

Computer Engineering Department  
Faculty of Engineering  
Deanery of Higher Studies  
Islamic University – Gaza  
Palestine



## **Blind and Reversible Color Image Watermarking Schemes for Content Authentication and Copyright Protection**

**Nader H. H. Aldeeb**

**Supervisor**

**Prof. Ibrahim S. I. Abuhaiba**

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Computer Engineering

1433H (2012)



## نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ نادر حمد حامد الديب لنيل درجة الماجستير في كلية الهندسة / قسم هندسة الحاسوب وموضوعها:

### Blind and Reversible Color Image Watermarking Schemes for Content Authentication and Copyright Protection

وبعد المناقشة التي تمت اليوم الأحد 03 رمضان 1433هـ، الموافق 2012/07/22م الساعة العاشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. إبراهيم سليمان أبو هيبه	مشرفاً ورئيساً	أ.د. إبراهيم أبو هيبه
د. أيمن أحمد أبو سمرة	مناقشاً داخلياً	أ.د. أيمن أحمد أبو سمرة
د. فادي إبراهيم النحال	مناقشاً خارجياً	د. فادي إبراهيم النحال

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية الهندسة / قسم هندسة الحاسوب.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد الدراسات العليا  
د. فؤاد علي العاجز

## DEDICATION

*To my Father's soul, who was my source of success and inspiration, I'm proud of being his son.*

*To my beloved Mother, who supported me with endless silent prayers, and encouragement.*

*To my Wife, who fills my life with all meanings of kindness, love, peace, and empathy.*

*To the candle who enlighten my life, to my daughter JOOD.*

*To those who gave me special kind of feeling, my brothers, and sisters.*

*To all, I dedicate this work.*

## ACKNOWLEDGMENT

First of all and always I thank almighty “Allah”, who helped me with strength, faith and capability to accomplish my thesis in this way. Secondly, I owe a lot to my supervisor, Prof. Ibrahim S. I. Abuhaiba, for his continuous encouragement, guidance, support, helpful suggestions and supervision offered during my graduate studies, research, and dissertation work. This work would not have been completed without his encouragement, patience, faith, and dedication. At last but not least, I gratefully acknowledge and express my deep thanks to my dear family and beloved friends who were always with me during the hard and the good times, I am very grateful and I deeply appreciate your support, encouragement and everything you have done for me.

# TABLE OF CONTENTS

COMMITTEE DECISION	ii
DEDICATION	iii
ACKNOWLEDGMENT	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	xi
ARABIC ABSTRACT	xii
ABSTRACT	xiii
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Overview and Statement of the Problem	1
1.2 Digital Watermarking Applications	2
1.3 Objective of the Study	5
1.4 Thesis Questions	6
1.5 Methodology	8
1.6 Thesis Significance and Contribution	9
1.7 Thesis Organization	10
<b>CHAPTER 2: BACKGROUND</b>	<b>11</b>
2.1 Overview	11
2.2 Theoretical Model of Digital Watermarking System	11
2.3 Requirements (Properties) of Digital Watermarking	12
2.4 Categories (Types) of Digital Watermarking	15
2.4.1 Classification According to Robustness against Attacks	15
2.4.2 Classification According to Visibility of the Watermark	16
2.4.3 Classification According to Watermark Extraction	16
2.4.4 Classification According to the Watermark Embedding Domain	17
2.4.5 Classification According to Ability of Recovering the Original Image	17
2.5 Watermarking Techniques on which we based Our Models	18
2.5.1 Chrysochos et al. Watermarking Scheme	18
2.5.2 Poonkuntran and Rajesh Watermarking Scheme	19

2.6 Attacks on Digital Watermarking	20
2.7 Some of the Used Image Processing Filters	23
2.8 Performance Measurements of a Watermarking Algorithm	24
<b>CHAPTER 3: LITERATURE REVIEW</b>	<b>26</b>
3.1 Overview	26
3.2 Related Works	26
3.2.1 Spatial Domain Watermarking Techniques	27
3.2.2 Transform Domain Watermarking Techniques	32
<b>CHAPTER 4: PROPOSED SCHEMES</b>	<b>35</b>
4.1 The Proposed Content Authentication Watermarking Scheme	35
4.1.1 Overview	35
4.1.2 The Conceptual Model of the Proposed CA Watermarking Scheme	37
4.1.3 Messy Watermark Generation Process	38
4.1.4 The Watermark Embedding Process of the Proposed CA Watermarking Scheme	40
4.1.5 The Watermark Extraction and Image Restoration Process of the Proposed CA Watermarking Scheme	49
4.1.6 Demonstrating the Reversibility of our Proposed CA Scheme	55
4.2 The Proposed Copyright Protection Watermarking Scheme	57
4.2.1 Overview	58
4.2.2 The Conceptual Model of the Proposed CP Watermarking Scheme	58
4.2.3 The Watermark Embedding Process of the Proposed CP Watermarking Scheme	59
4.2.4 The Watermark Extraction Process of the Proposed CP Watermarking Scheme	64
4.2.5 The Restoration Process of the Proposed CP Watermarking Scheme	67
4.3 The Proposed Multipurpose Watermarking Scheme for Both CA and CP of Color Images	69
4.3.1 Overview	69
4.3.2 The Conceptual Model of the Proposed Multipurpose Watermarking Scheme	71
<b>CHAPTER 5: EXPERIMENTAL RESULTS</b>	<b>73</b>
5.1 Experimental Results of the Proposed CA Watermarking Scheme	74
5.1.1 Comparing the Embedding Capacity	74
5.1.2 Comparing the Effect of Watermark Embedding on the Host Image	75

5.1.3	Comparing Watermark Spreading and Fragility against Small Modifications	77
5.1.4	Comparing the Fragility against Attacks	80
5.1.5	Comparing the Embedding Time	81
5.1.6	Comparing the Overhead Due to Using the Location Map	82
5.2	Experimental Results of the Proposed CP Watermarking Scheme	83
5.2.1	Comparing the Embedding Capacity	83
5.2.2	Comparing the Effect of Watermark Embedding on the Host Image	84
5.2.3	Comparing the Embedding Time	86
5.2.4	Robustness Against Geometrical Attacks	87
5.3	Experimental Results of the Proposed Multipurpose Watermarking Scheme	88
5.3.1	Comparing the Effect of Watermark Embedding on the Host Image	89
5.3.2	Comparing the Embedding Time	90
5.3.3	Testing the Fragility of CA Watermarking, and the Robustness of CP Watermarking against Geometrical Attacks	91
<b>CHAPTER 6: CONCLUSION</b>		<b>94</b>
6.1	Summary and Concluding Remarks	94
6.2	Recommendations and Future Work	97
<b>REFERENCES</b>		<b>99</b>

## LIST OF TABLES

Table No.	Caption	Page
4.1	List of Bins Triple Patterns (BTPs), used in our proposed CP watermarking scheme.	61
4.2	The embedding rule of our proposed CP watermarking scheme.	62
4.3	Illustration Example, histogram of a host color plane.	63
4.4	Illustration Example, steps of embedding a binary watermark by modifying the histogram of the host color plane.	64
4.5	The extraction rule of our proposed CP watermarking scheme.	65
4.6	Illustration Example, histogram of the watermarked plane.	66
4.7	Illustration Example, steps of extracting a binary watermark based on the histogram of the watermarked color plane.	66
4.8	Illustration Example, steps for restoring the original histogram.	68
4.9	Illustration Example, recovered histogram of the watermarked plane.	68
5.1	Comparing the average embedding capacity between Poonkuntran and Rajesh scheme and our proposed CA scheme.	74
5.2	Comparing the quality, PSNR, of the watermarked image for both Poonkuntran and Rajesh scheme, and our proposed CA scheme.	76
5.3	Comparing the watermark spreading in the watermarked images using both Poonkuntran and Rajesh scheme, and our proposed CA scheme.	78
5.4	Comparing fragility against different attacks between Poonkuntran and Rajesh scheme, and our proposed CA scheme.	80
5.5	The experimental results for testing the robustness of the proposed CP watermarking scheme, against some geometric attacks.	88
5.6	The results of testing the fragility of the CA watermark and the robustness of the CP watermark of the proposed multipurpose watermarking scheme.	92



## LIST OF FIGURES

Figure No.	Caption	Page
2.1	Watermark embedding process.	11
2.2	Watermark extraction process.	12
2.3	Example of visible watermarking.	16
2.4	Example of invisible watermarking.	16
2.5	Different stages of Chrysochos et al. watermarking scheme.	19
2.6	Different stages of Poonkuntran and Rajesh watermarking scheme.	20
4.1	The conceptual model of the proposed CA watermarking scheme.	37
4.2	Full watermark generation using messy system. The green color plane is used as a reference plane.	40
4.3	Illustration example, some inputs to the embedding process of the proposed CA watermarking scheme.	44
4.4	Illustration example, the working environment for the proposed vertical embedding stage of the watermark embedding process of the proposed CA scheme.	45
4.5	Illustration example, the working environment for the proposed horizontal embedding stage of the watermark embedding process of the proposed CA scheme.	46
4.6	Illustration example, the working environment for the proposed diagonal embedding stage of the watermark embedding process of the proposed CA scheme.	48
4.7	Example demonstrating watermark verification, image restoration, and tamper detection.	57
4.8	The conceptual model of the proposed CP watermarking scheme.	59
4.9	The conceptual model of the proposed multipurpose watermarking scheme.	71
5.1	A set of randomly selected samples, for the purpose of demonstrations.	73
5.2	Comparing the quality of the watermarked image for both Poonkuntran and Rajesh scheme, and our proposed CA scheme, after being embedded with a large watermark size.	76
5.3	Example demonstrating the benefit of watermark spreading, in detecting image modifications.	80
5.4	Comparing average watermark embedding time, between Poonkuntran and Rajesh watermarking scheme, and our proposed CA scheme; based on different watermark sizes.	81
5.5	Comparing the average embedding capacity between Chrysochos et al. scheme, and our proposed CP scheme.	84
5.6	Comparing the quality of the watermarked image for both Chrysochos et al. scheme, and our proposed CP scheme.	85
5.7	Comparing watermark embedding time between Chrysochos et al. scheme, and our proposed CP scheme.	86
5.8	Binary watermark (Size 20X20).	87

5.9	Comparing the quality, PSNR, of the watermarked images using the developed, and the non developed multipurpose watermarking schemes.	89
5.10	Comparing the time required to embed both the CA and the CP watermarks using the developed, and the non developed multipurpose watermarking schemes.	91
5.11	Illustration example, demonstrates the situation when an image is maliciously attacked.	93

## LIST OF ABBREVIATIONS

<b>CA</b>	Content Authentication
<b>CP</b>	Copyright Protection
<b>IT</b>	Integer Transform
<b>IIT</b>	Inverse Integer Transform
<b>PSNR</b>	Peak Signal to Noise Ratio
<b>NCC</b>	Normalized Cross Correlation
<b>EC</b>	Embedding capacity
<b>MSE</b>	Mean Square Error
<b>ROI</b>	Regions of Interest
<b>RONI</b>	Regions of Non Interest
<b>LSB</b>	Least Significant Bit
<b>DCT</b>	Discrete Cosine Transform
<b>DWT</b>	Discrete Wavelet Transform
<b>SVD</b>	Singular Value Decomposition
<b>DFT</b>	Discrete Fourier Transform
<b>CCI</b>	Copy Control Information
<b>HVS</b>	Human Visual System
<b>BTP</b>	Bits Triple Pattern
<b>BCP</b>	Bits Couple Pattern

## طرق عمياء قابلة للعكس للتعليم المائي على الصور الملونة للتحقق من سلامة المحتوى وحماية حقوق النسخ

نادر حمد حامد الديب

### الملخص

يتمثل الإسهام الرئيس لهذه الرسالة في الكشف عن أي تعديل، أو تزوير، أو تلاعب غير شرعي في الصور، بالإضافة إلى التمييز بين الهجمات الخبيثة والتعديلات العرضية للوصول إلى هذا الهدف، تم اقتراح ثلاث طرق. ولقد تم تقييم الطرق المقترحة الثلاثة بالاعتماد على مجموعة بيانات تحتوي على 1050 صورة ملونة مقسمة إلى إحدى عشرة صنف. ولقد أجريت كل الاختبارات باستخدام جهاز كمبيوتر محمول بمعالج ثنائي النواة وبسرعة (2GHz Core 2 Duo)، وذاكرة (2 GB)، وذاكرة محول عرض (384 MB)، ولقد تم استخدام الأدوات Visual studio 2011 و Matlab 7.8 في تنفيذ الطرق المقترحة ونظائرها.

الطريقة المقترحة الأولى؛ هي طريقة العلامّة المائيّة للتحقق من سلامة المحتوى، والتي تقوم بالتعليم الضعيف والقابل للإزالة بدون الحاجة للصورة الأصلية؛ بهدف الكشف عن التلاعب في الصور الملونة. فهي تولد العلامّة بشكل فريد لكل صورة باستخدام نموذج مُشْتَبِت، ويتم تضمين تلك العلامّة بشكل تراكمي؛ لضمان انتشارها على مساحة الصورة ككل، وبشكل متجانس؛ لضمان الحصول على صورة مُعلّمة ذات جودة عالية. الطريقة المقترحة هي تطوير لطريقة علامة مائية أُقترحت مؤخراً، ولقد تفوقت الطريقة المقترحة على نظيرتها من حيث؛ السعة، والجودة، ومدى انتشار العلامّة، وهشاشة العلامّة، والوقت اللازم للتعليم. حيث تمت زيادة نسبة الحمل في الصورة المُضيفة من 81.71% إلى 93.82%، وتمت زيادة أقل جودة (PSNR) للصورة المُعلّمة من 27.15 ديسيبل إلى 31.76 ديسيبل، وتمت زيادة نسبة انتشار العلامّة، أو النسبة المَحْمِيَة من ال (Pixels)، بشكل ملحوظ، والطريقة المُقترحة حساسة جداً إلى التعديلات في كل مكان على الصورة؛ حتى لو كانت صغيرة، وأخيراً، إن طريقتنا المقترحة أسرع من نظيرتها في التعليم، لقد حصلنا على معدل انخفاض في الوقت يساوي 15 جزء من الثانية.

الطريقة المقترحة الثانية؛ هي طريقة العلامّة المائيّة لحماية حقوق النسخ، والتي تقوم بالتعليم القوي والقابل للإزالة بدون الحاجة للصورة الأصلية؛ بهدف حماية حقوق نسخ الصور الملونة ضد الهجمات الهندسية. وهذه الطريقة المقترحة هي أيضاً تطوير على طريقة علامّة مائيّة أخرى موجودة. يعتمد تضمين العلامّة في هذه الطريقة، كما هو أيضاً في نظيرتها، بشكل أساسي على تبديل درجات المُدرج التكراري. ولكننا نقدم قاعدة تضمين جديدة، والتي بدورها أدت إلى زيادة سعة الحمل بحوالي 55 بت، وأيضاً أدت إلى زيادة جودة الصورة المُعلّمة من 35.01 ديسيبل إلى 38.05 ديسيبل. أيضاً، أظهرت العلامّة المُضمنة نسبة تحمل تساوي 100% ضد مجموعة متنوعة من الهجمات الهندسية مثل: التقليل بجميع أنواعه (أفقي، رأسي، كلاهما)، والتدوير بالزوايا ( $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ )، والبُعْثْرَة، واللَّفْت، والحرف، وتَوَلِيْفَاتْهَا. وأخيراً، لقد أظهرت طريقة حماية حقوق النسخ المقترحة عملية تضمين للعلامّة المائيّة أسرع من التي في نظيرتها بمعدل انخفاض في الوقت يساوي 4.84 ثانية.

الطريقة المقترحة الثالثة؛ هي طريقة العلامّة المائيّة متعددة الأهداف للتحقق من سلامة محتوى، ولحماية حقوق نسخ الصور الملونة، وهي أيضاً تقوم بالتأشير القابل للإزالة بدون الحاجة للصور الأصلية. وهي تجمع بين طريقتي العلامّة المائيّة المقترحة الأولى والثانية، كوسيلة لتحقيق الإسهام الرئيس لهذه الرسالة. وقد أظهرت نتائج التجارب أنه على الرغم من تضمين علامتين مائيتين في نفس الصورة المُستضيفة؛ الصور المُؤشّرة الناتجة من استخدام هذه الطريقة تتميز بقيمة جودة أعلى من 30 ديسيبل. لقد تم الحصول على معدل جودة 33.25 ديسيبل.

# Blind and Reversible Color Image Watermarking Schemes for Content Authentication and Copyright Protection

Nader H. H. Aldeeb

## ABSTRACT

The main contribution of this thesis is to detect any modification, forgery, or illegal manipulation in images, as well as distinguishing between malicious attacks and incidental manipulations. To achieve this goal, three schemes are proposed. The three proposed schemes are tested based on a dataset contains 1050 colored images divided into eleven categories. All tests are performed using a laptop with a 2 GHz core 2 duo processors, 2 GB memory, and 384 MB display adapter. Matlab 7.8 and Visual Studio 2011 are used in implementing the proposed schemes and counterparts.

The first proposed scheme is the Content Authentication (CA) watermarking scheme, which is a fragile, blind, and reversible watermarking scheme; aimed at detecting manipulation in color images. It generates the watermark uniquely, using a messy model. The generated watermark is embedded accumulatively; to obtain spreading over whole image's area, and homogeneously; to obtain a high quality watermarked image. Our proposed scheme is a development of a recently proposed watermarking scheme. Our proposed scheme surpassed its counterpart in terms of capacity, quality, watermark spreading, fragility, and embedding time. The payload of the host image increased from 81.71 % to 93.82 %. The minimum obtained PSNR value is increased from 27.15 dB to 31.76 dB. The watermark spreading percentage, or the percentage of the protected pixels, is noticeably increased. Our proposed scheme is very sensitive to modifications anywhere at the image, even if it is tiny. Finally, our proposed CA scheme is faster than its counterpart in embedding. We obtained an average reduction in time equals, 0.15 second.

The second proposed scheme is the Copyright Protection (CP) watermarking scheme, which is a robust, blind, and reversible watermarking scheme; aimed at protecting the copyright of color images against geometrical attacks. It is also a development of an already existing watermarking scheme. Watermark embedding in this scheme, as well as in its counterpart, mainly depends on the permutation of histogram bins. But, we present a new embedding rule, which increased the average capacity by about 55 bits, and also it increased the quality, PSNR, of the watermarked image from 35.01 dB to 38.05 dB. The embedded watermark demonstrates 100 % robustness against a variety of geometrical attacks, like Flipping (H, V, and Both), Rotation (90°, 180°, and 270°), Scattering, Warping, Skewing, and their combinations. Finally, our proposed CP watermarking scheme showed a faster watermark embedding process than that of its counterpart by an average reduction in time equals, 4.84 seconds.

The third proposed scheme is a blind and reversible multipurpose watermarking scheme, for both CA and CP of color images. It combines the first and the second proposed watermarking schemes, in a way, to fulfill the main contribution of this thesis. Experimental results showed that; despite of embedding two watermarks at the same host image, the generated watermarked images using this scheme are still having PSNR values of higher than 30 dB, the obtained average PSNR is 33.25 dB.

**Keywords:** *Content Authentication, Copyright Protection, Forgery Detection, Reversible, Blind, Geometrical Attacks, Messy, Accumulative, Homogeneous.*

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview and Statement of the Problem

A huge amount of digital information is moving around the world by means of the rapid growth in Internet technology and digital media. Digital media offers several distinct advantages, such as high quality, easy editing, and high fidelity copying. This ease, by which digital information can be manipulated and duplicated, has made publishers, authors, artists, and photographers afraid that their innovations and products will be modified illegally or claimed by others. For example, the contents of medical images, as well as the information related to it, such as patient's name, age, and initial diagnosing are usually assumed as sensitive data. Illegal manipulation of medical images, or its related information, could lead to dangerous problems, like false diagnoses of a specific disease. This is absolutely not accepted in medical applications. Thereby, healthcare institutions are ethically obliged to give up those maliciously modified images, because they are no longer suitable for taking serious decisions. Therefore, a technique for verifying content's integrity of digital media is needed. But, some images might be modified incidentally during transmission in a way, which allows its reuse. Examples of such modifications are: Flipping, Rotation, Warping, and other geometrical modifications, those only change pixels positions. Thus, it is also required to decide whether the modified images are still being usable or not.

Digital watermarking is a method of hiding information (watermark) into a host (cover) signal (image, audio, or video), so that the watermark can be detected or extracted later to make an assertion about the cover signal, for the purpose of Content Authentication (CA) [1], Copyright Protection (CP) [2], secure transmission of forensic questioned documents [3], content description, copy control, secret communication [4], and etc. Generally, the effective watermarking scheme should satisfy certain requirements to be reliable, such as invisibility, imperceptibility, unambiguity, low complexity, and either robustness, or fragility based on the intended watermarking application [5]. Also, digital watermarks should be difficult to remove

or modify without damaging the host signal. Digital watermarking is potentially useful in many modern applications like E-health and Telemedicine [4, 6].

## **1.2 Digital Watermarking Applications**

Digital watermarking techniques were initially used for limited intents, now it becomes a well-defined science, with its own resourceful schemes. Presently, it is the core of many modern applications. To name a few amongst its innumerable applications, digital watermarking is employed in: Copyright Protection [7], Content Authentication [8], Fingerprinting [9], Telemedicine or e-health [10], Copy Prevention or Control [4], Content Description [11], Secret Communication [12], and ID Card Security [13]. In the following subsections, we will try to describe each of these applications briefly.

### **☒ Copyright Protection (CP)**

Because traditional copyright notices, such as “©”, “date”, and “owner” are easily removed from the digital content when it is copied; and because copyright notices may cover important portions of the image; watermarking of digital images is used for CP instead. CP is one of the main applications of watermarking. The owner of the image’s content can be identified by using a hidden object, which is imprinted into the image. Owner’s own watermark is usually used as the hidden object. The watermark here, allows content’s owners to trace their contents and to detect unauthorized use or duplications of it.

Without watermarking, there would be no way to extend the control of the content’s owner, once his content leaves the protected digital domain or released to a different user [6].

### **☒ Content Authentication (CA)**

Nowadays, due to the advent of multimedia technology; multimedia contents, like image, video, audio, text, and graphics can easily be manipulated in a way, such that it is very difficult to detect what has been altered. Traditional watermarking techniques may be not adequate here, as they were not able to distinguish between malicious attacks and incidental manipulations [14, 15, and 16]. Therefore, a different intent of watermarking arises, namely, CA, in which watermarking is used by content owners and distributors for authentication and assuring integrity of the multimedia content. They embed digital watermarks directly into the digital contents. Later, the originality

of the digital content is verified by checking its extracted watermark. Authentication of multimedia content with a reasonable imperceptibility and high detection resolution is the challenge of today's research in the field of multimedia security [14].

### ☒ **Fingerprinting**

It is the application where multimedia content is electronically distributed over a network, and the content owner would like to prevent unauthorized duplication or distribution of his content. Here, digital watermarking is used in a way to trace the source of illegal copies [17]. In fingerprinting, the hidden message (watermark) is variable and depends on the recipient identity [18]. For example, if the owner of the content needs to send his content to multiple recipients; he may embed different watermarks, fingerprints, in the copies of the digital content, such that each embedded watermark is customized for each recipient. Later, if an unauthorized copy of the content is found, e.g. supplied to third parties, the origin of that copy can be determined by retrieving its associated fingerprint. Therefore, the watermark should be resistant to collusion. That is, a group of users have the same images, but containing different fingerprints, should not be able to interact and create a copy without any fingerprint. Thereby; they prevent the owner from detecting the origin of his image, when it is distributed illegally.

### ☒ **Telemedicine**

In the beginning of twentieth century people used the term e-health, which refers to the investment of modern data and communication technologies, in a way to meet the needs of citizens, patients, healthcare professionals, healthcare providers and policy makers [19]. Telemedicine, which is one form of the e-health applications, defined as the delivery of remote health care and medicine by the use of telecommunication and computer technologies together with medical expertise. Telemedicine technology may involve the use of computers, sound, video, and image processing [20].

Applications like telemedicine are promising; they can play a very important role in the range of provision of services, through connecting healthcare facilities and healthcare professionals, as well as improving services access, equity, and quality. By telemedicine, the geographical and physical limits are vanished [21].

With the rapid development in telemedicine systems, and according to the sensitivity and privacy of the information related to medical images, (patient's name, age, initial diagnosing, etc.), the security of the medical images, as well as its related information



becomes more and more important. Therefore, healthcare institutions are forced to take appropriate measures in order to maintain privacy of the patient information in the image. Also, the institutions must assure data integrity, which prevents others from tampering the image, when they mistakenly receive a copy of patient's medical images [22]. Watermarking provide us a good solution for these problems, by securely and imperceptibly embedding that sensitive information in patient's medical images. Later, the authorized recipients only can extract the hidden information. Also, watermarking has the benefit of disabling others from modifying, neither medical images, nor patient's information that is included in those images.

### ☒ **Copy Prevention or Control**

Copy control aims at preventing people from making illegal copies of a copyrighted content [23]. Copy control of the content could be achieved by inserting a watermark within it; the watermark can later be detected by a recording device. If a watermark is detected, the recorder will recognize it, then decide copying or not. Of course, for such a system to work, all manufactured recorders must include watermark detection circuitry [24].

In this application of watermarking, the watermark includes the information called Copy Control Information (CCI). This information refers to the rules, which the owner wishes to enforce about copying the content. Some examples are: "Copy Not Allowed", "Never Copy", "One Copy Allowed", and "Copy Not Restricted". Clearly, the embedded watermark must be robust against attacks [4].

### ☒ **Content Description**

In some systems, where automated retrieval of contents is performed, it is needed to attach descriptive information, such as content's name, title, size, and creation date along with the content to facilitate automatic indexing. This information usually attached to the content as a separate digital file; unfortunately, this makes it vulnerable to attacks. But when using watermarking, the vulnerability to attacks vanishes, because by watermarking; the information can be embedded hidden, and inseparable from the content [11].

### ☒ **Secret Communication**

Along with the demand for speed and integrity while exchanging information over the Internet, there is always a need for secrecy. Cryptography is increasingly used for

secure communication, where data is encrypted using security key, which is shared between participants [25]. Watermarking is also increasingly used for secure communication, where data is hidden inside the carrier media, so that the hidden data is transferred without drawing attentions.

### ☒ ID Card Security

Another application of watermarking is the ID card security, where the main personal information is not only written on the document, but also included in the person's photo, that appears on the document. Later, the ID card can be verified by extracting the embedded information and comparing it to the written text. By this application, a forged copy of the ID card, where the photo is replaced or the written text is modified, will be detected by the failure in extracting the watermark that matches the written text [6].

## 1.3 Objective of the Study

The primary objective of our proposed system in this thesis is to find a multipurpose watermarking scheme, for both CA and CP of color images. The scheme mainly aims at detecting any modification, which might infected the image, as well as deciding whether or not the modified images are still usable. These features are required in many applications. For example, the ability of detecting modifications at patient's medical images will prevent others from tampering images, when they mistakenly receive a copy of it. Also, this is supposed to end the exhausting problem of misdiagnosing, due to dependence on erroneous images.

As stated above, our algorithm is expected to be used in sensitive applications, like military and medical applications. This makes our algorithm different from any other available multipurpose watermarking algorithms in the literature, because those sensitive applications require some additional requirements. For example, in telemedicine systems, where medical images are shared between specialists for remote diagnosing, a lot of attention must be paid to protect the copyright of the patient's medical images, and to assure their content's integrity. Hence, most of previously designed watermarking techniques may no longer be sufficient for nowadays modern applications. Thus, our proposed algorithm must satisfy some special requirements.

## 1.4 Thesis Questions

From the intensive study, and the detailed analysis of the previously proposed image watermarking works, we may draw the following remarks and questions:

### **1. *Are the previous watermarking techniques well enough?***

It is well known that there are many watermarking algorithms in the literature. But in principle, with the huge advancements in both digital media and Internet technology, most of the previously designed watermarking techniques may no longer be sufficient now, as many modern applications have appeared. For example, consider the modern application, telemedicine, in which medical images are exchanged from one place to another, for remote diagnosis purposes. In such applications, any loss in the overall image's quality is not accepted. Hence, more attention must be paid when designing any new watermarking algorithm to be used in such modern applications. Thereby, in light of the aforementioned question, we propose a novel watermarking algorithm, to be suitable when used in nowadays applications.

### **2. *Can we guarantee both CP and CA of images using only spatial domain techniques?***

In spatial domain watermarking, the watermark is embedded directly by modifying the pixel values of the host image, without applying any transform to it. By contrast, in transform domain watermarking, the host image is first transformed from spatial to other domain, such as Discrete Cosine Transform (DCT), then the watermark is embedded by applying a specific watermarking algorithm, and finally the watermarked image is inverse-transformed back to the spatial domain [26]. Although transform domain watermarking can yield higher payload capacity limit and more robustness against attacks, its computational cost is higher than that of spatial domain watermarking. Accordingly, embedding the watermark in spatial domain component of the original image is a straightforward method. It has the advantage of low complexity, and easy implementation [27]. So, for obtaining high performance in our proposed scheme, we will answer the this question to test whether or not it is possible to find a dual watermarking algorithm that embeds two watermarks in spatial domain of the host image.

### **3. *Since robustness and transparency are the most important watermarking properties, the question is how and where to place the watermark, while keeping both of these requirements?***

Usually the embedding domain contains perceptually most significant components (low frequency components), and perceptually insignificant components (high frequency components). Such perceptibility is related to the Human Visual System (HVS). If we embed the watermark in the perceptually most significant components, our algorithm will be robust against attacks, but the watermark may be difficult to hide. On the other hand, if the watermark is embedded in the perceptually insignificant components the opposite occurs [28, 29]. Therefore, in removing this ambiguity, it is important to answer aforementioned question.

***4. What are the mechanisms that we should use, to guarantee removing the effect of the second embedding process on the information previously hidden in the first embedding process; thereby we claim a new meaning for “dual watermarking”?***

Some people use dual watermarking, where the term dual watermarking is often used in the literature referring to the use of different watermarks embedded in the same host signal, to serve different aims. For example, one watermark could be used for tamper detection in the host image, and the other could be used for the purpose of owner identification of the same image. Dual watermarking schemes, usually embed watermarks one after the other. But all these approaches have a problem in common, which is that the later embedding process affects the watermark embedded in the former one [30]! This question searches for new watermarking mechanisms, those guarantee removing the effect of each embedding stage on the other.

The main issues considered in answering this question are: Firstly, does the first embedded watermark have an effect on the second watermark, which will be embedded later? Secondly, what is the effect of the later embedded watermark on the former embedded one? Hence, we focus in determining the effects of each watermarking stage on the other. This will help in removing these effects, if it exists.

***5. What are the embedding strategies, which we should follow in order to preserve quality of the host image, while embedding two watermarks in it?***

It is important to answer this question, such that the degree of distortion introduced at the host image during watermark embedding process is inversely proportional to Peak Signal to Noise Ratio (PSNR). As the embedding capacity increases, the PSNR decreases [31]. In sensitive watermarking applications, any loss in the quality of images is not accepted.

**6. Which is the preferred mechanism and watermark embedding order in dual watermarking?**

In dual watermarking, according to the purpose of each individual watermark, whether it is for CA, CP or other, the mechanisms followed in embedding the watermarks differ. Some schemes embed the second watermark in the first one before being embedded in the host image [32]. In other schemes, the two watermarks are embedded into different resolution layers, or different embedding regions of the host image. The embedding regions and resolution layers are chosen carefully to achieve fragility in the first watermark, while achieving robustness for the second one [33]. The answer of the aforementioned question, will find the correct watermarks embedding order in our proposed watermarking scheme.

**7. Is it required to use a meaningful watermark for both CA and CP in dual watermarking?**

Actually, the watermark is a special symbolized image or text. Usually, watermarks are meaningful like a logo, sign, symbol, signature, or an animated image. Naturally, for the watermark, the increase in its meaning leads to an increase in its size.

**8. How effective our proposed algorithm will be, compared to recently available ones?**

In the literature, many effective watermarking algorithms have been proposed and implemented for digital images. But these algorithms may not keep their effectiveness when used in today's sensitive applications like, medical and military applications. Anyway, it is important for any new designed watermarking algorithm to be tested, evaluated, and compared to other previously generated ones. The answer of this question will clarify the position of our proposed schemes among others. It is evident that in answering this question, we will search for, and use good performance evaluation measurements.

## **1.5 Methodology**

One approach to get a multipurpose image watermarking scheme for both CA and CP is by cascading two available watermarking systems one after another at the same host image, the first system performs CP and the second performs CA, or vice versa.

Cascading is our promising solution to find a multipurpose watermarking scheme, but it is not that simple at all, the difficulty is summarized in answering the aforementioned questions, in Section 1.4.

A variety of excellent works can be found in the literature, some of them are specialized for CP, and others are specialized for CA of images. An intensive study and detailed analysis of the most promising proposed algorithms are performed, to find which of the proposed schemes can be adapted to meet our proposed algorithm requirements. According to their attractive properties (Secure, Blind, and Reversible); two algorithms are chosen, they are Poonkuntran and Rajesh algorithm [1] for CA, and Chrysochos et al. algorithm [34] for CP. Then, under the light of the previously mentioned questions, drawbacks and shortcomings of each algorithm are recorded. Finally, to achieve the proposed scheme, our work is divided into three phases. The first is CA phase, in which Poonkuntran and Rajesh algorithm [1] is implemented and improved, in order to meet our algorithm properties without any drawbacks. The second is CP phase, in which Chrysochos et al. algorithm [34] is implemented and improved such that it meets our algorithm properties without any drawbacks. The last phase is the multipurpose phase, in which we designed and implemented a new watermarking scheme that combines between the resulted schemes of the two previous phases; it utilizes the benefits of both Poonkuntran and Rajesh scheme, and Chrysochos et al. scheme. Thus, using different techniques and strategies, a new multipurpose watermarking algorithm is generated. The generated algorithm allows watermark embedding in a host image, and watermark extraction from a watermarked image, while allowing reversibility to fully restore the original host image.

## **1.6 Thesis Significance and Contribution**

This thesis adopts digital watermarking as an alternative solution for protection. Digital watermarking allows people to share their own digital images safely; it relates the digital images with their owners robustly, and it also protects images from illegal modifications. Thus, it protects the intellectual properties of content owners, even if the contents are shared along wide distances. Therefore, by the services provided in digital watermarking, people now can trust to do business electronically, because they do not worry about forgery.

This thesis discusses the potential of digital watermarking, and then provides valuable services in the range of sensitive information management. We provide new models through more flexibility in content use. Our main aim, as researchers, is to make a genuine contribution in the range of sensitive applications, by providing some data management issues like:

- a) More flexibility in content use.
- b) Promotion of protection of sensitive data.
- c) Accuracy in localizing images penetration.
- d) Relevant and Physician's authentication.
- e) Image integrity control, prevention of illegal modification.
- f) Noticeable increase at the payload of the cover image, with no degradation.
- g) Proficient image archiving and retrieval.

Although there are different alternatives, our proposed system suggests a new idea for protecting digital images, and solving the exhausting problems of forgery, illegal manipulation, and illegal distribution of digital contents.

## **1.7 Thesis Organization**

The rest of this thesis is organized as follows: Chapter 2 introduces watermarking concepts; definitions, models, properties, classifications, examples, techniques, and performance measurements. Chapter 3 introduces literature review, and related works. Chapter 4 leads to the proposed watermarking schemes. It gives a clear definition of the workflow of each proposed scheme, by means of flow charts, illustration examples, figures, and algorithms. Our proposed schemes experimental results, evaluations, interpretations, justifications, and analysis are presented in chapter 5. In chapter 6 we summarize our work, review of our concluding remarks, presenting recommendations, and give directions for future work.

## CHAPTER 2

### BACKGROUND

#### 2.1 Overview

Digital watermarking is today's modern solution considered by researchers for preventing unauthorized use, by embedding information, digital watermark, into the intended digital data. Later, in case of multiple claims, that embedded watermark plays the judge role. The owner of the original data proves his ownership by extracting his previously embedded watermark from the watermarked content (CP). Also the embedded data can effectively be used in protecting data contents from illegal manipulation (CA). Usually, the embedded information could be a logo, a meaningful message, or a random signal [35-37].

#### 2.2 Theoretical Model of Digital Watermarking System

Digital watermarking usually divided into two main processes: watermark embedding process and watermark extraction process.

Watermark embedding process, embeds the watermark in the host signal. The watermark and the host signal are the inputs of this process. The watermarked signal is the output of this process. Suppose that the host signal is an image,  $I$ , and we have a watermark,  $W$ , then, the result of the embedding function,  $E$ , is the watermarked image,  $I'$ , as shown mathematically in Equation 2.1 below.

$$E(I, W) = I' \quad (2.1)$$

In some watermarking systems, a security key is used as additional input for the embedding process; it adds a level of security to the watermarking process, and makes the watermark more robust against attacks. Figure 2.1 shows watermark embedding process, which is adopted from [38].

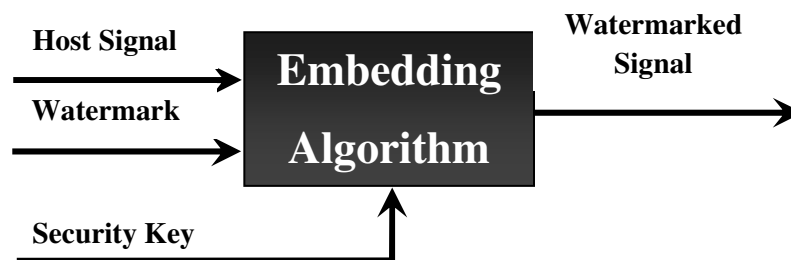


Figure 2.1: Watermark embedding process.



Watermark extraction process, extracts the watermark from the watermarked signal. The watermarked signal is the basic input of this process, the output of this process is the extracted watermark. Suppose we have a watermarked image,  $I'$ , from which we want to extract the watermark, the result of the detector, extractor, function,  $D$ , is the extracted watermark,  $W_e$ , as shown mathematically in Equation 2.2 bellow.

$$D(I') = W_e \quad (2.2)$$

If the embedding process used a security key, same key is needed in the extraction process; the extraction process is exactly the reverse of the embedding process.

Other inputs may be needed, based on which type of watermarking algorithms is applied. For example, in the CA and CP systems, the original watermark is also needed as an input to the extraction process for comparing with the extracted one, in order to determine whether the host signal is authentic or not. Comparison is performed using a comparator function,  $C_\delta$ , which is based on the correlation,  $C$ , of the two watermarks, and the threshold,  $\delta$ . The comparison process can be represented mathematically as shown in Equation 2.3 bellow.

$$C_\delta(W, W_e) = \begin{cases} 1, & C \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

Where,  $C$  is the correlation of  $W$  and  $W_e$ , it is calculated as shown in Equation 2.4.

$$C = \frac{\sum_i \sum_j (W(i,j) - M_W)(W_e(i,j) - M_{W_e})}{\sqrt{\sum_i \sum_j (W(i,j) - M_W)^2} \sqrt{\sum_i \sum_j (W_e(i,j) - M_{W_e})^2}} \quad (2.4)$$

Where,  $M_W$  and  $M_{W_e}$  are the mean values of the original and the extracted watermark respectively.

Finally, Figure 2.2 shows watermark extraction process, which is adopted from [38].

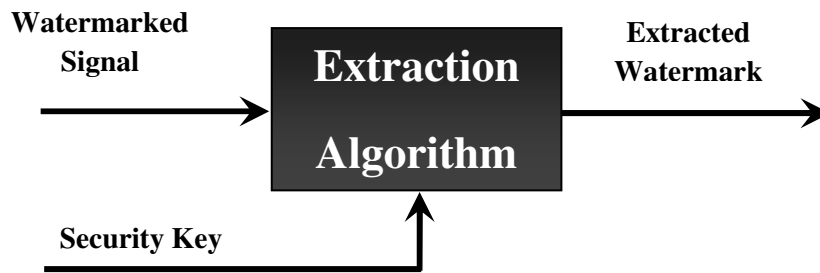


Figure 2.2: Watermark extraction process.

### 2.3 Requirements (Properties) of Digital Watermarking

A reliable and effective watermarking system should satisfy some requirements, the relative importance of these requirements depends on the intended application. Some

of the requirements are robustness, tamper resistance, fidelity, computational cost, and false positive rate. Practically, it is impossible for our watermarking system to satisfy all these properties, but instead, it is necessary to make tradeoffs between them based on the underlying application [39].

#### ❖ **Robustness**

Robustness of digital watermarking is the requirement that the watermark, which is embedded in the digital content, can still be detected intact even if the digital content has been modified by common signal processing operations or geometric distortions. Otherwise, it is a fragile watermark. Usually, a robust watermark must survive common signal processing only between the time of embedding and the time of detection [24]. Also, robustness is the requirement that the embedded watermark is hard to be detected, removed, or replaced illegally without obvious degradation in the quality of the content. Ideally, the degree of image distortions needed to remove the watermark should degrade the desired image quality to the point of becoming commercially useless. Robustness is often thought of as a single-dimensional value, but this is incorrect. A watermark that is robust against one process may be very fragile against another. In many applications, robustness to all possible processing is excessive and unnecessary [39]. The requirement that the watermark is robust can differ slightly from one application to another. Also, not all applications of watermarking demand all sorts of robustness [4]. Robust watermarking is mainly designed to resist malicious and non-malicious attacks, such as flipping, rotating, scaling, cropping, lossy compression, and others. Therefore, robust watermarking is mainly used in CP applications. But, on contrast, fragile watermarking is found to detect any small or large modification in the watermarked digital content, so it is mainly used in CA applications [40].

#### ❖ **Transparency (Imperceptibility or Fidelity)**

Transparency or imperceptibility of the watermark is that the watermark is not visible and does not affect the overall visual quality of the watermarked content. In other words, the watermark is neither visible by human eyes nor affects the carrier fidelity [41]. Some few applications require visible watermarks, but most others do not, this is why transparency is assumed as one of the basic requirements of digital watermarking [42].

Embedding the watermark in the digital content should not affect its quality. The fidelity of a watermarking system refers to the perceptual similarity between the original and watermarked versions of the digital content. It is clear that, the higher the fidelity, the better the watermarking algorithm.

#### ❖ **Capacity or Payload**

Capacity or payload of the watermarking system refers to the size of the watermark that the watermarking algorithm can embed within the digital content. Unfortunately, the higher capacity is usually obtained at the expense of either robustness, or imperceptibility, or both [11]. Therefore, it is necessary to make tradeoffs between these three properties based on the underlying application. In some applications, mildly perceptible watermarks are accepted in exchange for higher robustness or lower cost.

This requirement, payload, is highly dependent on three factors: first; the host medium, second; the intended application, third; the aimed resulted quality. Therefore, during the design of the watermarking scheme, an attention must be given to what minimum amount of information we need to embed. This to make sure that the host signal can carry that amount, while preserving imperceptibility [43].

#### ❖ **Embedding Effectiveness**

Embedding effectiveness of a watermarking system refers to the probability of detecting the embedded watermark immediately after embedding. This definition implies that a watermarking system might have an effectiveness of less than 100%, but always effectiveness of 100% is desirable [23].

#### ❖ **Computational Cost**

The requirement of having a fast embedding and extraction processes is based on the intended application. For example, the speed of detecting the watermark is not of much importance in applications of ownership verification. In such applications, even if it takes days to find the watermark, it is going to be worth waiting [39]. But when the intended application needs to run in a real time basis, speed becomes one of the most important factors [44]. Watermarking techniques are used mostly in real time applications; thus, low complexity requirements, and hence, low computational cost requirements, are mandatory to guarantee system efficiency and short time delay. A low complexity algorithm ensures that both the watermark embedding and extraction

processes are simple, and does not require too much time or computation [5]. With the presence of nowadays technology, computational cost is no longer being a problem.

#### ❖ **Interoperability**

The watermarked content shall still be interoperable and exchangeable, so that it can be seamlessly accessed through heterogeneous networks, and can be played on various play-out devices, whether those devices are watermark aware or not [45].

#### ❖ **False Positive Rate**

False Positive, is the term given to the situation when watermark detection is performed in a piece of media that is actually does not contain a watermark. False positive rate is the number of false positive expected to occur in a given number of runs of the detector. In other words, the false positive rate of a watermark detection system is the probability that it identifies an un-watermarked piece of data as a watermarked one [39].

## **2.4 Categories (Types) of Digital Watermarking**

Classification of digital watermarking can be made according to the ability of the watermark to resist attacks, visibility of the watermark, how the watermark is extracted, the domain in which the watermark is embedded, or according to the ability of recovering the original image [46].

### **2.4.1 Classification According to Robustness against Attacks**

Due to watermark's ability to resist attacks, there are two types of watermarking; namely, robust and fragile [1, 6].

- **Robust Watermark:** It is resistant to a specific image processing methods, and these watermarks can be extracted from the watermarked images, even if the watermarked images are heavily attacked by such methods. Since in CP, the embedded watermark should survive against attacks [1], robust watermark is used for CP of images.
- **Fragile Watermark:** it cannot resist attacks, and it is easily destroyed when the host image is modified. Since in CA, the embedded watermark should be sensitive to the attacks [1], fragile watermark is used for CA. It can provide information for image completeness [6]. Thus, if integrity of the host image has to be ensured, a fragile watermark would be applied.

## 2.4.2 Classification According to Visibility of the Watermark

- **Visible Watermarking:** Visible watermarking is the most primitive way of watermarking. A visible watermark is usually embedded in the host content in a way, such that the watermark is detectable and noticeable by Human Visual System (HVS). Figure 2.3 shows an example of visible watermarking.

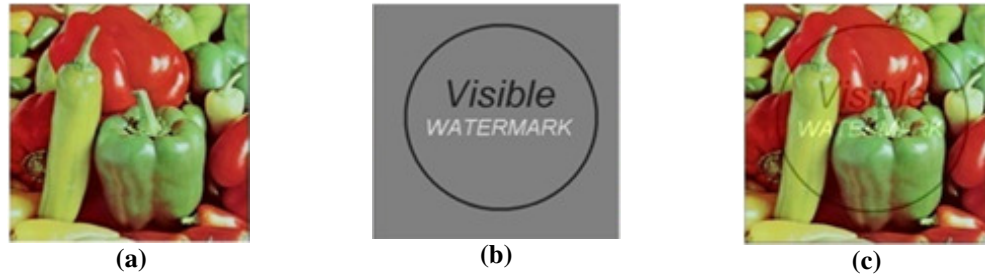


Figure 2.3: Example of visible watermarking. (a) Original Image. (b) Visible Watermark. (c) Watermarked image.

There are at least two disadvantages of visible watermarking:

- Visible watermark degrades the visual quality of the host content.
  - Visible watermark is not difficult to be removed [11].
- **Invisible Watermarking:** The watermark is called invisible if it is embedded in the digital content without making obvious degradation in it; or that is not visible to most, but can be detected under specific conditions. In invisible watermarking, it is difficult to distinguish between the original and watermarked content. The main advantage of invisible watermarking is that, it is difficult to remove or destroy the embedded watermark. Figure 2.4 shows an example of invisible watermarking.

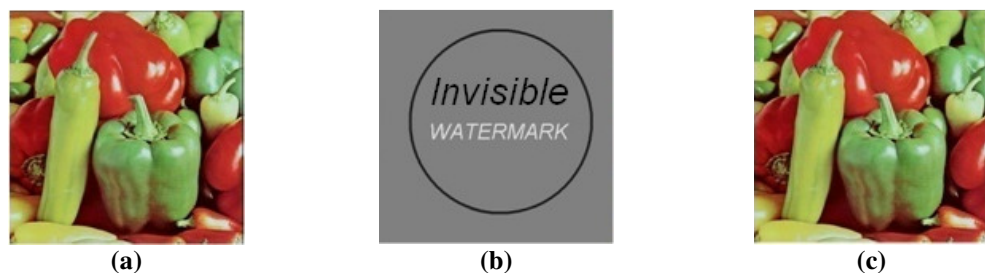


Figure 2.4: Example of invisible watermarking. (a) Original Image. (b) Invisible Watermark. (c) Watermarked image.

## 2.4.3 Classification According to Watermark Extraction [4, 6]

Any digital watermarking algorithm is either, non-blind, semi-blind, or blind; based on what type of information is required during watermark extraction stage.

- **Non-Blind:** In Non-blind techniques, the watermark can be extracted by the aid of the original image and the secret key(s). In other words, the watermark can only

be detected by those who have a copy of the original image and the secret key(s). It guarantees better robustness, but may lead to multiple claims of ownerships.

- **Semi-Blind:** In Semi-Blind techniques, the watermark can be extracted by the aid of secret key(s) and the watermark bit sequence. Sometimes, part of original image may be used in these techniques.
- **Blind:** In Blind techniques, the watermark detection and extraction depends on neither the availability of the original image nor on the availability of the watermark. The drawback in this type of watermarking is that, when the watermarked image is seriously altered, the watermark detection, as well as the original image recovery; will become very difficult.

#### **2.4.4 Classification According to the Watermark Embedding Domain**

- **Spatial Domain Watermarking:** In spatial domain schemes, the watermark is embedded directly by modifying the pixel values of the host image without applying any transform to it. Because in these schemes watermarks are often fragile and can be used to detect tamper in the host image, spatial domain watermarking is often used for the purpose of CA. These schemes are computationally less complex than transform domain schemes [47].
- **Transform Domain Watermarking:** In transform domain schemes, the watermark is embedded by first applying a certain transform to the host image, such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Singular Value Decomposition (SVD), or other. The watermark is then embedded by altering certain coefficients in the transform domain [4]. These schemes are more robust to attacks, when compared to spatial domain schemes [48]; therefore, transform domain watermarking is often used for the purpose of CP and proof of ownership.

#### **2.4.5 Classification According to Ability of Recovering the Original Image [49-52]**

- **Reversible Watermarking:** Reversible watermarking is a digital watermarking with the additional feature that, when the watermarked content found to be authentic, i.e, the extracted watermark is correct, the watermark can be removed completely to retrieve the original, un-watermarked, content. Such reversibility is highly desirable in sensitive applications.

- **Non-Reversible Watermarking:** By contrast, the non-reversible watermarking is a digital watermarking that cannot restore the watermarked content to its original, un-watermarked, form.

## 2.5 Watermarking Techniques on which we based Our Models

In this section we will discuss, briefly, the details of the used techniques in building our proposed methods.

### 2.5.1 Chrysochos et al. Watermarking Scheme [34]

It is a new reversible watermarking scheme, used to embed a binary watermark in a gray scale image. Watermark extraction process of this scheme does not need the original image. Watermark embedding in this scheme mainly based on the modification of the histogram of the host image. In other words, Embedding in this scheme based on the permutation of the histogram bins. Two keys are used in this scheme; the first key is called public key, which is a real number that specifies the watermark embedding area. This key is also needed for the watermark detection and extraction. The second key is called the private key, which is used for the full restoration of the original image, from the watermarked one, after being verified. The public key is the most important parameter needed to embed the watermark into the host image. The integer part of this key is called *start*; it indicates the embedding starting point in the histogram of the host image. The decimal part of this key, multiplied by ten, is called *step*, which defines the minimum distance a couple of histogram bins may have. Each watermark bit,  $w$ , is embedded by first locating a couple  $(a, b)$  of intensity values, chosen according to *start* and *step* values. The two intensity values,  $a$  and  $b$ , are chosen, such that the corresponding histogram bins,  $hist(a)$  and  $hist(b)$ , are not equal. If the watermark bit,  $w$ , equals *zero*, then the histogram values,  $hist(a)$  and  $hist(b)$ , are forced to be in ascending order; otherwise, they are forced to be in descending order, as seen in the following rule:

$$(w=0) \rightarrow hist(a) < hist(b) \quad (2.5)$$

$$(w=1) \rightarrow hist(a) > hist(b) \quad (2.6)$$

In the watermark extraction stage, again, the same public key is used to locate the intensity pairs,  $(a, b)$ , and for each pair the value of the watermark bit,  $w$ , is determined according to the rule:

$$hist(a) < hist(b) \rightarrow (w=0) \quad (2.7)$$



$$hist(a) > hist(b) \rightarrow (w=1) \quad (2.8)$$

The private key, used for the full restoration of the watermarked image, is generated during the embedding process. For each selected intensity pair,  $(a, b)$ , chosen for embedding, a bit ( $pk$ ) of the private key is generated. The value of  $pk$  is set to zero if  $hist(a)$  and  $hist(b)$  are originally in ascending order, otherwise  $pk$  is set to one. Later, to recover the original image, the steps were used in watermark extraction, are followed to get the intensity pairs  $(a, b)$  one after another. At each selected pair, the value of the corresponding private key bit,  $pk$ , is checked, if  $pk$  equals zero then  $hist(a)$  and  $hist(b)$  should be in ascending order, otherwise, they should be in descending order. Thus, the histogram of the host image, original image, is recovered to its original state. Watermark embedding and extraction stages of Chrysochos et al. algorithm are shown in Figure 2.5, (a) and (b) respectively.

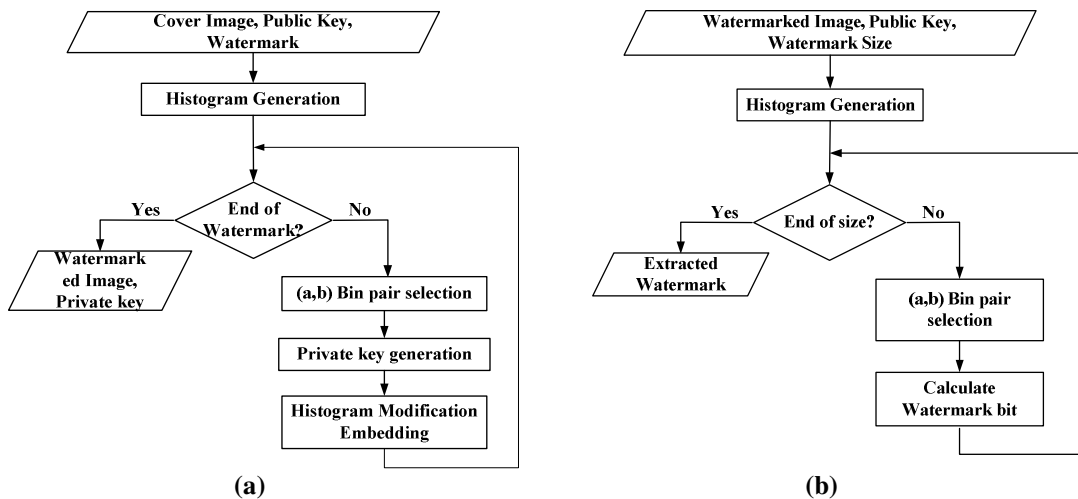


Figure 2.5: Different stages of Chrysochos et al. watermarking scheme [34]. (a) Watermark Embedding. (b) Watermark Extraction.

## 2.5.2 Poonkuntran and Rajesh Watermarking Scheme [1]

It is a watermarking scheme proposed for the authentication of color medical images. In this scheme, the watermark is generated dynamically using messy models. The generated watermark is embedded inside the host image by expanding the difference between any two color planes of it, in a method called intra plane difference expansion, which was used in [53]. In this scheme, Poonkuntran and Rajesh scheme, the watermark is generated using a hybrid bi stable messy system, which was used in [54], based on the Green color plane, as a seed to the messy system. In the watermark embedding process, each bit is embedded by expanding the difference between a corresponding pixel pair from Red and Blue color planes. Difference expansion is



performed through generating a new difference by appending the value of the bit to the binary representation of the old difference, such that the bit becomes the new Least Significant Bit (LSB). The new generated difference is used to generate a new pixels pair. But before difference expansion is performed, this scheme checks whether it is possible to expand that difference or not. This check is used to guarantee that the new generated difference will not lead new pixel values, which are outside the gray scale, [0,255]. A location map is used in this scheme, to refer to the locations where embedding takes place. The location map is an image of binary pixels. Each pixel in the location map is set to one, when the corresponding difference is expandable; otherwise, that pixel is set to zero. In watermark extraction, that location map is checked to locate pixel pairs, those were previously embedded. The LSB of the difference of each located pair gives the corresponding embedded watermark bit. Watermark embedding and extraction processes of Poonkuntran and Rajesh algorithm, are shown in Figure 2.6, (a) and (b) respectively.

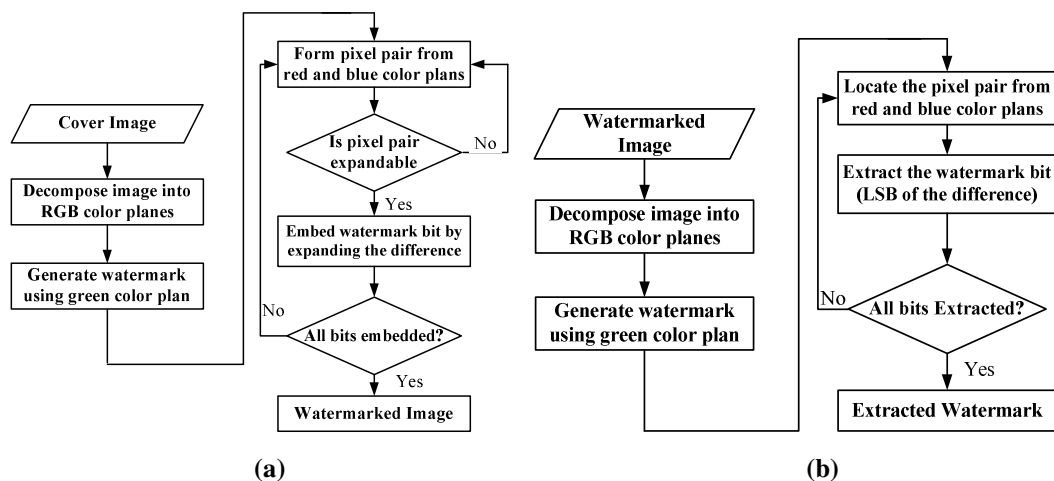


Figure 2.6: Different stages of Poonkuntran and Rajesh watermarking scheme [1]. (a) Watermark Embedding. (b) Watermark Extraction.

## 2.6 Attacks on Digital Watermarking

Like any other circulating digital media, watermarked content may be processed in some way before it reaches the receiver. Such processing modifies the content either intentionally or non-intentionally. For intentional modifications, the aim of the modifier is to hinder watermark reception, while non-intentional modifications are those occur accidentally during natural working environment.

In watermarking terminology, an attack is any processing that may impair either watermark detection or communication of information conveyed by the watermark.

Therefore, any processed watermarked data is called attacked data and the processing itself is called an attack. Some of the most common attack categories are given below [55]:

### ☒ Removal Attacks

The main aim of these attacks is the complete removal of the watermark information from the watermarked content, without the need to crack the security of the watermarking algorithm, i.e., without using the security key, which was used in embedding. After the content being attacked, it is impossible to recover the watermark information. This category includes de-noising, quantization, re-modulation, and collusion attacks. The complete removal of the watermark information may not be accomplished by some of these attacks; however, they may damage the watermark information significantly. Generally, these attacks try some optimization methods to impair the embedded watermark, while keeping the quality of the attacked content high enough. The higher the attacked document quality, the more sophisticated the removal attack [55]. For instance, low-pass filtering, does not introduce considerable degradation in watermarked document, but can dramatically damage the embedded watermark [56].

### ☒ Geometrical Attacks

Geometric attacks mainly based on the geometric transformations, which aims at modifying the spatial relationships between pixels of an image. For example, suppose that an image  $f$  with pixel coordinates  $(x, y)$  and width  $w$  undergoes geometric attack to produce an image  $g$  with coordinates  $(x', y')$ . this attack can be expressed as  $x' = r(x, y)$ , and  $y' = s(x, y)$ , where  $r(x, y)$  and  $s(x, y)$  are the spatial transformations that produced the geometrically attacked image,  $g(x', y')$ . Some of these transformations may lead an image  $g$  with histogram that is different than that of  $f$ . While others, those change only pixels positions; generate an attacked image  $g$  having same histogram as that of  $f$ . For example, if  $r(x, y) = x/2$  and  $s(x, y) = y/2$ , the attack is simply shrinking the size of the image by one half in both spatial directions. This of course changes the histogram of the original image, because the number of pixels is reduced by shrinking. But, if  $r(x, y) = |x-w|$  and  $s(x, y) = y$ , the attack is simply flipping the image horizontally, which does not affect its histogram [57]. Based on such transformations, geometric attacks aim to distort the synchronization between the embedded watermark and its detector. After the attack takes place, the detector can only recover

the embedded watermark when perfect resynchronization is regained, but usually the resynchronization process is very complex and too difficult to be practical. Some of the attacks fall under this category: rotation, translation, scaling, cropping, etc. [38]. The following are some of the geometrical attacks used in this thesis [58, 59]:

- ✓ **Flipping:** Flipping is the transformation in which a plane figure is flipped or reflected across a line, creating a mirror image of the original figure. The line across which the figure is reflected is called the line of reflection or axis of reflection. If the line of reflection is the x-axis, then it is horizontal flipping, on contrast, if the line of reflection is the y-axis, then it is a vertical flipping.
- ✓ **Rotation:** Rotation is the transformation which turns a figure about a fixed point. The fixed point around which the figure is rotated is called as centre of rotation. The degree to which the figure is rotated is called as angle of rotation.
- ✓ **Scattering:** Scattering is the transformation which aims at separating selected regions of the image and swapping them with other regions of the same image, in different directions.
- ✓ **Warping:** Image warping is, in essence, a transformation that defines how each point in the source image should be translated to produce the warped image, thus it changes the spatial configuration of an image. Using this definition a simple displacement of an image by, for example, five pixels in the x-direction would be considered a warp.
- ✓ **Skewing:** Skewing is the act of making a rectangular image slanted. That is to make it fit into a parallelogram instead of a rectangle.

#### ☒ **Cryptographic Attacks**

These attacks aim at cracking the security methods those were employed in watermarking schemes. One example of these attacks is brute-force search; it tries to find the embedded secret information. Another attack falls under this category is the so-called oracle attack. When the watermark detector device is available, oracle attack can be used to create a non-watermarked signal from the watermarked one. Due to their high computational complexity, the application of these attacks is limited [55].

#### ☒ **Protocol Attacks**

These attacks aim at attacking the entire concept of the watermarking application. Based on the idea of invertible watermarks, the attacker might embed his own watermark in the restored data and then he claims to be the owner of the data by

extracting his own watermark from it. This creates ambiguity with respect to the true owner of the data. The solution to this problem is to use a non invertible watermarking technology. This solution is important in CP applications [55].

## 2.7 Some of the Used Image Processing Filters [60-62]

In image processing, filters are mainly used to suppress either the high frequencies in the image, i.e. smoothing the image, or the low frequencies, i.e. enhancing or detecting edges in the image. Some of the used image filters are listed below:

- **Random Jitter:** It performs a displacement at each pixel position by a random amount, which is given according to an arbitrary distribution. Each pixel displacement is followed by an interpolation over the modified sampling grid.
- **Average Filter:** It is usually used for smoothing images, i.e. reducing the amount of intensity variation between one pixel and the next. The idea of this filter is simply to replace each pixel value in an image with the mean, average, value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings.
- **Disk Filter:** It is a circular averaging filter, which is based on the average filter.
- **Motion Filter:** It is a smoothing filter, and approximates the linear motion of a camera. It can be achieved by blurring in only one direction.
- **Gaussian Filter:** It is a 2-D convolution operator that is used to blur images and remove detail and noise. In this sense, it is similar to the average filter, but it uses a different kernel that represents the shape of a Gaussian, bell-shaped, hump.
- **Laplacian Filter:** It is a 2-D measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change, and is therefore often used for edge detection.
- **Log Filter:** It is the Laplacian of Gaussian filter, where the Laplacian filter is applied to an image that has first been smoothed using a Gaussian smoothing filter, in order to reduce its sensitivity to noise.
- **Prewitt Filter and Sobel Filter:** Both are used for edge detection. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. It is based on convolving the image

with a small, separable, and integer valued filter in horizontal and vertical direction.

- **Un-sharp Filter:** It is a simple sharpening operator, which derives its name from the fact that it enhances edges and other high frequency components in an image via a procedure, which subtracts a smoothed version of an image from the original image.

Note that, the strength of each of the aforementioned filters is determined by its size. The size of the filter usually takes a value from the discrete interval  $[0, max]$ , where the filter of size=0 has no effect on the target image, and the filter of size =max is the filter which destroys the target image completely. We can determine percentage of destruction at the filtered image by comparing it to the original un-filtered image.

## 2.8 Performance Measurements of a Watermarking Algorithm

The following are some of the measurements used to measure the performance of any proposed watermarking scheme.

### a) Normalized Cross Correlation (NCC).

NCC is an important performance parameter in any extracting module. Sometimes, it is needed to have a robust watermarking algorithm. Robustness means to have, approximately, undistorted extracted watermark, even if the watermarked image is subjected to attacks. The NCC used to verify the robustness of the watermarking systems, by expressing the comparability between extracted watermark and original watermark quantitatively [63]. NCC is defined as in Equation 2.9 bellow [64].

$$NCC = \frac{\sum_x \sum_y W(x,y)W'(x,y)}{\sqrt{(\sum_x \sum_y [W(x,y)^2]).(\sum_x \sum_y [W'(x,y)^2])}} \quad (2.9)$$

Where,  $W(x, y)$ ,  $W'(x, y)$  are the original watermark image and the extracted watermark image respectively.  $NCC$  is a value between 0 and 1. The larger the  $NCC$  value, the higher the watermark robustness.

### b) Embedding capacity (EC)

It is a measure to determine the ratio of information that can be embedded in the host image; it is defined in Equation (2.10) bellow:

$$EC = \frac{Ne}{N} \quad (2.10)$$

Where,  $N$  and  $N_e$  denote the total number of pixels, and the total number of embedded pixels, respectively.

**c) Mean Square Error (MSE):**

It is one of the simplest functions used to measure the distance between the host image and its watermarked version. Suppose we have an image  $I$  of size  $M \times N$ , and its watermarked version is  $I'$ . The  $MSE$  is defined in Equation (2.11) below:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [(R(i,j) - R'(i,j))^2 + (G(i,j) - G'(i,j))^2 + (B(i,j) - B'(i,j))^2] \quad (2.11)$$

Where,  $R(i,j)$ ,  $G(i,j)$ ,  $B(i,j)$ ,  $R'(i,j)$ ,  $G'(i,j)$ , and  $B'(i,j)$  are the pixels located at  $i^{th}$  column and  $j^{th}$  row of the host image  $I$  and watermarked image  $I'$  of Red, Green, and Blue components, respectively.

**d) Peak Signal to Noise Ratio (PSNR)**

PSNR is used to measure how much the watermarked version of an image is similar to the original image. Suppose we have an image  $I$ , its watermarked version  $I'$ , and the  $MSE$  values of  $R$ ,  $G$ , and  $B$  components are  $MSE_R$ ,  $MSE_G$ ,  $MSE_B$ , respectively. The  $PSNR$  is defined as shown in Equation 2.12 below:

$$PSNR \text{ (dB)} = 10 \log_{10} \frac{\max I^2}{(MSE_R + MSE_G + MSE_B)/3} \quad (2.12)$$

Where,  $\max I$  is the maximum pixel value of the original image. Internationally,  $PSNR$  is measured in decibel units (dB). And the bigger the  $PSNR$  value is, the better the watermark conceals [63]. In general, the processed image is acceptable to the human eyes if its  $PSNR$  is greater than 30 dB [64]. At that level, the processed or watermarked image will be visually very close to the original image.

## CHAPTER 3

### LITERATURE REVIEW

#### 3.1 Overview

The issue of secretly communicating is as old as communication itself. Steganography methods first introduced, and described in documents by Herodotus [65]. Steganography, “Covered Message”, means hiding secret messages within another carrier [66]. Watermarking is based on the science of steganography. It is a method of hiding information, known as watermark, into a host, known as cover, multimedia object such as image, audio, or video [1]. As seen, both watermarking and steganography are alike in seeking to embed information into a carrier. The main difference is in the intent of use. In watermarking, the issue of concern is the copyright, content protection, or license of the carrier. But in steganography, the issue of concern is the security of the embedded message rather than the carrier.

Almost, the use of watermarks appeared simultaneously with the making of handmade paper, nearly 720 years ago, since finding the oldest watermarked paper, dates back to 1292 in Italy, which was considered as the birthplace of watermarks [67]. Traditional watermarking techniques were based on placing a visible mark in a paper to ensure its originality. Such techniques are available in official documents, money bills, and stamps. Nowadays, digital world expanded this early concept of watermarks to include digital watermarking. Digital image watermarking arises at 1990 by Tanaka et al. [68]. The principle in digital watermarking is similar to the principle of traditional paper watermarking. In digital watermarking both the watermark and the paper are now digitally represented. One simple example of digital watermarking would be a visible company logo sealed over a digital image for the purpose of copyright control. Digital watermarking has been introduced as a complementary protection technology of encryption. Encryption alone, on which protection of digital content has relied for a long time, is not sufficient enough to protect digital data all along its lifetime [69].

#### 3.2 Related Works

According to the domain in which the watermark is embedded, digital image watermarking techniques are classified into either spatial domain watermarking techniques, or transform domain watermarking techniques.

### 3.2.1 Spatial Domain Watermarking Techniques

In spatial domain watermarking techniques, the watermark is embedded directly by modifying intensities or color values of pixels in the host image.

One of the simplest spatial domain watermarking schemes is that of Schyndel et al. [70], in which digital watermarking is applied to 512x512 gray scale images. It embeds the watermark in the LSBs of some randomly selected pixels. The watermark is imperceptible to human eyes, but it can easily be destroyed if some image processing techniques, such as averaging, is applied.

Coatrieux et al. [71] presented a retina images watermarking algorithm, as an application of medical image watermarking. The watermark insertion process adds or subtracts at most one gray level to or from the pixels in the cover image. And hence, they claim that their algorithm has no distortions on the watermarked image. But Coatrieux et al. algorithm has some inherent shortcomings, additive watermark insertion is not robust against image attacks, and transparency is also not enough high.

Wang et al. [72], proposed a multipurpose watermarking scheme for both CP and CA of color images and videos. They embed a robust watermark, used for CP, and a reversible fragile watermark, used for CA, into different color components of a color image, or video original frame. The robust watermark is embedded in the block's mean values of the green component of the color image, and then the digest (MD5 hash value) of the whole copyright-watermarked image is calculated, to be used as fragile authentication watermark, which is reversibly embedded in the LSBs of blue component blocks of the copyright-watermarked image. Here, reversible embedding of the fragile watermark is used to eliminate its influence on the robust watermark, which is embedded before. Thus, in the extraction process, the bits of authentication (digest) are first extracted and then, the original copyright-watermarked image is perfectly recovered in order to generate the original digest, which is compared with the extracted one. If, by making a match, it is proved that the image is not tampered, the copyright watermark is then extracted and verified.

But, the payload capacity limit of the copyright watermark in this scheme is very low. For example, according to the division shown in their scheme, if we have an image of size 512x512, it is divided into 8x8 sub-blocks division. Then, the mean of each sub-block is calculated. Thus, now we get a matrix of 64x64 mean values. Now each 8x8 sized value group is embedded by 1 bit; thus we maximally can embed 8x8 sized



binary watermark bits in the whole image. That is, 64 bits, which is very small to represent a copyright watermark. In our proposed work, we will allow larger copyright watermarks to be used.

Wang et al. embed both watermarks, a robust watermark for CP and a fragile watermark for CA, in spatial domain. Since transform domain methods are more robust against attacks than in spatial domain methods as stated in [73], it is a good choice by Wang et al. to place the CA watermark in spatial domain to get watermark fragility, but the copyright watermark should be embedded by a way that guarantee that the watermark will resist attacks. This issue will be addressed on our work.

Also, Wang et al. embed the copyright watermark in the block's mean values of the green component of the color image irreversibly. But this will distort the overall behavior of the image permanently. This distortion might not be acceptable in applications working with sensitive imagery, like medical imaging [74]. By investigating some works in the literature, we found that the algorithms those embed the watermark based on, quantization like in Kundur and Hatzinakos [75], bit-replacement like in Memon and Gilani [76], and truncation like in Karras [77], the original image could not be recovered from its watermarked version. In our work, we will find a completely reversible watermarking algorithm, despite of embedding two watermarks in the same host image.

Liew and Zain [74], proposed a reversible watermarking scheme for tamper detection and recovery of ultrasound grayscale images. Due to the characteristics of ultrasound images, they used four rectangles organized as a pyramid, to locate Regions of Interest (ROI). The rest of regions are assumed as Regions of Non Interest (RONI). In watermark embedding, bits are embedded in a way by replacing the LSBs of some ROI blocks with the watermark bits. The LSBs were replaced in watermark embedding process are stored in RONI, for the reason of restoration of ROI to its original state. They used a mapping sequence locating the blocks from where bits are removed and where bits are stored in.

From our point of view, the used mechanism by Liew and Zain for the reversibility of original bits of ROI is inappropriate. By any small attack at the RONI the stored bits will be lost. Also, along with the limitation of their scheme to grayscale images, their proposed scheme is dedicated to images with a pyramid shaped ROI, precisely, ultrasound grayscale images.

Tian [53], proposed a reversible watermarking method for gray scale images, which is based on difference expansion. In Tian's work, the difference between each neighboring pixel values of the image is calculated. Some of the calculated differences are selected for difference expansion and watermark bit embedding. Only expandable differences are selected, to avoid both overflow and underflow. The pixel's pairing could be horizontally, vertically, etc. If it is found that the difference  $h$  is expandable, it is replaced by a new difference  $h'=2h+bit$ , where  $bit$  is the watermark bit to be embedded. New pixel pair values are then calculated based on the new difference,  $h'$ . After embedding, a location map of all expanded differences is created, to be used as a guide in the watermark extraction stage. The size of the location map is equal to the number of differences, or pixel pairs. A value of "1" is assigned in the location map to correspond to an expandable difference. Otherwise a value of "0" is assigned.

Li et al. [78], described a reversible watermark embedding algorithm for CP of tongue color images, which is based on the prediction-error of the calculated four neighbor's context prediction for both **Red** and **Green** components. To achieve reversibility, they used the same procedure followed by Tian [53]. The only difference is that, now they pass the prediction error of **Red** and **Green** components as input for difference expansion, rather than passing the values of the neighboring pixels. Also, they extend Tian's difference expansion to deal with negative values.

From our point of view, the main drawback in Tian [53], and Li et al. [78] is that extra information needs to be embedded other than the watermark, that is map of location, which is used as an indicator in the watermark extraction process, to indicate where the watermark is embedded. Additional drawbacks in Li et al. [78] are: The watermark is embedded in a selected square area; same area is required to be selected in the extraction process. This requires that, the location and the dimensions of that square area must be published either by embedding or by transmission to the extractor. This will increase the payload and complexity of the algorithm. Also, Li et al. algorithm exploits the high correlation that is inherent among the neighboring pixels of tongue images, in order to guarantee accurate prediction of pixel based on its neighbors, and to achieve small prediction-error. Thus, Li et al. algorithm could not be used for other image types.

Chrysochos et al. [34], presented a reversible watermarking scheme, aimed at embedding a binary watermark into a gray scale image. Watermark embedding in this

scheme is mainly depends on the permutation of the histogram bins of the host image. Two keys are needed in this scheme. The first key is called public key, which is a real number used to specify the watermark embedding area at the histogram of the host image. The second key is called private key, which is used for the restoration of the image. The integer part of the public key is called *start*, and it indicates the embedding starting point in the histogram of the host image. The decimal part of this key, multiplied by ten, is called *step*, which defines the minimum distance a couple of histogram bins may have. Each watermark bit,  $w$ , is embedded by first locating a couple  $(a, b)$  of intensity values, chosen according to *start* and *step* values, such that the corresponding histogram values,  $hist(a)$  and  $hist(b)$ , are not equal. If the watermark bit,  $w$ , equals *zero*, then the histogram values,  $hist(a)$  and  $hist(b)$ , are forced to be in ascending order, otherwise they are forced to be in descending order. In the watermark extraction stage, the public key is used to locate the same couple,  $(a, b)$ , then, according to the values of  $hist(a)$  and  $hist(b)$ , the previously embedded bit,  $w$ , is extracted. The private key is generated during the embedding process. For each intensity pair,  $(a, b)$ , chosen for embedding, a bit,  $pk$ , of the private key is generated. The value of  $pk$  is set to zero if  $hist(a)$  and  $hist(b)$  are originally in ascending order, otherwise  $pk$  is set to one. The authors of this scheme claim that their scheme shows robustness against a variety of geometrical attacks.

Unfortunately, the maximum payload capacity of this scheme is very low. At the best case, where all histogram bin pairs are assumed as candidates for embedding, the payload capacity is 128 bits. If the scheme is applied for color images, in a way, such that each color component carries a portion of the watermark bits, the maximum payload capacity is 384 bits.

Yalman and Erturk [79], proposed a new histogram modification based data hiding technique, which modifies the histogram of the cover image for data hiding. The data hiding in their proposed scheme is based on the number of iterations ( $IN$ ) of each brightness value ( $BV$ ) of the cover image. Their scheme mainly depends on the arithmetic modulo operator. After determining the histogram of the cover image, the lowest and the highest  $BVs$  are determined and named, Lower Limit Value ( $LLV$ ), and Upper Limit Value ( $ULV$ ) respectively. These two values specify the area where the data is going to be embedded. Each histogram's  $IN$  is used to embed a watermark *bit*. For each *bit* to be embedded, the modulo2 of the corresponding  $IN$  is checked if it equals the value of the *bit*, the *bit* is assumed embedded successfully. Otherwise one

pixel of the image with the corresponding *BV* to that *IN* is changed to the next following *BV*, to enforce both the value of the *bit*, and the value of modulo2 of the corresponding *IN* are equal. For example, suppose we have 3 bits (**010**) to be embedded, and the histogram of the cover image has 3 brightness values *BVs* [58, 59, and 60], and a corresponding numbers of iterations *INs*, [20, 24, and 27]. Therefore, *LLV*=58, and *ULV*=60. At the first step the algorithm takes the first *bit* “0”, and the *IN* correspond to the *LLV*, which is 20. Now because both (**20 mod 2**) and *bit* are equal, then the *bit* is assumed as embedded successfully. At the second step, because the modulo2 of the following *IN*, and the second *bit* are not equal, ( $1 \neq (24 \bmod 2)$ ), one pixel of the image whose *BV* is 59 is changed to the next *BV*, which is 60. And hence, *INs* changed from (24, 27) to (23, 28). Here we actually enforced the second *bit* and the corresponding (***IN mod 2***) to have equal values. Thereby, the second *bit* is now embedded successfully. This process continues until reaching *ULV*.

The maximum embedding capacity of this scheme is 256 bits, provided that the image histogram is uniformly distributed from “0” to “255”, and hence, the 256 bins are assumed as candidates for embedding. In addition, the maximum payload capacity is 768 bits if the scheme is applied for color images, in a way such that each color plane carries a portion of the watermark bits. Therefore, the embedding capacity of Yalman and Erturk scheme is approximately twice the embedding capacity of the aforementioned, Chrysochos et al. scheme [34]. This is due to the embedding strategy followed in Yalman and Erturk scheme, in which each histogram bin is embedded by a watermark bit. But in Chrysochos et al. scheme, each non-equal couple of bins is embedded by a watermark bit. But unfortunately, the scheme of Yalman and Erturk is not reversible, also from our point of view, we see that their scheme is not that robust, because any change to a pixel value will lead to a different corresponding number of iterations, *IN*, and hence, leading to a different modulo2 value.

Thus, according to its attractive properties with respect to robustness, low computational cost, blind, and reversible the scheme of Chrysochos et al. [34], will be improved in our work, to solve its limitations. After refinement, it will contribute significantly in our proposed CP scheme.

Poonkuntran and Rajesh [1], proposed a watermarking scheme for the authentication of color medical images. Their proposed scheme generates the watermark, dynamically, using a hybrid bi stable messy system, which was used in [54], and based on the green color plane of the host image, as a seed to the messy system. Bits

of the generated watermark are embedded by expanding the expandable differences between the corresponding pixel pairs from *Red* and *Blue* color planes, in a method called intra plane difference expansion, which was used in Tian's scheme [53]. A location map is used to refer to the locations where embedding takes place. The location map is an image of binary pixels. Each pixel in the location map is set to one, when the corresponding difference is expandable, otherwise that pixel is set to zero. In watermark extraction, the location map is checked to locate the previously embedded pixel pairs. The LSB of the difference of each located pair gives the corresponding embedded watermark bit.

The most important point in Poonkuntran and Rajesh work is using the messy system; by which, the watermark is generated securely and unique to each image.

Generally, it is required to detect any modification in any location at the medical image, even if it is tiny. We can obtain this high sensitivity to image tampering by allowing the watermark to cover whole the image area. Poonkuntran and Rajesh scheme is for the purpose of CA of medical images. But unfortunately, the difference between the color planes may be large to some extent, leading to generate a non-expandable difference. Therefore, the generated watermark might not cover whole the image area. Thus, fragility is not enough high in Poonkuntran and Rajesh scheme. Image modifications might not be detected in all image locations.

Because it is secure, blind, reversible, and easily implemented; we decided to improve the idea discussed in Poonkuntran and Rajesh work, solving its drawbacks, and finally use its improved version in our proposed CA scheme.

### **3.2.2 Transform Domain Watermarking Techniques**

In transform domain watermarking techniques, the watermark is embedded in the transformed host image, then after watermark embedding the watermarked image is inverse-transformed back to spatial domain [26].

Hua et al. [80], proposed a fragile watermarking algorithm, which is based on both Discrete Wavelet Transform (DWT), and Discrete Cosine Transform (DCT), to achieve content integrity protection, CA, of color images. Original image's brightness component is first three-discrete-wavelet decomposed. The sub-bands LL3 and LH3 are then extracted and divided into blocks. Then, DCT is applied to each block. A specific coefficient (namely non-zero minimum value) in LH3's block's DCT is replaced by a corresponding coefficient (namely non-zero minimum value) in LL3's

block's DCT. A reverse DCT and DWT is done to get the watermarked image. Later, to detect whether the image has been tampered or not, those two replaced coefficients in each block are extracted and compared, if they are too close, then no tamper is detected in that block, otherwise, the block was tampered. Hence, the distortion will be located in each block independently.

In this work, the embedded watermark is represented by changing the LH3's block's DCT's coefficient value by LL3's block's DCT's coefficient value. Since the selected coefficients for replacement are independent and may have very different values (large in-between distance); replacement of one coefficient value by another (to have same values), of course, will distort original block's quality and hence, distort original image's quality permanently. Also, this work represents only a mechanism for image's tamper detection, CA, and not for image's CP.

When watermarking is used for the purpose of CP, or data hiding, it is needed to protect the embedded watermark. To increase the security of the watermark, Hongqin and Fangliang [81] proposed a new watermarking algorithm for color images, where Arnold transform is employed in the stage of watermark construction. The watermark is then embedded by modifying the coefficients of original image's Discrete Integer Wavelet Transform, DIWT. Their algorithm is resistance to JPEG compression.

Based on Singular Value Decomposition (SVD) and a tiny genetic algorithm, Lai [27] proposed a robust digital image watermarking scheme. In watermark embedding: Firstly, the SVD is employed in the cover image,  $\mathbf{A}$ , to obtain the three matrices,  $\mathbf{U}$  (left singular vectors of  $\mathbf{A}$ ),  $\mathbf{V}$  (right singular vectors of  $\mathbf{A}$ ), and  $\mathbf{S}$  (diagonal matrix of singular values). Secondly, a watermark image,  $\mathbf{W}$ , is inserted into  $\mathbf{S}$ , as  $\mathbf{S} + \alpha\mathbf{W}$ , where  $\alpha$  is a scaling factor used to control the watermark embedding strength, its value determined systematically without making any assumption, using a Tiny-GA. Thirdly, SVD is employed in the new generated matrix to obtain the matrices  $\mathbf{U}_w$ ,  $\mathbf{S}_w$ , and  $\mathbf{V}_w$ . Finally, the watermarked image,  $\mathbf{A}_w$ , is obtained by multiplying  $\mathbf{U}$ ,  $\mathbf{S}_w$ , and  $\mathbf{V}^T$ . In watermark extraction, given a watermarked image  $\mathbf{A}_w$ , the watermark can be extracted by reversing the embedding procedure, but here the extraction process requires some data previously generated in embedding:  $\mathbf{U}_w$ ,  $\mathbf{S}$ ,  $\mathbf{V}_w$ , and the scaling factor  $\alpha$ . Thus, it is a non-blind watermarking algorithm. Actually, Lai scheme [27] is a conventional watermarking scheme based on SVD, except the use of Tiny-GA to determine the optimum value of the scaling factor,  $\alpha$ . According to the experimental results denoted in their paper, there is approximately no difference between using the

scaling factor  $\alpha$  as generated by their idea (using Tiny-GA), or by assigning it a constant value (as in conventional SVD based methods). Thus, we see that the use of the genetic algorithms in this paper is useless. In addition, it increases the complexity of the algorithm.

In order to achieve blindness, a new way for embedding a grayscale watermark in a grayscale cover image, based on SVD, was introduced by Ma and Shen [2]. First, Arnold chaos map is employed to scramble the watermark. Secondly, according to the size of the watermark, the cover image is divided into blocks (non-overlapping). Then, SVD is employed to decompose each block. The scrambled watermark is then embedded into the biggest singular values of the decomposed blocks, by the quantization. The embedding process to get the new biggest singular value,  $S_w$ , based on the old biggest singular value,  $S$ , of the block is:  $S_w = S - S \bmod T + W_g * T$ , where  $T$  is a predefined quantization coefficient, used to adjust the embedding strength and  $W_g$  is a watermark's gray pixel value, scaled to fall in the interval  $[0,1]$ . Each embedded block is inverse transformed to generate the watermarked image.

The original cover image is not needed during watermark extraction procedure. The suspected image is treated as was done in the embedding stage, to get the biggest singular value,  $S^*$ , of each block. The watermark pixel's gray value, which was embedded in each block's biggest singular value,  $S^*$ , can be obtained by:  $W_g = (S^* \bmod T) / T$ , the results are assembled to get the scrambled watermark, which is inverse-Arnold transformed to get the unscrambled watermark. Actually, Ma and Shen achieve blindness by their proposed algorithm, but after applying inverse SVD, we note that the change in blocks singular value may lead to double values. In order to represent correct pixel values, these double values must be converted into integers. Unfortunately, by this numbering conversion, we will lose the modification done to singular values during embedding. Also, despite of its attractive properties regarding robustness and blindness, SVD based embedding method in [2], has the drawback of being non-reversible. Hence, the original image could not be recovered. Since in general SVD embedding has the drawback of being non-reversible [82], and because reversibility is a basic requirement in our algorithm, we will find a new way for embedding other than conventional SVD based watermarking.

Finally, and after reviewing those recently proposed watermarking schemes, we conclude that we still need new watermarking schemes, to cope with today's requirements. And thus, we answered our first research question.



## CHAPTER 4

### PROPOSED SCHEMES

In this thesis, three watermarking schemes are proposed. The first is the CA watermarking scheme. It is a reversible, secure, blind, and fragile watermarking scheme, which aims at detecting tampers in color images, even if the tamper is tiny. The second is the CP watermarking scheme. It is a reversible, secure, and blind watermarking scheme. In contrast to the first proposed scheme, this scheme is a robust watermarking scheme. Thus, it aims at protecting the copyright of color images. It will be robust against some geometrical attacks. The third is a multipurpose watermarking scheme, for both CA and CP of color images. The third scheme will be employed to detect whether or not the watermarked image is modified, as well as distinguishing between malicious attacks, and incidental manipulations. These three watermarking schemes will be discussed in the next sections.

#### 4.1 The Proposed Content Authentication Watermarking Scheme

In this section, we discuss our proposed fragile watermarking scheme. It will be used for tiny tamper detection of color images. The terms, "CA watermarking scheme" and "first proposed watermarking scheme" are used interchangeably, henceforth.

##### 4.1.1 Overview

We propose a CA watermarking scheme, which is a development of the technique proposed by Poonkuntran and Rajesh [1]. Their proposed scheme aimed at authenticating medical images, especially fundus images. They used a hybrid bi stable messy system, which was used in [54], to generate a watermark using a messy model, which is based on the green color plane as a seed to the messy system. The watermark embedding process is then carried out using intra-plane difference expanding, based on Integer Transform (IT) and Inverse Integer Transform (IIT). Each watermark bit is embedded by expanding the difference of the corresponding pixel pair, which is composed from Red and Blue color planes. Inverse method is used to restore the original image, after extracting and verifying the watermark. Watermark embedding and extraction processes of Poonkuntran and Rajesh algorithm can be seen in Figure 2.6 (a), and (b) respectively. The researchers claim that their proposed scheme is very sensitive to the jittering, geometrical and various filtering attacks. Actually, we



performed an intensive study, implementation, and tests to their proposed scheme, and from our point of view, we found two drawbacks in their proposed scheme:

***a) Poonkuntran and Rajesh's scheme first drawback***

Generally, any small change in the medical image may lead to different diagnose, and hence, a different treatment. This is absolutely not accepted in medicine. Thus, it is required to detect any alteration in medical images even if it is small and either if it is intentional or not. In such cases of modification, the attacker is interested in small portions of the watermarked image, usually are certain important details in the image. The only way to survive such attacks is by allowing the watermark to spread over the whole image's area. But unfortunately, that is not the situation in the scheme proposed by Poonkuntran and Rajesh [1]. The difference between pixels of the color planes may be large to some extent, leading to generate a non-expandable difference. Therefore, the embedded watermark might not cover whole the image. Thus, fragility is not enough high in Poonkuntran and Rajesh scheme. In other words, modifications might be detected in some image locations, but not in all locations.

***b) Poonkuntran and Rajesh's scheme second drawback***

Also, if we arguably assumed that the underlying image gives expandable locations over whole area of the image, and hence a spreading watermark is generated. Unfortunately, the way of embedding followed by Poonkuntran and Rajesh [1] will generate a watermarked image with a distorted quality if it is used to embed that large watermark. The reason is that, their scheme is based on the difference between pixels among two different color planes, which is generally not a small quantity. Based on that large difference, Poonkuntran and Rajesh substitute a pair of pixels, among two different color planes, with a new completely different pixels pair; to embed each single bit. Thus, having a spreading watermark over whole image area will completely change an amount of pixels equal twice its size, and distributed over two different color planes, Red and Blue color plans. Therefore, using large watermark sizes in Poonkuntran and Rajesh scheme will introduce low PSNR, and hence, generate a distorted image quality. This problem prohibits the embedding of any additional watermark for any other purpose, like CP. Because embedding additional bits will introduce extra image distortion, and may lead to a useless image, especially when talking about sensitive applications. Finally, it is worth mentioning that, this generated image distortion is not by an attacker, but it is only by embedding the watermark!

These two drawbacks are addressed in our first proposed watermarking scheme.

#### 4.1.2 The Conceptual Model of the Proposed CA Watermarking Scheme

The first proposed watermarking scheme is a reversible and fragile watermarking scheme. It is used for watermarking color images, aiming to detect any modification in it. The main functionalities of Poonkuntran and Rajesh scheme [1] are developed in our proposed scheme; to handle the previously mentioned drawbacks. The conceptual model of the proposed technique is summarized in Figure 4.1.

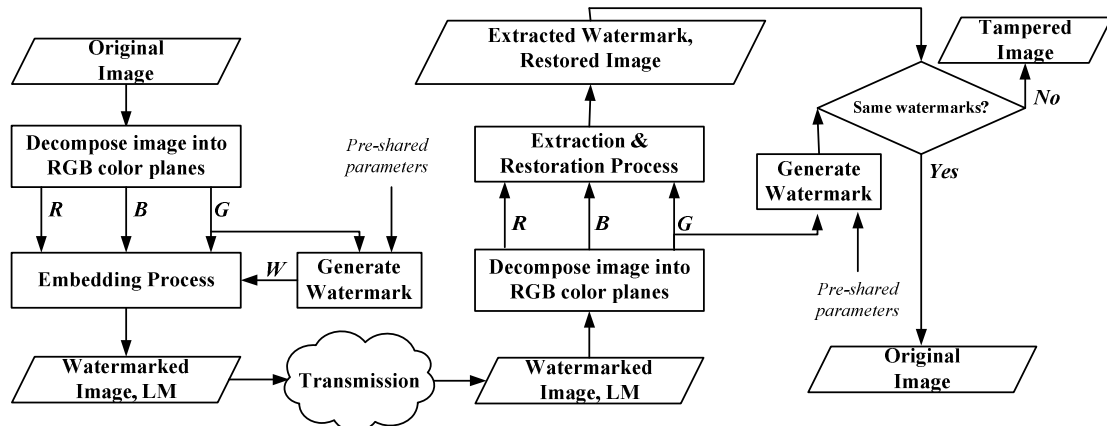


Figure 4.1: The conceptual model of the proposed CA watermarking scheme.

Our technique mainly depends on image's three color planes Red, Green, and Blue. Thus, decomposing the original image is the first process in our model. Then, a specific color plane, among the three, is chosen as a base for generating the watermark ( $W$ ), to be embedded. According to [1], in some color images; the Green channel contains more important details than other channels. Thus, in our CA scheme, the Green color plane is chosen as a reference plane for watermark generation process. One of the other two color planes can be used to embed the generated watermark. In our scheme, the Red color plane is used as a host for embedding the generated watermark, in a method called inter-plane difference expanding. After watermark embedding, the three color planes are all used to reconstruct the watermarked image. The watermarked image and the Location Map (LM) are the outputs of the embedding process. Since only a portion of image locations will be used for embedding, the LM is a one-bit bitmap, carrying information about the selected embedding locations.

At the other end, the receiver can make an assertion about the integrity of the received image. Firstly, the three color planes are obtained by decomposing the received watermarked image. Secondly, the watermark generation process is also conducted

here, based on the green color channel as before. Finally, the watermarked color plane, Red color plane, along with Green and Blue color planes are brought to the watermark extraction and image restoration process. The LM will be used in this process as a guide to refer to the locations where the watermark was embedded. The extracted watermark is then compared with the generated one, if a match is found, the restored image is the original image; otherwise, it is assumed a tampered image with precise tampering location. The next subsections discuss the core processes in our proposed CA scheme.

### 4.1.3 Messy Watermark Generation Process

In our proposed CA watermarking scheme, the watermark is generated using a messy, chaotic, system shown in Equation 4.1, which was also used in Poonkuntran and Rajesh scheme [1], and Ni et al. scheme [54]. The input variable,  $x_n$ , refers to the current input of the system, and  $x_{n+1}$  refers to the output of the messy system, which may be used as the next input to the system. Using a messy system has several benefits. First, the system complexity is alone providing a secure watermark generation, especially when using a security key and parameters those are unknown by others. Secondly, although it is deterministic, the behavior of the messy system appears random. Finally, based on the reference color plane of the host image, the watermark is generated dynamically. In other words, a unique watermark is generated for each different image.

$$x_{n+1} = f(x_n) = 4 \sin^2(x_n - 2.5) \quad (4.1)$$

As stated before, the green color plane will be used to generate the watermark. Therefore, it is used as a seed to the messy system of Equation 4.1. Generally, each pixel value will iteratively enter the system until obtaining a sequence of values, with a certain messy status. But before entering the messy system, each pixel value is first converted to a corresponding initial value, using Equation 4.2.

$$seq(k, 0) = a * floor \left[ \frac{s(k)}{2^l} \right] * 2^l + b * pos + c * key \quad (4.2)$$

Where,  $seq(k, 0)$  refers to the initial value of the  $k^{th}$  pixel,  $s(k)$  is the value of  $k^{th}$  pixel,  $a$ ,  $b$ , and  $c$  are predefined constants,  $l$  refers to the embedding depth,  $pos$  refers to the position index of the  $k^{th}$  pixel, and  $key$  is the security key used for securing the watermarking method, it could be any positive integer value. These parameters are pre-shared among legitimate parties, and it could be changed by their agreement.

Because the messy system is highly sensitive to its initial value; the variable  $pos$ , is used to achieve the requirement that, equal pixels on different positions in the same reference color plane should produce different messy sequences. The value of  $pos$  is set to the distinct pixel's position index,  $(M*i + j)$ , which ranges from  $0$  to  $MN-I$ . Where,  $i$  and  $j$  refer to the  $i^{th}$  column and  $j^{th}$  row of the current pixel respectively. Also,  $M$  is number of rows and  $N$  is the number of columns of the host image. Thereby, regardless of what values are assigned to the other parameters in Equation 4.2, equal pixels in the same reference color plane will generate different initial values, and hence, different messy sequences. Also, the constants  $a$ ,  $b$ , and  $c$  are used to fulfill the requirement that, equal pixels at same positions of different reference color planes should generate different messy sequences. The constants  $a$ ,  $b$ , and  $c$  are assigned different values for each processed image. Thereby, equal pixels those falls in the same positions along two different reference color planes, will generate different initial values, and hence, different messy sequences. The embedding depth,  $l$ , was used in [54], and it referred to the number of LSBs used to embed the authentication watermark bits in every selected pixel at the host image. But because in our scheme we embed bits by expanding the difference between pixels, rather than embedding in their LSBs; this parameter has a different meaning in our scheme. Anyway, we assign it a small quantity ( $l = 2$ ), to enable each pixel to contribute significantly in calculating its initial value.

***An illustration example for generating pixel's initial value is expressed below:***

Suppose that we have an image, which is decomposed into the three color planes Red, Green, and Blue. Assume that the  $k^{th}$  pixel index is 15 in the Green color plane, with value  $s(k) = 150$ , and we wish to generate its corresponding initial value,  $seq(k,0)$ , using Equation 4.2. Let  $a = b = c = 10$ ,  $l = 2$ , and  $key = 679$ .

$$seq(k,0) = 10 * \left\lfloor \frac{150}{2^2} \right\rfloor * 2^2 + 10(15) + 10 * 679 = 8420.$$

Now, to obtain the messy sequence of the  $k^{th}$  pixel, we substitute its initial value,  $seq(k,0)$ , for  $x_n$  in Equation 4.1 to obtain  $seq(k,1)$ , which is used as a new input to the same messy system to obtain  $seq(k,2)$ , and so on. This process is repeated a reasonable number of iterations,  $I$ , for the  $k^{th}$  pixel, until attaining the messy status, which will be discussed later. The generated sequence of the  $k^{th}$  pixel is referred to as  $seq(k,i)$ ,  $i=1,2,3,4,\dots,I$ . It is evident that the generated sequence contains a floating numbers. The sequence is then converted to a binary sequence  $b\_seq(k,i)$  using

Equation 4.3. Where,  $T$  is a threshold set to  $8/3$ , to attain approximately equal number of 0's and 1's, according to [1].

$$b\_seq(k, i) = \begin{cases} 1, & seq(k, i) > T \\ 0, & otherwise \end{cases} \quad (4.3)$$

In the binary representation of sequences, the probability of having equal binary sequences is high if the length of the sequence is very small. Therefore, in our work, we defined the messy status as the situation, where for a given set of generated binary sequences, the probability of encountering equal consecutive sequences is as small as possible. After some experiments, we observed that we could attain a messy status when the length of the sequences is not less than 5 bits, thus, we choose  $I$  to be 5.

The generated binary sequence is then summarized to 1 bit by applying XOR operation. The generated bit is the messy watermark bit for the  $k^{th}$  pixel. The same procedure is repeated with the rest of pixels in the reference color plane, Green plane, to obtain the binary watermark for entire the image, as shown in Figure 4.2.

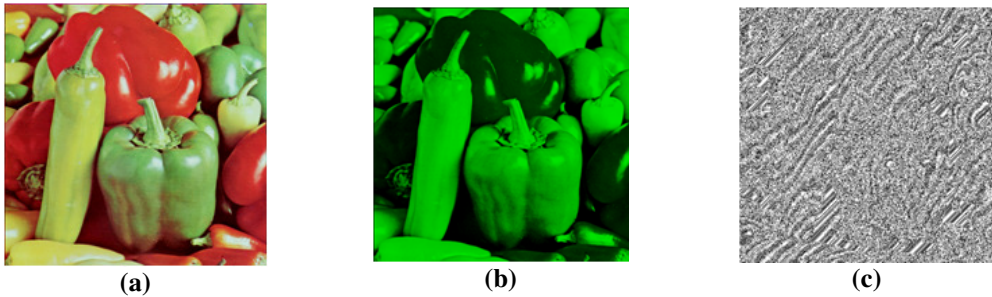


Figure 4.2: Full watermark generation using messy system. The green color plane is used as a reference plane. (a) Original image, (b) Reference color plane, (c) Generated watermark.

In CA applications, the role of the CA watermark is to sense modifications those may be applied to the watermarked image. Therefore, it is not important to have a meaningful CA watermark. As is the case in our proposed CA scheme, the watermark is generated using the messy system. On contrast, the CP watermark usually carries meaningful information, such as the logo of the company, to which the watermarked content belongs. This answers our seventh research question in this thesis.

#### 4.1.4 The Watermark Embedding Process of the Proposed CA Watermarking Scheme

A new embedding strategy is proposed aiming to solve the previously mentioned drawbacks of Poonkuntran and Rajesh scheme [1]. It will be conducted to embed a watermark based on the spatial domain of the original image. It is an accumulative

and homogenous embedding strategy. It is designed accumulative, aiming at spreading the watermark over whole image's area, and designed homogenous, aiming at producing a high quality, undistorted, watermarked image. Homogeneity is one of the characteristics of the embedding strategy, which we are searching for in the fifth research question of this thesis. To obtain homogeneity we embed the watermark by expanding the differences between the neighboring pixels at the same color plane, in a method called inter-plane difference expanding, rather than expanding the difference between pixels from different color planes, as in Poonkuntran and Rajesh scheme. We can choose either the Red or Blue color plane for embedding the watermark. The Red color plane is chosen for watermark embedding in our work. Inter-plane difference expanding is performed by employing the integer transform, IT, and its inverse, IIT, which will be discussed later.

In addition to homogeneity, we propose an accumulative watermark embedding, which aims at wide and non-uniform spreading of the embedded watermark; and hence, it will sense any modifications in the watermarked image, even if it is very small. To obtain an accumulative watermark embedding, the watermark embedding process is divided into three stages; namely, vertical, horizontal, and diagonal. These stages are run one after another. The output of each watermarking stage is the input to the next stage. As shown in Figure 4.1, the three color planes, obtained after decomposing the original image to be watermarked, and the watermark, which is generated using the messy system discussed in the previous subsection, are the inputs to the embedding process.

The detailed steps of the embedding process are illustrated in Algorithm 4.1. This algorithm will perform an accumulative watermark embedding by running the three embedding stages one after the other. Each stage is responsible for embedding some selected watermark bits, homogeneously, in a predefined direction. Each stage takes three inputs, namely, the host color plane, the binary watermark,  $W$ , and the location map matrix,  $LM$ . As seen in the Algorithm, the host color plane of each stage is the output of the preceding stage. Except for the first stage, the host color plane is the original *Red* color plane. The same binary watermark,  $W$ , enters all the three stages. Each stage will embed a portion of that watermark into its input host color plane, in the predefined direction. The last input, the location map matrix,  $LM$ , is passed to the three stages. Every stage will contribute in updating this matrix for each corresponding embedding location. After running the three embedding stages one

after another, the output of the last stage is used to reconstruct the watermarked image,  $WI$ . The three embedding stages share the same homogeneous embedding mechanism, which is embedding by inter-plane difference expanding, which is performed by employing the integer transform, IT, and its inverse, IIT.

---



---

**Algorithm 4.1:** The embedding process of the proposed CA scheme.

---



---

**Purpose:** Accumulative and homogeneous watermark embedding.

**Input:** *Red*, *Green*, and *Blue* color planes of the original image,  
Binary Watermark,  $W$ .

**Output:** The watermarked image,  $WI$ , and the location map,  $LM$ .

**Procedure:**

- a) Define  $LM$ , an empty matrix of size = size of the *Red* color plane.
- b) Based on the *Red* color plane and the watermark,  $W$ , the vertical embedding stage generates the vertical embedded component,  $VEC$ , and modifies the location map,  $LM$ , accordingly.
- c) Based on the vertical embedded component,  $VEC$ , and the watermark,  $W$ , the horizontal embedding stage generates the vertical and horizontal embedded component,  $VHEC$ , and modifies the location map,  $LM$ , accordingly.
- d) Based on the vertical and horizontal embedded component,  $VHEC$ , and the watermark,  $W$ , the diagonal embedding stage generates the vertical, horizontal, and diagonal embedded component,  $VHDEC$ , and modifies the location map,  $LM$ , accordingly.
- e) The watermarked image,  $WI$ , is generated by combining the vertical, horizontal, and diagonal embedded component,  $VHDEC$ , along with the *Green* and *Blue* color planes.

**End**

---



---

#### ❖ *Watermark embedding by inter-plane difference expanding*

Suppose that we have a pixel pair  $(p1, p2)$ , such that  $p1$  and  $p2$  are both integers and satisfying  $0 \leq p1, p2 \leq 255$ . Let  $d$  is the difference between  $p1$  and  $p2$ , and  $m$  is their average. The IT defines a one-to-one correspondence between  $(p1, p2)$  and  $(m, d)$ . The integer transform to obtain  $(m, d)$  from  $(p1, p2)$  is defined as shown in Equations 4.4 and 4.5 bellow.

$$m = \left\lfloor \frac{p1+p2}{2} \right\rfloor \quad (4.4)$$

$$d = p1 - p2 \quad (4.5)$$



Where,  $\lfloor v \rfloor$  is the floor operator, which return the greatest integer less than or equal  $v$ . The inverse integer transform to obtain  $(p1, p2)$  from  $(m, d)$  is defined as shown in Equations 4.6 and 4.7 bellow.

$$p1 = m + \left\lfloor \frac{d+1}{2} \right\rfloor \quad (4.6)$$

$$p2 = m - \left\lfloor \frac{d}{2} \right\rfloor \quad (4.7)$$

To embed a watermark *bit* by expanding the difference between  $p1$  and  $p2$ , a new difference  $d'$  is generated as seen in Equation 4.8, which is equivalent to appending the *bit* to the binary representation of  $d$ , such that *bit* becomes the new LSB.

$$d' = 2 * d + bit \quad (4.8)$$

That new difference,  $d'$ , along with the mean,  $m$ , will be used to generate a new pixel pair,  $(p1', p2')$ , to replace the old one, using the IIT shown in Equations 4.6 and 4.7. But before generating the new pixel pair, the suitability of that new difference is checked, to avoid both overflow and underflow of the new calculated pixels, i.e., to avoid having pixel values those fall outer the interval  $[0,255]$ . Particularly, the difference,  $d$ , is expandable if and only if, the new difference,  $d'$ , satisfies either part of Equation 4.9, which is derived by bounding  $p1$  and  $p2$  of Equations 4.6 and 4.7 respectively; such that  $0 \leq p1, p2 \leq 255$ .

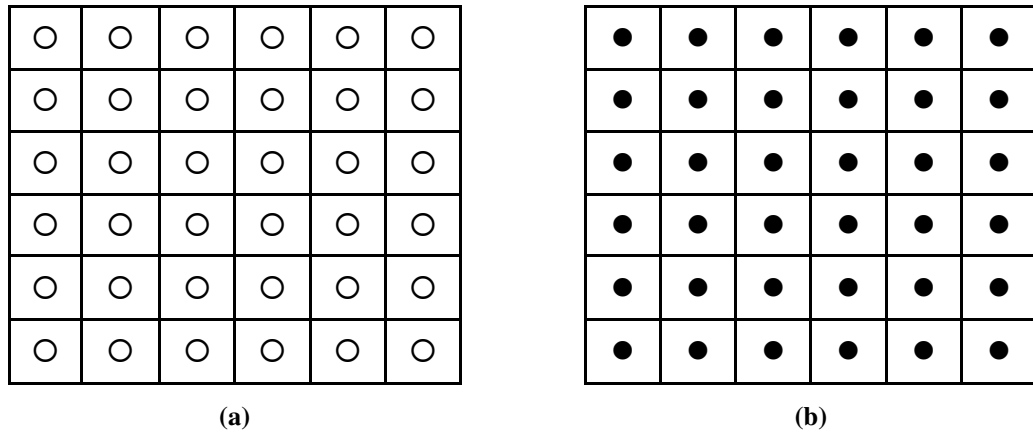
$$\begin{cases} |d'| \leq 2 * (255 - m), & \text{if } 128 \leq m \leq 255 \\ |d'| \leq (2 * m) + 1, & \text{if } 0 \leq m \leq 127 \end{cases} \quad (4.9)$$

For illustration, assume that we need to embed the *bit=1* by replacing the pixels pair  $(150,100)$ , which has the difference  $d=50$  and the average  $m=125$ . A new difference is generated,  $d'=2(50) +1=101$ , which is checked using the second part of Equation 4.9, according to the value of  $m$ . Since  $|101| \leq (2*125) +1$ , we conclude that  $d$  is an expandable difference, and hence,  $d'$  is a suitable difference for embedding. Finally, using IIT in Equations 4.6 and 4.7, and based on pixels mean,  $m$ , and the new difference,  $d'$ , we compute the new pixels pair,  $(176, 75)$ . On contrast, the pair  $(100, 30)$ , which has the difference  $d=70$  and the average  $m=65$ , is not a suitable pair for embedding *bit=1*; because  $d'=141$  does not satisfy Equation 4.9.

Now, and after being familiar with the general embedding mechanism followed by the three embedding stages, we discuss each stage individually. For better understanding, we will clarify the idea by the aid of an illustration example. Suppose that we have a colored image,  $I$ , of size  $(6 \times 6)$ , which is decomposed to have the three color planes



**Red**, **Green**, and **Blue**. As stated before, in watermark embedding we are interested in the **Red** color plane. Assume that the **Green** color plane is used as a seed to generate a (6x6) binary watermark image, **W**, based on the messy system discussed in section 4.1.3. Figure 4.3 (a) shows the **Red** color plane, and (b) shows the generated binary watermark image, **W**, of our illustration example.



**Figure 4.3: Illustration example, some inputs to the embedding process of the proposed CA watermarking scheme. (a) Red color Plane. (b) Binary watermark image W.**

Our goal is to embed the binary watermark image, **W**, into the **Red** color plane in a way that the embedded watermark covers whole image's area. Using inter-plane difference expanding method, each watermark bit will be embedded by expanding the difference between a two selected neighboring pixels from the **Red** plane. The three embedding stages share the task of embedding whole watermark bits. Each stage will be responsible for embedding some reserved watermark bits, using its own predetermined pixel pairing type. At each stage run, the previously generated watermarked plane is assumed as a new host plane for the current stage.

#### ***a) The proposed vertical embedding stage***

It is the first embedding stage in our proposed watermark embedding process of the proposed CA scheme. The **Red** color plane and the generated binary watermark, **W**, are inputs to this stage; they were shown in Figure 4.3 of our illustration example. Also, the location map, **LM**, is the third input to this stage. The vertical embedded component, **VEC**, mentioned in Algorithm 4.1, is the output of this stage. This stage is responsible for embedding some selected watermark bits, each of which is embedded by expanding the difference between two vertically neighbored pixels. Practically, in our illustration example, Figure 4.4 (a) shows the vertical pixels pairing

of the input Red color plane. And Figure 4.4 (b) shows the candidate watermark bits for vertical embedding, colored red.

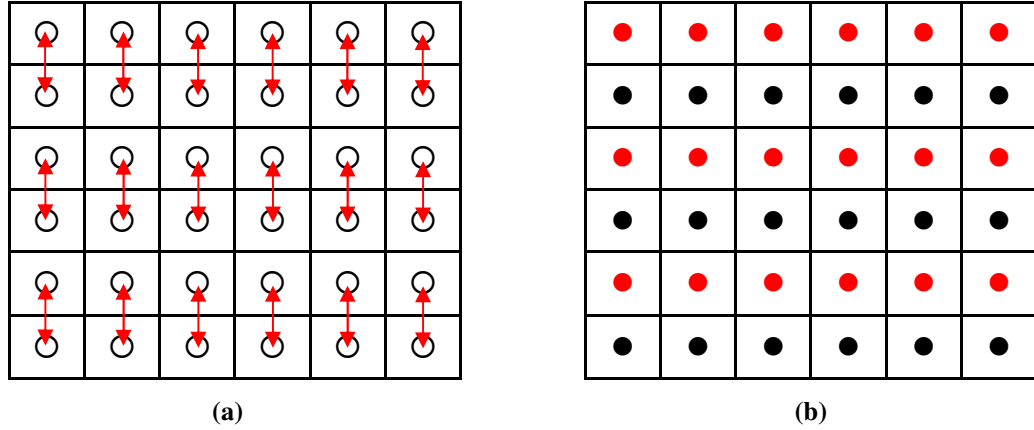


Figure 4.4: Illustration example, the working environment for the proposed vertical embedding stage of the watermark embedding process of the proposed CA scheme. (a) Vertical pixels pairing. (b) Candidate bits for V embedding.

As seen in Figure 4.4, each vertical pairing of pixels has a corresponding bit for embedding, colored red. The remaining non-embedded bits, colored black, will be handled by the participation of the other two embedding stages.

The detailed steps of the vertical embedding stage are illustrated in Algorithm 4.2.

---

**Algorithm 4.2:** The vertical embedding stage of the proposed CA scheme.

---

**Purpose:** Vertical embedding of some selected watermark bits.

**Input:** *Red* color plane, Binary Watermark,  $W$ , and Location Map,  $LM$ .

**Output:** Vertical Embedded Component,  $VEC$ .

**Procedure:** //Indexes:  $i$  for columns and  $j$  for rows

Initialize:  $VEC = Red$ ,  $Width = W.width$ ,  $Height = W.Height$

**FOR**  $i = 0 ; i < Width ; i++$

**FOR**  $j = 0 ; j < Height - 1 ; j += 2$

        ✓ Calculate the difference,  $d = VEC[i, j] - VEC[i, j+1]$

        ✓ Calculate the new difference,  $d' = 2(d) + W[i, j]$

**IF**  $d'$  satisfies Equation 4.9, i.e.  $d$  is expandable

            ✓ Update the location map as:  $LM[i, j] = 1$

            ✓ Replace the pixels pair ( $VEC[i, j], VEC[i, j+1]$ ) with a new pixels pair, using IIT, which is based on the new difference,  $d'$ , and the average,  $m$ , of the pair, as seen in Equations 4.6 and 4.7.

**END IF**

**END FOR**

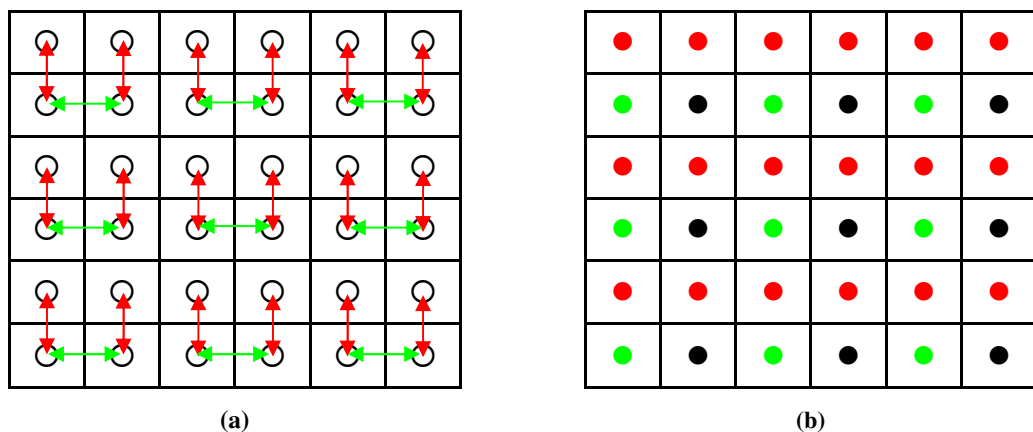
**END FOR**

---

As seen in Algorithm 4.2, the initial values, and the way of incrementing the variables  $i$  and  $j$  enforces vertical moving through the **Red** color plane, and hence generating vertical pixels pairing. Also, the variables  $i$  and  $j$  are responsible for relating each generated pixel pair with its corresponding watermark bit. Finally, in the algorithm, the location map,  $LM$ , is updated just before each bit embedding takes place.

**b) The proposed horizontal embedding stage**

It is the second embedding stage in our proposed watermark embedding process. The same watermark,  $W$ , and location map,  $LM$ , are also used as inputs to this stage. But, the host color plane is changed in this stage. The new host color plane is the output of the vertical embedding stage; namely,  $VEC$ . The output of this stage is  $VHEC$ , which will carry both vertical and horizontal embedding, shown practically in Figure 4.5 (a). And also, Figure 4.5 (b) shows the addition of new candidate watermark bits for horizontal embedding, colored green.



**Figure 4.5: Illustration example, the working environment for the proposed horizontal embedding stage of the watermark embedding process of the proposed CA scheme. (a) V and H pixels pairing. (b) Candidate bits for V&H embedding.**

As seen in Figure 4.5 (b), there still non-embedded bits, colored black. Those remaining bits will be embedded by the last embedding stage, diagonal embedding stage. The detailed steps of the horizontal embedding stage are illustrated in Algorithm 4.3, which embeds some selected watermark bits by expanding the difference between horizontally neighbored pixels through the vertical embedded component,  $VEC$ .

---

---

**Algorithm 4.3:** The horizontal embedding stage of the proposed CA scheme.

---

---

**Purpose:** Horizontal embedding of some selected watermark bits.

**Input:** Vertical Embedded Component,  $VEC$ , Watermark,  $W$ , Location Map,  $LM$ .

**Output:** Vertical and Horizontal Embedded Component,  $VHEC$ .

**Procedure:** //Indexes:  $i$  for columns and  $j$  for rows

Initialize:  $VHEC = VEC$ ,  $Width = W.width$ ,  $Height = W.Height$   
**FOR**  $i = 0 ; i < Width - 1 ; i += 2$   
    **FOR**  $j = 1 ; j < Height ; j += 2$   
        ✓ Calculate the difference,  $d = VHEC[i, j] - VHEC[i+1, j]$   
        ✓ Calculate the new difference,  $d' = 2(d) + W[i, j]$   
        **IF**  $d'$  satisfies Equation 4.9, i.e.  $d$  is expandable  
            ✓ Update the location map as:  $LM[i, j] = 1$   
            ✓ Replace the pixels pair ( $VHEC[i, j], VHEC[i+1, j]$ ) with a new  
            pixels pair, using IIT, which is based on the new difference,  $d'$ , and  
            the average,  $m$ , of the pair, as seen in Equations 4.6 and 4.7  
        **END IF**  
    **END FOR**  
**END FOR**  
**END**

---

---

### c) The proposed diagonal embedding stage

It is the last embedding stage in our proposed watermark embedding process. The output of the previous horizontal embedding stage; namely  $VHEC$ , is the new host color plane of this stage. The watermark,  $W$ , and location map,  $LM$ , are also used as inputs to this stage. Along with the previously embedded bits, the  $VHEC$  will be embedded with the rest of bits those were not embedded till now. The output of this stage is  $VHDEC$ , which will carry vertical, horizontal, and diagonal embedding. Practically, Figure 4.6 (a) shows the addition of the new embedding dimension, diagonal, colored blue. And also, Figure 4.6 (b) shows the addition of new candidate watermark bits for diagonal embedding, colored blue.

Assuming that all differences are expandable, by the completion of the diagonal embedding stage; it is guaranteed that all the watermark bits are embedded vertically, horizontally, and diagonally. In our illustration example, it is colored red, green, and blue respectively, as seen in Figure 4.6 (b). But, it is worth to mention that this

situation is only true when both the number of columns and the number of rows of the host image are even. The other situations are summarized in the following two points:

- ❖ If the number of rows of the host image is odd; then all the bits of the watermark's last row will be ignored.
- ❖ If the number of columns of the host image is odd; then half the bits of the watermark's last column will be ignored.

Actually, this is not a problem, because; firstly, we are not embedding a meaningful watermark, like a logo, secondly, we are interested in protecting important pixels, those are usually found far from the boundary of the image. Thereby, ignoring watermark bits those falls at the boundary of the watermark is not a big matter.

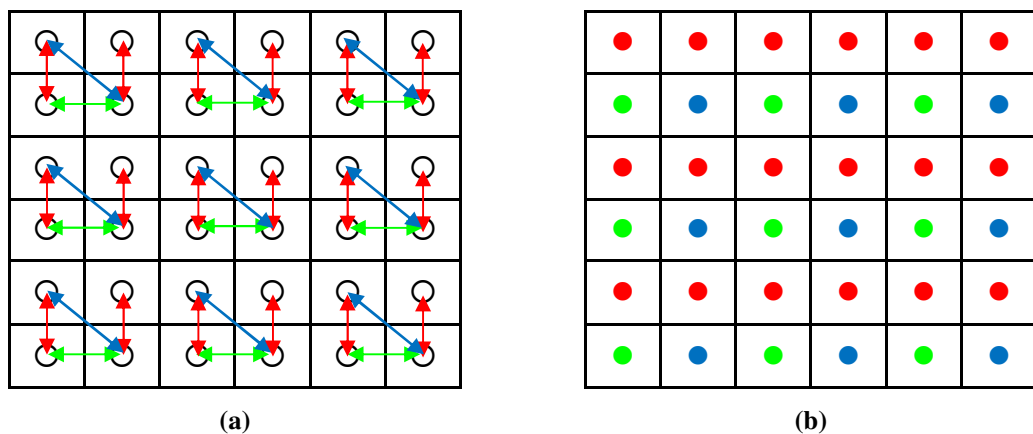


Figure 4.6: Illustration example, the working environment for the proposed diagonal embedding stage of the watermark embedding process of the proposed CA scheme. (a) V, H, and D pixels pairing. (b) Candidate bits for V, H and D embedding.

The detailed steps of the diagonal embedding stage are illustrated in Algorithm 4.4. As seen in the algorithm, it embeds bits by expanding the difference between diagonally paired pixels of the vertical and horizontal embedded component, *VHEC*.

According to Algorithm 4.1, the watermarked image, *WI*, is obtained by combining the original *Green* and *Blue* color planes with the accumulatively embedded component, *VHDEC*, which is the output of algorithm 4.4. Also, by the completion of the three embedding stages, the location map, *LM*, is obtained.

Finally, it is worth mentioning that, embedding by inter-plane difference expansion leads to larger number of expandable differences than those obtained by intra-plane difference expansion used in Poonkuntran and Rajesh scheme [1], because now pairing is based on neighboring pixels at the same color plane, which leads to smaller

differences. This also solves the second drawback of their scheme, low PSNR, because we replace old pixels with new pixels those are very close to the old ones.

---



---

**Algorithm 4.4:** The diagonal embedding stage of the proposed CA scheme.

---



---

**Purpose:** Diagonal embedding of some selected watermark bits.

**Input:** Vertical and Horizontal Embedded Component, *VHEC*, Binary Watermark, *W*, and Location Map, *LM*.

**Output:** Vertical, Horizontal, and Diagonal Embedded Component, *VHDEC*.

**Procedure:** //Indexes: *i* for columns and *j* for rows

Initialize: *VHDEC* = *VHEC*, *Width*=*W.width*, *Height*= *W.Height*

**FOR** *i* = 1 ; *i* < *Width* ; *i* += 2

**FOR** *j* = 1 ; *j* < *Height*; *j* += 2

        ✓ Calculate the difference,  $d = VHDEC[i, j] - VHDEC[i-1, j-1]$

        ✓ Calculate the new difference,  $d' = 2(d) + W[i, j]$

**IF**  $d'$  satisfies Equation 4.9, i.e.  $d$  is expandable

        ✓ Update the location map as:  $LM[i, j] = 1$

        ✓ Replace the pixels pair ( $VHDEC[i, j], VHDEC[i-1, j-1]$ ) with a new pixels pair, using IIT, which is based on the new difference,  $d'$ , and the average,  $m$ , of the pair, as seen in Equations 4.6 & 4.7

**END IF**

**END FOR**

**END FOR**

**END**

---



---

#### 4.1.5 The Watermark Extraction and Image Restoration Process of the Proposed CA Watermarking Scheme

To coincide with the new former proposed watermark embedding process, a new process is proposed. Namely, watermark extraction and image restoration process. It aims to extract the previously accumulatively embedded watermark, simultaneously with removing the effect of the embedded bits. The detailed steps of the watermark extraction and image restoration process are illustrated in Algorithm 4.5. As seen in the algorithm, The **Red**, **Green**, and **Blue** color planes of the watermarked image, along with the location map, *LM*, which was constructed during the embedding process are inputs to the watermark extraction and image restoration process. Three stages are employed to extract, then to remove, the watermark, which was embedded accumulatively in the **Red** plane. Each stage is responsible for detecting the

watermark bits, those were embedded in a specific direction, and also it performs a restoration from that embedding. The order of the three stages is reverse to that followed by the embedding process. In other words, here, we start with the diagonal direction, where it is the last handled direction in the watermark embedding process. Finally, the restored image, *RI*, and the extracted watermark, *EW*, are the outputs of this process.

---

---

**Algorithm 4.5:** Watermark extraction and image restoration process of the proposed CA scheme.

---

---

**Purpose:** Watermark detection, and image restoration

**Input:** The color planes *Red*, *Green*, and *Blue*; and the Location Map, *LM*.

**Output:** The Restored Image, *RI*, and the Extracted Watermark, *EW*.

**Procedure:**

- a) Initialize  $VHDEC = Red$
- b) Based on the accumulatively embedded component, *VHDEC*, and the location map, *LM*, the diagonal extraction and restoration stage extracts the diagonally embedded bits, and modifies the extracted watermark, *EW*, accordingly. Then it restores from diagonal embedding, and generates the vertical and horizontal embedded component, *VHEC*.
- c) Based on the vertical and horizontal embedded component, *VHEC*, and the location map, *LM*, the horizontal extraction and restoration stage extracts the horizontally embedded bits, and modifies the extracted watermark, *EW*, accordingly. Then it restores from the horizontal embedding, and generates the vertical embedded component, *VEC*.
- d) Based on the vertical embedded component, *VEC*, and the location map, *LM*, the vertical extraction and restoration stage extracts the vertically embedded bits, and modifies the extracted watermark, *EW*, accordingly. Then it restores from the vertical embedding, and generates the restored component, *RC*.
- e) The restored image, *RI*, is generated by combining the restored component, *RC*, along with the *Green* and *Blue* color planes.

**END**

---

---

Now, we discuss each stage of the proposed watermark extraction and image restoration process individually.

**a) The proposed diagonal extraction and restoration stage**

This stage is responsible for both detecting and removing the diagonally embedded watermark bits from the vertically, horizontally, and diagonally embedded component, **VHDEC**, which is the **Red** component of the watermarked image.

The detailed steps of this stage are illustrated in Algorithm 4.6.

---



---

**Algorithm 4.6:** The diagonal extraction and restoration stage of the proposed CA scheme.

---



---

**Purpose:** Detecting and removing the watermark bits those were embedded diagonally.

**Input:** The vertical, horizontal, and diagonal embedded component, **VHDEC**, and the location map, **LM**.

**Output:** The vertical and horizontal embedded component, **VHEC**, and the extracted watermark, **EW**, which will only have the bits were embedded diagonally.

**Procedure:** //Indexes: *i* for columns and *j* for rows

Initialize: **VHEC** = **VHDEC**, **Width**= **VHEC.Width**, **Height**= **VHEC.Height**

**FOR** *i* = 1 ; *i* < **Width** ; *i* += 2

**FOR** *j* = 1 ; *j* < **Height**; *j* += 2

        Take the diagonal pixels **Pair** = (**VHEC** [*i*, *j*], **VHEC** [*i*-1, *j*-1])

**IF** **LM** [*i*, *j*] ==1, i.e., The **Pair** was previously embedded

**Extraction:**

        ✓ Calculate the new difference, **d'** = **VHEC** [*i*, *j*] - **VHEC** [*i*-1, *j*-1]

        ✓ Based on the new difference, **d'**, and using Equation 4.10, extract the diagonally embedded watermark **bit** as: **EW**[*i*, *j*] = (**d'** % 2)

**Restoration:**

        ✓ Based on the new difference, **d'**, use Equation 4.8 to calculate the old difference, as  $\mathbf{d} = \left\lfloor \frac{\mathbf{d}'}{2} \right\rfloor$ .

        ✓ Using Equation 4.4 of the IT, calculate the average, **m**, of the current **Pair**.

        ✓ Based on the old difference, **d**, and the average, **m**, apply the IIT in Equations 4.6 and 4.7, to get a new pair to replace the current **Pair**.

**END IF**

**END FOR**

**END FOR**

**END**

---



---



The algorithm first searches for a diagonal pair, for which the corresponding location map,  $LM$ , value is 1, i.e., the diagonal pair was previously embedded. Then it extracts the embedded bit using Equation 4.10, which finds the LSB of the pair difference,  $d'$ . Then, it restores from the diagonal embedding of that bit. Particularly, the restoration is simply replacing the current pixels pair with the old one, using Equations 4.6 and 4.7 of the IIT, which is based on the old difference,  $d$ , and the average,  $m$ . The old difference,  $d$ , can be obtained based on the new difference,  $d'$ , using Equation 4.8.

$$bit = (d' \% 2) \quad (4.10)$$

For example, suppose we have a diagonal pair of pixels,  $(p'_1, p'_2) = (30, 13)$ , in the watermarked plane. First we compute the new difference  $d'=17$ , and the average  $m=21$ . Then we extract  $bit = (17 \% 2) = 1$ . Then we restore from the diagonal embedding of this bit by replacing the watermarked pair,  $(p'_1, p'_2)$ , by the original pair,  $(p_1, p_2)$ , based on the old difference  $d = floor(d'/2) = 8$ , and IIT to get  $p_1$ , and  $p_2$  as follows:

$$p_1 = 21 + \left\lfloor \frac{8+1}{2} \right\rfloor = 25, \quad p_2 = 21 - \left\lfloor \frac{8}{2} \right\rfloor = 17$$

After the completion of this stage; the extracted watermark,  $EW$ , will only have the watermark bits, those were embedded diagonally. Therefore, the same extracted watermark parameter,  $EW$ , will also pass to the next two stages, to obtain the full extracted watermark, which will have the diagonal, horizontal, and vertical bits.

#### ***b) The proposed horizontal extraction and restoration stage***

It is the second stage in the extraction and restoration process of our proposed CA scheme, and it is responsible for both detecting and removing the horizontally embedded watermark bits from the plane previously generated by the diagonal extraction and restoration stage, namely, vertical and horizontal embedded component,  $VHEC$ . The detailed steps of this stage are illustrated in Algorithm 4.7.

---

---

**Algorithm 4.7:** The horizontal extraction and restoration stage of the proposed CA scheme.

---

---

**Purpose:** Detecting and removing the watermark bits those were embedded horizontally.

**Input:** The vertical and horizontal embedded component, *VHEC*, and the location map, *LM*.

**Output:** The vertical embedded component, *VEC*, and the extracted watermark, *EW*, which will have both the diagonally and horizontally embedded bits.

**Procedure:** //Indexes: *i* for columns and *j* for rows

Initialize: *VEC* = *VHEC*, *Width* = *VEC.Width*, *Height* = *VEC.Height*

**FOR** *i* = 0 ; *i* < *Width* - 1 ; *i* += 2

**FOR** *j* = 1 ; *j* < *Height* ; *j* += 2

        Take the horizontal pixels *Pair* = (*VEC* [*i*, *j*], *VEC* [*i*+1, *j*])

**IF** *LM* [*i*, *j*] == 1, i.e., The *Pair* was previously embedded

**Extraction:**

- ✓ Calculate the new difference,  $d' = VEC[i, j] - VEC[i+1, j]$ .
- ✓ Based on the new difference,  $d'$ , and using Equation 4.10, extract the horizontally embedded watermark *bit* as:  $EW[i, j] = (d' \% 2)$ .

**Restoration:**

- ✓ Based on the new difference,  $d'$ , use Equation 4.8 to calculate the old difference, as  $d = \left\lfloor \frac{d'}{2} \right\rfloor$ .
- ✓ Using Equation 4.4 of the IT, calculate the average,  $m$ , of the current *Pair*.
- ✓ Based on the old difference,  $d$ , and the average,  $m$ , apply the IIT in Equations 4.6 and 4.7, to get a new pair to replace the current *Pair*.

**END IF**

**END FOR**

**END FOR**

**END**

---

---

### c) The proposed vertical extraction and restoration stage

It is the last stage in the extraction and restoration process of our proposed CA scheme. It is responsible for both detecting and removing the vertically embedded watermark bits from the vertical embedded component, *VEC*, which was generated by the previous stage, namely, horizontal extraction and restoration stage. The detailed steps of the vertical extraction and restoration stage are illustrated in Algorithm 4.8.

---

---

**Algorithm 4.8:** The vertical extraction and restoration stage of the proposed CA scheme.

---

---

**Purpose:** Detecting and removing the watermark bits those were embedded vertically.

**Input:** The vertical embedded component, *VEC*, and the location map, *LM*.

**Output:** The restored component, *RC*, and the extracted watermark, *EW*, which will have the diagonally, horizontally, and vertically embedded bits.

**Procedure:** //Indexes: *i* for columns and *j* for rows

Initialize: *RC* = *VEC*, *Width* = *RC.Width*, *Height* = *RC.Height*

**FOR** *i* = 0 ; *i* < *Width* ; *i* ++

**FOR** *j* = 0 ; *j* < *Height* - 1 ; *j* += 2

        Take the vertical pixels *Pair* = (*RC* [*i*, *j*], *RC* [*i*, *j*+1])

**IF** *LM* [*i*, *j*] == 1, i.e., The *Pair* was previously embedded

**Extraction:**

        ✓ Calculate the new difference,  $d' = RC[i, j] - RC[i, j+1]$ .

        ✓ Based on the new difference,  $d'$ , and using Equation 4.10, extract the horizontally embedded watermark *bit* as:  $EW[i, j] = (d' \% 2)$ .

**Restoration:**

        ✓ Based on the new difference,  $d'$ , use Equation 4.8 to calculate the old difference, as  $d = \left\lfloor \frac{d'}{2} \right\rfloor$ .

        ✓ Using Equation 4.4 of the IT, calculate the average,  $m$ , of the current *Pair*.

        ✓ Based on the old difference,  $d$ , and the average,  $m$ , apply the IIT in Equations 4.6 and 4.7, to get a new pair to replace the current *Pair*.

**END IF**

**END FOR**

**END FOR**

**END**

---

---

As seen in Algorithm 4.8, the steps are mainly as those in the previous stages, except, now the algorithm deals with vertical pairs of pixels. The outputs of this stage are the restored component, *RC*, and the extracted watermark, *EW*.

As stated in Algorithm 4.5, shown before, the restored image, *RI*, is generated by combining the restored component, *RC*, along with the *Green* and *Blue* color components. If we assumed that the watermarked image in not modified; then the

restored image,  $RI$ , is identical to the original, un-watermarked, image. The next subsection discusses and proves this fact, mathematically.

#### 4.1.6 Demonstrating the Reversibility of our Proposed CA Scheme.

Assume that we have an original image, and we wish to protect it by employing our proposed CA watermarking scheme. Our CA scheme first generates the messy watermark based on the *Green* component of the host image, and then it tries to embed the watermark bits into the *Red* component of the same image. Each watermark bit is embedded, independently, by expanding the difference of the corresponding pixels pair. Therefore, what applies to bit processing, also applies to whole watermark processing. In this section, we will demonstrate, mathematically, how a bit is embedded, how the bit is extracted, and finally, we prove the reversibility of our proposed CA scheme, via recovering the original, un-watermarked, pixels pair from the watermarked pixels pair.

##### (a) Embedding by expanding the difference

Assume that we have a pixel pair,  $(p_1, p_2)$ , and a watermark *bit* = 1; and we wish to embed *bit* by employing our proposed CA scheme. Using the IT in Equations 4.4 and 4.5, we calculate the pair average,  $m$ , and difference,  $d$ , as follows:

$$m = \left\lfloor \frac{p_1 + p_2}{2} \right\rfloor, d = p_1 - p_2$$

Using Equation 4.8, which is based on the value of *bit*, and the difference,  $d$ ; we find the new difference,  $d'$ , which is an expansion of the old difference  $d$ .

$$d' = 2(p_1 - p_2) + 1$$

Using IIT in Equations 4.6 and 4.7, based on the average,  $m$ , and the new difference,  $d'$ , we embed the *bit* by generating a new pixels pair,  $(p'_1, p'_2)$ :

$$p'_1 = \left\lfloor \frac{p_1 + p_2}{2} \right\rfloor + \left\lfloor \frac{2(p_1 - p_2) + 2}{2} \right\rfloor = \left\lfloor \frac{p_1 + p_2}{2} \right\rfloor + (p_1 - p_2) + 1$$

$$p'_2 = \left\lfloor \frac{p_1 + p_2}{2} \right\rfloor - \left\lfloor \frac{2(p_1 - p_2) + 1}{2} \right\rfloor = \left\lfloor \frac{p_1 + p_2}{2} \right\rfloor - (p_1 - p_2)$$

Thus, we embedded the *bit* by replacing the pair,  $(p_1, p_2)$ , with the new pair  $(p'_1, p'_2)$ .

##### (b) Extracting the previously embedded bit

Assume that we have the watermarked pixel pair,  $(p'_1, p'_2)$ , and we wish to extract the embedded *bit* by employing our proposed CA scheme. Using Equation 4.5, we calculate the difference,  $d'$ , between the pixels of the pair:

$$\mathbf{d}' = \mathbf{p}'_1 - \mathbf{p}'_2 = 2(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{1}$$

It is evident that the difference,  $\mathbf{d}'$ , is always odd. Therefore, according to Equation 4.10, we conclude that the embedded bit was  $\mathbf{bit} = 1$ , which is the LSB of the difference,  $\mathbf{d}'$ . Of course, this extracted bit equals the previously embedded bit only if the watermarked pixels pair,  $(\mathbf{p}'_1, \mathbf{p}'_2)$ , is not modified. This can be used as an indicator to refer to the modified locations at the watermarked image. Thus, to verify whether the watermarked image is altered or not, the extracted bit is compared with the previously embedded bit. The previously embedded bit is generated using the messy system, and based on the **Green** color component of the watermarked image. Thereby, at the level of the whole image, any mismatch between the extracted watermark and the generated watermark, corresponds to a tamper location. Knowing that the image is not modified, we can restore from watermarking to recover the original image, otherwise our restored image is a tampered version of the original image.

***(c) Restoring from watermarking and recovering the original pair***

Using Equation 4.8; based on the new difference,  $\mathbf{d}'$ , and the extracted  $\mathbf{bit} = 1$ , we calculate the old difference, which we name,  $\mathbf{d}''$ .

$$\mathbf{d}'' = \lfloor \mathbf{d}' / 2 \rfloor = \lfloor (2(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{1}) / 2 \rfloor = (\mathbf{p}_1 - \mathbf{p}_2)$$

Based on the watermarked pair,  $(\mathbf{p}'_1, \mathbf{p}'_2)$ , and using Equation 4.4, we find the new average,  $\mathbf{m}'$ .

$$\mathbf{m}' = \left\lfloor \frac{\mathbf{p}'_1 + \mathbf{p}'_2}{2} \right\rfloor = \left\lfloor \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \right\rfloor = \mathbf{m}$$

Finally, the restored pixels pair,  $(\mathbf{p}''_1, \mathbf{p}''_2)$ , is found using the IIT in Equations 4.6 and 4.7, based on the difference,  $\mathbf{d}''$ , and the average,  $\mathbf{m}'$ ,

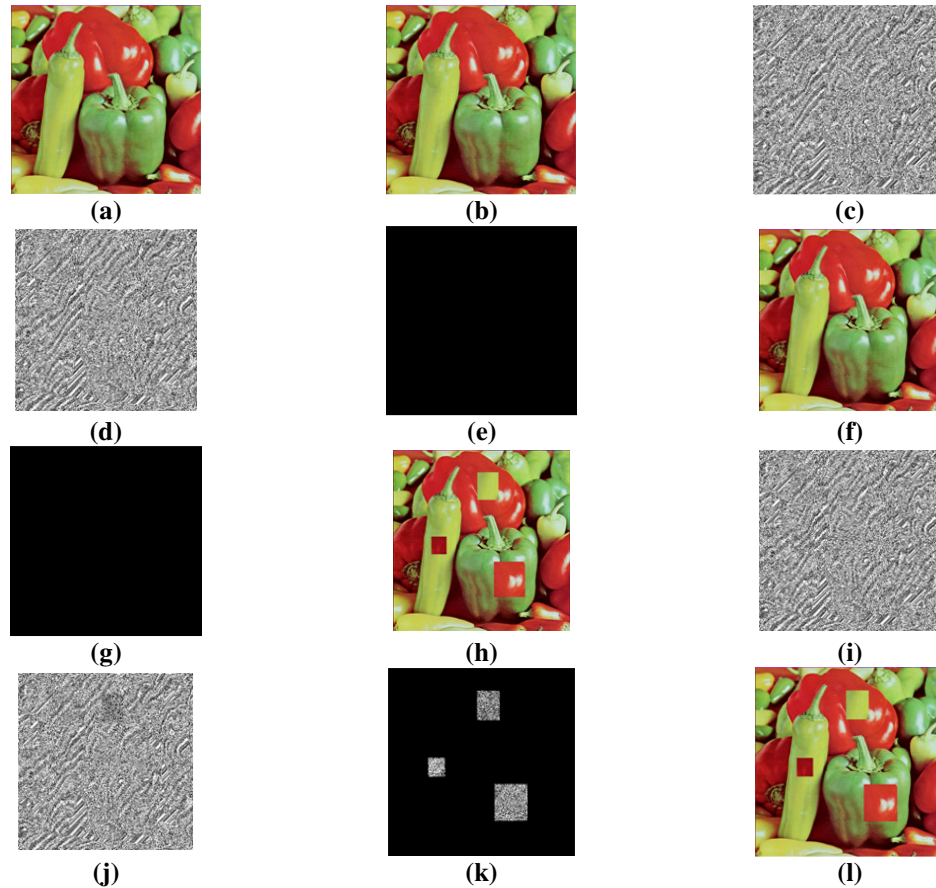
$$\mathbf{p}''_1 = \left\lfloor \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \right\rfloor + \left\lfloor \frac{(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{1}}{2} \right\rfloor = \left\lfloor \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \right\rfloor + \left\lfloor \frac{\mathbf{p}_1 - \mathbf{p}_2}{2} \right\rfloor = \left\lfloor \frac{2\mathbf{p}_1}{2} \right\rfloor = \mathbf{p}_1$$

$$\mathbf{p}''_2 = \left\lfloor \frac{\mathbf{p}_1 + \mathbf{p}_2}{2} \right\rfloor - \left\lfloor \frac{(\mathbf{p}_1 - \mathbf{p}_2)}{2} \right\rfloor = \left\lfloor \frac{2\mathbf{p}_2}{2} \right\rfloor = \mathbf{p}_2$$

Thus, the restored pixels pair,  $(\mathbf{p}''_1, \mathbf{p}''_2)$ , leads the original pixel pair,  $(\mathbf{p}_1, \mathbf{p}_2)$ . This actually proves the reversibility of our proposed CA watermarking scheme.

An illustration example demonstrating watermark verification, image restoration, and tamper detection is shown in Figure 4.7. When exact match is found between the generated watermark (c) and the extracted watermark (d) of the watermarked image (b); the error image (e) will be zero anywhere and the restored image (f) is equivalent

to the original image (a). This is evident in the error image (g) between the original image (a) and the restored image (f). But, when there is a mismatch between the generated watermark (i) and the extracted watermark (j) of the tampered watermarked image (h); the locations of alterations are detected precisely in the error image (k). Therefore, the restored image (l) is a tampered version of the original image (a) with precise tamper locations.



**Figure 4.7: Example demonstrating watermark verification, image restoration, and tamper detection. (a) Original image. (b) Watermarked version of (a). (c) Generated watermark based on green component of (b). (d) Extracted watermark from red component of (b). (e) Error image between (c) and (d). (f) Restored Image from (b). (g) Error image between (a) and (f). (h) Tampered version of (b). (i) Generated watermark based on green component of (h). (j) Extracted watermark based on red component of (h). (k) Error image between (i) and (j). (l) Restored image from (h).**

## 4.2 The Proposed Copyright Protection Watermarking Scheme

In this section, we discuss our proposed robust watermarking scheme, for the CP of color images. We start with an overview, which will, briefly, discuss the emergence of this scheme, and then we focus on the drawback the scheme will handle. Henceforth, the terms "CP watermarking scheme" and "second proposed watermarking scheme" are used interchangeably.

### 4.2.1 Overview

We propose a CP watermarking scheme, which is a development of the reversible watermarking technique proposed by Chrysochos et al. [34]. Their proposed scheme aimed at proving the ownership of digital images by embedding a robust watermark in it. The basic principle of their proposed scheme is based on the permutation of the histogram bins, according to a specific rule. In other words, each bit is embedded by permuting a corresponding couple of the histogram bins according to that rule. The couples of the histogram bins are chosen based on a public key, which is a real number. The integer part of this key is called (*start*), and it defines the embedding starting point in the histogram of the host image. The decimal part of this key, multiplied by ten, is called (*step*), and it defines the minimum distance a couple of histogram bins may have. The public key is needed in both watermark embedding and extraction processes. The authors claim that their proposed scheme is resistant to some geometrical attacks and has low computational complexity. Watermark embedding and extraction processes of Chrysochos et al. algorithm can be seen in Figure 2.5 (a), and (b) respectively. The authors of this scheme refer to an inherent drawback in their proposed scheme:

#### *The drawback of Chrysochos et al. scheme*

The maximum payload capacity of the scheme is rather low. At the best case, where all selected grayscale image histogram bin pairs are assumed as candidates for embedding, the payload capacity is 128 bits. If the scheme is applied for color images, in a way, such that each color component carries a portion of the watermark bits, the maximum payload capacity is 384 bits.

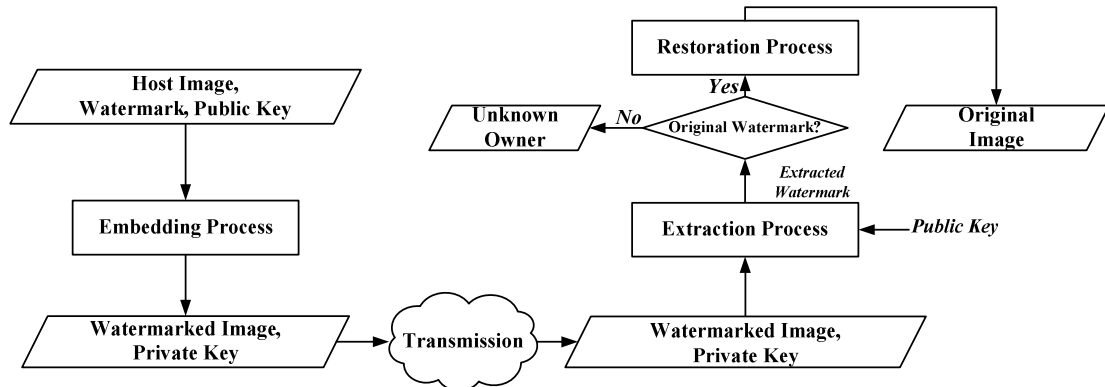
Usually, robustness to geometrical attacks is obtained at the expense of the payload capacity [83, 84]. However, we performed an intensive study, implementation, and testing to their proposed scheme, in an attempt to address this drawback. We have reached a new idea for embedding, which is employed in our second proposed watermarking scheme. The next subsections demonstrate our achieved idea.

### 4.2.2 The Conceptual Model of the Proposed CP Watermarking Scheme

The second proposed watermarking scheme is a reversible and robust watermarking scheme. It is used for watermarking color images aiming at protecting them from illegal distribution. It is a blind watermarking scheme; the embedded watermark can



be detected apart from the original image. The main functionalities of Chrysochos et al. scheme [34] are developed to handle the previously mentioned drawback. The conceptual model of the proposed CP technique is summarized in Figure 4.8.



**Figure 4.8: The conceptual model of the proposed CP watermarking scheme.**

As seen in Figure 4.8, the conceptual model of our proposed CP watermarking scheme consists of three main processes. The first main process is the embedding process, which is responsible for embedding the watermark in a way, such that it is embedded robustly. This process depends on three inputs, the original image to be watermarked, the watermark, and a public key. Two outputs are produced, the watermarked image, and the private key, which will be used later for image restoration. The second process is the extraction process, which is responsible for detecting the watermark previously embedded. The watermarked image and the same public key, used for embedding, are the two inputs used by this process. The extracted watermark is its output. Finally, the third process in our proposed scheme is the restoration process, which is responsible for removing the previously embedded bits from the watermarked image completely, and recovering the original, un-watermarked, image. This process runs only if the originality of the extracted watermark, which is the output of the extraction process, is verified. In addition to the public key, the restoration process is also based on the private key, which is generated during the embedding process. The next subsections discuss each of these three main processes individually.

### **4.2.3 The Watermark Embedding Process of the Proposed CP Watermarking Scheme**

Aiming at increasing the maximum payload capacity of Chrysochos et al. scheme [34], a new embedding mechanism is proposed. The public key and the permutation



of the histogram bins are still used in our proposed scheme. But the embedding rule is completely different. Now, a couple of bits are embedded at a time, by permuting three histogram bins. While in Chrysochos et al. scheme, one bit is embedded at a time, by permuting two histogram bins. In our scheme, the public key is still a real number, but it has a different use than that in Chrysochos et al. scheme. The integer part of this key is called (*begin*), which refers to the point at the histogram, where the embedding process will begin choosing triples, rather than couples, of intensity values (*a*, *b*, and *c*) with a corresponding histogram bins (Hist (*a*), Hist (*b*), and Hist (*c*)). Thus, the integer part of the public key, *begin*, could be any integer value in the interval [0, 255]. The decimal part of the public key is called (*step*), which is a single digit, used to define the intensity interval the three intensity values may occupy. Equation 4.11, demonstrates how to calculate the value of *begin*, by extracting the integer part of a *publicKey*. And Equation 4.12 demonstrates how to calculate the value of *step*, by extracting the decimal part of that key.

$$\mathbf{begin = publicKey/1} \quad (4.11)$$

$$\mathbf{step = (publicKey \% 1) * 10} \quad (4.12)$$

Based on the calculated values of *begin* and *step*, the three intensity values, *a*, *c*, and *b* are calculated as shown in Equations 4.13, 4.14, and 4.15 respectively.

$$\mathbf{a = begin} \quad (4.13)$$

$$\mathbf{c = (begin + step) \% 256} \quad (4.14)$$

$$\mathbf{b = \left\lfloor \frac{(a+c)}{2} \right\rfloor} \quad (4.15)$$

As seen in the equations, the intensity value *b* is chosen as a midpoint between the two values *a*, and *c*. Therefore, a sufficient distance between *a*, and *c* is needed. This is obtained by restricting the value of *step* in the interval [2, 9]. Moving to the next triple is simply achieved by cyclically incrementing each intensity value by one, as seen in Equation 4.16.

$$\mathbf{a = (a + 1)\%256; b = (b + 1)\%256; c = (c + 1) \%256} \quad (4.16)$$


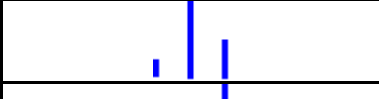
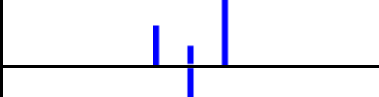


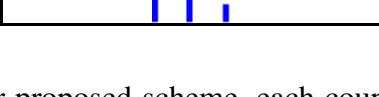
At each triple calculation, we test whether *c* is less than *b*; to avoid the situation where *a*, and *b* are at the end of the histogram, while *c* is at the beginning. If that occurs, the triple is moved to the beginning of the histogram as seen in Equation 4.17.

$$\mathbf{a = 0; c = step; b = \left\lfloor \frac{step}{2} \right\rfloor} \quad (4.17)$$

Each intensity values triple, (*a*, *b*, *c*), has a corresponding histogram bins triple, (Hist(*a*), Hist(*b*), Hist(*c*)). In our proposed scheme, to obtain a different pattern for

each permutation in the bins triple; we are concerned only with intensity triples, whose corresponding histogram bins are strictly non equal; Mean:  $\text{Hist}(a) \neq \text{Hist}(b) \neq \text{Hist}(c)$ . Excluding triples those have equal histogram bins, the magnitude of each histogram bin within a triple might be higher than, lower than, or in between the other two histogram bins at the same triple. Consequently, each intensity triple will have a corresponding Bins Triple Pattern (**BTP**), which will be one of the six patterns shown in Table 4.1.

**Table 4.1: List of Bins Triple Patterns (BTPs), used in our proposed CP watermarking scheme.**

Bins Triple Pattern (BTP)	Shape of the pattern	Interpretation of the pattern
0		Low, Between, High
1		Low, High, Between
2		Between, Low, High
3		Between, High, Low
4		High, Low, Between
5		High, Between, Low

As mentioned before, in our proposed scheme, each couple of bits will be embedded in a selected triple of histogram bins. Each couple of bits has a corresponding Bits Couple Pattern (**BCP**), which may equal “00”, “01”, “10”, or “11”. Embedding each selected couple is performed by the permutation of the corresponding triple of histogram bins according to the rule seen in Table 4.2. As seen in the table, the rule maps each **BCP** (“00”, “01”, “10”, and “11”) to a corresponding **BTP** (0, 1, 4, and 5). Also, to achieve lower computational complexity, the remaining two **BTPs** (2 and 3) are also exploited to represent the **BCPs** “00” and “11”. Hence, if the current couple of bits to be embedded has a **BCP** = “00” and the corresponding triple of bins has a **BTP** = 2, no permutation is performed and the couple is assumed as embedded successfully. This also applies to the situation, where the current couple of bits to be embedded has a **BCP** = “11” and the corresponding triple of bins has a **BTP** = 3. This of course, will reduce the computational complexity of our proposed CP watermarking scheme, especially when we have noticed that the two **BCPs**, “00” and “11”, are highly repeated in binary watermarks.

Because we embed the watermark in couples, it is required to have even watermark sizes. If the size of the watermark to be embedded is odd, the watermark is appended by a new bit, equals zero, from the right most side.

**Table 4.2: The embedding rule of our proposed CP watermarking scheme.**

Bits Couple Pattern (BCP)	The corresponding rule for embedding
“00”	<b>IF</b> (BTP = 0 <b>OR</b> BTP = 2), do nothing; <b>Otherwise</b> , permute the current bins triple until having a corresponding <b>BTP = 0</b> .
“01”	<b>IF</b> (BTP = 1), do nothing; <b>Otherwise</b> , permute the current bins triple until having a corresponding <b>BTP = 1</b> .
“10”	<b>IF</b> (BTP = 4), do nothing; <b>Otherwise</b> , permute the current bins triple until having a corresponding <b>BTP = 4</b> .
“11”	<b>IF</b> (BTP = 5 <b>OR</b> BTP = 3), do nothing; <b>Otherwise</b> , permute the current bins triple until having a corresponding <b>BTP = 5</b> .

We can embed the watermark in either histogram of the three color components of the host image. Or embed the watermark collectively in the histograms of the three color components, in a way, such that each color component carries a portion of the watermark bits. Anyway, we will explain the general idea of our histogram-based embedding process. The detailed steps for watermark embedding process of our proposed CP watermarking scheme are illustrated in Algorithm 4.9 bellow.

---

**Algorithm 4.9:** The embedding process of the proposed CP watermarking scheme.

---

**Purpose:** Watermark embedding, based on the permutation of the bins of the histogram of host color plane.

**Input:** The host color plane, the watermark to be embedded, and the *publicKey*.

**Output:** The watermarked image, and the private key, **PK**.

**Procedure:**

- a) The histogram of the host color plane is computed.
- b) Based on the *publicKey*, the values of *begin* and *step* are calculated using Equations 4.11, and 4.12 respectively.
- c) If no intensity triple is selected yet, an intensity triple (*a*, *b*, *c*) is calculated based on the values of *begin* and *step*, and using Equations 4.13, 4.14, and 4.15; otherwise, the next intensity triple, (*a*, *b*, *c*), is calculated using Equation 4.16. In either case, the selected intensity triple must have a corresponding strictly not equal histogram bins, or reject the selected triple and select the next one using Equation 4.16. An attention must be paid at each triple

selection; to avoid colloid with the previously selected triples, i.e., to avoid using a previously embedded triple or we will lose that old embedding.

- d) For each selected intensity triple,  $(a, b, c)$ , having a corresponding triple of bins,  $(\text{Hist}(a), \text{Hist}(b), \text{Hist}(c))$ , with bins triple pattern, **BTP**; a corresponding watermark's bit couple with bits couple pattern, **BCP**, is embedded according to the rule shown in Table 4.2. If the rule asks to permute the current triple of bins until having a new corresponding **BTP**, the image pixels with intensities  $a, b$ , and  $c$  are interchanged accordingly.
- e) The private key (**PK**), which is necessary for image restoration, is generated accumulatively during the watermark embedding. For each selected intensity triple,  $(a, b, c)$ , the original **BTP** is appended to the **PK**. (*This step is optional; it is done only if we are interested in restoring the original image, later*)
- f) Steps c, d, and e are repeated until all watermark bits are embedded in the color plane, or the algorithm reaches the capacity limit of the host color plane, i.e., all the candidate triples of the histogram bins are already embedded.

**END**

**Example illustrating watermark embedding process of the proposed CP scheme:**

Suppose we have a binary watermark  $W = "010011"$ , a host color plane, and a **publicKey** = 2.3. Now, to embed the watermark,  $W$ , into the host color plane by modifying its histogram based on the **publicKey**, we follow the following steps:

- a) The histogram of the host color plane is computed. Assume that the computed histogram is represented numerically as seen in Table 4.3 bellow.

**Table 4.3: Illustration Example, histogram of a host color plane.**

<b>No. of Pixels</b>	19	23	24	17	23	13	23	13	78	73	66	50	83	58
<b>Intensity Value</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13

- b) According to Equations 4.11 and 4.12, we compute  $begin = 2.3/1 = 2$ . And  $step = (2.3 \% 1) * 10 = 3$ .
- c) According to Equations 4.13 to 4.15, we calculate  $a = 2$ ;  $c = 2 + 3 = 5$ ;  $b = \left\lfloor \frac{2+5}{2} \right\rfloor = 3$ . Thereby, the intensity triple,  $(a, b, c) = (2, 3, 5)$ . According to the histogram shown in Table 4.3, this intensity triple has a corresponding triple of histogram bins,  $(24, 17, 13)$ , which has strictly non equal bins, and

interpreted as (*High, Between, Low*). According to Table 4.1, it has a **BTP** equals 5.

- d) The **BCP** of the first couple of the watermark, **W**, to be embedded, is “**01**”. According to the rule shown in Table 4.2, the current **BCP**=”01” and **BTP**=5 are inconsistent with that rule. Thus the current triple of histogram bins is permuted until obtaining the suitable **BTP**. In other words, the triple of histogram bins is permuted until having a **BTP** equals 1, in our example, which has the interpretation of (*Low, High, Between*). Thus, the pixels with intensities (2, 3, 5) are interchanged until having a corresponding triple of histogram bins equal (13, 24, 17). That is achieved by the interchanges: 5→2, 2→3, and 3→5. Where the arrow sign, →, means becomes.
- e) The private key, **PK**, is appended by the original **BTP** = 5, to be used later for image restoration.
- f) We continue to embed the rest of watermark bits, but now the next intensity triples are selected according to Equation 4.16. Also, at each triple selection we check to avoid colloid with the previously selected triples. Table 4.4 summarizes the process of embedding all the watermark bits.

**Table 4.4: Illustration Example, steps of embedding a binary watermark by modifying the histogram of the host color plane.**

Intensity Triple	Histogram Triple	BTP	PK	BCP	Rule BTP	Permute?	New H Triple
(2, 3, 5)	(24, 17, 13)	5	5	“01”	1	Yes	(13, 24, 17)
(6, 7, 9)	(23, 13, 73)	2	52	“00”	0 or left 2	No	(23, 13, 73)
(10,11,13)	(66,50,58)	4	524	“11”	5 or left 3	Yes	(66, 58, 50)

Finally, because our proposed CP scheme embeds the watermark in the histogram of the host image, the embedded watermark will be robust against any attack, which might infected the watermarked image, as long as it does not change its histogram. Also embedding by this strategy preserves the quality of the host image, because it merely interchanges close intensity values at the same image. Thus, histogram permutation is what we are searching for in our third and fifth research questions.

#### **4.2.4 The Watermark Extraction Process of the Proposed CP Watermarking Scheme**

It is a blind process, which aims at detecting the previously embedded watermark from the watermarked image without the presence of the original image. The detailed steps for extracting the watermark from a watermarked color plane are illustrated in

Algorithm 4.10. As seen in the algorithm, the public key and the watermark size, along with the watermarked plane, are the parameters needed for watermark extraction process. The extracted watermark is the process output.

---



---

**Algorithm 4.10:** The extraction process of the proposed CP watermarking scheme.

---



---

**Purpose:** Watermark extraction, based on the histogram of watermarked color plane.

**Input:** The watermarked color plane, watermark size,  $S$ , and the *publicKey*.

**Output:** The extracted watermark,  $EW$ .

**Procedure:**

- a) The histogram of the watermarked color plane is computed.
- b) Based on the *publicKey*, the values of *begin* and *step* are calculated using Equations 4.11, and 4.12 respectively.
- c) If no intensity triple is selected yet, an intensity triple  $(a, b, c)$  is calculated based on the values of *begin* and *step*, and using Equations 4.13, 4.14, and 4.15; otherwise, the next intensity triple,  $(a, b, c)$ , is calculated using Equation 4.16. In either case, the selected intensity triple must have a corresponding strictly not equal histogram bins, or reject the selected triple and select the next one. An attention must be paid at each triple selection; to avoid colloid with the previously selected triples, i.e., to avoid extracting from a previously used triple or we will duplicate the extracted bits.
- d) From each selected intensity triple,  $(a, b, c)$ , having a corresponding triple of bins,  $(\text{Hist}(a), \text{Hist}(b), \text{Hist}(c))$ , with bins triple pattern, **BTP**, the embedded couple of bits, with bits couple pattern, **BCP**, is extracted according to the rule shown in Table 4.5, which is reverse of the embedding rule in Table 4.2. The extracted couple of bits are appended to the extracted watermark,  $EW$ .
- e) Steps c and d are repeated until all watermark bits are extracted, i.e., until the size of the extracted watermark,  $EW$ , equals  $S$ .

**END**

---



---

**Table 4.5:** The extraction rule of our proposed CP watermarking scheme.

<b>BTP</b>	<b>The corresponding extracted Bits Couple Pattern, BCP.</b>
<b>IF BTP = 0 OR BTP = 2</b>	<b>BCP = "00"</b>
<b>IF BTP = 1</b>	<b>BCP = "01"</b>
<b>IF BTP = 4</b>	<b>BCP = "10"</b>
<b>IF BTP = 5 OR BTP = 3</b>	<b>BCP = "11"</b>

**Example illustrating watermark extraction process of the proposed CP scheme:**

For better understanding, let's return to the example discussed in the embedding process. Therefore, assume that we have obtained the watermarked plane, the size of the watermark,  $S = 6$ , and the *publicKey* = 2.3. Then, to extract the previously embedded watermark we follow the following steps:

- a) The histogram of the watermarked plane is computed. In our example, it is the histogram, which was generated in the embedding process, after watermark embedding. The histogram is seen in Table 4.6 bellow.

**Table 4.6: Illustration Example, histogram of the watermarked plane.**

<b>No. of Pixels</b>	19	23	13	24	23	17	23	13	78	73	66	58	83	50
<b>Intensity Value</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13

- b) Using Equations 4.11 and 4.12, those based on the *publicKey*, we compute the values of  $begin = 2.3/1=2$  and  $step = (2.3\%1)*10= 3$ .
- c) According to Equations 4.13 to 4.15, we calculate  $a = 2$ ;  $c = 2 + 3 = 5$ ;  $b = \left\lfloor \frac{2+5}{2} \right\rfloor = 3$ . Thereby, we obtained the intensity triple,  $(a, b, c) = (2, 3, 5)$ . As seen in Table 4.6, this intensity triple has a corresponding triple of histogram bins,  $(13, 24, 17)$ , which has strictly non equal bins, and interpreted as,  $(Low, High, Between)$ . According to Table 4.1, it has a **BTP** equal to 1.
- d) According to the extraction rule, shown in Table 4.5, knowing that **BTP** equals 1, we conclude that the previously embedded couple of bits, has a **BCP** = "01". We append that couple to the extracted watermark, *EW*.
- e) We continue until extracting the whole watermark of size,  $S=6$ , but the next intensity triples are selected according to Equation 4.16, at each triple selection we check to avoid colloid with the previously selected triples. Table 4.7 summarizes the process of extracting all the watermark bits.

**Table 4.7: Illustration Example, steps of extracting a binary watermark based on the histogram of the watermarked color plane.**

<b>Intensity Triple</b>	<b>Histogram Triple</b>	<b>BTP</b>	<b>Extracted BCP</b>
(2, 3, 5)	(13, 24, 17)	1	"01"
(6, 7, 9)	(23, 13, 73)	2	"00"
(10,11,13)	(66, 58, 50)	5	"11"

Finally, the extraction process can only be successful if its selected triples are same as those selected during the embedding process.



#### 4.2.5 The Restoration Process of the Proposed CP Watermarking Scheme

In some applications, it is needed to restore the watermarked image after being verified. Restoration is the process of recovering the original, un-watermarked, image from its watermarked version. The detailed steps of the restoration process of our proposed CP watermarking scheme are listed in Algorithm 4.10. This process mainly depends on the private key, **PK**, which was generated during the watermark embedding process; it carries the original list of bins triple patterns **BTPs**.

---

---

**Algorithm 4.11:** The restoration process of the proposed CP watermarking scheme.

---

---

**Purpose:** Recovering the original, un-watermarked, image from the watermarked one.

**Input:** The watermarked plane, the *publicKey*, and the private key, **PK**.

**Output:** Restored Image.

**Procedure:**

- a) The histogram of the watermarked plane is computed.
- b) Based on the *publicKey*, the values of *begin* and *step* are calculated using Equations 4.11 and 4.12 respectively.
- c) If no intensity triple is selected yet, an intensity triple (*a*, *b*, *c*) is calculated based on the values of *begin* and *step*, and using Equations 4.13, 4.14, and 4.15; otherwise, the next intensity triple, (*a*, *b*, *c*), is calculated using Equation 4.16. In either case, the selected intensity triple must have a corresponding strictly not equal histogram bins, or reject the selected triple and select the next one. An attention must be paid at each triple selection; to avoid colloid with the previously selected triples.
- d) The bins triple pattern, **BTP**, of the selected intensity triple, (*a*, *b*, *c*), is compared with the corresponding bins triple pattern, **BTP**, which was stored in the private key, **PK**. If a match is found, then the **BTP** of the selected intensity triple is original and left as it. Otherwise, the **BTP** of the selected intensity triple is permuted to match that in **PK**; and consequently, pixels with intensities (*a*, *b*, *c*) are interchanged according to that permutation.
- e) Steps c and d are repeated until reaching the end of the private key, **PK**.

**END**

---

---

**Example illustrating the restoration process of the proposed CP scheme:**

Also here, we proceed with the same previously mentioned example. Assume that we have obtained the watermarked plane, the private key, **PK = 524**, and the **publicKey = 2.3**. Also, assume that the watermarked plane is verified to be true in the extraction process. To recover the original, un-watermarked, plane based on the watermarked plane, we follow the following steps:

- The histogram of the watermarked plane is computed. It can be seen in Table 4.6, which was viewed in the previous example.
- Using Equations 4.11 and 4.12, and based on the **publicKey**, we compute the values of **begin**= 2.3/1=2 and **step**= (2.3%1)\*10= 3.
- Based on the values of **begin** and **step**, and according to Equations 4.13 to 4.15, we calculate **a = 2; c = 2 + 3 = 5; b =  $\lfloor \frac{2+5}{2} \rfloor = 3$** . Thereby, we obtained an intensity triple, **(a, b, c) = (2, 3, 5)**.
- The intensity triple, **(2, 3, 5)**, has a corresponding triple of strictly non equal histogram bins, **(13, 24, 17)**, and interpreted as, **(Low, High, Between)**. According to Table 4.1, it has a **BTP** equal 1. But, the value of the corresponding bins triple pattern, **BTP**, which was stored in the private key, **PK**, is **BTP=5**, and interpreted as, **(High, Between, Low)**. Therefore, we need to permute **BTP** from 1 to 5. Thus, the pixels with intensities **(2, 3, 5)** are interchanged until having a corresponding triple of histogram bins equal **(24, 17, 13)**. That is achieved by the interchanges: **2→5, 3→2, and 5→3**.
- We continue to recover the rest of bins triple patterns, but from now, the next intensity triples are selected according to Equation 4.16. Also, at each triple selection we check to avoid colloid with the previously selected triples. Table 4.8 summarizes the process of recovering the original histogram. Table 4.9 shows the recovered histogram, it is identical to the original one, Table 4.3.

**Table 4.8: Illustration Example, steps for restoring the original histogram.**

Intensity Triple	Histogram Triple	BTP	PK's BTB	Permute?	New H Triple
(2, 3, 5)	(13, 24, 17)	1	5	Yes	(24, 17, 13)
(6, 7, 9)	(23, 13, 73)	2	2	No	(23, 13, 73)
(10,11,13)	(66, 58, 50)	5	4	Yes	(66,50,58)

**Table 4.9: Illustration Example, recovered histogram of the watermarked plane.**

No. of Pixels	19	23	24	17	23	13	23	13	78	73	66	50	83	58
Intensity Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13

### **4.3 The Proposed Multipurpose Watermarking Scheme for Both CA and CP of Color Images**

In this section we discuss our third proposed watermarking scheme; namely, the multipurpose watermarking scheme for both CA and CP of color images.

#### **4.3.1 Overview**

The first proposed watermarking scheme, CA watermarking scheme, detects any alteration in images, whether it is intentional or not, and even if it is small. In many applications, those are characterized as sensitive applications, like medical and military applications; images are abandoned once it is proved that they were maliciously modified. There are some other image modifications, where images may still usable, despite of being modified. For example, flipping medical images does not hide its content. Those modifications always descend from the geometrical attacks. Thus, it is needed to distinguish the attack type in such applications. Unfortunately, despite of its proved excellence, distinguishing between the benign and malicious attacks is not possible using our proposed CA watermarking scheme alone.

The second proposed watermarking scheme, CP watermarking scheme, can be used to solve this problem. It provides a watermarking mechanism, which allows embedding of a CP watermark in images, such that it is robust to geometrical attacks those change pixels positions. After that, if the identification of that embedded CP watermark is not possible; this means it is destroyed by a malicious attack, and hence, the image is attacked maliciously. On contrast, if we could identify the CP watermark, this means the image is either not modified, or geometrically transformed. Unfortunately, after identifying the CP watermark, our proposed CP watermarking scheme, alone, could not distinguish whether the image is geometrically transformed, or not modified.

Thus, the primary objective of our third proposed watermarking scheme is to find a multipurpose image watermarking algorithm for both CA and CP of color images. In doing so, the two aforementioned and proposed watermarking schemes in sections 4.1 and 4.2 are cascaded one after another, each of which is responsible for embedding a special purpose watermark in the host image. The first will embed the CA watermark, which will sense any image modification despite of its type. The second will embed the CP watermark, which will be robust to some geometrical attacks; those change only the positions of the pixels. Cascading these two proposed schemes is possible,

because cascading is the main factor taken into consideration during their developments in our work.

***The idea of our proposed multipurpose watermarking scheme works as follows:***

Suppose that a specialist received a medical image, which was embedded with both the CA watermark, and the CP watermark using our proposed multipurpose watermarking scheme. The disability of the receiver to identify the CP watermark of the received image means that it is indeed attacked by a malicious attack; here the medical image should be ignored. In contrast, identifying the CP watermark of the received image does not necessarily mean that it is not modified. It could be geometrically transformed. Determining whether the received image is transformed or not, is assigned to the extracted CA watermark, which will notify the receiver with the presence or the absence of any modifications.

Thereby, in this phase of work, multipurpose phase, we combine the evolved features of the proposed schemes in the previous two phases; namely, CA phase, and CP phase to obtain a multipurpose watermarking scheme. As seen, by cascading, each embedded watermark plays an individual role; we could take a decision by examining only one extracted watermark. Also, by cascading, our proposed multipurpose scheme is easily implemented. Thus, it is our preferred mechanism to achieve the indented idea of the proposed multipurpose scheme. Also, the preferred watermark embedding order is to embed the CA watermark, then the CP watermark. Of course, the watermarks are extracted in the reverse order. By this order, we actually saving time; because once we discovered that the extracted CP watermark is anonymous, we ignore that image, there is no need to extract the CA watermark. The CA watermark is only extracted if the CP watermark of the received image is authentic. With this, the sixth research question of this thesis is answered. Finally, it is important to note that the used mechanism and the watermark embedding order are application-based issues.

***Features of our proposed multipurpose watermarking scheme:***

Our proposed multipurpose watermarking scheme has the following features:

- ✓ Provides CP for color images (robust to some geometrical attacks).
- ✓ Provides CA for color images (with high sensitivity to attacks).
- ✓ Excellent watermarked image quality, despite of embedding two watermarks in it.
- ✓ Fragile: CA watermarking provides a precise and overall tampering localization.
- ✓ Robust: CP watermarking is robust against some geometrical attacks.

- ✓ Invisible: embeds watermarks in a way, such that they are imperceptible to HVS.
- ✓ Reversible: watermarked image can be restored to its original state completely.
- ✓ Blind: the watermark can be extracted apart from the original image.
- ✓ Provides high randomization and dynamicity degree in CA watermarking.
- ✓ Has a low computational cost, fast embedding speed, high embedding capacity, and easy to implement.

In the proposed multipurpose watermarking scheme, by the reversibility of the second embedded watermark, CP watermark, we guarantee removing its effect on the first embedded watermark, CA watermark. Thus, reversibility is the property we are searching for in the fourth research question of this thesis. Our proposed scheme allows watermark embedding in a host image, and watermark extraction from a watermarked image; while allowing reversibility to fully restore the original host image, despite of being embedded by two watermarks. Finally, our proposed multipurpose approach works robustly with attacks those change pixels positions.

#### 4.3.2 The Conceptual Model of the Proposed Multipurpose Watermarking Scheme

The third proposed watermarking scheme is a multipurpose watermarking scheme. It is used for watermarking color images aiming at both identifying whether or not images are modified, and distinguishing between malicious attacks and incidental manipulations. The model of the proposed technique is shown in Figure 4.9.

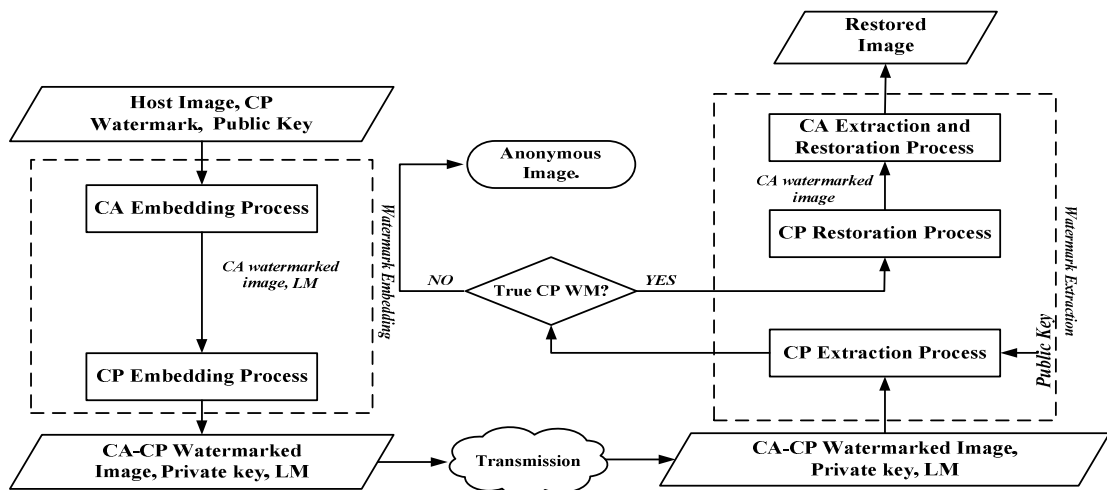


Figure 4.9: The conceptual model of the proposed multipurpose watermarking scheme.

As seen in Figure 4.9, the main idea of our proposed multipurpose watermarking scheme is simply cascading our previously proposed schemes; CA watermarking

scheme, and CP watermarking scheme one after another for both watermark embedding and extraction processes. The first main process in our proposed model is the CA embedding process, which is responsible for embedding a fragile watermark into the host image. The main output of this process is the CA watermarked image, which is then entered to the next process in the model, CP embedding process, which is responsible for embedding the CP watermark in a way, such that it is robust to geometrical attacks those change pixels positions. The main output of this process is CA-CP Watermarked Image, which carries both robust and fragile watermarks.

On the other hand, the extraction processes are also cascaded on the received image, but now the order is reversed. We begin by CP extraction process, which was conducted lastly. This process is responsible for extracting the CP watermark, which is then verified. If it is found to be incorrect, the received image is assumed an anonymous image. Otherwise, the CP restoration process recovers the CA watermarked image, which is entered to the CA extraction and restoration process, which generates the restored image that is either an original or a geometrically transformed image. All the processes used in this model are discussed previously in sections 4.1 and 4.2; thus, there is no need for their repetition.

Finally, in our proposed multipurpose watermarking scheme, we combine the CA and the CP watermarking schemes, each of which embeds the watermark using the spatial domain. Thus, we obtained both CP and CA of images, using only spatial domain techniques. By this, we answered the second research question of this thesis.

## CHAPTER 5

### EXPERIMENTAL RESULTS

The experimental results of the proposed watermarking schemes are listed in this chapter. The performance of our proposed schemes is explored through comparing them with the previous related works. All tests are performed using a laptop running windows XP operating system, with a 2 GHz core 2 duo processors, 2 GB memory, and 384 MB display adapter. Matlab 7.8 and Visual Studio 2011 are the main software components used in our work.

The broad goal of our proposed schemes is to protect color images; therefore, a general purpose image database is obtained, which includes 1000 color images, those are given in IPEG format, with size 384 x 256 or 256 x 384. This database was previously used in [85], and we downloaded it from [86]. Database images are grouped into ten categories; including, African people, Beach, Buildings, Cars, Dinosaurs, Elephants, Flowers, Houses, Mountains, and Food. Also, because our proposed schemes may be directed to medical applications; a set of 50 colored medical images is established and joined to the aforementioned database, under a new category named, Medical. This medical set is obtained randomly from the Science Photo Library [87], and given in JPEG format, in RGB color space. Therefore, our proposed schemes are evaluated using a dataset, which contains 1050 samples.

For the purpose of visual demonstrations; we have formed a set of ten samples, shown in Figure 5.1; each sample is selected randomly from its category.



Figure 5.1: A set of randomly selected samples, for the purpose of demonstrations.

Finally, in evaluation of our proposed watermarking schemes, we used some performance measurements. Including, Normalized Cross Correlation (NCC, Equation 2.9), Embedding Capacity (EC, Equation 2.10), Mean Square Error (MSE, Equation 2.11), and Peak Signal to Noise Ratio (PSNR, Equation 2.12).



## 5.1 Experimental Results of the Proposed CA Watermarking Scheme

In this section, the experimental results of the first proposed watermarking scheme are demonstrated and analyzed. Since our proposed scheme is a development of Poonkuntran and Rajesh scheme [1], different comparisons are conducted to evaluate our proposed scheme relative to their proposed scheme. Therefore, we implemented Poonkuntran and Rajesh scheme. And the following subsections demonstrate the results of our tests.

### 5.1.1 Comparing the Embedding Capacity

Each sample in our dataset is embedded by the watermark, which is generated using the messy system, which is based on the green component of the sample. Watermark embedding is performed using Poonkuntran and Rajesh scheme on the one hand, and using our proposed CA scheme on the other hand. The percentage of the embedded bits at each sample is recorded. The average of that percentage is determined for each image category. Table 5.1 summarizes the obtained results for both Poonkuntran and Rajesh scheme, and our proposed CA scheme. From which we conclude that the average embedding capacity for all image categories is 81.71% when using Poonkuntran and Rajesh scheme, and is 93.82% when using our proposed CA scheme. Therefore, for each introduced watermark to our proposed scheme, about 90 % of the watermark bits will be embedded, while, for each introduced watermark to their proposed scheme, about 80 % of the watermark bits will be embedded.

**Table 5.1: Comparing the average embedding capacity between Poonkuntran and Rajesh scheme and our proposed CA scheme.**

Category Name	Average (%) of embedded bits, using P. and R. scheme	Average (%) of embedded bits, using our proposed CA scheme
African people	89.24	96.07
Beach	85.93	95.09
Buildings	86.82	92.24
Cars	75.38	86.64
Dinosaurs	93.67	97.04
Elephants	89.82	95.96
Flowers	72.69	98.01
Houses	88.06	92.80
Mountains	86.77	94.52
Food	66.09	92.57
Medical	64.28	91.03
<b>Average</b>	<b>81.71</b>	<b>93.82</b>

The interpretation for this increase at the embedding capacity, in our proposed CA scheme, is due to the use of inter-plane difference expansion, rather than using intra-

plane difference expansion for watermark embedding. In using inter-plane difference expansion, we embed each watermark bit by expanding the difference between two neighboring pixels, which are, naturally, highly correlated. This leads to a small difference, which is mostly expandable. In other words, the new generated pixels, based on that small difference, will remain in the interval  $[0, 255]$ . And hence, using inter-plane difference expansion, leads a larger amount of expandable differences than those obtained when using intra-plane difference expansion, as in Poonkuntran and Rajesh scheme.

Finally, Poonkuntran and Rajesh mentioned that the embedding capacity of their scheme can be increased by using a multilayer difference expansion. Means, the same pairs of pixels are selected for further data embedding. But, it is worth mentioning that, this solution can also further increases the embedding capacity of our proposed CA scheme. Thus, for a fair comparison, our recorded results are all based on a single layer embedding for both schemes. Means, each pixels pair is used only once.

### **5.1.2 Comparing the Effect of Watermark Embedding on the Host Image**

Sometimes, the presence of a watermark in the host image might distort its quality. Therefore, in this section, the quality of the watermarked image is our main concern. The PSNR, seen in Equation 2.12, is used as a measurement of the quality of the watermarked image. Particularly, PSNR is used to measure how much the watermarked image is similar to the original, un-watermarked, image. The bigger the PSNR value, the better the watermark invisibility. Generally, if the PSNR is not less than 30 dB, the processed image is assumed within acceptable degradation levels, and is acceptable to the HVS [64].

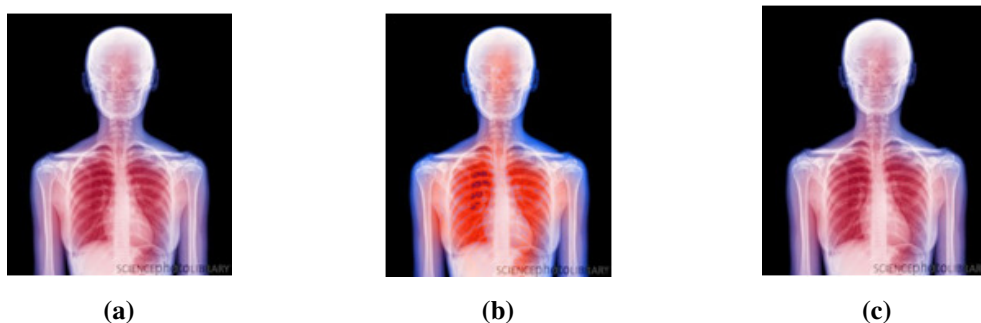
To compare our proposed scheme with Poonkuntran and Rajesh scheme [1], each of the 1050 samples in our dataset is watermarked by its corresponding generated watermark using the messy system. But because the quality, PSNR, is proportional with the watermark size; for fair comparisons, each sample is embedded by a fixed watermark size using either of the two watermarking schemes. The quality, PSNR, of each watermarked sample is calculated and recorded. Also, the average PSNR is calculated per each category. Table 5.2 shows the obtained results of this experiment. As seen in the table, for each category, our proposed CA scheme has higher average PSNR than that obtained using Poonkuntran and Rajesh scheme. Also, over all sample categories, our proposed CA scheme has an average PSNR equals 34.09 dB, while it

is 29.37 dB for Poonkuntran and Rajesh scheme, which is under the level of acceptable degradation, 30 dB. Thus, we conclude that Poonkuntran and Rajesh scheme fails to generate high quality watermarked images when the watermark size is large. This contrasts with the fact that, large watermark size is a mandatory issue in CA watermarking schemes. Finally, in our proposed CA scheme, we obtained a minimum PSNR of 31.76 dB, which is 4.61 dB larger than the minimum PSNR of 27.15 dB obtained by using Poonkuntran and Rajesh scheme.

**Table 5.2: Comparing the quality, PSNR, of the watermarked image for both Poonkuntran and Rajesh scheme, and our proposed CA scheme.**

Category Name	Average PSNR (dB), using P. and R. scheme	Average PSNR (dB), using our proposed CA scheme
African people	28.93	32.82
Beach	27.72	33.80
Buildings	28.90	31.76
Cars	30.48	32.62
Dinosaurs	32.52	35.55
Elephants	27.15	32.67
Flowers	31.66	40.38
Houses	29.30	32.04
Mountains	29.02	33.70
Food	27.85	33.32
Medical	29.60	36.29
<b>Average</b>	<b>29.37</b>	<b>34.09</b>

An example demonstrating how the watermarked image is affected when embedded by a large watermark size using Poonkuntran and Rajesh scheme in one hand and using our proposed CA scheme on the other hand is shown in Figure 5.2 bellow.



**Figure 5.2: Comparing the quality of the watermarked image for both Poonkuntran and Rajesh scheme, and our proposed CA scheme, after being embedded with a large watermark size. (a) Original, un-watermarked, image. (b) Watermarked image using Poonkuntran and Rajesh scheme, PSNR= 25.60. (c) Watermarked image using proposed CA scheme, PSNR= 35.75.**

In this experiment, the original un-watermarked sample, shown in part (a) of the figure, is embedded by a watermark of size equals 90204 bits. A highly distorted watermarked image, with PSNR= 25.60 dB, is generated when using Poonkuntran

and Rajesh scheme, as seen in part (b) of the figure. While, a high quality watermarked image, seen in part (c) with PSNR= 35.75 dB, is obtained when using our proposed CA watermarking scheme. Due to lack of quality; looking at part (b), one may erroneously think of a presence of a new disease. Finally, the PSNR obtained using our scheme is larger than that obtained using their scheme by 10.15 dB.









































This increase in the quality of the watermarked images, using our proposed CA scheme, is due to the homogeneity of its embedding strategy, by which, pairing is based on neighboring pixels at the same color plane, rather than different color planes. This leads to smaller differences, and hence watermark embedding is performed by replacing old pixels with new pixels those are very close to the old ones.

### **5.1.3 Comparing Watermark Spreading and Fragility against Small Modifications**

#### **(a) Comparing Watermark Spreading**

As mentioned before, in many applications; it is required to detect any modification in images, even if it is very small, in a way for countering the attacker, who is interested in a certain small important portions of the watermarked images. The only way to survive such attacks is by allowing the watermark to spread over the whole image. Whenever the watermark spreads enough, the ability to detect modifications anywhere is increased, as well as the fragility of the watermarking scheme increased. This part is dedicated to compare the spreading of the watermark when being embedded using Poonkuntran and Rajesh scheme from one side, and when using our proposed CA watermarking scheme from the other side. At this experiment, each of the ten samples shown in Figure 5.1 is embedded by the maximally possible watermark size using either of the two watermarking schemes. The watermark spreading degree, which is generated when using each watermarking scheme is captured and listed in Table 5.3. As seen in the table, watermark spreading degree is expressed by referring to the embedded locations at the host image. Note that, the embedded locations are represented by black color, while non-embedded locations are left white. Also, the percentage of the embedded locations, (Total number of embedded locations/ Total number of image locations), is mentioned under each watermark spreading image. The value of the PSNR is displayed under each watermarked image.

**Table 5.3: Comparing the watermark spreading in the watermarked images using both Poonkuntran and Rajesh scheme, and our proposed CA scheme.**

No.	P. and R. Watermarked image	P. and R. Embedded locations	The proposed CA Watermarked image	The proposed CA Embedded locations
1	 PSNR= 23.06	 55.89%	 PSNR= 31.15	 91.52%
2	 PSNR= 23.84	 45.24%	 PSNR = 25.59	 95.21%
3	 PSNR= 27.82	 45.61%	 PSNR= 30.14	 87.8%
4	 PSNR= 24.59	 84.91%	 PSNR= 30.36	 95.94%
5	 PSNR=22.1	 83.93%	 PSNR= 28.84	 95.2%
6	 PSNR= 32.98	 46.51%	 PSNR= 37.24	 99.01%
7	 PSNR=27.87	 90.2%	 PSNR= 30.22	 97.22%
8	 PSNR=24.98	 85.49%	 PSNR= 29.66	 95.98%
9	 PSNR=28.48	 63.56%	 PSNR= 28.51	 91.17%
10	 PSNR=25.60	 86.49%	 PSNR=35.31	 93.68%



The average percentage of the watermark spreading over all the watermarked images using Poonkuntran and Rajesh scheme is 68.78 %. Thus, only about 68.78 % of pixels of the watermarked image are protected using Poonkuntran and Rajesh scheme. Whereas, the average percentage of the watermark spreading over all the watermarked images using our proposed CA scheme is 94.27 %, which is the percentage of the protected pixels in our proposed CA scheme. Generally, the percentage of the watermark spreading varies from one image to the other, based on the amount of expandable locations found on it.

It is worth mentioning that, even with larger watermark sizes embedded using our proposed CA scheme, we obtained watermarked images with higher quality than those obtained when using Poonkuntran and Rajesh scheme.

### **(b) Comparing the Fragility against Small watermarks Modifications**

As seen in Table 5.3, in contrast to our proposed CA scheme, the embedded watermark using Poonkuntran and Rajesh scheme is limited. Actually, the vulnerability of the watermarking scheme is not only on the limitation of the watermark spreading, but also related to where the watermark is concentrated. In protecting images of valuable contents, watermarks need to spread over whole image's area, or at least concentrate at valuable positions of the image. In view of the foregoing; some of the aforementioned watermarked samples using Poonkuntran and Rajesh scheme are assumed as not protected yet. Take for example, Sample 6, in which the watermark is concentrated at the background of the sample. Therefore, any alteration on the un-watermarked space will not be detected. Thus, this watermarking is useless. Practically, Figure 5.3 (a) shows an original image, (b) shows a modified Poonkuntran and Rajesh scheme-based watermarked version of (a) , and (c) shows a modified CA scheme-based watermarked version of (a). The error image between the generated and the extracted watermarks in Poonkuntran and Rajesh scheme is zero anywhere, and hence, it does not sense the modifications as seen in (d). While, the modifications are detected precisely using our proposed CA scheme, as seen in (e). The failure to detect this attack in Poonkuntran and Rajesh scheme is due to the limited embedding locations, rather than being spread as in our proposed scheme. Thus, we can say that our proposed CA scheme is exclusive in detecting small modifications.

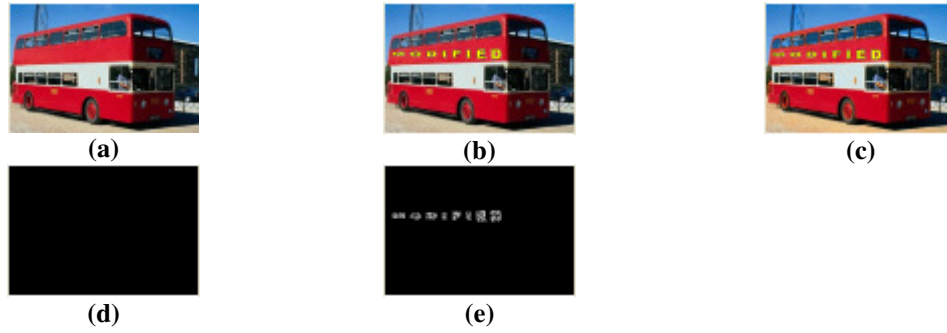


Figure 5.3: Example demonstrating the benefit of watermark spreading, in detecting image modifications. (a) Original image. (b) Watermarked image using P. and R. scheme, attacked. (c) Watermarked image using the proposed CA scheme, attacked. (d) Watermarks error image of P. and R. scheme. (e) Watermarks error image of the proposed CA scheme.

### 5.1.4 Comparing the Fragility against Attacks

Fragility refers that the embedded watermark should be sensitive to various attacks [1]. Means, the watermark should be easily destroyed when the host image is modified. Fragile watermarks can provide information about image completeness [6]. In this section, we compare the degree of fragility of Poonkuntran and Rajesh scheme, with that of our proposed CA scheme. In this experiment, each sample in our dataset is watermarked by its corresponding generated messy watermark. A list of attacks is used in this experiment, each of which is applied to every previously generated watermarked image. The percentage of defect is recorded at every attacked image, after applying each attack. It is the percentage of mismatches between the generated and the extracted watermarks of the attacked image. The average of that percentage is estimated over all attacked images, for each attack type. This experiment is repeated two times, by applying one of the two watermarking schemes at each time. The results obtained in this experiment are shown numerically in Table 5.4 below.

Table 5.4: Comparing fragility against different attacks between Poonkuntran and Rajesh scheme, and our proposed CA scheme.

Attack Name	Percentage of defect for P. and R. scheme.	Percentage of defect for CA scheme.
Random Jitter	45.54	49.13
Rotation	51.38	51.06
Average Filter	51.65	54.03
Disk Filter	50.38	52.76
Gaussian Filter	40.19	41.58
Laplacian Filter	50.43	51.13
Log Filter	50.74	51.18
Motion Filter	49.24	49.86
Prewitt Filter	52.19	53.77
Sobel Filter	52.36	53.94
Un-sharp Filter	50.07	50.15
<b>Average</b>	<b>49.47</b>	<b>50.78</b>



As seen in the table, although it is a slight difference, our proposed CA scheme outperforms Poonkuntran and Rajesh scheme in sensitivity to attacks. Generally speaking, we can say that approximately both schemes have an equal degree of fragility against attacks. Using either scheme for generating a watermarked image; about 40-50 % of its embedded watermark will be destroyed after being attacked.

Finally, in this experiment, each attack is applied with strength ranges from 5% to 95%, and then the results are averaged to get a single value for each attack application. For example, the rotation attack is applied in the range, ( $9^\circ$  to  $171^\circ$ ), which is 5% to 95% of the total rotation range, ( $0^\circ$  to  $180^\circ$ ).

### 5.1.5 Comparing the Embedding Time

Poonkuntran and Rajesh watermarking scheme, and our proposed CA watermarking scheme, are used in turn to embed a watermark into each of the 1050 samples in our dataset. For each sample, the watermark is generated using the messy system, which is based on the sample's green color plane. Also, for fair comparison, the size of each embedded watermark has been fixed, so that it is equal for both schemes. The embedding time for each sample is calculated and recorded. Finally, we calculated the average embedding time for each category in our dataset. Figure 5.4 shows the obtained results of this experiment.

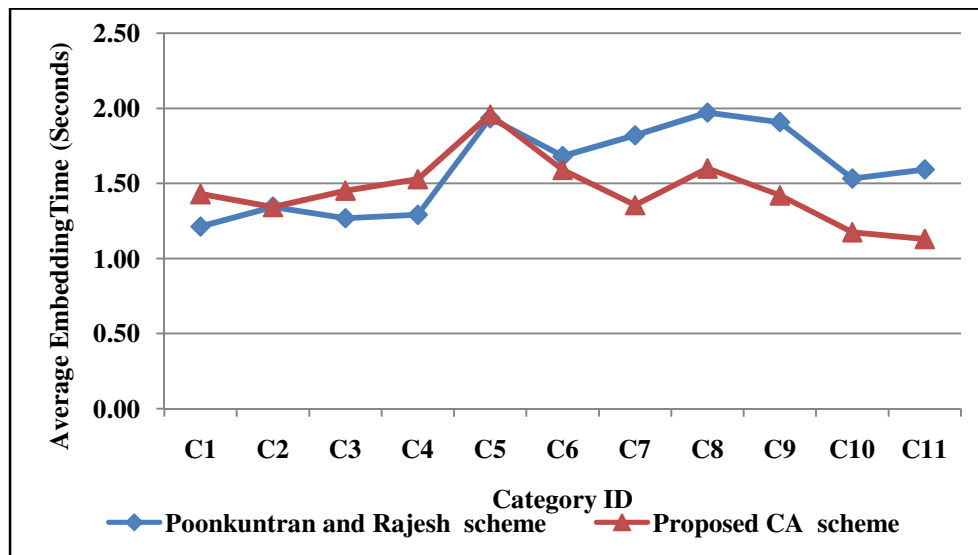


Figure 5.4: Comparing average watermark embedding time, between Poonkuntran and Rajesh watermarking scheme, and our proposed CA scheme; based on different watermark sizes.

As seen in Figure 5.4, at most of sample categories, our proposed CA algorithm is superior to, or at least the same as, Poonkuntran and Rajesh algorithm, regarding the average embedding time, despite a small difference. At the other categories, namely,

C1, C3, and C4, the average embedding time of our proposed algorithm exceeds that of their algorithm by no more than 0.25 second. Finally, the recorded average embedding time for all samples, using our proposed CA scheme is 1.45 seconds, while using Poonkuntran and Rajesh scheme is 1.60 seconds.

During each watermark embedding process, both schemes visit all the locations of the host image; forming pairs of pixels. At each watermark bit embedding, the expandability of the difference of the corresponding pixels pair is checked. If it is found expandable, the bit is embedded by replacing the pixels of that pair. Otherwise, both the bit and its corresponding pixels pair are ignored, and both schemes move to the next bit and pixels pair. Thus, at each watermark bit embedding, the corresponding image location is accessed two times. The first time is merely for reading the pixels pair, the second is for writing the new pixels pair. Because Poonkuntran and Rajesh watermarking scheme forms pixels pairs among two different color planes of the host image, while our proposed CA watermarking scheme forms pairs using the neighboring pixels at the same color plane, our proposed CA algorithm is faster than Poonkuntran and Rajesh algorithm.

### **5.1.6 Comparing the Overhead Due to Using the Location Map**

The role of the location map in both Poonkuntran and Rajesh scheme, and our proposed CA watermarking scheme is to refer to the locations at the host image where the embedding occurs. The ability to embed a bit in Poonkuntran and Rajesh watermarking scheme depends on the expandability of the corresponding difference between pixels pair of different color planes. While, in our proposed CA watermarking scheme, the ability to embed a bit depends on the expandability of the corresponding difference between the neighboring pixels pair at the same color plane. Anyway, at both schemes, for each watermark bit, a value of “1” is assigned in the location map to correspond to an expandable difference. Otherwise a value of “0” is assigned. Thus, the size of the location map is equal to the number of differences, or pixel pairs. Because, at both schemes, each watermark bit has a corresponding pixel pair; the size of the location map at both schemes is the same, which is equal to the size of the binary watermark. For each watermarked image, the corresponding location map is kept with the legitimate users. Later, if those users need to verify a received watermarked image, they use its corresponding location map. Finally, by the aid of the location map, the False Positive Rate of the CA scheme is zero.

## 5.2 Experimental Results of the Proposed CP Watermarking Scheme

The experimental results of the second proposed watermarking scheme are demonstrated and analyzed in this section. To confirm its position among the others; our proposed CP watermarking scheme is compared with its counterpart, Chrysochos et al. watermarking scheme [34], which was developed in generating our proposed CP watermarking scheme. So, we implemented Chrysochos et al. scheme, and then our tests are performed based on the 1050 sample images of our dataset. The three color planes of the host image share the process of embedding; each color plane takes a portion of the watermark according to its capacity.

Different tests are performed including, measuring and comparing the embedding capacities, comparing the two schemes regarding the effect of the embedding process on the quality of the watermarked image, comparing the embedding times, and finally the robustness of the embedded watermarks to some geometrical attacks is evaluated. The results of our tests and analysis are displayed in the subsequent subsections.

### 5.2.1 Comparing the Embedding Capacity

Theoretically, the maximum possible payload capacity per color plane using our proposed CP scheme is 170 bits. Particularly, we have  $(256/3) = 85$  triples of histogram bins, each of which will be embedded with a couple of bits. On the other hand, the maximum possible payload capacity per color plane using Chrysochos et al. scheme it is 128 bits. Particularly, we have  $(256/2) = 128$  pair of histogram bins, each of which will be embedded with a single bits. If the three color planes are used for embedding, these capacities may rise to three times the old capacities, i.e., it becomes 510 bits in our scheme, and 384 bits in their scheme. Anyway, we conclude that our proposed CP scheme outperforms Chrysochos et al. scheme, regarding the maximum payload capacity.

Practically, Chrysochos et al. watermarking scheme and our proposed CP watermarking scheme, each is used in turn, to embed a watermark of size 512 bits into each of the samples in our dataset. A variety of public keys, (5.2, 10.3, 25.4, 50.5, 75.6, 100.7, 125.8, 150.9, 175.2, 200.5, 250.7, and 255.9), are used in this experiment. Thus, the embedding capacity for each watermarked sample is the average of the obtained capacities, using those different public keys. Also, the average embedding capacity is calculated over each category in our dataset. Figure 5.5 shows the results of this experiment. As seen in the figure, in each category, the maximum payload

capacity obtained when using our proposed CP watermarking scheme surpasses that of Chrysochos et al. watermarking scheme. Numerically, the average payload capacity, over all samples in the dataset, using our proposed CP scheme is 417 bits, while it is 362 bits when using Chrysochos et al. scheme. Thus, our proposed CP watermarking scheme increased the average payload capacity, per each sample, by about 55 bits.

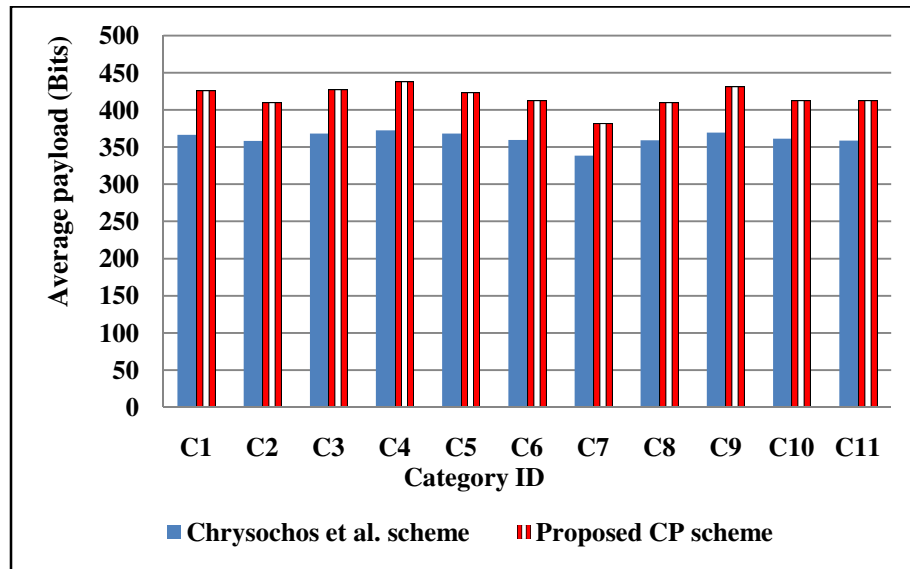


Figure 5.5: Comparing the average embedding capacity between Chrysochos et al. scheme, and our proposed CP scheme.

The interpretation for this increase at the payload capacity, obtained using our proposed CP watermarking scheme, is that, we embed every two watermark bits by permuting three histogram bins. While, Chrysochos et al. embed each one bit of the watermark by permuting two histogram bins. Thus, our embedding mechanism exploits the histogram bins more than its counterpart. Finally, we noted that there is no direct relation between the *step* value of the public key and the embedding capacity of the two watermarking schemes.

### 5.2.2 Comparing the Effect of Watermark Embedding on the Host Image

Preserving the quality of the host image after being watermarked is one of the main success factors of any watermarking scheme. Therefore, the quality of the watermarked image is our main concern in this section. Measuring the quality of the watermarked image is an estimate of how much the watermarked image is similar to the original, un-watermarked, one. The PSNR, seen in Equation 2.12, is used as a measurement of the quality of the watermarked image.

Both the Chrysochos et al. watermarking scheme and our proposed CP watermarking scheme, are used in turn to embed a fixed watermark into each of the samples in our dataset. The public key used in this experiment equals, 5.7. The quality, PSNR, of each watermarked sample is calculated. Finally, the calculated values of the PSNR are averaged over every category of samples. Figure 5.6 shows the obtained results.

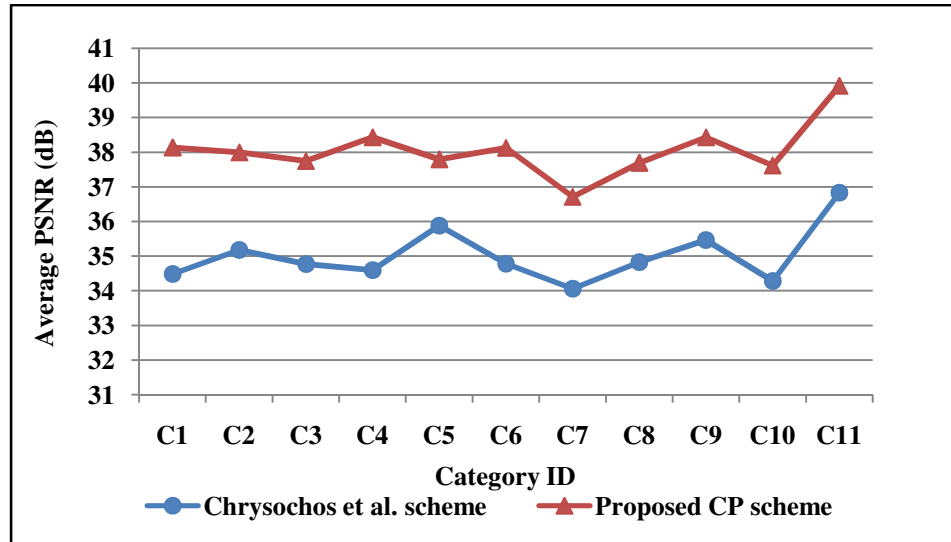


Figure 5.6: Comparing the quality of the watermarked image for both Chrysochos et al. scheme, and our proposed CP scheme.

As seen in Figure 5.6, for all categories, our proposed CP scheme generates higher quality watermarked images than those obtained when using Chrysochos et al. scheme. The average PSNR over all watermarked samples using CP scheme is, 38.05 dB, while using Chrysochos et al. scheme is, 35.01 dB. This increase in the quality is due to the embedding strategy of our proposed scheme, which performs a smaller number of histogram permutations than those performed using Chrysochos et al. scheme, in embedding the same watermark. Our proposed CP scheme permutes three histogram bins to embed two watermark bits, while Chrysochos et al. scheme permutes two histogram bins for embedding one bit. In other words, embedding two bits using our proposed CP scheme requires at most permuting three histogram bins, while embedding two bits using Chrysochos et al. scheme requires at most permuting four histogram bins. In addition, during watermark embedding in our proposed CP scheme, some Bins Triple Patterns, **BTP**, namely 2 and 3, are left unchanged when facing the Bits Couple Patterns, **BCPs** “00” and “11”. Thus, our proposed scheme has fewer side effects on the host image if compared with its counterpart.

Finally it is noticed in this experiment that, increasing the *step* of the public key; decreases the quality of the watermarked image using Chrysochos et al. scheme more quickly than when using our proposed scheme. The interpretation is that, in Chrysochos et al. scheme, when we increase the step of the public key; the distance occupied by the two selected histogram bins increased. Thereby, all the interchanged pixels, for embedding, will have far intensity values. But because our proposed CP scheme depends on three histogram bin, that occupied distance is divided into two smaller distances. Thereby, some of the interchanged pixels will have closer intensity values.

### 5.2.3 Comparing the Embedding Time

Chrysochos et al. watermarking scheme and our proposed CP watermarking scheme, each is used in turn to embed a fixed watermark into each of the samples in our dataset. The value of the used public key is, 5.7. We recorded the embedding time at each sample in the dataset. Finally, the average embedding time is calculated for each category. Figure 5.7 shows the obtained results for the both schemes.

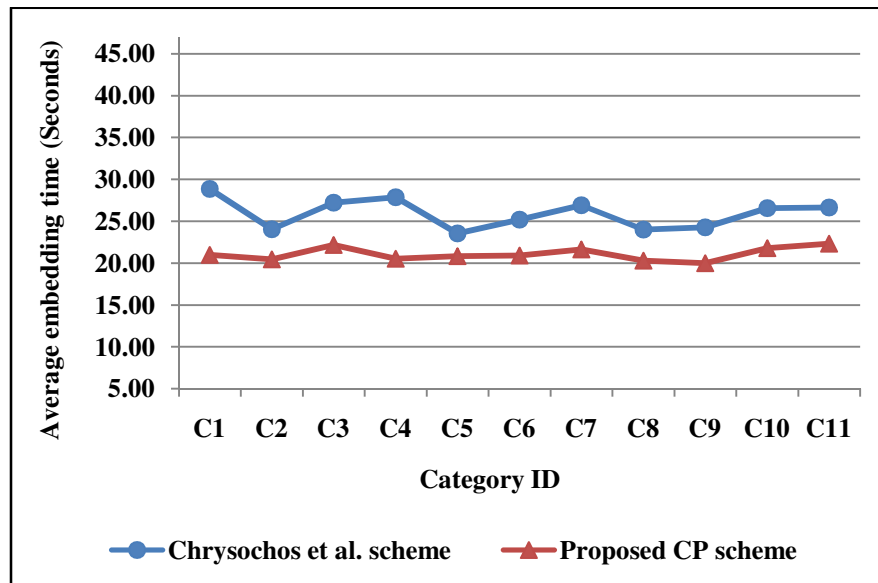


Figure 5.7: Comparing watermark embedding time between Chrysochos et al. scheme, and our proposed CP scheme.

As seen in Figure 5.7, it is evident that our proposed CP algorithm is faster than Chrysochos et al. algorithm in embedding watermarks. The average embedding time over all samples using the proposed CP scheme is, 21.08 seconds, while using Chrysochos et al. scheme is, 25.92 seconds. The interpretation of the superiority of our proposed CP watermarking scheme to its counterpart, regarding obtaining less

embedding time, comes as follows: Usually, any algorithm time is mostly consumed in performing operations like reading, writing, processing, and etc. Assume that we seek to embed a watermark based on the histogram of a host image, using both schemes in turn. The total number of the histogram bins permuted, and consequently, the total number of pixels interchanged using our proposed CP watermarking scheme is less than that when using Chrysochos et al. watermarking scheme. That is because, three histogram bins are permuted in our proposed CP scheme to embed two bits, while two histogram bins are permuted in Chrysochos et al. scheme to embed one bit. For example, if we seek to embed a binary watermark of 6 bits, at most we need to permute 9 bins using our proposed scheme, while 12 bins are permuted using Chrysochos et al. scheme.

#### 5.2.4 Robustness Against Some Geometrical Attacks

Robustness of the watermarking scheme against geometrical attacks means that the embedded watermark must survive against geometrical attacks those might be applied to the host image. In other words, despite of geometrically transforming the host image, the embedded watermark must still be detectable by the watermarking scheme. Both Chrysochos et al. scheme and our proposed CP scheme embed watermarks based on the histogram of the host image. Therefore, both schemes are robust only against geometrical attacks those change pixels positions, but not against those change the histogram of the host image. In this section, the robustness of the proposed CP watermarking scheme against such geometrical attacks is tested. In this test, each sample in the dataset is embedded with the copyright watermark shown in Figure 5.8, which is a binary watermark of size 20x20. The used public key is, 16.7.



**Figure 5.8: Binary watermark (Size 20X20).**

Now to evaluate the robustness of our proposed CP watermarking scheme against some geometric attacks, those change only pixels positions, each watermarked sample is subjected to some geometrical attacks; including, Flipping, Rotation, Scattering, Warping, and Skewing. For each attack application at a given watermarked sample, the embedded watermark is extracted and evaluated using the NCC. The NCC values are averaged over all the samples, those were modified by a given attack type. Table 5.5 summarizes the obtained results.



**Table 5.5: The experimental results for testing the robustness of the proposed CP watermarking scheme, against some geometric attacks.**

Attack Name	Parameters	The NCC of the extracted watermark
Flipping	H, V, and Both	1
Rotation	90°, 180°, and 270°	1
Scattering	Any degree	1
Warping	Any degree	1
Skewing	Any degree	1

As seen in the table, our proposed CP watermarking scheme showed 100 % robustness against some geometrical attacks, i.e., the detected watermark exactly matches the original embedded one. Such attacks are flipping (horizontally, vertically, both), rotation (90°, 180°, 270°), scattering, warping and skewing, as well as their combinations.

The interpretation for this 100% robustness of our proposed CP watermarking scheme against such geometrical attacks, is due to the fact that our proposed scheme embeds the watermark by modifying the histogram of the host image, which is mostly not affected after applying the aforementioned attacks to the host image. Those attacks change only the positions of the pixels. Mean, every pixel in the original image goes to a predefined point. This predefinition of new pixels positioning, specifies the type of the transformation. For example, at the horizontal flipping, the pixels of the original image are mirrored across a horizontal axis. The histogram of the image does not depend on the positions of pixels, but it depends on the number of pixels. Therefore, such image modifications will not change the histogram of the watermarked image. Thereby, we conclude that our proposed CP watermarking scheme is robust 100% against geometrical attacks those only change pixels positions, but not against those change the histogram of the watermarked image.

### **5.3 Experimental Results of the Proposed Multipurpose Watermarking Scheme**

The experimental results for evaluating the third proposed watermarking scheme are demonstrated and analyzed in this section. We will also demonstrate the importance of our performed developments at each individual participating party of the proposed multipurpose watermarking scheme. In doing so, we evaluate our proposed multipurpose watermarking scheme based on the developed versions of Poonkuntran and Rajesh scheme [1], and Chrysochos et al. scheme [34] on one hand; and based on the non developed versions of them on the other hand. We will refer to these by

“developed multipurpose scheme” and “non-developed multipurpose scheme”. Experiments are performed based on the 1050 sample images of our dataset, and also based on the CP watermark, seen in Figure 5.8, which was used in evaluating the second proposed watermarking scheme. Finally, the value of the public key used during experiments in this section is 5.7.

### 5.3.1 Comparing the Effect of Watermark Embedding on the Host Image

First of all we evaluate the quality, PSNR, of the watermarked images. This evaluation is crucial here, where we embed two watermarks at the same host image. To compare the quality of the watermarked images of the developed multipurpose scheme with that of the non-developed multipurpose scheme; each scheme is used separately to embed both the CA and the CP watermarks into each sample in our dataset. To achieve fair comparisons, the size of each embedded watermark is fixed for both schemes. The quality, PSNR, of each watermarked sample is calculated and recorded. The results obtained in this experiment are shown in Figure 5.9, which shows the average PSNR for each category of samples.

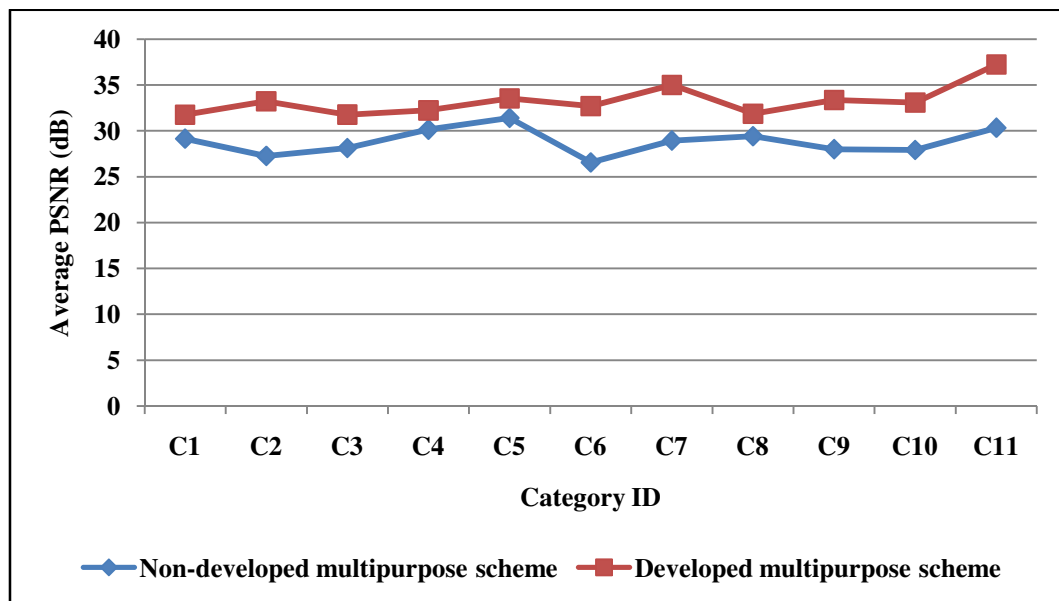


Figure 5.9: Comparing the quality, PSNR, of the watermarked images using the developed, and the non developed multipurpose watermarking schemes.

As seen in the figure, for all sample categories, using the developed multipurpose scheme leads higher quality than that when using the non-developed multipurpose scheme. In addition, for most categories, using the non-developed multipurpose scheme leads an average PSNR that is under the acceptance level of quality, (30 dB). The recorded average PSNR, over all samples, using the developed multipurpose

scheme is 33.25 dB, and using the non-developed multipurpose scheme is 28.85 dB. Therefore, under view of the foregoing, we conclude that our proposed multipurpose watermarking scheme would not be possible without our developments made at each of its participating parties. Actually, the superiority of the developed multipurpose watermarking scheme stems from the developments made at each of its cascaded schemes, individually, especially the new embedding mechanisms; those increased the embedding capacity, as well as the quality of the watermarked images.

### 5.3.2 Comparing the Embedding Time

In this section, we compare the embedding time of the non-developed and the developed multipurpose watermarking schemes. In doing so, each scheme is used separately to embed both the CA and CP watermarks into a given host sample. The CA watermark is generated using the messy system, which is based on the green component of the host sample. The CP watermark is the binary watermark shown in Figure 5.8. The size of each embedded watermark is fixed for both schemes. The embedding process is repeated for all the samples in our dataset. The time elapsed by each scheme is recorded for each sample, and then, the average embedding time for each category of samples is calculated. Figure 5.10 shows the recorded results in this experiment. As seen in the figure, the developed multipurpose scheme surpassed the non-developed one; regarding obtaining smaller embedding time. We obtained an average reduction in the embedding time equals, 4 seconds. The average embedding time using the developed multipurpose scheme is 23.48 seconds, while using the non-developed multipurpose scheme is 27.48 seconds. Actually, these results were expected; because the developed multipurpose scheme combines both the proposed CA and CP watermarking schemes, those were surpassed their counterparts in having a smaller embedding times. The developed multipurpose watermarking scheme embeds the CA watermark by expanding the differences of the pixel pairs those were formed using the neighboring pixels at the same color plane, rather than using two different color planes, as in the non-developed multipurpose watermarking scheme. Also, the developed multipurpose watermarking scheme embeds the CP watermark by performing a smaller number of histogram permutations than those performed using the non-developed one. This explains the smaller embedding time in our developed multipurpose watermarking scheme.

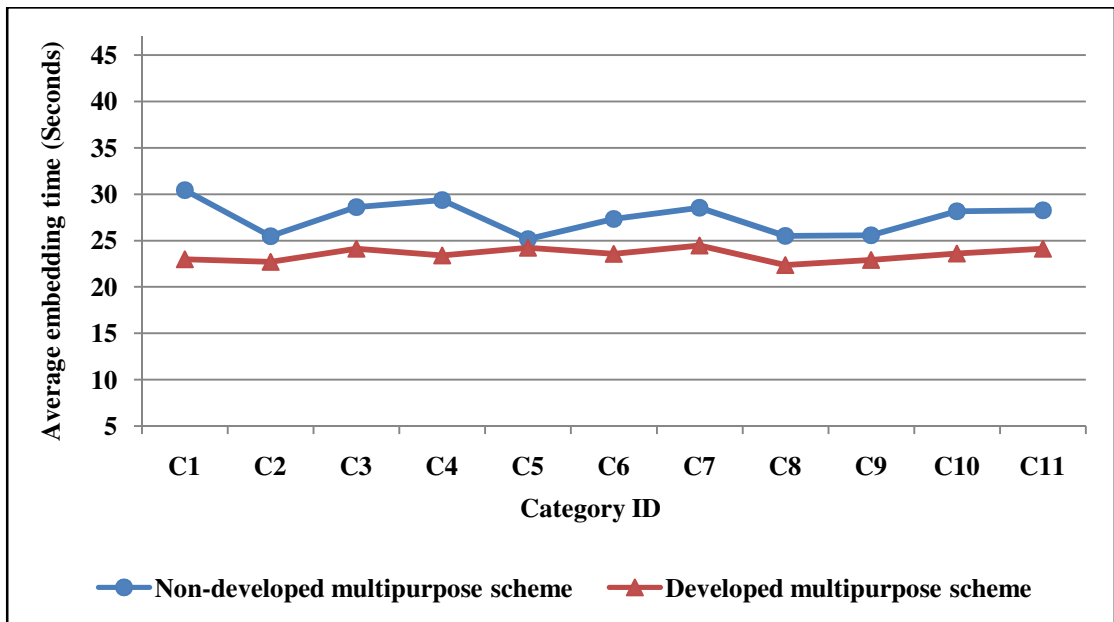


Figure 5.10: Comparing the time required to embed both the CA and the CP watermarks using the developed, and the non developed multipurpose watermarking schemes.

### 5.3.3 Testing the Fragility of CA Watermarking, and the Robustness of CP Watermarking against Geometrical Attacks

The proposed multipurpose watermarking scheme embeds two watermarks at the same host image. The first is the CA watermark, which is embedded in a way such that it is destroyed easily; to notify the extractor with the presence of a modification in the image. The CA watermarking in our proposed scheme is supposed to sense any modification in the watermarked image, even if it is tiny. The second embedded watermark is the CP watermark, which is embedded in away such that it is robust to attacks as much as possible; in order to provide the extractor with image's copyright information, even if the image is attacked. The CP watermark in our proposed scheme will be robust only to the geometrical attacks those change pixels positions. In this section we measure the fragility of the CA watermarking, as well as the robustness of the CP watermarking against such geometrical attacks. In doing so, each sample in our dataset is embedded by both the CA and the CP watermarks using our proposed multipurpose watermarking scheme. The CA watermark is generated using the messy system, which is based on the green color plane of the sample. The CP watermark is the binary watermark shown in Figure 5.8. Each watermarked sample is subjected to a list of such geometric attacks. After each attack application, to test the effect of each attack on the embedded watermarks, we extracted and verified both the CP and CA watermarks. The extracted CP watermark is evaluated using the NCC, while the

extracted CA watermark is evaluated by comparison with the generated messy watermark of the sample, each mismatch between the watermarks is assumed as a defect, the total number of defects divided by the total number of watermark bits is the percentage of defect. For each attack type, the obtained results are averaged over all the samples. The results of this experiment are listed in Table 5.6.

**Table 5.6: The results of testing the fragility of the CA watermark and the robustness of the CP watermark of the proposed multipurpose watermarking scheme.**

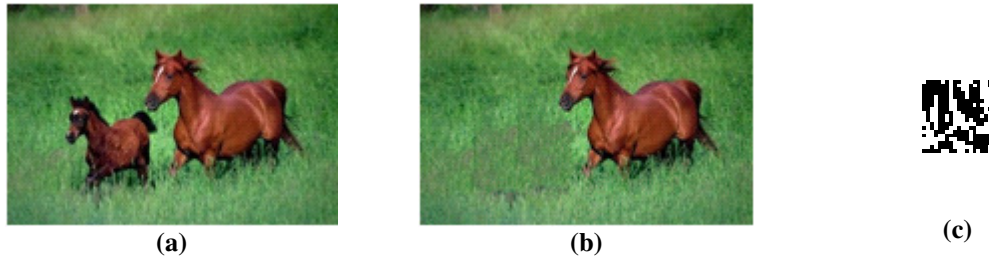
Attack Type	Average % of defect of the extracted CA watermark	Average NNC of the extracted CP Watermark
Flipping Attack H,V, Both	50.23	1
Rotate Attack 90°,180°,270°	50.37	1
Scattering attack	49.75	1
Warping	50.08	1
Skewing	51.62	1
<b>Average</b>	<b>50.41</b>	<b>1</b>

As seen in the table, the CA watermarking could sense the geometrical modifications those change the positions of the pixels. Approximately 50 % of the watermark is destroyed due to such geometrical attacks. It is a significant percentage, especially when dealing with binary watermarks. Also, the extracted CP watermark demonstrates 100 % robustness to those modifications.

From our point of view, we see that this type of robustness, i.e. robustness to attacks those change pixels positions, is effective especially in medical applications. Sometimes, medical images are transformed during their circulation from one place to the other. These transformed images may still be used if the transformation is one of which our scheme is robust against. In such case, our proposed multipurpose scheme could extract the copyright information of those transformed images, and hence, the specialist could decide to use the transformed images. On the other hand, because our scheme is not robust against other types of attacks; if the image is attacked maliciously, our scheme will extract an anonymous copyright. In such case, the specialist is informed with the occurrence of a malicious attack, and hence, he will ignore that attacked image.

An example demonstrating the situation where the image is maliciously attacked is shown in Figure 5.11, which shows a watermarked image, along with its modified version (the little horse is hidden), and the corresponding extracted CP watermark. As seen in the figure, the extracted watermark is anonymous, having a very low NCC = 0.49. This indicates that the watermarked image is maliciously attacked. Thereby, it is

logical to ignore such attacked image, because the attack hides some important portions in it.



**Figure 5.11: Illustration example, demonstrates the situation when an image is maliciously attacked. (a) Watermarked image. (b) Attacked version of (a). (c) The extracted CP watermark,  $NCC = 0.49$ .**

## CHAPTER 6

### CONCLUSION

#### 6.1 Summary and Concluding Remarks

Images might be subject to attacks those violate their contents, as well as their copyright. Those attacked images may no longer be suitable for taking serious decisions. Thus, it is required to find a mechanism to detect any modification in images, as well as decide whether the modified images are still being usable or not. In this thesis, we proposed a multipurpose watermarking scheme for both CA and CP of color images to fulfill this requirement. It combines other two watermarking schemes, those are also proposed in this thesis; namely, CA watermarking scheme and CP watermarking scheme.

The proposed CA watermarking scheme is a fragile, blind, and reversible watermarking scheme for tiny tamper detection of color images. It is a development of the technique proposed by Poonkuntran and Rajesh [1]. Two drawbacks of Poonkuntran and Rajesh scheme are solved in our proposed CA scheme. The first drawback is regarding the lack of fragility of their scheme. Due to their mechanism of embedding by expanding the differences between two different color planes; their embedded watermark might not cover whole image's area, because of encountering non expandable differences. Thus, modifications in some specific locations might not be detected using their scheme. We overcome this drawback by proposing an accumulative watermark embedding process, which aims at spreading the watermark over whole image's area. It is divided into three stages namely, vertical, horizontal, and diagonal. Each stage is responsible for embedding some dedicated bits of the watermark by expanding the differences between pixels in the corresponding direction. A wide and non-uniform spreading of the embedded watermark is accomplished in our proposed CA watermarking scheme, and hence, any modification in the watermarked image is necessarily sensed, even if it is very small.

The second drawback of Poonkuntran and Rajesh scheme is regarding the low quality of its generated watermarked images. Because of embedding by expanding the differences between two different color planes, which is generally not a small quantity, old pixel values are changed with new distant values. Consequently, a distorted image quality, low PSNR, will be produced if their scheme is used to embed

large watermarks sizes. This of course, prevents embedding any additional other purpose watermark in the watermarked image; because it will introduce an extra image distortion or lead to a useless image, especially in sensitive applications, like medical and military. We overcome this drawback by proposing a homogenous watermark embedding process, which aims at producing a high quality, undistorted, watermarked image. It embeds the watermark by expanding the differences between the neighboring pixels at the same color plane, in a method called inter-plane difference expanding, which is performed based on the Integer Transform (IT).

Our proposed CA watermarking scheme is evaluated and compared with its counterpart, Poonkuntran and Rajesh scheme. Tests are performed based on a dataset of 1050 samples, from eleven categories. Results are averaged over all the dataset samples. Evaluation is done in terms of capacity, quality, spreading, fragility, and time.

- a) Regarding the capacity, for each introduced watermark for embedding, about 90% of the watermark bits will be embedded using our proposed CA scheme, which becomes 80% when using their proposed scheme.
- b) Regarding the quality, we obtained a minimum PSNR of 31.76 dB using our proposed CA scheme, which is 4.61 dB larger than that obtained by using Poonkuntran and Rajesh scheme, 27.15 dB.
- c) Regarding the watermark spreading, the observed average percentage of the watermark spreading using our proposed CA scheme is 94.27 %, whereas, the average percentage of the watermark spreading using Poonkuntran and Rajesh scheme is 68.78 %. Therefore the percentage of protected pixels using our proposed CA scheme is much higher than that when using Poonkuntran and Rajesh scheme.
- d) Regarding the fragility, approximately both schemes have an equal degree of fragility against filtering attacks. Using either scheme for watermark embedding; about 40-50 % of the watermark is destroyed after attacking the watermarked image by filtering attacks. But, our proposed CA scheme outperforms its counterpart in obtaining a much higher degree of fragility against attacks those interested in small portions of images.
- e) Regarding the embedding time, our proposed CA scheme exploits time more than that in its counterpart.



- f) Finally, because our proposed CA scheme depends on the location map for detecting the previously embedded watermark; it has a zero False Positive Rate, provided that the location map is correct.

The proposed CP watermarking scheme is a robust, blind and reversible watermarking scheme for CP of color images. It is a development of the scheme proposed by Chrysochos et al. [34]. Their scheme mainly depends on the permutation of the image's histogram bins. For each watermark bit to be embedded, a two histogram bins are permuted according to a given rule. The authors referred to an inherent drawback in their proposed scheme; the theoretical maximum embedding capacity of their scheme is rather low (128 bits per color plane, and 384 bits per color image). An attempt for adapting this drawback is presented in our proposed CP watermarking scheme. A new embedding mechanism is proposed, which is still using the permutation of the histogram bins for watermark embedding, but with a completely different embedding rule. In our proposed scheme, for each couple of the watermark bits to be embedded a three histogram bins are permuted. Theoretically, the maximum payload capacity of our proposed CP watermarking scheme is 510 bits, which is larger than that of Chrysochos et al. scheme by 126 bits.

Our proposed CP watermarking scheme is evaluated and compared with its counterpart, Chrysochos et al. scheme. Different tests are performed including, measuring and comparing the embedding capacities, comparing the watermarked image quality, comparing the embedding time, and finally evaluating the robustness of the embedded watermarks against some geometrical attacks. Tests are performed based on the 1050 sample images of our dataset.

- a) Regarding the capacity, at each watermarked sample, our proposed CP scheme increased the average payload capacity, per each sample, by about 55 bits over that obtained using Chrysochos et al. scheme.
- b) Regarding the quality, our proposed CP scheme generates watermarked images with a higher quality, PSNR= 38.05 dB, than that obtained when using Chrysochos et al. scheme, PSNR=35.01 dB. This is due to the embedding strategy of our proposed scheme, which permutes 3 histogram bins to embed 2 watermark bits, rather than permuting 2 bins for embedding 1 bit as in their scheme.
- c) Regarding the embedding time, the experiments showed that our proposed CP algorithm is faster than Chrysochos et al. algorithm in embedding watermarks,

especially with large watermark sizes. The cause is that when the two schemes are invoked to embed the same watermark size; a smaller number of histogram permutations will be performed using our proposed scheme than those performed using Chrysochos et al. scheme.

- d) Regarding the robustness, experiments showed that both Chrysochos et al. scheme and our proposed CP scheme are robust against geometrical attacks those change pixels positions, but not those change image histogram. This is natural, since both schemes embed bits based on the identity of the histogram bins. Our scheme demonstrates a high robustness against a variety of such geometrical attacks like, Flipping (H, V, Both), Rotation (90°, 180°, 270°), scattering (any degree), Warping (any degree), and Skewing.
- e) Because our proposed CP scheme depends on the public key for detecting the previously embedded watermark; it has a zero False Positive Rate, provided that the public key is correct.

Finally, the three proposed schemes work in the spatial domain, which requires a lower computational cost than that required in transform domain based schemes.

## 6.2 Recommendations and Future Work

There are a number of recommendations, which are derived from this thesis:

- a) We recommend using digital watermarking instead of data encryption for preventing illegal copying of digital images. Actually, encryption solves the problem of illegal copying by restricting the display of the content only to people who have the security key, by which the content was encrypted. Also in encryption, once the content is decrypted; it is no longer protected from illegal copying. On the other hand digital watermarking embeds watermarks in the contents without their encapsulation, and the embedded watermark can never be removed during the normal usage of the watermarked images.
- b) When it is possible to achieve the intended goal using both spatial and transform domain based watermarking techniques, we recommend using the spatial domain based ones. In addition of having a relatively low computational cost, spatial domain based schemes are easily implemented.
- c) In order to maintain the quality of the host image after being watermarked, we recommend considering the main issue of interest at each bit embedding as the

question: By what values pixels will be substituted? Because the vulnerability of the watermarked image quality is not in the amount of pixels changed at each bit embedding, as far as how the pixels will be manipulated.

- d) In dual watermarking, we recommend embedding the CA watermark first then embedding the CP watermark in the generated watermarked image. This order reduces both time and cost. In the extraction process this ordering is reversed. Once the copyright information is detected and verified to be correct an assertion about the integrity of the image content is started by extracting and verifying the CA watermark. Otherwise, there is no need to make an assertion about an anonymous image.

As of this point, the goals of this research have been achieved. Further, we would like to conclude with a road map for further work. Future works will be devoted to the following points:

- a) Reducing the reliance at the information used for reversibility of both CA and CP watermarking schemes, such as keys.
- b) Increasing the robustness of the proposed CP watermarking scheme, against other geometric attacks, such as arbitrary rotation angles.
- c) Further increasing the maximum payload capacity of the proposed CP watermarking scheme.

## REFERENCES

- [1] S. Poonkuntran and R. S. Rajesh, "A Messy Watermarking for Medical Image Authentication," in *Proc. of 2011 Int. Conf. Communications and Signal Processing*, Kerala, pp. 418-422.
- [2] X. Ma and X. Shen, "A novel blind grayscale watermark algorithm based on SVD," in *Proc. Int. Conf. Audio, Language and Image Processing*, (ICALIP '08), Shanghai, July 2008, pp. 1063-1068.
- [3] N. Yadav, N. Pahal, P. Kalra, B. Lall, and S. Chaudhury "A Novel Approach for Securing Forensic Documents Using Rectangular Region-of-Interest (RROI)," in *Proc.2nd Int. Conf. Emerging Applications of Information Technology*, (EAIT), Kolkata, 2011, pp. 198-201.
- [4] M. Prasad and Sh. Koliwad "A Comprehensive Survey of Contemporary Researches in Watermarking for Copyright Protection of Digital Images," *International Journal of Computer Science and Network Security*, vol. 9, no. 4, pp. 91-107, Apr. 2009.
- [5] E. Brannock, M. Weeks, and R. Harrison "The Effect of Wavelet Families on Watermarking," *Journal of Computers*, vol. 4, no. 6, pp. 554-566, Jun. 2009.
- [6] Mrdjenovic and Ljiljana "Digital watermarking in the generalized discrete cosine transform domain," M.S. thesis, Dept. Comput. Sci., York University, Toronto, Ontario, Jan. 2010.
- [7] L. Fan, T. Gao, Q. Yang and Y. Cao, "A copyright-protection watermark mechanism based on generalized brain-state-in-a-box neural network and error diffusion halftoning," *IEEE Int. Conf. on Multimedia and Expo*, (ICME), China, 2011, pp. 1-6.
- [8] C. Xiaoling, and Z. Huimin "A Novel Video Content Authentication Algorithm Combined Semi-fragile Watermarking with Compressive Sensing," *2nd Int. Conf. on Intelligent System Design and Engineering Application*, (ISDEA), 2012, pp. 134-137.
- [9] Y. Hu, H. Lee, and H. Zeng, "Curve Watermarking Technique for Fingerprinting Digital Maps," *8th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, (IIHMSP), 2008, pp. 223-226.
- [10] A. Giakoumaki, K. Perakis, A. Tagaris, D. Koutsouris, "Digital Watermarking in Telemedicine Applications - Towards Enhanced Data Security and Accessibility," *Proc. of the 28th IEEE Annual Int. Conf. on Engineering in Medicine and Biology Society*, (EMBS), New York City, USA, 2006, pp. 6328 – 6331.
- [11] F. H. Wang, J. S. Pan, and L. C. Jain, "Digital Watermarking Techniques," in *Innovations in Digital Watermarking Techniques*, Berlin, Heidelberg: Springer-Verlag, 2009, pp. 11-26.

- [12] B. Harjito, S. Han, V. Potdar, E. Chang, M. Xie, "Secure Communication in Wireless Multimedia Sensor Networks using Watermarking," *4th IEEE Int. Conf. on Digital Ecosystems and Technologies*, (DEST), 2010, pp. 640-645.
- [13] E. Bollain-y-Goytia, M. Nakano-Miyatake and H. Pérez-Meana, "Authentication of Identification Card Using Watermarking," *48th Midwest Symposium on Circuits and Systems*, 2005, pp. 1422 - 1425.
- [14] R. Chamlawi and A. Khan, "Digital image authentication and recovery: Employing integer transform based information embedding and extraction," *Journal of Information Sciences*, vol. 180, no. 24, pp. 4909-4928, 2010.
- [15] R. Chamlawi, A. Khan, and A. Idris, "Wavelet based image authentication and recovery," *Journal of Computer Science and Technology*, vol. 22, no. 6, pp. 795-804, 2007.
- [16] A. Piva, F. Bartolini, and R. Caldelli, "Self-recovery authentication of images in the DWT domain," *International Journal of Image and Graphics*, vol. 5, no. 1, pp. 149-165, 2005.
- [17] M. Kuribayashi and H. Tanaka, "Fingerprinting Protocol for Images Based on Additive Homomorphic Property," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2129-2139, Dec. 2005.
- [18] J. Domingo-Ferrer and J. Herrera-Joancomarti, "Simple collusion-secure fingerprinting schemes for images," in *Proc. 2000 Int. Conf. Information Technology: Coding and Computation*, ITCC'2000, Catalonia, pp. 128-132.
- [19] EU 2003, "e-Health Ministerial Declaration," *e-Health Ministerial Conference*, May 2003.
- [20] D. Ziadlou, A. Eslami, and H.R. Hassani, "Telecommunication Methods for Implementation of Telemedicine Systems in Crisis," in *Proc. 3rd Int. Conf. Broadband Communications, Information Technology & Biomedical Applications*, Gauteng, Nov. 2008, pp. 268-273.
- [21] A. Horsch, "Telemedicine and e-Health in recent years: Meeting the challenges," *Proc. 3rd Int. Conf. on Information Communication Technology in Health*, (ICICTH'05), Greece, 2005.
- [22] W. Gang and R. Ni-ni, "A Fragile Watermarking Scheme for Medical Image," in *Proc. 27th Annu. Int. Conf. Engineering in Medicine and Biology Society*, China, Shanghai, 17-18 Jan. 2006, pp. 3406-3409.
- [23] I. Cox, M. Miller, and J. Bloom, "Applications and Properties," in *Digital watermarking*, 1st ed., USA: Morgan Kaufman Publishers, 2002, ch. 2, sec. 2.2.1, pp. 26-27.

- [24] S. Oliveira, M. Nascimento, and O. Zaiane, "Digital Watermarking: Status, Limitations and Prospects," *Technical Report TR02-01*, University Of Alberta, January 2002.
- [25] J. M. Guo and T. N. Le, "Secret Communication Using JPEG Double Compression," *IEEE Signal Process. Lett.* vol. 17, no. 10, pp. 879-882, Oct. 2010.
- [26] D. Bhattacharyya, P. Chakraborty, F. Alisherov, and T. h. Kim, "Quantum Watermarking: A Review," *International Journal of Security and Its Applications*, vol. 4, no. 3, 2010.
- [27] Ch. Ch. Lai, "A digital watermarking scheme based on singular value decomposition and tiny genetic algorithm," *Digital Signal Processing*, vol. 21, no. 4, pp. 522-527, Jul. 2011.
- [28] F. Liu, Sh. Liang, and C. Wang, "A Novel Blind Watermark Algorithm," in *Proc. Int. Conf. Information Engineering and Computer Science*, (ICIECS), China, Wuhan, 2009, pp. 1-4.
- [29] N. Narawade, and R. Kanphade, "Reversible Watermarking: A Complete Review," *International Journal of Computer Science and Telecommunications*, vol. 2, no.3, 2011, pp. 46-50.
- [30] M. Schlauweg, D. Pröfrock, B. Zeibich, and E. Müller, "Dual watermarking for protection of rightful ownership and secure image authentication," in *Proc. 4th ACM international workshop on Contents protection and security*, New York, 2006.
- [31] K. Ramanjaneyulu and K. Rajarajeswari, "An Oblivious Image Watermarking Scheme Using Multiple Description Coding and Genetic Algorithm," *International Journal of Computer Science and Network Security*, (IJCSNS), vol. 10 no. 5, pp. 167-174, 2010.
- [32] M. Sharkas, D. Elshafie, and N. Hamdy, "A Dual Digital-Image Watermarking Technique," in *Proc. 3rd World Enformatika Conf. Computer Graphics and Image Processing*, Istanbul, Turkey, 2005, pp.136-139.
- [33] W. N. Lie, T. L. Hsu, G.S. Lin, and W.J. Ho, "Fragile Watermarking for JPEG-2000 Images," in *Proc. 16th Conf. Computer Vision, Graphics and Image Processing*, (CVGIP 2003), China, Kinmen, 2003, pp. 823-826.
- [34] E. Chrysochos, V. Fotopoulos, A. Skodras, and M. Xenos, "Reversible Image Watermarking Based on Histogram Modification," in *Proc. 11th Panhellenic Conf. Informatics with international participation*, (PCI 2007), vol. B, Patras, Greece, May 2007, pp. 93-104.
- [35] L. Xie and R. Arce, "A Class of Authentication Digital Watermarks for Secure Multimedia Communication," *IEEE Trans. on Image Processing*, vol. 10, no. 11, pp. 1754-1764, 2001.

- [36] W. Zeng and S. Lei, "Efficient Frequency Domain Selective Scrambling of Digital Video," *IEEE Trans. Multimedia*, vol.5, no. 1, pp. 118-129, 2003.
- [37] C. Moucary and B. Hassan, "A Novel Blind Digital Watermarking Technique for Stegano-Encrypting Information Using Nine-AC-Coefficient Prediction Algorithm with an Innovative Security Strategy," *WSEAS Trans. on Signal Processing Journal*, Vol. 5, No. 11, pp. 359-368, 2009.
- [38] S. Mohanty, "Digital Watermarking: A Tutorial Review ," *Master Project Report*, Dept. of Electrical Engineering, India, Institute of Science, DANGALORE, India, 1999.
- [39] I. J. Cox, M. L. Miller, and J. A. Bloom, "Watermarking applications and their properties," in *Proc. Int. Conf. Information Technology: Coding and Computing*, USA, Las Vegas, 2000, pp. 6-10.
- [40] Sh. Weng, Y. Zhao, and J. Sh. Pan, "A Novel Reversible Data Hiding Scheme," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 2, pp. 351-358, Feb. 2008.
- [41] Y. Zhang, "Digital Watermarking Technology: A review," in *Proc. Int. Conf. Future Computer and Communication*, (FCC'09), Kuala Lumpur, Malaysia, Apr. 2009, pp. 250-252.
- [42] V. Vallabha, "Multiresolution Watermark Based on Wavelet Transform for Digital images," *on Cranes Software International Limited*, 2003.
- [43] G. Voyatzis, and I. Pitas, "The use of watermarks in the protection of multimedia Products," *Proc. of the IEEE Journal*, Department of informatics, University of Thessaloniki, Vol. 87, No. 7, pp. 1197–1207, 1999.
- [44] M. Barni, "Digital watermarking for copyright protection: a communication perspective," *IEEE Communication Magazine*, 2001, pp 90-91.
- [45] B. Biswas, white paper, Digital Watermarking. [Online]. Available: <http://www.tataelxsi.com/whitepapers/digitalwatermarking.pdf>, Accessed 8 April 2012.
- [46] L. Liu, "A survey of digital watermarking technologies," *Tech. Rep., Computer Vision Laboratory*, Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA, 2005.
- [47] N. Dharwadkar, B. Amberker, and A. Gorai, "Secure Watermarking Scheme for Color Image Using Intensity of Pixel and LSB Substitution," *Journal of Computing*, vol. 1, no. 1, pp. 1-6, 2009.
- [48] N. V. Dharwadkar, B. B. Amberker, and A. Gorai, "Non-blind Watermarking scheme for color images in RGB space using DWT-SVD," in *proc. Int. Conf. Communications and Signal Processing*, (ICCSP), 2011, pp. 489 – 493.



- [49] J. Tian, "Reversible watermarking by Difference Expansion," *In Proc. Workshop on Multimedia and Security: Authentication, Secrecy, and Steganalysis*, 2002, pp. 19–22.
- [50] A. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform," *IEEE Trans. on Image Processing*, Vol. 13, No. 8, 2004, pp. 1147- 1156.
- [51] I. Cox, M. Miller, and J. Bloom, "Digital watermarking," *1st ed., USA: Morgan Kaufman Publishers*, 2002.
- [52] M. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible Data Hiding," *in Proc. IEEE Int. Conf. Image Processing*, Vol. 2, 2002, pp. 157-160.
- [53] J. Tian, "Reversible Watermarking Using a Difference Expansion," *IEEE Trans. Circuits Syst. for Video Technology Video Technol.*, vol. 13, no. 8, pp. 890-896, 2003.
- [54] R. Ni, Q. Ruan, and Y. Zhao, "Pinpoint authentication watermarking based on a chaotic system," *Forensic Science International Journal*, vol. 179, no. 1, pp. 54-62, 2008.
- [55] S. Voloshynovskiy, S. Pereira, T. Pun, J. Eggers, and J. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE Communications Magazine*, vol. 39, pp. 118–126, 2001.
- [56] F. Gonzalez, and J. Hernandez, "A Tutorial on Digital Watermarking," in *Proc. of 33rd IEEE Annual Carnahan Conf. on Security Technology*, 1999.
- [57] R. C. Gonzalez, and R. W. Woods, "Geometric Transformations," in *Digital Image Processing*, 2nd Edition, Prentice Hall, 2002, ch. 5, sec. 11, pp. 270-277.
- [58] [http://www.icoachmath.com/math\\_dictionary/](http://www.icoachmath.com/math_dictionary/) *Math Dictionary*, Accessed 4 July 2012.
- [59] H. Bowles, K. Mitchell, R. Sumner, J. Moore, and M. Gross "Iterative Image Warping," *Computer Graphics Forum*, vol. 31, no. 2, pp. 237–246, 2012.
- [60] V. Licks, F. Ourique, R. Jordan, F. Perez-Gonzalez "The effect of the random jitter attack on the bit error rate performance of spatial domain image watermarking," *Proc. Int. Conf. on Image Processing*, (ICIP), 2003, pp. II 455-458.
- [61] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm>, *Image Processing Learning Resources*, Accessed 3 July 2012.
- [62] MathWorks, "Create predefined 2-D filter", *Image Processing Toolbox User's Guide 6*, pp.17 169- 174.

- [63] M. Jiansheng, L. Sukang, and T. Xiaomei, "A Digital Watermarking Algorithm Based On DCT and DWT," *Proc. of the Int. Symposium on Web Information Systems and Applications*, (WISA'09), Nanchang, P. R. China, 2009, pp. 104-107.
- [64] A. Zeki and A. Manaf, "A novel digital watermarking technique based on ISB (Intermediate Significant Bit)," *World Academy of Science, Engineering and Technology*, 50. 2009, pp. 989-996.
- [65] Neil F. Johnson, "Steganography," *Technical Report*, Nov. 1995, [Online]. Available: <http://www.jjtc.com/stegdoc/sec202.html> , Accessed 27 March 2012.
- [66] S. Sachdeva and A. Kumar, "Colour Image Steganography Based on Modified Quantization Table," in *Proc. Int. Conf. Advanced Computing & Communication Technologies*, Rohtak, Haryana, India, 2012, pp. 309-313.
- [67] F. Hartung, and M. Kutter, "Multimedia Watermarking Techniques," in *Proc. IEEE Int. Conf. on Image Processing*, Germany, vol. 87, no. 7, 1999, pp. 1079-1107.
- [68] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding Secret Information into a Dithered Multilevel Image," in *Proc. IEEE Conf. on Military Communication*, Monterey, CA, vol.1, 1990, pp. 216-220.
- [69] G. Doerr, and L. Dugelay, "What is Digital Watermarking" in *Handbook of Video Databases: Design and Applications*, 1st ed., CRC Press Publisher, 2003, ch. 42, sec. 2, pp. 333-334.
- [70] R. G. Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital watermark," in *Proc. Int. Conf. Image Processing*, vol. 1, 1994, pp. 86-90.
- [71] G. Coatrieux, M. Lamard, W. Daccache, J. Puentes, C. Roux, "A Low Distorsion and Reversible Watermark: Application to Angiographic Images of the Retina," in *Proc. 27th IEEE Conf. on Engineering in Medicine and Biology*, Shanghai, China, 2005, pp. 2224-2227.
- [72] Z. Wang, B. Yang, X. Niu, and Y. Zhang, "A Practical Multipurpose Watermarking Scheme for Visual Content Copyright Protection and Authentication," in *Proc. 2006 Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, (IIH-MSP'06), China, 2006, pp. 461-464.
- [73] A. A. Reddy, and B. N. Chatterji "A new wavelet based logo-watermarking scheme," *Pattern Recognition Letters*, vol. 26, pp. 1019-1027, 2005.
- [74] S. C. Liew and J. M. Zain, "Reversible Medical Image Watermarking For Tamper Detection And Recovery," in *Proc. Int. Conf. Computer Science and Information Technology*, (ICCSIT), Chengdu, 2010, pp. 417-420.

- [75] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tampering Proofing and Authentication," *IEEE*, vol. 87, no. 7, 1999, pp. 1167-1180.
- [76] N.A. Memon, S.A.M. Gilani, and A. Ali, "Watermarking of Chest CT Scan Medical Images for Content Authentication," in *Int. Conf. Information and Communication Technologies*, Karachi, Aug. 2009, pp. 175-180.
- [77] D. A. Karras, "A Second order Spread Spectrum Modulation Scheme for Wavelet based Low Error Probability Digital Image Watermarking," *International Journal on Graphics, Vision and Image Processing*, (GVIP), vol. 5, no. 3, 2005.
- [78] L. Li, Y. Fan, and Ch. Chang, "A Reversible Watermarking Algorithm Based on Four-Neighbors Context Prediction for Tongue Images," *International Journal of Intelligent Information Processing*, (IJIP), vol. 2, no. 2, pp. 22 - 28, 2011.
- [79] Y. Yalman, and I. Erturk, "A new histogram modification based robust image data hiding technique," *24<sup>th</sup> Int. Symposium on Computer and Information Sciences*, (ISCIS), Guzelyurt, Turkey, 2009, pp. 39-43.
- [80] Y. Hua, B. Wu, and G. Wu, "A color image fragile watermarking algorithm based on DWT-DCT," in *Proc. 2010 Chinese Control and Decision Conf.*, (CCDC), Xuzhou, 2010, pp. 2840-2845.
- [81] S. Hongqin and L. Fangliang, "A Blind Digital Watermark Technique for Color Image Based on Integer Wavelet Transform," in *Proc. Int. Conf. Biomedical Engineering and Computer Science*, (ICBECS), 2010, pp. 1-4.
- [82] K. Ohzeki, "Discontinuity in SVD Embedding Mapping Used for Watermarks," *International Journal of Computer Science and Applications*, vol. 7, no. 3, pp. 9-17, 2010.
- [83] Z. Fang, Y. Zhao "Image Watermarking Resisting to Geometrical Attacks Based on Histogram," *Int. Conf. on Intelligent Information Hiding and Multimedia*, pp. 79-82, 2006.
- [84] M. Dainaka, S. Nakayama, I. Echizen, and H. Yoshiura "Dual-Plane Watermarking for Color Pictures Immune to Rotation, Scale, Translation, and Random Bending," *Int. Conf. on Intelligent Information Hiding and Multimedia*, pp. 93-96, 2006.
- [85] Jia Li and James Z. Wang, "Real-time Computerized Annotation of Pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985-1002, 2008.
- [86] <http://wang.ist.psu.edu/docs/related.shtml/>, *James Z. Wang Research Group*, Accessed 17 June 2012.
- [87] <http://www.sciencephoto.com/>, *online image library*, Accessed 21 April 2012.