected graph $G[\pi]$ directly from a permutation π we do as follows: The graph $G[\pi]$ has n vertices numbered from 1 to n. Two vertices are joined by an edge if the larger of their corresponding numbers is to the left of the smaller in π (i.e. they occur out of their proper order reading left to right). In the example in figure 3.1, both 4 and 3 are connected to 1 since they are each larger and to the left of 1, whereas neither 5 nor 2 is connected to 1 (see Figure 7.1). The graph $G[\pi]$ is sometimes called the inversion graph of π [30].

More formally if π is a permutation of the numbers $1, 2, \ldots, n$, then the graph $G[\pi] = (V, E)$ is defined as follows:

$$V = \{1, 2, \ldots, n\}$$

and

$$ij \in E \Leftrightarrow (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0.$$

3.1.2 Self-inverting Permutations

Self-inverting parmutations are a subclass belonging to the class of the permutations. As we have already seen, permutations may be represented in many ways but the most straightforward as we saw at the previous subsection is simply a rearrangement of the elements of the set $N_n = \{1, 2, \ldots, n\}$; in this way we think of the permutation $\pi = (5, 6, 9, 8, 1, 2, 7, 4, 3)$ as a rearrangement of the elements of the set N_9 such that "1 goes to 5", "2 goes to 6", "3 goes to 9", "4 goes to 8", and so on [80, 30]. Hereafter, we shall say that π is a permutation over the set N_9.

So, which is the extra property which distinguishes the subclass of the self-inverting permutations, or for short hereafter, SiP from the class of the permutations? The answer is at the following definition.

Definition 3.1.1. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be a permutation over the set N_n, where $n > 1$. The inverse of the permutation π is the permutation $q = (q_1, q_2, \ldots, q_n)$ with $q_{\pi_i} = \pi_{q_i} = i$. A *self-inverting permutation* (or, for short, SiP) is a permutation that is its own inverse: $\pi_{\pi_i} = i$.

By definition, a permutation is a SiP (self-inverting permutation) if and only if all its cycles are of length 1 or 2; for example, the permutation $\pi = (4, 3, 2, 1, 5, 7, 6)$ is a SiP with cycles: $(1, 4)$, $(2, 3)$, (5) and $(6, 6)$.

index	1	2	3	4	5	6	7
permutation	4	3	2	1	5	7	6

Figure 3.2: SiP's indexes and permutation numbers.

As you see at figure 3.2 the SiP has indeed its own inverse as, $\pi_1 = 4$ and $\pi_4 = 1$, $\pi_2 = 3$ and $\pi_3 = 2$, $\pi_5 = 5$ and $\pi_5 = 5$ and last $\pi_6 = 7$ and $\pi_7 = 6$.

Hereafter we shall represent SiPs with π^*.

We selected to use self-inverting permutations, as there is a way of representing every integer number to a self inverting permutation [15] which will be further analyzed at the following subsection.

Moreover as we will also later see, self-inverting permutations have error-correcting properties enabling extraction even after attacks. Such properties are for example the symmetric property and the fact that there is always a circle of length one.

But that's something that we we will thoroughly see and analyze in the meantime after first describing the algorithms for converting an integer to a SiP and representing a permutation in the 2D space.

3.1.3 Encoding Numbers as SiPs

There are several systems that correspond integer numbers into permutations or self-inverting permutations [80]. Chroni and Nikolopoulos [15] have proposed algorithms for such a system which efficiently encode an integer w into a self-inverting permutation π^* and efficiently decode it.

The encoding (w_to_SiP) and the decoding (SiP_to_w) algorithms of the codec system run in $O(n)$ time, where n is the length of the binary representation of the integer w, while the key-idea behind its algorithms is mainly based on mathematical objects, namely, bitonic permutations.

Following, is the step-by-step description below the two codec algorithms (encode and decode) that correspond integer numbers into self-inverting permutations (SiP).

After the description of the algorithms there will be also a demonstration of the correspondence between the integer $w = 12$ and the self-inverting permutation $\pi^* = (5, 6, 9, 8, 1, 2, 7, 4, 3)$ by the help of two examples. One fore the encoding process and one for the decoding process.

Algorithm w_to_SiP (Chroni and Nikolopoulos [15])
Input: a watermark integer w;
Output: the self-inverting permutation π^*;

1. Compute the binary representation $B = b_1 b_2 \dots b_n$ of w;
2. Construct the binary number $B' = 00\dots0 \| B \| 1$ of length $2n + 1$, and then the binary sequence $B^* = (b_1, b_2, \dots, b_{n'})$ of $flip(B')$;
3. Find the sequence $X = (x_1, x_2, \dots, x_k)$ of the positions of 0's and the sequence $Y = (y_1, y_2, \dots, y_m)$ of the positions of 1's in B^* from left-to-right;
4. Construct the bitonic permutation $\pi = (x_1, x_2, \dots, x_k, y_m, y_{m-1}, \dots, y_1)$ on $n' = 2n + 1$ numbers;
5. Let $(z_1, z_2, \dots, z_k, z_{k+1}, z_{k+2}, \dots, z_{n'}) = (x_1, x_2, \dots, x_k, y_m, y_{m-1}, \dots, y_1)$;
   - Case 1: n' even: select $n'/2$ pairs $(z_1, z_{n'})$, $(z_2, z_{n'} - 1), \dots, (z_{n'/2}, z_{(n'+3)/2})$;
     for each selected pair (z_i, z_j), do the following:
     $\pi_{z_j} = z_i$ and $\pi_{z_i} = z_j$;
   - Case 2: n' odd: select $\lfloor n'/2 \rfloor$ pairs $(z_1, z_{n'}), (z_2, z_{n'-1}), , (z_{\lfloor n'/2 \rfloor}, z_{\lfloor n'/2 \rfloor + 2})$
     and the number z n'/2 +1;
     for each selected pair (z_i, z_j), do the following:
     $\pi_{z_j} = z_i$ and $\pi_{z_i} = z_j$;
     $\pi_{z_{\lfloor n'/2 \rfloor + 1}} = z_{\lfloor n'/2 \rfloor + 1}$;
6. Return the self-inverting permutation $\pi^* = (\pi_1, \pi_2, \dots, \pi_{n'})$ on $n' = 2n+1$ numbers;

Algorithm SiP_to_w (Chroni and Nikolopoulos [15])
Input: a self-inverting permutation π^*;
Output: the watermark integer w;

1. Compute the cycle representation $C = c_1 c_2 \dots c_k$ of the self-inverting permutation $\pi^* = (\pi_1, \pi_2, , \pi_{n'})$, where $n' = 2n + 1$;
2. Initially, $i = 1$, $j = n'$ and all the cycles of C are unmarked;
3. While there exists an unmarked cycle c in C, do the following:

   Select the first unmarked cycle c of C from left-to-right;

   - Case 1: the selected cycle c has length 2 and let $c = (a, b)$:
     $\pi_i = a$ and $\pi_j = b$; mark cycle c; $i = i + 1$ and $j = j - 1$;
   - Case 2: the selected cycle c has length 1 and let $c = (a)$:
     $\pi_i = a$; mark cycle c; $i = i + 1$;

4. Find the first increasing subsequence $X = (x_1, x_2, \dots, x_k)$ and then the first decreasing subsequence $Y = (y_1, y_2, \dots, y_m)$ of π;
5. Construct the binary sequence $B^* = (b_1, b_2, , b_{n'})$ as follows: set 0's in positions $x_1, x_2, \dots, x_k$ and 1's in positions $y_1, y_2, \dots, y_m$;
6. Compute $B' = flip(B*) = (b_1, b_2, , b_n, b_{n+1}, , b_{2n}, b_{2n+1})$;
7. Return the integer w of the binary number $B = b_{n+1}, b_{n+2}, \dots, b_{2n}$;

Example w_to_SiP: Let $w = 12$ be the given watermark integer. We first compute the binary representation $B = 1100$ of the number 12; then we construct the binary number $B' = 0000||1100||1$ and the binary sequence $B^* = (1, 1, 1, 1, 0, 0, 1, 1, 0)$ by flipping the elements of B'; we compute the sequences $X = (5, 6, 9)$ and $Y = (1, 2, 3, 4, 7, 8)$ by taking into account the indices of 0s and 1s in B^*, and then the bitonic permutation $\pi = (5, 6, 9, 8, 7, 4, 3, 2, 1)$ on $n' = 9$ numbers by taking the sequence $X||Y^R$; since n' is odd, we select 4 cycles $(5, 1), (6, 2), (9, 3), (8, 4)$ of lengths 2 and one cycle (7) of length 1, and then based on the selected cycles construct the self-inverting permutation $\pi = (5, 6, 9, 8, 1, 2, 7, 4, 3)$.

Example SiP-to-W: Let $\pi = (5, 6, 9, 8, 1, 2, 7, 4, 3)$ be the given self-inverting permutation produced by our method. The cycle representation of π is the following: $(1, 5), (2, 6), (3, 9), (4, 8), (7)$; from the cycles we construct the permutation $\pi = (5, 6, 9, 8, 7, 4, 3, 2, 1)$; then, we compute the first increasing subsequence $X = (5, 6, 9)$ and the first decreasing subsequence $Y = (8, 7, 4, 3, 2, 1)$; and construct the binary sequence $B^* = (1, 1, 1, 1, 0, 0, 1, 1, 0)$ of length 9; we flip the elements of B^* and construct the sequence $B' = (0, 0, 0, 0, 1, 1, 0, 0, 1)$; the binary number 1100 is the integer $w = 12$.

3.1.4 2D/2DM Representations

We first define the two-dimensional representation (2D representation) of a permutation π over the set $N_n = \{1, 2, \ldots, n\}$, and then its 2DM representation which is more suitable for efficient use in our codec system.

In the 2D representation, the elements of the permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ are mapped in specific cells of an $n \times n$ matrix A as follows:

$$\text{number} \quad \pi_i \quad \longrightarrow \quad \text{entry} \quad A(\pi_i^{-1}, \pi_i)$$

or, equivalently, the cell at row i and column π_i is labeled by the number π_i, for each $i = 1, 2, \ldots, n$.

Figure 3.3(a) shows the 2D representation of the self-inverting permutation $\pi = (6, 3, 2, 4, 5, 1)$.

Note that, there is one label in each row and in each column, so each cell in the matrix A corresponds to a unique pair of labels; see, [80] for a long bibliography on permutation representations and also in [13] for a DAG representation.

Based on the previously defined 2D representation of a permutation π, we next propose a two-dimensional marked representation (2DM representation) of π which is an efficient tool for watermarking images.

In our 2DM representation, a permutation π over the set $N_n = \{1, 2, \ldots, n\}$ is represented by an $n \times n$ matrix A^* as follows:

- the cell at row i and column π_i is marked by a specific symbol, for each $i = 1, 2, \ldots, n$;
- in our implementation, the used symbol is the asterisk, i.e., the character "*".

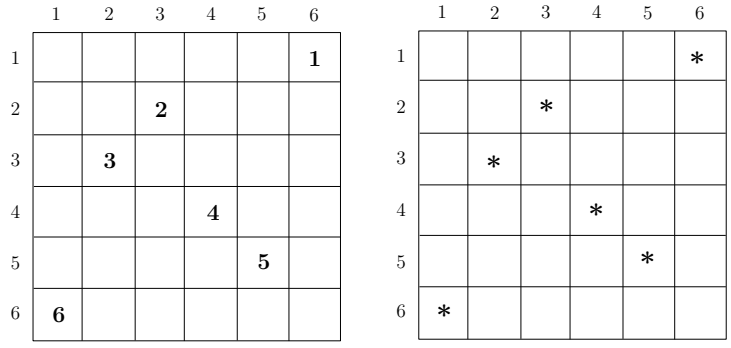


Figure 3.3: The 2D and 2DM representations of the self-inverting permutation $\pi = (6, 3, 2, 4, 5, 1)$.

Figure 3.3(b) shows the 2DM representation of the permutation π. It is easy to see that, since the 2DM representation of π is constructed from the corresponding 2D representation, there is also one symbol in each row and in each column of the matrix A^*.

We next present an algorithm which extracts the permutation π from its 2DM representation matrix. More precisely, let π be a permutation over N_n and let A^* be the 2DM

representation matrix of π (see, Figure 3.3(b)); given the matrix A^*, we can easily extract π from A^* in linear time (i.e., linear in the size of matrix A^*) by the following algorithm:

Algorithm Extract_π_from_2DM
Input: the 2DM representation matrix A^* of π;
Output: the permutation π;

1. For each row i of matrix A^*, $1 \leq i \leq n$, and
   for each column j of matrix A^*, $1 \leq j \leq n$,
   if the cell (i, j) is marked then $\pi_i \leftarrow j$;

2. Return the permutation π;

Remark 3.1.1. It is easy to see that the resulting permutation π, after the execution of Step 1, can be taken by reading the matrix A^* from top row to bottom row and write down the positions of its marked cells. Since the permutation π is a self-inverting permutation, its 2D matrix A has the following property:

- $A(i, j) = j$ if $\pi_i = j$, and
- $A(i, j) = 0$ otherwise, $1 \leq i, j \leq n$.

Thus, the corresponding matrix A^* is symmetric:

- $A^*(i, j) = A^*(j, i) =$ "mark" if $\pi_i = j$, and
- $A^*(i, j) = A^*(j, i) = 0$ otherwise, $1 \leq i, j \leq n$.

Based on this property, it is also easy to see that the resulting permutation π can be also taken by reading the matrix A^* from left column to right column and write down the positions of its marked cells.

Hereafter, we shall denote by π^* a SiP and by n^* the number of elements of π^*.

3.1.5 Color Images

A digital image is a numeric representation of a 2-dimensional image; it has a finite set of values, called *picture elements* or *pixels*, that represent the brightness of a given color at any specific point in the image. [31].

A digital image contains a fixed number of rows and columns of pixels which are usually stored in computer memory as a two-dimensional matrix I of numeric values; in our implementation the numeric values are integers from 0 to 255. When we say that an image has a *resolution* of $N \times M$ we mean that its two-dimensional matrix I contains N rows and M columns and the value of each entry $I(i, j)$, i.e., the value of each pixel, is an integer k_0 (grayscale image), or a triple of integers (k_1, k_2, k_3) (color image), $0 \leq k_0, k_1, k_2, k_3 \leq 255$.

There are several models used for representing color. In our implementation, we use the *RGB* model; it is an additive color model in which *red*, *green*, and *blue* light is added

together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, Red, Green, and Blue [31, 65].

In our implementation we use the *RGB* model, where the name comes from the initials of the three additive colors Red, Green and Blue. The range of colors can be represented on the Cartesian 3-dimensional system. The axes x, y and z are used for the red green and blue color respectively

- on the x-axis (R-axis) we have the brightness of the **red** colour,
- on the y-axis (G-axis) we have the brightness of the **green** colour, and
- on the z-axis (B-axis) we have the brightness of the **blue** colour.

Figure 3.4 shows the 3D topology of the colors. For example, the white color (255, 255, 255) is located in the front upper right point of the color cube.

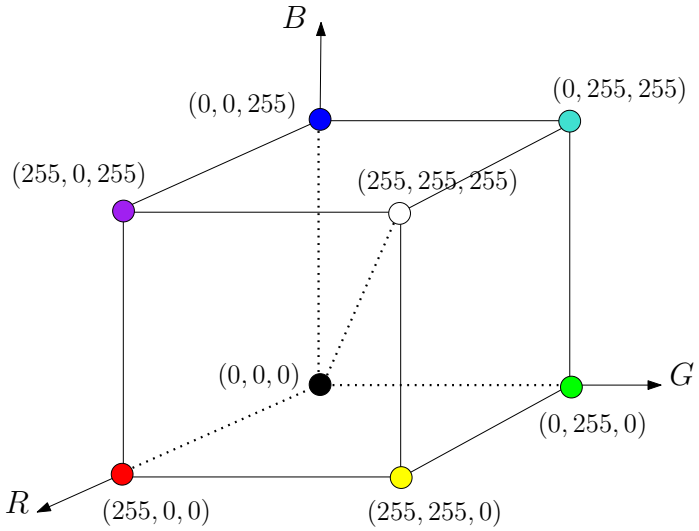


Figure 3.4: The range of colors represented on the Cartesian 3-dimensional system.

In our system, since a color is a triple of integers (x, y, z), a digital image I of resolution $N \times M$ (i.e., it contains N rows and M columns of pixels) is stored in a three-dimensional matrix Img of size $N \times M \times 3$ as follows:

if the pixel $I(i, j)$ of the image I has (x, y, z) color, then $Img(i, j, 1) = x$, $Img(i, j, 2) = y$, and $Img(i, j, 3) = z$.

For example, let (240, 29, 35) be the color of the upper left pixel of an image I, i.e., $I(1, 1) = (240, 29, 35)$. Then, in our system $Img(1, 1, 1) = 240$, $Img(1, 1, 2) = 29$, and $Img(1, 1, 3) = 35$.

3.2 Marking in the Spatial Domain

As mentioned, our initial idea of embedding watermarks using the 2DM representation of the permutations was based on using marks in the spatial domain of the image.

Briefly that was done by altering pixels' values at specific areas of the image. Embedding and extracting is based on the idea of comparing the values (difference) between the certain pixels. Marked are considered the areas where this "difference" betweens pixels is the greatest.

That was the idea that helped us develop the enhanced method in the frequency domain, us the idea of representing permutations in the 2D space remained the same, but instead of using marks in the spatial domain different kind of marks were used in the frequency domain to enable marking of specific areas of the image, but more information are about to follow in the next pages.

In the next subsections you will find more details concerning the embedding and the extracting algorithms of our initial method in the spatial domain. Those are described step by step.

3.2.1 Embed Watermark into Image

We next describe the algorithm Encode_SIP-to-IMAGE of our codec system which embeds a self-inverting permutation (SIP) π^* into an image I; recall that, in our system we use a SIP π^* over the set N_{n^*} for encoding the watermark w, where $n^* = 2n + 1$ and n is the length of the binary representation of the integer w (author's technique); see, Subsection 3.1.3.

The algorithm takes as input a SIP π^* and an image I, in which the user wants to embed the watermark $w = \pi^*$, and produces the watermarked image I_w; it works as follows:

Step 1: The algorithm first computes the 2DM representation of the permutation π^*, that is, it computes the $n^* \times n^*$ array A (see, Subsection 3.1.4); the entry (i, π_i^*) of the array A contains the symbol "*", $1 \leq i \leq n^*$.

Step 2: Next, the algorithm computes the size $N \times M$ of the input image I and do the following: if N is an even number it removes the pixels from the bottom row of I and reduces N by 1, while if M is an even number it removes the pixels from the right column of I and reduces M by 1. The resulting image has size $N^* \times M^*$, where N^* and M^* are both odd numbers.

Step 3: Let n^* be the size of the SIP π^* and let $N^* \leq M^*$. Now the algorithm takes the input image I and places on it an imaginary grid $\mathcal{G}$, which covers almost the whole image I, having

$$n^* \times n^* \text{ grid-cells } C_{ij}(I)$$

each $C_{ij}(I)$ of size

$$\lfloor N^*/n^* \rfloor \times \lfloor N^*/n^* \rfloor$$

where, $1 \leq i, j \leq n^*$.

It places the imaginary grid $\mathcal{G}$ on I as follows: it first locates the central pixel P^0_{cent} of the image I, which is at position $(\lfloor N^*/2 \rfloor + 1, \lfloor M^*/2 \rfloor + 1)$, then locates the central pixel p^0_{ii} of the central grid-cell $C_{ii}(I)$, where $i = \lfloor n^*/2 \rfloor + 1$, and places the grid $\mathcal{G}$ on image I such that both P^0_{cent} and p^0_{ii} have the same position in I.

Step 4: Then it scans the image and goes to each grid-cell $C_{ij}(I)$ (there are always $n^* \times n^*$ grid-cells in any image) and locates the central pixel p^0_{ij} of the grid-cell $C_{ij}(I)$ and also the four pixels p^1_{ij}, p^2_{ij}, p^3_{ij}, and p^4_{ij} around it, $1 \leq i, j \leq n^*$; hereafter, we shall call these four pixels *cross* pixels.

Then, it computes the difference between the brightness of the central pixel p^0_{ij} and the average brightness of the twelve pixels around it, that is, the pixels $p^{\ell 1}_{ij}$, $p^{\ell 2}_{ij}$, and $p^{\ell 3}_{ij}$ $(\ell = 1, 2, 3, 4)$, and stores this value in the variable $\text{dif}(p^0_{ij})$ (see, Figure 3.5).

Finally, it computes the maximum absolute value of all $n^* \times n^*$ differences $\text{dif}(p^0_{ij})$, $1 \leq i, j \leq n^*$, and stores it in the variable $\text{Maxdif}(I)$.

Step 5: The algorithm goes again to each central pixel p^0_{ij} of each grid-cell C_{ij} and if the corresponding entry $A(i, j)$ contains the symbol "*", then it increases

- the brightness k^0_{ij} of the central pixel p^0_{ij}, and
- the brightness k^1_{ij}, k^2_{ij}, k^3_{ij}, and k^4_{ij} of its cross pixels.

Actually, it first increases the central pixel p^0_{ij} by the value e^0_{ij} so that it surpasses the image's maximum difference $\text{Maxdif}(I)$ by a constant c; that is,

- $k^0_{ij} + e^0_{ij} = \text{Maxdif}(I) + c$

and, then, it sets the brightness of the four cross pixels p^1_{ij}, p^2_{ij}, p^3_{ij}, and p^4_{ij} equal to k^0_{ij}.

In our implementation we use c for robustness, so the brightness k^0_{ij} of the central pixel of each grid-cell C_{ij} is increased by e^0_{ij}, where

$$e^0_{ij} = \text{Maxdif}(I) - k^0_{ij} + c \tag{3.1}$$

where, $1 \leq i, j \leq n^*$.

Let I_w be the resulting image after increasing the brightness of the n^* central and the corresponding cross pixels, with respect to π, of the image I. Hereafter, we call the n^* central pixels of I as *2DM-pixels*; recall that, p^0_{ij} is a 2DM-pixel if $A(i, \pi_i) =$ "*", or, equivalently, the cell (i, π_i) of the matrix A is marked.

Step 6: The algorithm returns the watermarked image I_w.

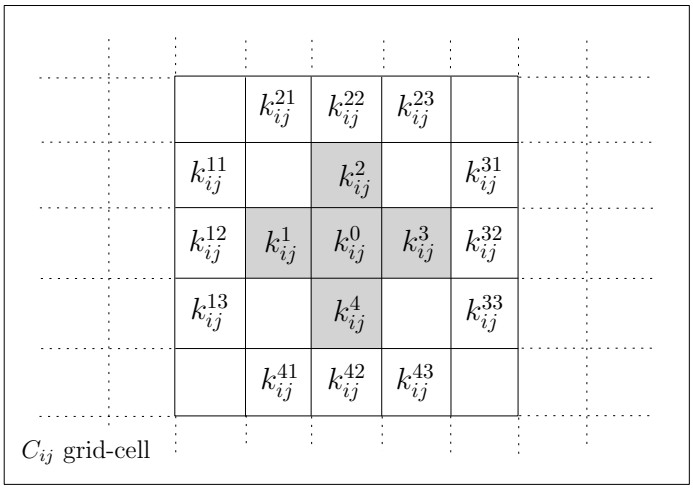


Figure 3.5: The brightness k_{ij}^{ℓ} of the central and cross pixels p_{ij}^{ℓ} of the grid-cell $C_{ij}(I)$, $0 \leq \ell \leq 4$, and the brightness $k_{ij}^{\ell m}$ of the cycle-cross pixels $p_{ij}^{\ell m}$, $1 \leq \ell \leq 4$ and $m = 1, 2, 3$.

3.2.2 Extract Watermark from Image

Next we describe our decoding algorithm which is responsible for extracting the watermark $w = \pi^*$ form image I_w. In particular, the algorithm, which we call Decode_IMAGE-to-SIP, takes as input a watermarked image I_w and returns the SIP π^* which corresponds to integer watermark w; the steps of the algorithm are the following:

Step 1: The algorithm places again the same imaginary $n^* \times n^*$ grid on image I_w and locates the central pixel p_{ij}^0 of each grid-cell $C_{ij}(I)$, $1 \leq i, j \leq n^*$; there are $n^* \times n^*$ central pixels in total. Then, it finds the n^* central pixels $p_1^0, p_2^0, \ldots, p_{n^*}^0$, among the $n^* \times n^*$, with the maximum brightness using a known sorting algorithm [18].

Step 2: In this step, the algorithm takes the n^* grid-cell $C_1, C_2, \ldots, C_{n^*}$ of the image I_w which correspond to n^* central pixels $p_1^0, p_2^0, \ldots, p_{n^*}^0$, and compute an $n^* \times n^*$ matrix A^* as follows:

- Initially, set $A^*(i, j) \leftarrow 0$, $1 \leq i, j \leq n^*$;
- For each grid-cell C_m, $1 \leq m \leq n^*$, do:

  if (i, j) is the position of the grid-cell C_m in the grid $\mathcal{G}$ then set $A^*(i, j) \leftarrow$ "$*$";

It is easy to see that, the $n^* \times n^*$ matrix A^* is exactly the 2DM representation of the self-inverting permutation π^* embedded in image I_w by the algorithm Encode_SIP-to-IMAGE.

Then, the permutation π^* can be extracted from the matrix A^* using the algorithm Extract_π_from_2DM; see, Subsection 3.1.4.

Step 3: Finally, the algorithm returns the self-inverting permutation π^*.

3.2.3 The WaterIP system

Along with Maria Chroni and Stavros D. Nikolopoulos we implemented a system for educational purposes based on the above algorithms.

The purpose of this system was to enable students understand the notion of watermarking through embedding and extracting watermarks from images. We believe that by making them participate actively in the whole process motivate them and give them the will to learn more about ideas like intellectual property and watermarking [13].

To give some information about this system, we have named it WaterIP and it provides a student with the two following main working levels:

- **Embed level**: Through a friendly graphical user interface, the student creates a secret key (i.e., the watermark w) and selects a picture I in which he wants to embed the watermark; in our system the watermark w is a permutation π over the set N_6 and it is embedded into the original picture I, using the 2DM representation, resulting the watermarked picture I_w.

- **Mark level**: The student, in order to prove that he is the owner of the picture I_w, inputs the watermarked picture I_w into the system which makes the marks visible to the student so that he will be able to easily extract the watermark w (i.e., his secret key) just by looking at the marks; in particular, the system returns the marked picture I_m to the student.

Our WaterIP system consists of two main components. The usability of the system is based on a friendly to the student graphical user interface. Using it he can easily produce his watermark w, i.e., a permutation π, using his mouse without making any mistake. He can also choose an image I from his computer and he can either embed a watermark into I resulting the watermarked image I_w or make the marks of I_w visible so that he will be able to prove to his teacher that the picture belongs to him.

The first component is responsible for embedding the desired watermark $w = \pi$ into the image I using the 2DM representation of π, while the second one is responsible for making the marks of a watermarked image I_w visible to the student so that he will be able to easily extract the watermark w by hand.

There could have been a third component for extracting the permutation from the watermarked image but we chose not to include it because we consider important, for pedagogical reasons, that the student must participate interactively in the process of proving ownership.

All the system's algorithms have been initially developed and tested in MATLAB [32] and then redeveloped and also tested in JAVA.

3.3 Marking in the Frequency Domain

Having described an efficient method for encoding integers as self-inverting permutations using the 2DM representation of self-inverting permutations, we next describe codec algorithms that efficiently encode and decode a watermark into the image's frequency domain [83, 54, 27, 31].

3.3.1 Embed Watermark into Image

We next describe the embedding algorithm of our proposed technique which encodes a self-inverting permutation (SiP) π^* into a digital image I.

Recall that, the permutation π^* is obtained over the set N_{n^*}, where $n^* = 2n + 1$ and n is the length of the binary representation of an integer w which actually is the image's watermark [15]; For more information go back to Subsection 3.1.3.

The watermark w, or equivalently the self-inverting permutation π^*, is invisible and it is inserted in an image I by placing marks in the frequency (Fourier) domain of specific areas of the image that are denoted by an imaginary grid. More precisely, we mark the DFT's magnitude of an image's area using two ellipsoidal annuli, denoted hereafter as "Red" and "Blue" (see, Figure 3.6). The ellipsoidal annuli are specified by the following parameters:

- P_r, the width of the "Red" ellipsoidal annulus,
- P_b, the width of the "Blue" ellipsoidal annulus,
- R_1 and R_2, the radiuses of the "Red" ellipsoidal annulus on y-axis and x-axis, respectively.

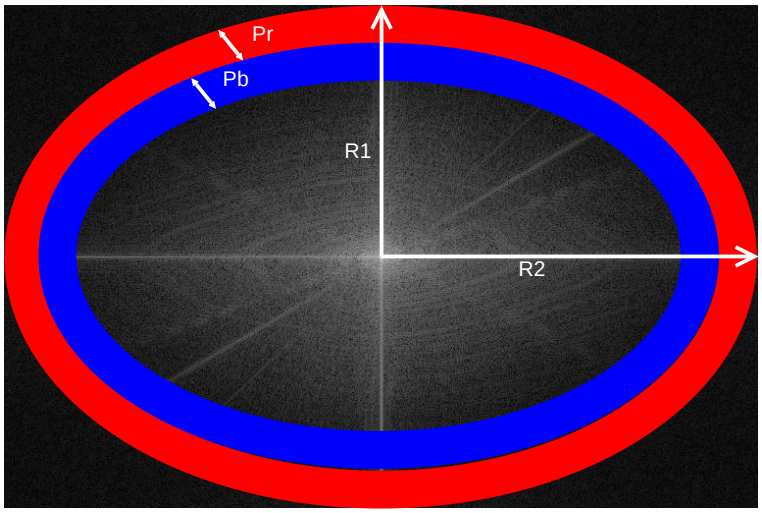


Figure 3.6: The "Red" and "Blue" ellipsoidal annuli.

The algorithm takes as input a SiP π^* and an image I, in which the user embeds the watermark, and returns the watermarked image I_w; it consists of the following steps.

Algorithm Embed_SiP-to-Image
Input: the watermark $\pi^* \equiv w$ and the host image I;
Output: the watermarked image I_w;

Step 1: Compute first the 2DM representation of the permutation π^*, i.e., construct an array A^* of size $n^* \times n^*$ such that the entry $A^*(i, \pi_i^*)$ contains the symbol "*", $1 \leq i \leq n^*$.

Step 2: Next, calculate the size $N \times M$ of the input image I and cover it with an imaginary grid C with $n^* \times n^*$ grid-cells C_{ij} of size $\lfloor \frac{N}{n^*} \rfloor \times \lfloor \frac{M}{n^*} \rfloor$, $1 \leq i, j \leq n^*$.

Step 3: For each grid-cell C_{ij}, compute the Discrete Fourier Transform (DFT) using the Fast Fourier Transform (FFT) algorithm, resulting in a $n^* \times n^*$ grid of DFT cells F_{ij}, $1 \leq i, j \leq n^*$.

Step 4: For each DFT cell F_{ij}, compute its magnitude M_{ij} and phase P_{ij} matrices which are both of size $\lfloor \frac{N}{n^*} \rfloor \times \lfloor \frac{M}{n^*} \rfloor$, $1 \leq i, j \leq n^*$.

Step 5: Then, the algorithm takes each of the $n^* \times n^*$ magnitude matrices M_{ij}, $1 \leq i, j \leq n^*$, and places two imaginary ellipsoidal annuli, denoted as "Red" and "Blue", in the matrix M_{ij} (see, Figure 3.6). In our implementation,

- the "Red" is the outer ellipsoidal annulus while the "Blue" is the inner one. Both are concentric at the center of the M_{ij} magnitude matrix and have widths P_r and P_b, respectively;
- the radiuses of the "Red" ellipsoidal annulus are R_1 (on the y-axis) and R_2 (on the x-axis), while the "Blue" ellipsoidal annulus radiuses are computed in accordance to the "Red" ellipsoidal annulus and have values $(R_1 - P_r)$ and $(R_2 - P_r)$, respectively;
- the inner perimeter of the "Red" ellipsoidal annulus coincides to the outer perimeter of the "Blue" ellipsoidal annulus;
- the values of the widths of the two ellipsoidal annuli are $P_r = 2$ and $P_b = 2$, while the values of their radiuses are $R_1 = \lfloor \frac{N}{2n^*} \rfloor$ and $R_2 = \lfloor \frac{M}{2n^*} \rfloor$.

The areas covered by the "Red" and the "Blue" ellipsoidal annuli determine two groups of magnitude values on M_{ij} (see, Figure 3.6).

Step 6: For each magnitude matrix M_{ij}, $1 \leq i, j \leq n^*$, compute the average of the values that are in the areas covered by the "Red" and the "Blue" ellipsoidal annuli; let $AvgR_{ij}$ be the average of the magnitude values belonging to the "Red" ellipsoidal annulus and $AvgB_{ij}$ be the one of the "Blue" ellipsoidal annulus.

Step 7: For each magnitude matrix M_{ij}, $1 \leq i, j \leq n^*$, compute first the variable D_{ij} as follows:

- $D_{ij} = |AvgB_{ij} - AvgR_{ij}|$, if $AvgB_{ij} \leq AvgR_{ij}$
- $D_{ij} = 0$, otherwise.

Then, for each row i of the magnitude matrix M_{ij}, $1 \leq i, j \leq n^*$, compute the maximum value of the variables $D_{i1}, D_{i2}, \ldots, D_{in^*}$ in row i; let $MaxD_i$ be the max value.

Step 8: For each cell (i, j) of the 2DM representation matrix A^* of the permutation π^* such that $A^*_{ij} =$ "$*$" (i.e., marked cell), mark the corresponding grid-cell C_{ij}, $1 \leq i, j \leq n^*$; the marking is performed by increasing all the values in magnitude matrix M_{ij} covered by the "Red" ellipsoidal annulus by the value

$$AvgB_{ij} - AvgR_{ij} + MaxD_i + c, \tag{3.2}$$

where $c = c_{opt}$. The additive value of c_{opt} is calculated by the function $f()$ (see, Subsection 3.3.3) which returns the minimum possible value of c that enables successful extracting.

Step 9: Reconstruct the DFT of the corresponding modified magnitude matrices M_{ij}, using the trigonometric formula [31], and then perform the Inverse Fast Fourier Transform (IFFT) for each marked cell C_{ij}, $1 \leq i, j \leq n^*$, in order to obtain the image I_w.

Step 10: Return the watermarked image I_w.
In Figure 3.7, we demonstrate the main operations performed by our embedding algorithm. In particular, we show the marking process of the grid-cell C_{44} of the Lena image; in this example, we embed in the Lena image the watermark number w which corresponds to SiP $(6, 3, 2, 4, 5, 1)$.

3.3.2 Extract Watermark from Image

In this section we describe the decoding algorithm of our proposed technique. The algorithm extracts a self-inverting permutation (SiP) π^* from a watermarked digital image I_w, which can be later represented as an integer w.

The self-inverting permutation π^* is obtained from the frequency domain of specific areas of the watermarked image I_w. More precisely, using the same two "Red" and "Blue" ellipsoidal annuli, we detect certain areas of the watermarked image I_w that are marked by our embedding algorithm and these marked areas enable us to obtain the 2D representation of the permutation π^*. The extracting algorithm works as follows:

Algorithm Extract_SiP-from-Image
Input: the watermarked image I_w marked with π^*;
Output: the watermark $\pi^* = w$;

Step 1: Take the input watermarked image I_w and calculate its $N \times M$ size. Then, cover it with the same imaginary grid C, as described in the embedding method, having $n^* \times n^*$ grid-cells C_{ij} of size $\lfloor \frac{N}{n^*} \rfloor \times \lfloor \frac{M}{n^*} \rfloor$.

Step 2: Then, again for each grid-cell C_{ij}, $1 \leq i, j \leq n^*$, using the Fast Fourier Transform (FFT) get the Discrete Fourier Transform (DFT) resulting a $n^* \times n^*$ grid of DFT cells.

Step 3: For each DFT cell, compute its magnitude matrix M_{ij} and phase matrix P_{ij} which are both of size $\lfloor \frac{N}{n^*} \rfloor \times \lfloor \frac{M}{n^*} \rfloor$.

Step 4: For each magnitude matrix M_{ij}, place the same imaginary "Red" and "Blue" ellipsoidal annuli, as described in the embedding method, and compute as before the average values that coincide in the area covered by the "Red" and the "Blue" ellipsoidal annuli; let $AvgR_{ij}$ and $AvgB_{ij}$ be these values.

Step 5: For each row i of M_{ij}, $1 \leq i \leq n^*$, search for the j_{th} column where $AvgB_{ij} - AvgR_{ij}$ is minimized and set $\pi_i^* = j$, $1 \leq j \leq n^*$.

Step 6: Return the self-inverting permutation π^*.

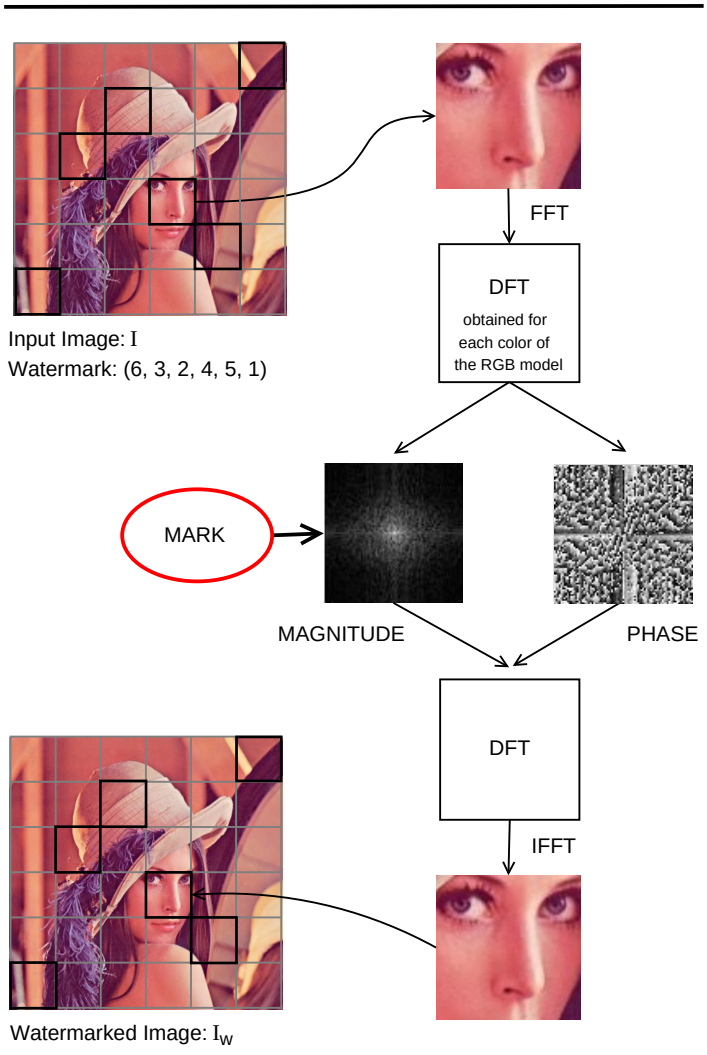


Figure 3.7: A flow of the embedding process.

3.3.3 Function f

Having presented the embedding and extracting algorithms, let us next describe the function f which returns the additive value $c = c_{opt}$ (see, Step 8 of the embedding algorithm Embed_SiP-to-Image).

Based on our marking model, the embedding algorithm amplifies the marks in the "Red" ellipsoidal annulus by adding the output of the function f. What exactly f does is returning the optimal value that allows the extracting algorithm under the current requirements, such as JPEG compression, to still be able to extract the watermark from the image.

The function f takes as an input the characteristics of the image and the parameters R_1, R_2, P_b, and P_r of our proposed marking model (see, Step 5 of embedding algorithm and Figure 3.6), and returns the minimum possible value c_{opt} that added as c to the values of the "Red" ellipsoidal annulus enables extracting (see, Step 8 of the embedding algorithm). More precisely, the function f initially takes the interval $[0, c_{max}]$, where c_{max} is a relatively great value such that if c_{max} is taken as c for marking the "Red" ellipsoidal annulus it allows extracting, and computes the c_{opt} in $[0, c_{max}]$.

Note that, c_{max} allows extracting but because of being great damages the quality of the image (see, Figure 4.1). We mentioned relatively great because it depends on the characteristics of each image. For a specific image it is useless to use a c_{max} greater than a specific value, we only need a value that definitely enables the extracting algorithm to successfully extract the watermark.

We next describe the computation of the value c_{opt} returned by the function f; note that, the parameters P_b and P_r of our implementation are fixed with the values 2 and 2, respectively. The main steps of this computation are the following:

(i) Check if the extracting algorithm for $c = 0$ validly obtains the watermark $\pi^* = w$ from the image I_w; if yes, then the function f returns $c_{opt} = 0$;

(ii) If not, that means, $c = 0$ doesn't allow extracting; then, the function f uses binary search on $[0, c_{max}]$ and computes the interval $[c_1, c_2]$ such that:

- $c = c_1$ doesn't allow extracting,
- $c = c_2$ does allow extracting, and
- $|c_1 - c_2| < 0.2$;

(iii) The function f returns $c_{opt} = c_2$;

As mentioned before, the function f returns the optimal value c_{opt}. Recall that, optimal means that it is the smallest possible value which enables extracting $\pi^* = w$ from the image I_w. It is important to be the smallest one as that minimizes the additive information to the image and, thus, assures minimum drop to the image quality.

Chapter 4

Evaluation

4.1 Testing Environment

In this section we present the experimental results of the proposed watermarking method which we have implemented using the general-purpose mathematical software package Matlab (version 7.7.0) [32].

We experimentally evaluated our codec algorithms on digital color images of various sizes and quality characteristics. Many of the images in our image repository where taken from a web image gallery [67] and enriched by some other images different in sizes and characteristics. Our experimental evaluation is based on two objective image quality assessment metrics namely Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) [89].

There are three main requirements of digital watermarking: *fidelity*, *robustness*, and *capacity* [21]. Our watermarking method appears to have high fidelity and robustness against JPEG compression.

Before moving on, some things should be noted concerning the initial method with marking in the spatial domain. In that method, for images with general characteristics and relatively large size according to the results the method delivers optically good results.

By saying "good results" we mean that the modifications made are quite invisible for various images and different sizes. Also the method's algorithms run really fast as they simply access a finite number of pixels. Furthermore, both the embedding and extracting algorithms are easy to modify and adjust for various scenarios. And last, it the method was also tested for images compressed using JPEG compression and extraction was successful for those cases.

On the other hand, the method fails to deliver "good results" either for relatively small images, or for images that depict something smooth which allows the eye to distinct the modifications on the image. Even though in smooth images the modification is the least possible that smoothness makes even the slightest modifications of certain pixels get visible. As for very small images the problem is the fact that as images get smaller the percentage of modified pixels gets greater and greater.

Another reason that made us to decide to move to the new method was the fact that the positions of the crosses are centered at strictly specific positions causing difficulties in the extracting algorithm even for the smallest geometric changes such as scaling or rotation. In those cases it was quite difficult to detect the crosses due to the fact that even movement by one pixel or loss of pixels is a problem.

Now, as for the latest enhanced method using marks in the frequency domain of the images, more thorough results will be presented. Those results have been acquired as we have tested our codec algorithms on various 24-bit digital color images of various sizes (from 200×130 up to 4600×3700) and various quality characteristics [11].

In our implementation we set both of the parameters P_r and P_b equal to 2; see, Subsection 3.2.1. Recall that, the value 2 is a relatively small value which allows us to modify a satisfactory number of values in order to embed the watermark and successfully extract it without affecting images' quality. There isn't a distance between the two ellipsoidal annuli as that enables the algorithm to apply a small additive information to the values of the "Red" annulus. The two ellipsoidal annuli are inscribed to the rectangle magnitude matrix, as we want to mark images' cells on the high frequency bands.

We mark the high frequencies by increasing their values using mainly the additive parameter $c = c_{opt}$ because alterations in the high frequencies are less detectable by human eye [45]. Moreover, in high frequencies most images contain less information.

In this work we used JPEG images due to their great importance on the web. In addition, they are small in size, while storing full color information (24 bit/pixel), and can be easily and efficiently transmitted. Moreover, robustness to lossy compression is an important issue when dealing with image authentication.

Notice that the design goal of lossy compression systems is opposed to that of watermark embedding systems. The Human Visual System model (HVS) of the compression system attempts to identify and discard perceptually insignificant information of the image, whereas the goal of the watermarking system is to embed the watermark information without altering the visual perception of the image [97].

The quality factor (or, for short, Q factor) is a number that determines the degree

of loss in the compression process when saving an image. In general, JPEG recommends a quality factor of 75–95 for visually indistinguishable quality loss, and a quality factor of 50–75 for merely acceptable quality. We compressed the images with Matlab JPEG compressor from `imwrite` with different quality factors; we present results for $Q = 90$, $Q = 75$ and $Q = 60$.

The quality function f returns the factor c, which has the minimum value c_{opt} that allows the extracting algorithm to successfully extract the watermark. In fact, this value c_{opt} is the main additive information embedded into the image; see, Formula 3.2.

Depending on the images and the amount of compression, we need to increase the watermark strength by increasing the factor c. Thus, for the tested images we compute the appropriate values for the parameters of the quality function f; this computation can be efficiently done by using the algorithm described in Subsection 3.3.3 of the previous chapter.

To demonstrate the differences on watermarked image human visual quality, with respect to the values of the additive factor c, we watermarked the original images Lena and Baboon and we embedded in each image the same watermark with $c = c_{max}$ and $c = c_{opt}$, where $c_{max} >> c_{opt}$; the results are demonstrated in Figure 4.1.

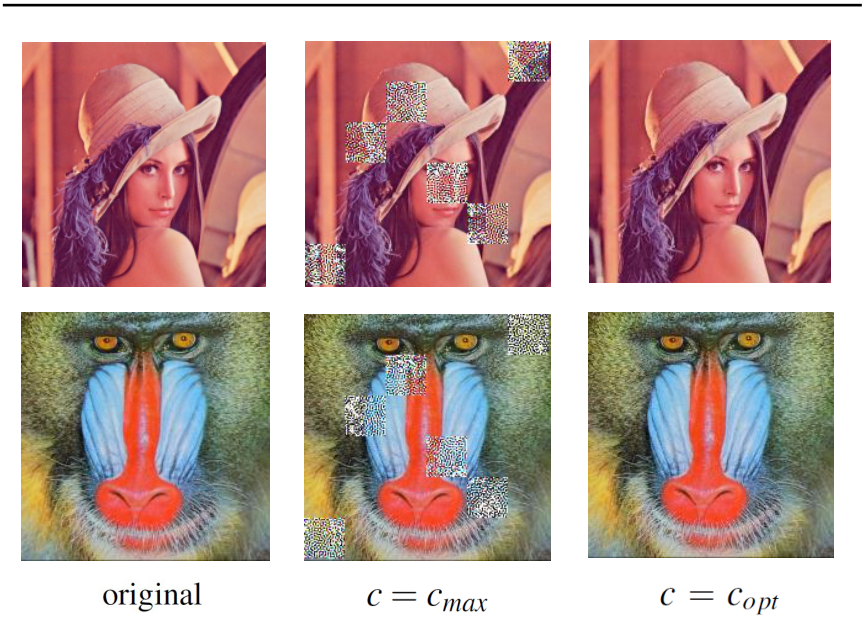


Figure 4.1: The original images of Lena and Baboon followed by their watermarked images with additive values $c = c_{max}$ and $c = c_{opt}$; both images are marked with the same watermark $(6, 3, 2, 4, 5, 1)$.

4.2 Design Issues

noindent We tested our codec algorithms on various 24-bit digital color images of various sizes (from 200×130 up to 4600×3700) and various quality characteristics.

In our implementation we set both of the parameters P_r and P_b equal to 2; see, Subsection 3.2.1. Recall that, the value 2 is a relatively small value which allows us to modify a satisfactory number of values in order to embed the watermark and successfully extract it without affecting images' quality.

There isn't a distance between the two ellipsoidal annuli as that enables the algorithm to apply a small additive information to the values of the "Red" annulus. The two ellipsoidal annuli are inscribed to the rectangle magnitude matrix, as we want to mark images' cells on the high frequency bands.

We mark the high frequencies by increasing their values using mainly the additive parameter $c = c_{opt}$ because alterations in the high frequencies are less detectable by human eye [45]. Moreover, in high frequencies most images contain less information.

In this work we used JPEG images due to their great importance on the web. In addition, they are small in size, while storing full color information (24 bit/pixel), and can be easily and efficiently transmitted. Moreover, robustness to lossy compression is an important issue when dealing with image authentication.

Notice that the design goal of lossy compression systems is opposed to that of watermark embedding systems. The Human Visual System model (HVS) of the compression system attempts to identify and discard perceptually insignificant information of the image, whereas the goal of the watermarking system is to embed the watermark information without altering the visual perception of the image [97].

The quality factor (or, for short, Q factor) is a number that determines the degree of loss in the compression process when saving an image. In general, JPEG recommends a quality factor of 75–95 for visually indistinguishable quality loss, and a quality factor of 50–75 for merely acceptable quality. We compressed the images with Matlab JPEG compressor from `imwrite` with different quality factors; we present results for $Q = 90$, $Q = 75$ and $Q = 60$.

The quality function f returns the factor c, which has the minimum value c_{opt} that allows the extracting algorithm to successfully extract the watermark. In fact, this value c_{opt} is the main additive information embedded into the image; see, Formula 3.2. Depending on the images and the amount of compression, we need to increase the watermark strength by increasing the factor c. Thus, for the tested images we compute the appropriate values for the parameters of the quality function f; this computation can be efficiently done by using the algorithm described in Subsection 3.3.3.

To demonstrate the differences on watermarked image human visual quality, with respect to the values of the additive factor c, we watermarked the original images Lena and Baboon and we embedded in each image the same watermark with $c = c_{max}$ and $c = c_{opt}$, where $c_{max} >> c_{opt}$; the results are demonstrated in Figure 4.1.

4.3 Image Quality Assessment

In order to evaluate the watermarked image quality obtained from our proposed watermarking method we used two objective image quality assessment metrics, that is, the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Metric (SSIM). Our aim was to prove that the watermarked image is closely related to the original (image fidelity), because watermarking should not introduce visible distortions in the original image as that would reduce images' commercial value.

The PSNR metric is the ratio of the reference signal and the distortion signal (i.e., the watermark) in an image given in decibels (dB); PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g., for image compression). The higher the PSNR value the closer the distorted image is to the original or the better the watermark conceals. It is a popular metric due to its simplicity, although it is well known that this distortion metric is not absolutely correlated with human vision.

For an initial image I of size $N \times M$ and its watermarked image I_w, PSNR is defined by the formula:

$$\mathrm{PSNR}(I, I_w) = 10 \log_{10} \frac{N_{max}^2}{MSE}, \tag{4.1}$$

where N_{max} is the maximum signal value that exists in the original image and MSE is the Mean Square Error given by

$$\mathrm{MSE}(I, I_w) = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i,j) - I_w(i,j))^2. \tag{4.2}$$

The SSIM image quality metric [89] is considered to be correlated with the quality perception of the HVS [37]. The SSIM metric is defined as follows:

$$\mathrm{SSIM}(I, I_w) = \frac{(2\mu\mu_w + C_1)(2\sigma(I, I_w) + C_2)}{(\mu^2 + \mu_w^2 + C_1)(\sigma(I)^2 + \sigma(I_w)^2 + C_2)}, \tag{4.3}$$

where μ and μ_w are the mean luminances of the original and watermarked image I respectively, $\sigma(I)$ is the standard deviation of I, $\sigma(I_w)$ is the standard deviation of I_w, whereas C_1 and C_2 are constants to avoid null denominator. We use a mean SSIM (MSSIM) index to evaluate the overall image quality over the M sliding windows; it is given by the following formula:

$$\mathrm{MSSIM}(I, I_w) = \frac{1}{M} \sum_{i=0}^{M} SSIM(I, I_w). \tag{4.4}$$

The highest value of SSIM is 1, and it is achieved when the original and watermarked images, that is, I and I_w, are identical.

Our watermarked images have excellent PSNR and SSIM values. In Figure 4.2, we present three images of different sizes, along with their corresponding PSNR and SSIM

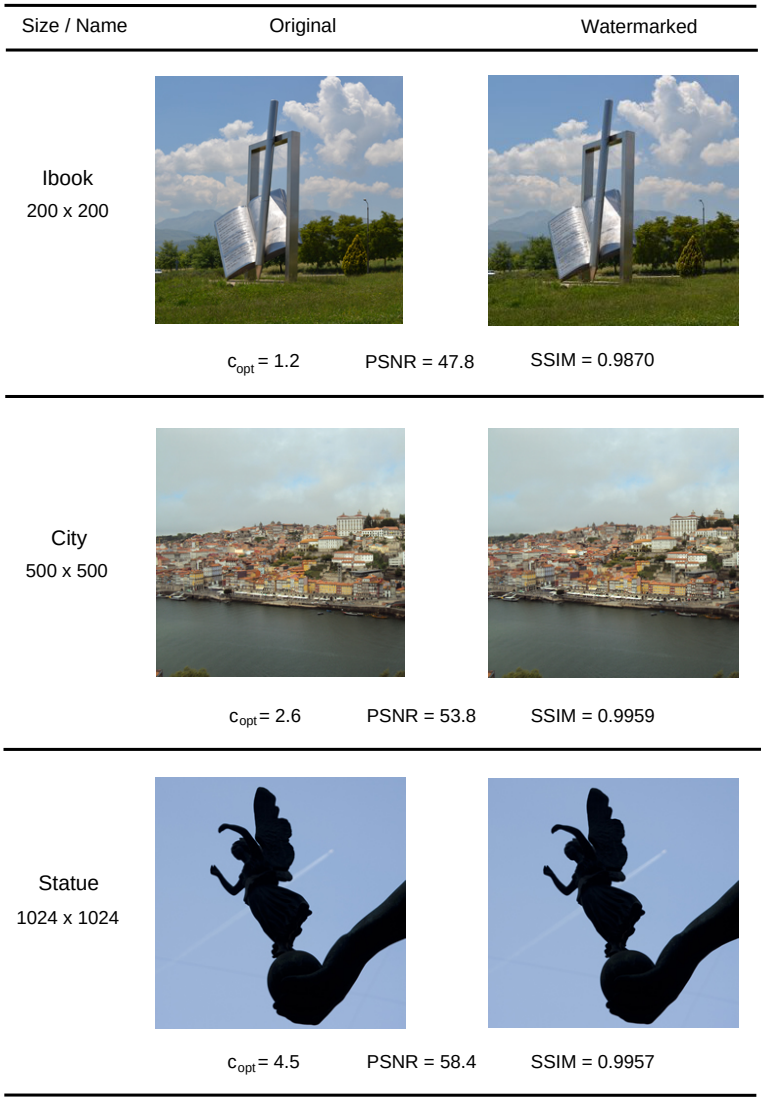


Figure 4.2: Sample images of three size groups for JPEG quality factor $Q = 75$ and their corresponding watermarked ones; for each image, the c_{opt}, PSNR and SSIM values are also shown.

values. Typical values for the PSNR in lossy image compression are between 40 and 70 dB, where higher is better. In our experiments, the PSNR values of 90% of the watermarked images were greater than 40 dB. The SSIM values are almost equal to 1, which means that the watermarked image is quite similar to the original one, which proves the method's high fidelity.

In Table 4.1 and 4.2, we demonstrate the PSNR and SSIM values of some selected images of various sizes used in our experiments. We observe that both values, PSNR and SSIM, decrease as the quality factor of the images becomes smaller.

That happens because as the quality of the watermarked images drops watermarks fade. Because of that we need "stronger", more robust watermarks. To do that the additive value gets greater and the "stronger" watermarks cause greater distortions to the images. And that of course gives as the expected results concerning the PSNR and

PSNR VALUES

Image	Size	Qual. 90	Qual. 75	Qual. 60
Ibook	200	54.7	47.8	42.9
City		52.6	47.3	43.6
Statue		52.3	46.2	42.6
Ibook	500	58.2	54.5	46.5
City		58.7	53.8	44.7
Statue		60.7	51.5	49.6
Ibook	1024	65.6	57.9	52.0
City		64.4	56.7	49.6
Statue		67.5	58.4	51.4

Table 4.1: The PSNR values of watermarked images of different sizes under JPEG qualities $Q = 90$, 75 and 60.

SSIM VALUES

Image	Size	Qual. 90	Qual. 75	Qual. 60
Ibook	200	0.9972	0.9870	0.9670
City		0.9959	0.9860	0.9705
Statue		0.9898	0.9664	0.9419
Ibook	500	0.9981	0.9957	0.9782
City		0.9985	0.9959	0.9743
Statue		0.9978	0.9838	0.9767
Ibook	1024	0.9995	0.9975	0.9913
City		0.9995	0.9974	0.9884
Statue		0.9995	0.9957	0.9813

Table 4.2: The SSIM values of watermarked images of different sizes under JPEG qualities $Q = 90$, 75 and 60.

SSIM quality metrics.

Moreover, the additive value c that enables robust marking under qualities $Q = 90$, 75 and 60 does not result in a significant image distortion as Tables 4.1 and 4.2 suggest; see also the watermarked images on Figure 4.2.

Closing, we mention that Lena and Baboon images of Figure 4.1 are both of size 200×200. Lena image has PSNR values 55.4, 50.1, 46.2 and SSIM values 0.9980, 0.9934, 0.9854 for $Q = 90$, 75 and 60, respectively, while Baboon image has PSNR values 52.7, 46.2, 42.5 and SSIM values 0.9978, 0.9908, 0.9807 for the same quality factors.

4.4 Overcoming Geometrical Attacks

Thanks to certain properties of the self-inverting permeations, as well as properties of their 2D/2DM representation we are able to extract watermarks even if the image has been subject to certain geometrical attacks.

Specifically, such attacks might be rotation and cropping. When an image is rotated or cropped, we would not like to extract an invalid watermark as the sequence of the marks is now different or not compete resulting in a false extracted 2DM representation. And of course a false 2DM representation might lead to numerical watermark value of to nothing at all.

Such kind of problems like the ones from abive, can be avoided as the sequence of the marks is not random when we have a self-inverting permutation. That enables us to detect the valid angle of the image in case of rotations. As for cropping now, the symmetric property of the 2DM representation enables recovering marked areas thanks to the symmetric respective marked areas. More details are about to follow in the next two subsections.

4.4.1 Rotation attacks

Beginning with the correction of the watermark in case of error caused from rotation attack we give the following demonstration.

We have an initial watermarked image with the extracted watermark indicated as see as seen in figure 4.3. The embedded watermark is the one resulted from the numerical watermark $w = 12$ or if you prefer with the self-inverting permutation $\pi^* = (5, 6, 9, 8, 1, 2, 7, 4, 3)$.

Concerning this case, we can detect wether the watermarked image has been subject to rotations thanks to the following two properties having to to with the marked areas of the grid:

Figure 4.3: the initial watermarked image.

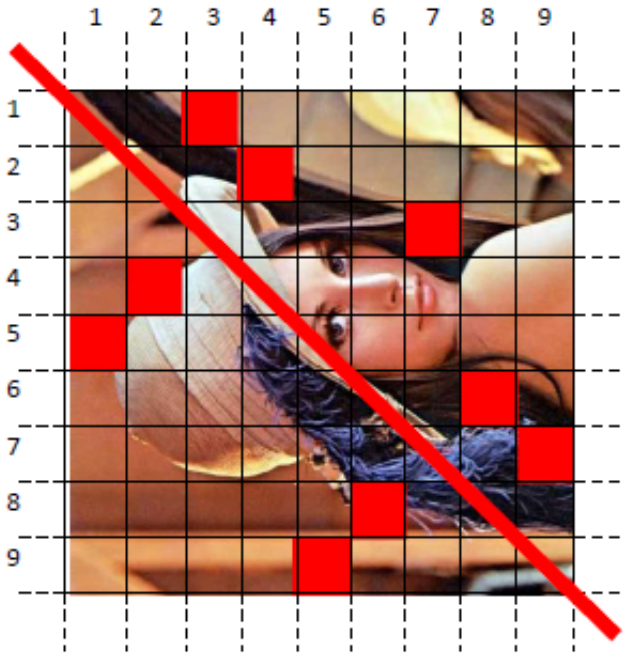


Figure 4.4: 90 degrees angled image.

- The main diagonal of the $n^* \times n^*$ symmetric matrix A^* have always one and only one marked cell.

- The marked cell on the diagonal is always in the entry (i, i) of A^* where: $i = \lceil \frac{n*}{2} \rceil + 1, \lceil \frac{n*}{2} \rceil + 2, \ldots n^*$.

If the image is rotated by 90 degrees as demonstrated in figure 4.5 you can notice that the main diagonal has not any cells marked which means that this is not a valid watermark as there should have been at exactly one marked cell. Thus we come up with the assumption that the image have been subjected to rotation.

Extraction should be again attempted with another angle and checking should be made once again whether with the two conditions above are valid.

Last, if the image is rotated by 180 degrees as in figure 4.5 you can notice that a cell is marked in the main diagonal in a position (i, i). At that very position $i < \lceil \frac{n^*}{2} \rceil$.

That proves once again that this is not valid watermark and that the image have been subjected to 180 degree rotation.

And again extraction should be again attempted with another angle and checking should be made again with the two conditions above.

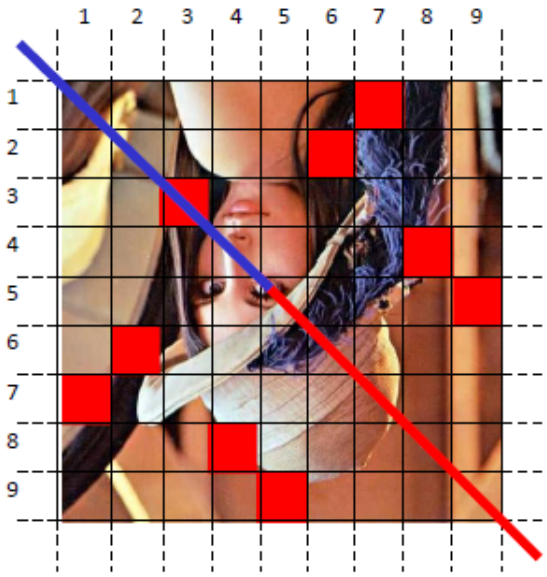


Figure 4.5: 180 degrees angled image.

4.4.2 Replace Part of the Image

Cropping recovery can be achieved by the following property of the 2D/2DM representation of a self-inverting permutation.

- The The $n^* \times n^*$ matrix A* is symmetric on its main diagonal.

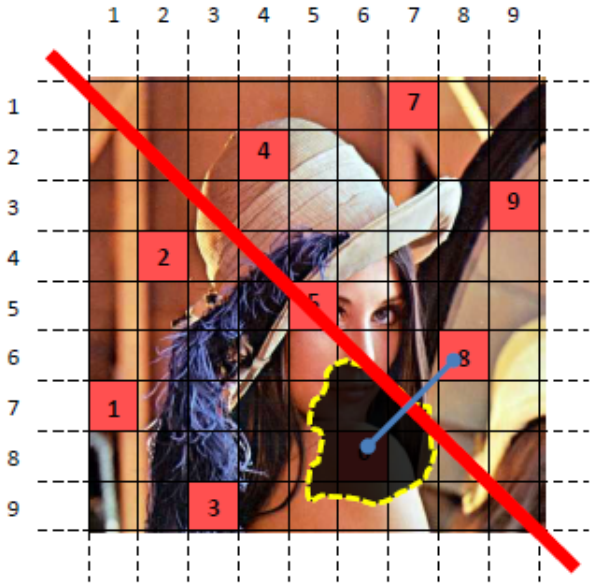


Figure 4.6: Watermarked image with removed part

As you see on figure 4.6 the removed marked part of the image can be recovered as marks using a self inverting permutation are symmetric on A^* with its main diagonal. Thus, in our example, because of the fact that $(6, 8)$ is marked, $(8, 6)$ is marked as well.

4.5 Other Experimental Outcomes

In the following, based on our experimental results, we discuss several impacts concerning characteristics of the host images and our embedding algorithm, and also we justify them by providing explanations on the observed outcomes.

The Additive Value Influences. As the experimental results show the PSNR and SSIM values decrease after embedding the watermark in images with lower quality index in its JPEG compression; see, Tables 4.1 and 4.2. That happens since our embedding algorithm adds more information in the frequency of marked image parts. By more information we mean a greater additive factor c; see, Equetion 3.2.

We next discuss an important issue concerning the additive value $c = c_{opt}$ returned by function f; see, Subsection 3.3.3. In Table 4.3, we show a sample of our results demonstrating for each JPEG quality the respective values of the additive factor c_{opt}. The figures show that the c_{opt} value increases as the quality factor of JPEG compression decreases. It is obvious that the embedding algorithm is image dependent. It is worth noting that c_{opt} values are small for images of relatively small size while they increase as we move to images of greater size.

Additive values

Image	Size	Qual. 90	Qual. 75	Qual. 60
Ibook	200	0.4	1.2	2.3
City		0.5	1.2	2.0
Statue		0.6	1.5	2.4
Ibook	500	1.4	2.3	6.1
City		1.4	2.6	7.6
Statue		1.1	3.5	4.4
Ibook	1024	1.7	4.7	9.5
City		1.9	5.3	12.5
Statue		1.4	4.5	10.5

Table 4.3: The $c = c_{opt}$ values for watermarking image samples with respect to JPEG qualities $Q = 90$, 75 and 60.

Moving beyond the sample images in order to show the behaviour of additive value c_{opt} under different image sizes, we demonstrate in Figure 4.7 the average c_{opt} values of all the tested images grouped in three different sizes. We decided to select three representative groups for small, medium, and large image sizes, that is, 200×200, 500×500 and 1024×1024, respectively. For each size group we computed the average c_{opt} under the JPEG quality factors $Q = 90$, 75 and 60.

As the experimental results suggest the embedding process requires greater optimal values c_{opt} for the additive variable c as we get to JPEG compressions with lower qualities.

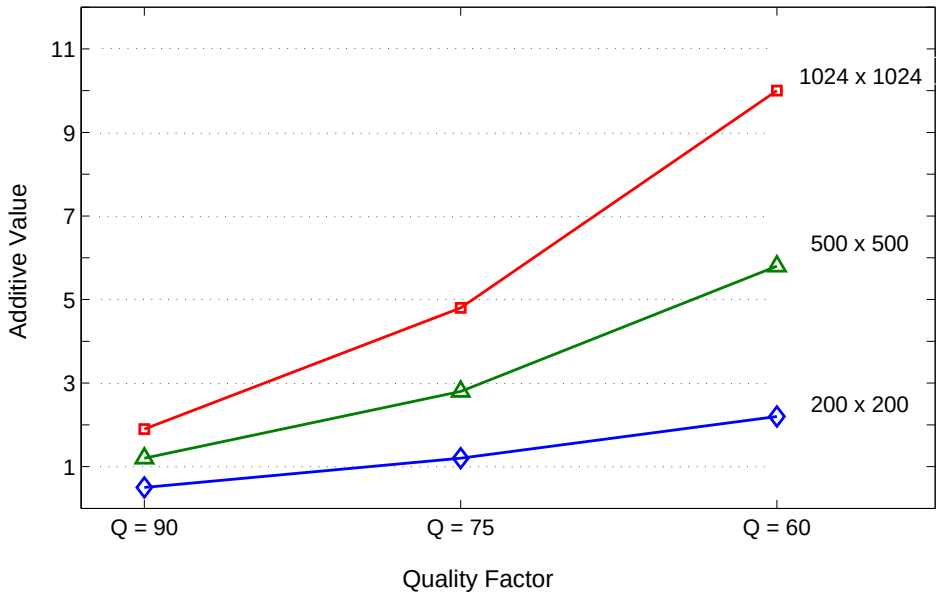


Figure 4.7: The average c_{opt} values for the tested images grouped in three deferent sizes under the JPEG quality factors $Q = 90$, 75 and 60.

The reason for that can be found looking at the three main steps of JPEG compression:

1. In the first step, the image is separated into 8×8 blocks and converted to a frequency-domain representation, using a normalized, two-dimensional discrete cosine transform (DCT) [1].

2. Then, quantization of the DCT coefficients takes place. This is done by simply dividing each component of the DCT coefficients matrix by the corresponding constant from the same sized Quantization matrix, and then rounding to the nearest integer.

3. In the third step, it's entropy coding which involves arranging the image components in a "zigzag" order employing run-length encoding (RLE) algorithm that groups similar frequencies together, inserting length coding zeros, and then using Huffman coding on what is left.

Focusing on the second step, we should point out that images with higher compression (lower quality) make use of a Quantization matrix which typically has greater values corresponding to higher frequencies meaning that information for high frequency is greatly reduced as it is less perceivable by human eye.

As we mentioned our method marks images in the higher frequency domain which means that as the compression ratio increases marks gradually become weaker and thus c_{opt} increases to strengthen the marks.

Furthermore, someone may notice that c_{opt} also increases for larger images. That is because regardless of the image size the widths of the ellipsoidal annuli remain the same

meaning that the larger the image the less frequency amplitude is covered by the constant sized annuli. That makes marks less robust and require a greater c_{opt} to strengthen them.

Frequency Domain Imperceptiveness. It is worth noting that the marks made to embed the watermark in the image are not just invisible in the image itself but they are also invisible in the image's overall Discrete Fourier Transform (DFT). More precisely, if someone suspects the existence of the watermark in the frequency domain and gets the image's DFT, it is impossible to detect something unusual. This is also demonstrated in Figure 4.8, which shows that in contrast with using the ellipsoidal marks in the whole image, using them in specific areas makes the overall DFT seem normal.

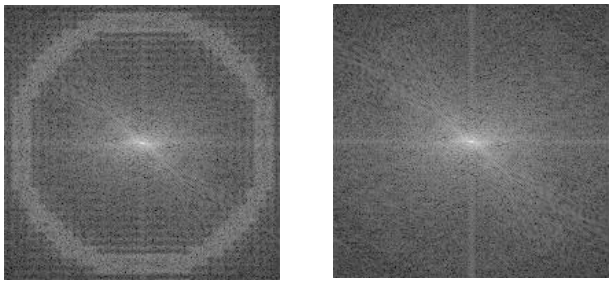


Figure 4.8: (a) The DFT of a watermarked image marked on the full image's frequency domain. (b) The DFT of a watermarked image marked partially with our technique.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we present a method for embedding invisible watermarks into images and their intention is to prove the authenticity of an image. The watermarks are given in numerical form, transformed into self-inverting permutations, and embedded into an image by partially marking the image in the frequency domain; more precisely, thanks to 2D representation of self-inverting permutations, we locate specific areas of the image and modify their magnitude of high frequency bands by adding the least possible information ensuring robustness and imperceptiveness.

We experimentally tested our embedding and extracting algorithms on color JPEG images with various and different characteristics; we obtained positive results as the watermarks were invisible, they didn't affect the images' quality and they were extractable despite the JPEG compression. In addition, the experimental results show an improvement in the frequency domain approach and they also depict the validity of our proposed codec algorithms.

It is worth noting that the proposed algorithms are robust against cropping or rotation attacks since the watermarks are in SiP form, meaning that they determine the embedding positions in specific image areas. Thus, if a part is being cropped or the image is rotated, SiP's symmetry property may allow us to reconstruct the watermark. Furthermore, our codec algorithms can also be modified in the future to get robust against scaling attacks. That can be achieved by selecting multiple widths concerning the ellipsoidal annuli depending on the size of the input image.

5.2 Future Work

Our codec algorithms can also be modified in the future to get robust against scaling attacks. That can be achieved by selecting multiple widths concerning the ellipsoidal annuli depending on the size of the input image.

It is fair to point out that although our technique is in a fully operational stage it can be later used in an graphical image watermarking software for web usage by incorporating its algorithms [84, 26, 87].

Finally, we should point out that the study of our quality function f remains a problem for further investigation; indeed, f could incorporate learning algorithms [77] so that to be able to return the c_{opt} accurately and in a very short computational time.

Bibliography

[1] N. Ahmed, T. Natarajan and K.R. Rao, Discrete cosine transform, *IEEE Transactions on Computers* C-23:1 (1974) pp. 90–93.

[2] R. Andersson, Information hiding, *Lectures In Computer Science* 1174 Springer-Verlag (1996).

[3] D. Ballard and C. Brown *Computer Vision*, Prentice-Hall, 1982.

[4] M. Barni, F. Bartolini, V. Cappellini, A. Lippi and A. Piva, A DWT-based technique for spatio-frequency masking of digital signatures, *Proc. SPIE/IS&T Int'l. Conf. of Security and Watermarking of Multimedia Contents* 3657 (1999) pp. 31–39.

[5] M. Barni, F. Bartolini, V. Cappellini and A. Piva, A DCT-domain system for robust image watermarking, *Signal Processing* 6:3 (1998) pp. 357-372.

[6] A.E. Bell, The dynamic digital disk, *IEEE Spectrum* 36:10 (1996) pp. 28–35.

[7] B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*, Oxford: Elsevier Science, 2003.

[8] G. Bouridane and M.K. Ibrahim, Digital image watermarking using balanced multiwavelets, *IEEE Transaction on Signal Processing* 54:4 (2006) pp. 1519–1536.

[9] E. Brannock, M. Weeks and R. Harrison, Watermarking with wavelets: simplicity leads to robustness, *Southeastcon IEEE* (2008) pp. 587–592.

[10] C. Carr and P.E O'Neill, Adding INSPEC to your chemical search strategy - let's get physical, *Database* 18:2 (1995) pp. 99–102.

[11] M. Chroni, A. Fylakis, and S.D. Nikolopoulos, Watermarking images in the frequency domain by exploiting self-inverting permutations, *Journal of Information Security* 4:2 (2013) pp. 80–91.

[12] M. Chroni, A. Fylakis, and S.D. Nikolopoulos, Watermarking images in the frequency domain by exploiting self-inverting permutations, *Proc. 9th Int'l Conf. on Web Information Systems and Technologies (WEBIST'13)* (2013).

[13] M. Chroni, A. Fylakis, and S.D. Nikolopoulos A Watermarking system for teaching intellectual property rights: implementation and performance, *Proc. IEEE 11th Int'l Conf. on Information Technology Based Higher Education and Training (ITHET'12)* (2012) pp. 1–8.

[14] M. Chroni, A. Fylakis, and S.D. Nikolopoulos, Watermarking images using 2D representations of self-inverting permutations, *Proc. 8th Int'l Conf. on Web Information Systems and Technologies (WEBIST'12)* (2012) pp. 380–385.

[13] M. Chroni and S.D. Nikolopoulos An Efficient Graph Codec System for Software Watermarking *Proc. IEEE 36th Int'l Conf. on Computers, Software, and Applications* (2012) pp. 595–600.

[15] M. Chroni and S.D. Nikolopoulos, Encoding watermark integers as self-inverting permutations, *Proc. 11th Int'l Conf. on Computer Systems and Technologies* (2010) pp. 125–130.

[16] L. Chun-Shien, H. Shih-Kun, S. Chwen-Jye, and M.L. Hong-Yuan, Cocktail watermarking for digital image protection, it IEEE Transactions on Multimedia 2:4 (2000) pp. 209–224.

[17] C. Collberg and J. Nagra, *Surreptitious Software*, Addison-Wesley, 2010.

[18] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, 2nd edition, MIT Press, 2001.

[19] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure spread spectrum watermarking for multimedia, *Image Processing, IEEE Transactions on* 6:12 (1997) pp. 1673–1687.

[20] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, A secure, robust watermark for multimedia, *Proc. 1st Int'l Workshop on Information Hiding* LNCS 1174 (1996) pp. 317–333.

[21] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd edition, Morgan Kaufmann, 2008.

[22] J.C. Davis, Intellectual property in cyberspace-what technological/legislative tools are necessary for building a sturdy global information infrastructure?, *Proc. IEEE Int'l Symposium on Technology and Society* (1997) pp. 66–74.

[23] G. Depovere, T. Kalker, J. Haitsma, M. Maes, L. de Strycker, P. Termont, J. Vandewege, A. Langell, C. Alm, P. Norman, G. O'Reilly, B. Howes, H. Vaanholt, R. Hintzen, P. Donnelly and A. Hudson, The VIVA project: digital watermarking for broadcast monitoring, *Proc. IEEE 2nd Int'l Conf. on Image Processing* (1999) pp. 202–205.

[24] A. De Rosa, M. Barni, F. Bartolini, V. Cappellini and A. Piva, Optimum decoding of non-additive full frame DFT watermarks, *Proc. 3rd Workshop of Information Hiding* (1999) pp. 159–171.

[25] V. Fotopoulos and A.N. Skodras, A Subband DCT approach to image watermarking, *Proc. X European Signal Processing Conference* (2000).

[26] E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[27] E. Ganic, S.D. Dexter, and A.M. Eskicioglu, Embedding multiple watermarks in the dft domain using low and high frequency bands, *Proc. Security, Steganography, and Watermarking of Multimedia Contents VII* (2005) pp. 175–184.

[28] E. Ganic and A.M. Eskicioglu Robust digital watermarking: Robust DWT-SVD domain image watermarking: embedding data in all frequencies, *Proc. of the 2004 multimedia and security workshop on Multimedia and Security* (2004) pp. 166–174.

[29] S. Garfinkel, *Web Security, Privacy and Commerce*, 2nd edition, O'Reilly, 2001.

[30] M.C. Golumbi *Graph Theory and Perfect Graphs*, 2nd edition, Elsevier, 2004

[31] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd edition, Prentice-Hall, 2007.

[32] R.C. Gonzalez, R.E. Woods, and S.L. Eddins, *Digital Image Processing using Matlab*, Prentice-Hall, 2003.

[33] D. Grover, *The Protection of Computer Software - Its Technology and Applications*, Cambridge University Press, New York, 1997.

[34] F. Hartung and M. Kutter, Multimedia watermarking techniques, *Proc. IEEE* 87:7 (1994) pp. 267–276.

[35] E.F. Hembrooke, Identification of sound and like signals, *U.S. Patent 3,004,104* (1961).

[36] J.R. Hernandez, M. Amado and F.P. Gonzalez DCT-Domain watermarking techniques for still images: Detector performance analysis and a new structure *IEEE Transactions of Image Processing* 9 (2000) pp. 55–68.

[37] A. Hore and D. Ziou, Image Quality Metrics: PSNR vs. SSIM *Proc. 20th Int'l Conf. on Pattern Recognition* (2010) pp. 2366–2369.

[38] X.Y. Huang, M.S. Tan, Y. Luo and D.Z. Lin, An image digital watermarking based on DCT in invariant wavelet domain, *IEEE 1st Int'l Conf. on Wavelet Analysis and Pattern Recognition* (2007) pp. 458–463.

[39] A. Hyvarinen, J. Karhunen and E. Oja *Independent Component Analysis*, Wiley-Interscience, 2001.

[40] K. Jain, M. Raghavan and S.K Jha, Study of the linkages between innovation and intellectual property, *Proc. Portland Int'l Conf. on Management of Engineering and Technology (PICMET'09)* (2009) pp. 1945–1953.

[41] N. Johnson and S. Katezenbeisser, A survey of steganographic techniques *Eds.Northwood, MA:Artec House* 43 (1999).

[42] N. Kaewkamnerd and K.R. Rao Multiresolution based image adaptive watermarking scheme *EUSIPCO online at www.ee.uta.edu/dip/paper/EUSIPCO_water.pdf* (2000).

[43] N. Kaewkamnerd and K.R. Rao, Wavelet based image adaptive watermarking scheme, *Proc. IEE Electronics Letters* 36 (2000) pp. 312–313.

[44] S.S. Katariya, Digital Watermarking: Review *International Journal of Engineering and Innovative Technology*, 1:2 (2012) pp. 143–153.

[45] M. Kaur, S. Jindal, and S. Behal, A Study of Digital Image Watermarking, *Journal of Research in Engineering and Applied Sciences*, 2 (2012) pp. 126–136.

[46] T. Kalker, J.P. Linnartz and G. Depovere On the reliability of detecting electronic watermarks in digital images *Proc. European Signal Processing Conference (EUSIPCO'98)* 1 (1998) pp. 13–16.

[47] E. Koch, J. Rindfrey and J. Zhao, Copyright protection for multimedia data, *Proc. Int'l Conference Digital Media and Electronic Publishing* (1994).

[48] N. Komatsu and H. Tominaga, Authentication system using concealed image in telematics, *Memoirs of the School of Science and Engineering, Waseda University* 52 (1988) pp. 45–60.

[49] D. Kundur and D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, *Proc. IEEE* (1999) pp. 1167–1180.

[50] D. Kundur and D. Hatzinakos, Towards Robust Logo Watermarking using Multiresolution Image Fusion, *IEEE Transactions on Multimedia*, 6:1 (2004) pp. 185–198.

[51] G. Langelaar, I. Setyawan and R.L. Lagendijk, Watermarking digital image and video data, *IEEE Signal Processing Magazine* 17 (2000) pp. 20–43.

[52] M.A. Lemley, Intellectual property, and free riding, *Texas Law Review* 83:1031 (2004).

[53] X. Li and X. Xue, Improved robust watermarking in DCT domain for color images, *Proc. IEEE 1st Int'l Conf. on Advanced Information Networking and Applications (IEEE-AINA'04)* (2004) pp. 53–58.

[54] V. Licks and R. Hordan", On digital image watermarking robust to geometric transformations, *Proceedings of the 3rd IEEE Int'l Conf. on Image Proceesing* (2000) pp. 690–693.

[55] C-S. Lu, S-K. Huang, C-J. Sze and H-Y. Liao A new watermarking technique for multimedia protection, *Multimedia Image and Video Processing* (2001) pp. 507–530.

[56] C-S. Lu, H-Y. Liao, H-Y., M. Huang and S-K. Sze Combined Watermarking for Images Authentication and Protection, *Proc. 1st IEEE Int'l Conf. on Multimedia and Expo* 3:30 (2000) pp. 1415–1418.

[57] C-Y. Lin, M. Wu, J.A. Bloom, I.J. Cox, M.L. Miller and T.M. Lui, Rotation, scale and translation resilient watermarking for images, *IEEE Transactions on Image Processing* 10:5 (2001) pp. 767–782.

[58] M.M. Macq and J.J. Quisquater, Cryptology for digital TV broadcasting, *Proc. IEEE* 83 (1995) pp. 944–957.

[59] P.B. Meggs and A.W. Purvis, *Meggs' History of Graphic Design*, 4th edition, Wiley, 2005.

[60] S.P. Mohanty, Digital Watermarking: A Tutorial Review, *online at: http://citeseer.ist.psu.edu/mohanty99digital.html* Unversity of South Florida (1999).

[61] M. Narasimha and A. Peterson, On the computation of the discrete cosine transform, *IEEE Transactions on Communications* 26:6 (1978) pp. 934–936.

[62] N. Nikolaidis and I. Pitas, Robust image watermarking in the spatial domain, *Signal Processing* 66:3 (1998) pp. 385–403.

[63] A. Noore, An improved digital watermarking technique for protecting JPEG images *Proc. IEEE Int'l Conf. on Electronics* (2003) pp. 222–223

[64] J.J.K. O'Ruanaidh, W.J. Dowling, and F.M. Boland, Watermarking digital images for copyright protection, *Proceedings of the IEE Vision, Image and Signal Processing* 143:4 (1996) pp. 250–256.

[65] D. Pascale *A Review of RGB Color Spaces ...from xyY to R'G'B'* The BabelColor Company, 2003.

[66] S. Pereira and T. Pun, Robust template matching for affine resistant image watermarks, *Image Processing, IEEE Transactions on* 9:6 (2000) pp. 1123–1129.

[67] F. Petitcolas, Image database for watermarking, it online at http://www.petitcolas.net/fabien/watermarking/, (2012).

[68] I. Pitas, A method for watermark casting in digital images, *IEEE Transactions on Circuits and Systems on Video Technology* 8:6 (1998) pp. 775–780.

[69] V.M. Potdar, S. Han and E. Chang, A survey of digital image watermarking techniques. *Proc. IEEE 3rd Int'l Conf. on Industrial Informatics* (2005) pp. 709–716.

[70] C.I. Podilchuk and E.J. Delp, Digital watermarking: algorithms and applications, *Signal Processing Magazine IEEE* (2001) pp. 33–46.

[71] M. Ramkumar, A.N. Akansu and A.A. Alatan A robust data hiding scheme for digital images using DFT, *Proc. IEEE ICIP* 2 (1999) pp. 211–215.

[72] M.S. Raval and P.P. Rege, Discrete wavelet transform based multiple watermarking scheme, *Conference on Convergent Technologies for Asia-Pacific Region (TENCON'03)* 3 (2003) pp.935–938.

[73] R. Raysman, E.A Pisacreta and K.A. Adler, *Intellectual Property Licensing: Dorms and Analysis*, Law Journal Press, 1999.

[74] G.B Rhoads Indentification/authentication coding method and apparatus, *World Intellectual Property Organization* IPO WO 95/14289 (1995).

[75] G.B. Rhodes, Steganography system, *U.S. Patent 5,850,481* (1998).

[76] J.J.K.O. Ruanaidh, W.J. Dowling and F.M. Borland, Phase watermarking of digital images, *Proc. IEEE Int'l Conf. Image Processing* (1996) pp. 239–242.

[77] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edition, Prentice-Hall, 2010.

[78] J.J.K.O. Ruanaidh and T. Pun, Rotation, scale and translation invariant spread spectrum digital image watermarking, *Signal Process* 66:3 (1998) pp. 303–317.

[79] V. Saxena, Digital image watermarking, *PhD thesis, Department of Computer Science and Engineering, Jaypee institute of information technology, India* (1998).

[80] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, 1996.

[81] F.Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, CRC Press, Boca Raton, FL, 2007.

[82] L.D. Smith, *online at: www.motherbedford.com.*

[83] V. Solachidis and I. Pitas, Circularly symmetric watermark embedding in 2-D DFT Domain, *IEEE Transactions on Image Processing* 10:11 (2001) pp. 1741–1753.

[84] I. Sommerville", *Software Engineering*, 9th edition, Addison-Wesley, 2010.

[85] G-M. Su, An overview of transparent and robust digital image watermarking, *online at: www.watermarkingworld.org/WMMLArchive/0504/pdf00000.pdf*

[86] P. Tao and A.M Eskicioglu, A Robust multiple watermarking scheme in the discrete wavelet transform domain, *Symposium on Internet Multimedia Management Systems V* (2004).

[87] R.N. Taylor, N. Medvidovic and E.M. Dashofy, *Software Architecture: Foundations, Theory, and Practice*, Wiley, 2009.

[88] G. Voyatzis and I. Pitas, Digital image watermarking using mixing systems, *Computer and Graphics, Elsevier* 22:4 (1998) pp. 405–416.

[89] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *Image Processing, IEEE Transactions on* 13:4 (2004) pp. 600–612.

[90] J. Wang, S. Lian, Z. Liu, Z. Ren, Y. Dai and H. Wang, Image Watermarking Scheme Based on 3-D DCT, *Proc. 1st IEEE Int'l Conf. on Industrial Electronics and Applications* (2006) pp. 1–6.

[91] A.B. Watson, DCT Quantization Matrices Visually Optimized for Individual Images, *Proc. Human Vision, Visual Processing, and Digital Display* IV (1992) pp. 202–216.

[92] R.B Wolfgang and E.J. Delp, A watermarking technique for digital imagery: Further studies, *Proc. Int'l Conf. on. Imaging Science, Systems and Technology* 3 (1997) pp. 112–118.

[93] P.W. Wong, and E.J. Delp, Security and watermarking of multimedia contents, *Society of Photo-Optical Intsrumentation Engineers* 3657 (1999).

[94] P.W. Wong, and E.J. Delp, Security and watermarking of multimedia contents II, *Society of Photo-Optical Intsrumentation Engineers* 3971 (2000).

[95] X-G. Xia, C.G Boncelet and G.R. Arce, A multiresolution watermark for digital images, *Proc. Int'l Conference on Image Processing* 1:1 (1997) pp. 548–551.

[96] W. Xiao, Z. Ji, J. Zhang and W. Wu, A watermarking algorithm based on chaotic encryption, *Proc. IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering (TENCON'02)* (2002) pp. 545–548.

[97] J.M. Zain, Strict authentication watermarking with JPEG Compression (SAW-JPEG) for medical images, *European Journal of Scientific Research* 42:2 (2010) pp. 250–256.

[98] Y. Zhao, P. Campisi and D. Kundur, Dual domain watermarking for authentication and compression of cultural heritage images *Proc. IEEE Transactions on Image Processing* 13:3 (2004) pp. 430–448.

[99] X. Zhu, Y. Gao and Y. Zhu, Image-adaptive watermarking based on perceptually shaping watermark blockwise *Proc. ACM Symposium on Information, computer and communications Security (ASIACCS'06)* (2006) pp. 175–181.

[100] W. Zhu,, Z. Xiong, Y-Q. Zhang and Y.-Q., Multiresolution Watermarking for Images and Video, *IEEE Trans. on circuit and System for Video Technology* 9:4 (1999) pp. 545–550.

Author's Publications

- **Journal Papers:**

  1. M. Chroni, A. Fylakis, and S.D. Nikolopoulos, "Watermarking Images in the Frequency Domain by Exploiting Self-Inverting Permutations", Journal of Information Security, Vol. 4, No. 2, 2013, pp. 80 - 91

- **Conference Papers:**

  1. M. Chroni, A. Fylakis, and S.D. Nikolopoulos, "Watermarking Images in the Frequency Domain by Exploiting Self-Inverting Permutations", 9th International Conference on Web Information Systems and Technologies (WEBIST'13), SciTePress Digital Library, 2013
  2. M. Chroni, A. Fylakis, and S.D. Nikolopoulos, "A Watermarking System for Teaching Intellectual Property Rights: Implementation and Performance", 11th Int'l Conference on Information Technology Based Higher Education and Training (ITHET'12), IEEE Proceedings, 2012
  3. M. Chroni, A. Fylakis, and S.D. Nikolopoulos, "Watermarking images using 2D representations of self-inverting permutations", 8th International Conference on Web Information Systems and Technologies (WEBIST'12), SciTePress Digital Library, 2012
  4. M. Chroni, A. Fylakis, and S.D. Nikolopoulos, "A watermarking system for teaching students to respect intellectual property rights", 4th Int'l Conference on Computer Supported Education (CSEDU'12), Poster paper, 2012

Short Vita

Angelos Fylakis received his B.Sc. degree in Computer Science (2011) from the Department of Computer Science of the University of Ioannina. His B.Sc. thesis was on algorithmic software watermarking techniques. He received M.Sc. degree from the Department of Computer Science and Engineering (2013) of the University of Ioannina and his M.Sc. thesis is on efficient algorithmic image watermarking techniques. During his M.Sc. studies he was part of the "Algorithms Engineering Lab" doing research on algorithmic theory and digital watermarking where he also published results. His research interests focus mainly on algorithm engineering, graph theory, information security, information hiding and image processing.