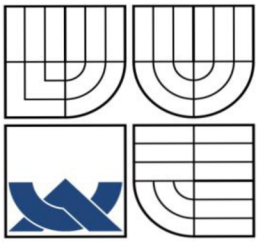
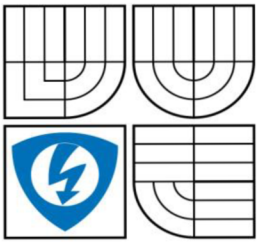


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DIGITAL IMAGE WATERMARKING

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR
ABDULLAH S. A. ALZAID

VEDOUCÍ PRÁCE
SUPERVISOR
ING. PETR ČÍKA

BRNO 2009

BRNO UNIVERSITY
OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

Department of Telecommunications

Bachelor thesis

bachelor's study field
Teleinformatics

Student: Al Zaid Abdullah Saleh ***ID:*** 96880
Year of study: 3 ***Academic year:*** 2008/09

TITLE OF THESIS:

Digital Image Watermarking

INSTRUCTION:

Peruse today's options of digital image watermarking in the frequency domain. Propose 2 methods for watermark embedding and extracting using the discrete wavelet transform and the discrete cosine transform. Compare these two methods by means of the Checkmark benchmark tool.

REFERENCE:

[1] LU, Chun-Shien. Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. London : Idea Group Publishing, c2005. 255 s. ISBN 978-1591402756.
[2] ARNOLD, Michael, WOLTHUSEN, Stephen D., SCHMUCKER, Martin. Echniques and Applications of Digital Watermarking and Content . Norwood : Artech House, Inc., c2003. 274 s. ISBN 978-1580531115.

Assigment deadline: 10.1.2009 ***Submission deadline***31.10.2009

Head of thesis: Ing. Petr Číka
Consultant:

prof. Ing. Kamil Vrba, CSc.
Subject Council chairman

WARNING:

The author of the bachelor thesis must not in creating this thesis infringe the copyrights of third parties, in particular, he must not infringe upon the rights of a foreign copyright personality in an illegal way and must be fully aware of the consequences of a breach of the provisions of Section 11 and the Copyright Act No. 121/2000 Coll., including possible criminal consequences resulting from the provisions of Section 152 of Criminal Law No. 140/1961 Coll.

Digital Image Watermarking

AL ZAID, A. S. *Digital Image Watermarking*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 82 s. Vedoucí bakalářské práce Ing. Petr Číka.

Abdullah S. A. Alzaid

Declaration of Originality

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma "**Digital Image Watermarking**" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení §11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne …….

_____

(podpis autora)

Abdullah S. A. Alzaid

Abstract

This is the Final Project Report as being composed of an extensive summary of activities and results made by the student Abdullah S. A. Alzaid while undertaking the Watermark Project. This project acquaints an algorithm of digital watermarking which is based on Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). In accord with the characters of human vision, the main objective of the project is to be focused on developing an image watermarking algorithm by taking advantage of both the DCT and DWT transforms and analysis of the algorithm on the basis of invisibility, distortion and robustness to attacks. The simulation results show that this algorithm is invisible and has good robustness for some common image processing operations. By the use of Matlab software, the two structures have been coded and then implemented properly.

Table of Contents

List of Figures

Glossary

2-dimensional Signal: the two-dimensional signal is decomposed into levels or scales using wavelets which are derived from the scaling function $\varphi(x)$ and its translates $\varphi(2x)$ 42

Blind Watermark Detection: is the watermark detecting method which does not require a cover image in order to extract the watermark from the watermarked image 17

Cover Image: is the original image that has been used in the watermarking operation 28

DCT: Discrete Cousin Transform 11

DFT: Discrete Fourier Transform. 23

DWT: Discrete Wavelet Transform. 11

FDCT: Forward Discrete Cousin Transform. 29

FDWT: Forward Discrete Wavelet Transform. 39

F_H: High-band frequencies. 30, 50

F_L: Low-band frequencies. 29

F_M: Middle-band frequencies. 30

Gaussian Noise: is statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution) 44

HH: High-high Frequency coefficients. 43

High-pass: is A filter that blocks signals below a specific frequency while allowing signals above that frequency to pass 40

HL: High-low Frequency coefficients. 43

HVS: Human Visual System. 16

IDCT: Inverse Discrete Cousin Transform. 23

IDWT: Inverse Discrete Wavelet Transform. 41

JPEG: Joint Photography Experts Group. 30

LH: Low-high Frequency coefficients. 43

Chapter 1: Introduction to Digital Watermarking

1.1 Motivation

At the present time, the aptitude in contemplation of accessing as well as sharing images has become progressively facile with the Internet allowing people to procure information remotely from anywhere in the entire world. Moreover, there has been also an expansion with regard to the number of the still digital images over the internet for the sake of the fact that a vast number of millions of people are capturing digital photos. This mentioned fact could bring forth the requirement for people to conserve their own images or intellectual properties. Given the motivation to protect intellectual property, Digital Watermarking technology has been referred to as fit for acceptance as a form of copyright protection and a preventing those who have such an ambition in order to get a hold of such multimedia date either image disproportionately.

Fundamentally, the procedure of digital watermarking can be delineated as a method for embedding information into another signal (a digital signal). In case of digital images, the embedded information can be either visible or hidden from the user. In this project, we will concentrate on imperceptible watermarks. The principal intention of digital watermarks is to provide copyright protection for intellectual property that is in digital format. Typical usage scenarios for watermarking are such as copyright protection and data authentication.

1.2 Aims

The main objective of the entire project is to design two methods by means of Matlab features. A huge part of the concentration of the research has covered the different techniques of how to create and well as how to insert a watermark into a digital still image, after all. There is a demand in order to carry forward this project into the domain of making the watermarks robust and secure. The vital aim of this project was to scrutinize the possible techniques that are robust enough as being applied into the transform domain and develop a watermarking program that would be able to implement the robust technique. A watermarking system was also submitted for the practice of using with digital images. Based on the digital watermarking for a still image, we are to formulate and encode the following methods:

- The embedding and detecting procedure for watermarking technique based on DCT transform.
- The embedding and detecting procedure for watermarking technique based on DWT transform.
- Computing PSNR function (peak single-to-noise ratio) the resultant watermarked images from both of the techniques DCT/DWT for the purpose of measuring the distinctive distortion between the cover image and the watermarked image.
- Applying the checkmark software by means of NCC function for the original watermarks and extracted watermarks from the DCT/DWT techniques.

Chapter 2: Applications of Watermarking

The technology of digital watermarking has been submitted to be implemented in such many different applications as shown in the next subsections.

2.1 Copyright protection

- Can be applied to most of the prominent implementations in existence at the present time for supplementing the vital issues of the copyright protection.
- It gives the kind of allowance for the owner to embed such information that relates to him/her for the purpose of preventing those without official authorizations from asserting such copyright.
- In consideration of this field of application, it would absolutely necessitate a great level of robustness which is one type of the watermarking requirements.

2.2 Copy protection

- The embedded watermark within this application has the future of disallowing whatever unauthorized duplication that might occur to the original cover.
- As an illustration, in case of an acquiescent DVD player, it will not playback, and in the same situation, there could be also such data that carry out the watermark sign (copy never) won't function out unless the multimedia item has been purchased from the owner.

2.3 Content Authentication

- This sort of implementation is performed for the intention of detecting no matter which potential modifications might occur to the cover item.
- Forasmuch as in this matter in particularly, the digital watermark will be a type of watermarking technique known as (fragile watermarking).

2.4 Tamper Detection and Localization

4

- The localization and tamper detection ability is more the same as being related to the data authentication application somehow.
- The main goal of the tamper detection is to reveal the possible alterations that could perhaps occur to the cover due to such manipulations or modifications and so on.
- In a matter of detecting the tamper in the multimedia item such as an image, then it would analyze the case of not being that object genuine.
- The tamper localization ability however, permits supplementary investigations by acting as a tamper which can lead straight ahead to identify the regions of the multimedia object which have been tampered.
- In similarity within the data authentication application, tamper and detection localization technique can also be attained by means of using either one of the watermarking requirements such that robust, fragile or even semi-fragile watermark.

2.5 Transaction Tracking

- This particular application is applied in contemplation of embedding the digital watermark in the interest of carrying out the desired information that relates to the legal recipient of the original cover.
- This scheme in vital for the purpose of either supervising or else investigating whichever copies of the original cover which are being produced illegally to the public either to individuals.
- This implementation however is usually referred to as (fingerprinting).

2.6 Broadcast Monitoring

- This application embeds the desired watermark into the cover and using an intentional monitoring to ascertain whether the cover has been broadcasted as it was agreed on or not.
- As well, the watermark can be embedded into the public announcement section.

2.7 Application Fields by Target Contents

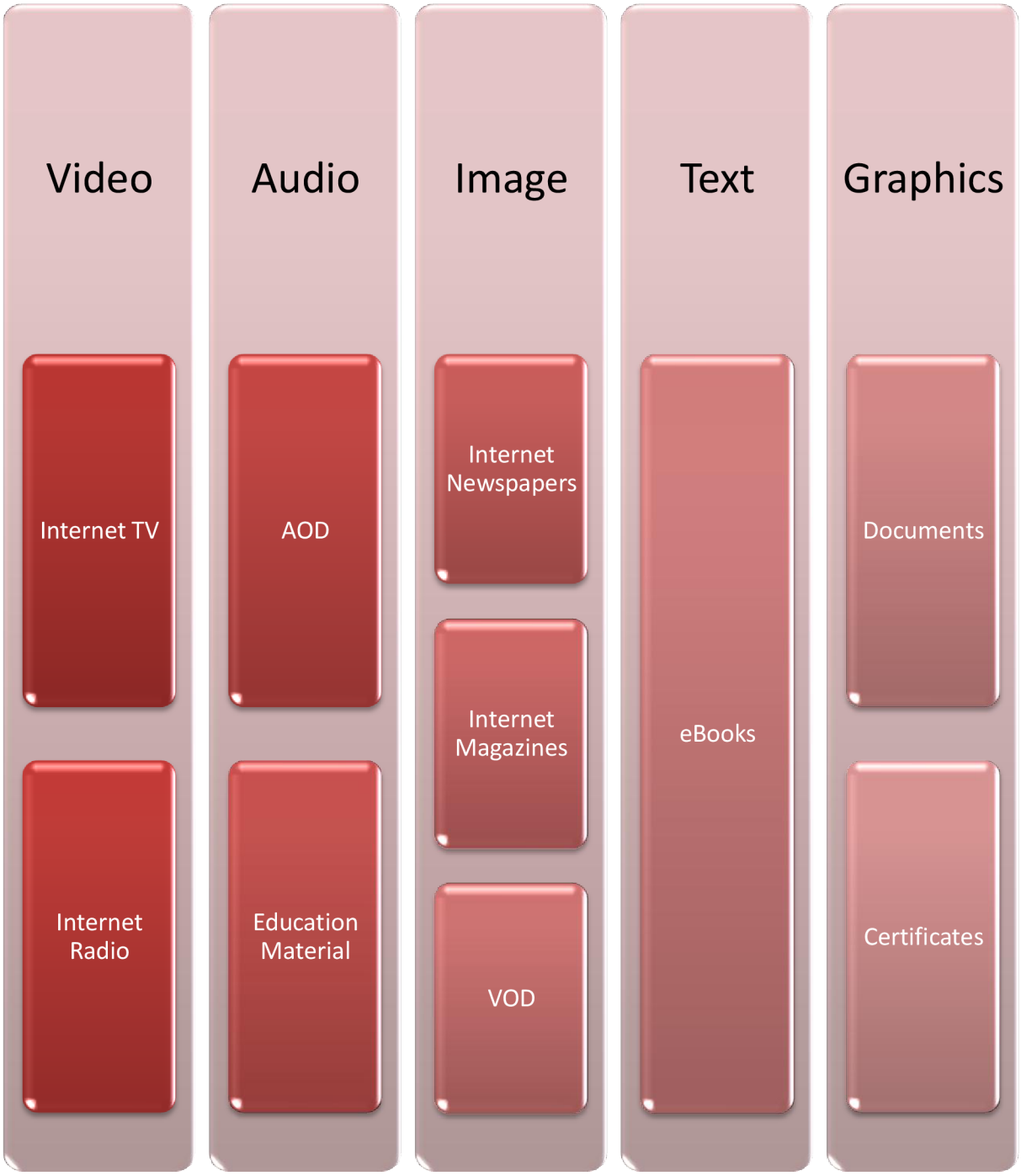


Figure 1. The possible application fields where digital watermarking technology can be involved with

Chapter 3: Digital Watermarking Concepts

3.1 The Fundamental of Digital Watermarking

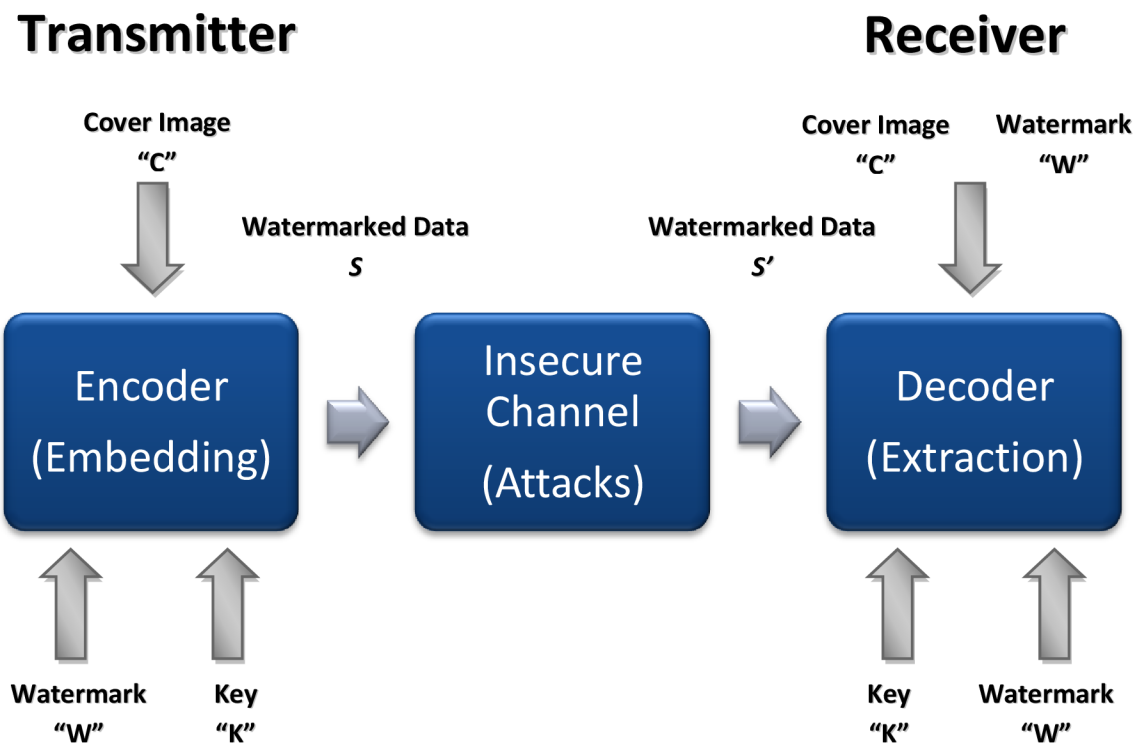


Figure 2. The generic scheme for digital watermarking technique

All digital watermarking schemes could possibly partake in the same generic principal of the watermarking implementation which are the two watermarking systems and known as embedding and extracting systems. The scheme's input is the watermark itself and it can be such an image or a secret key. The digital watermark can be formed in many different forms such as a text, a number or even an image. The real use of the key the scheme has is to compel the security which then can prevent those unauthorized parties from manipulating either from recovering the watermark. The watermark scheme will have an output which is the watermarked data [1].

3.2 The Requirements of Watermarking

There are several common requirements and properties being jointly held by most of all digital watermarking systems as we will describe below:

3.2.1 Perceptual transparency

The substance that has been watermarked similarly would have the same subjective quality as the original contents.

3.2.2 Imperceptibility

This is an important property which is usually called as imperceptibility of digital watermark or sometimes it is indicated as fidelity or else called perceptual transparency. This property however can be described as the characteristic of hiding a watermark so that it does not degrade the visual quality of the image. Moreover whichever modifications occurred by means of watermark embedding should be then below the perceptible threshold. During the watermark embedding process the models of the human visual system (HVS) can be applied for the reason of enhancing the imperceptibility as well as the robustness of the watermark itself.

3.2.3 Robustness

It's the capability of the watermark to withstand distortion that has been introduced by standard or malicious data processing. No person has the ability to eliminate, modify, or damage the watermark without a secret key.

3.2.4 Security

It is the ability that watermark can resist malicious attacks. A watermark is secure if knowing the algorithms for embedding and extraction, it would not help unauthorized party to detect or remove the watermark. Secret key determines the value of watermark and the locations where the watermark is embedded.

3.2.5 Payload

The payload of watermarking is the amount of information to be embedded. In other words, it is the number of bits which are encoded into a message or else the data payload can be thought as the encoded message size of a watermark in such an image.

3.2.6 Capacity

Capacity is the amount of watermark information in an image. Multiple watermarks can be embedded and extracted. For instance, if multiple watermarks are being embedded into an image, then the watermark capacity of the image is the sum of the individual watermark's data payload.

3.2.7 Blind Watermark Detection

The blind detection of the digital watermark is referred to the ability of detecting the invisible information without the need for the reference image. The reference image however can be either the original image (cover image) or it is possible to be the stego image with such distinctive digital watermark or else the possibility of being a non-distorted one (stego image). Blind detection is a vital practical feature of watermarking technology so that in order to implement the extracting system for instance, the watermarking method itself should not be relying on the reference image. On the other hand, it should supply the blind detection feature which then can use the image under test only. In a different meaning, we can simply detect the watermark by the use of the test image only based on the blind watermark detection feature. This detection technique will take the test image as an input, and then later on execute the appropriate algorithm for the detection process, and as a result, it will output out the watermark that has been detected.

3.2.8 Non-blind Watermark Detection

On the other hand, the non-blind detection of the digital watermark is more the same as the previous mentioned type that is the blind detection feature of the watermark, except that the non-blind detection type always demands for the reference image for the reason of extracting out the watermark. However, this future is somehow impractical due to the probability of being the reference image not readily attainable, but it can be more accurate for detecting the watermark signature [2].

3.3 Some Fields of Implementation

- **Text watermark**
  This field gives variety to spaces as subsequent to punctuation, and creating spaces in between lines of text, as well as spaces at the end of each sentence and so on.

- **Audio watermark**
  It is applied with such frail bit coding, also fortuitously undetectable noise, and contains fragile and robust watermarks.

- **Images watermark**
  It is dealing with the least-significant bit, as its treatments involve the random noise, and causing the features of filtering and masking techniques, and so forth.

Chapter 4: Digital Watermarking Techniques

4.1 Watermark Types

There are several procedures for the intention of classifying the methods of watermarking. Such one of the most widely adopted systematic arranging is based on the robustness of watermarking. Beneath this category, digital watermarking can be sorted into three types as described below:

- **Robust watermark**
  This type is the watermark that has the feature to oppose the non-malicious distortion; (was described in deep details in the above section 3.2).

- **Fragile watermark**
  The fragile watermarking classification can be easily destroyed by all image distortions.

- **Semi-fragile watermark**
  This type can be destroyed by certain types of distortions while it can withstand some other minor changes.

As long as robust watermarks can resist common image processing operation, they would be precisely suitable for copyright protection. Fragile watermark, on the other hand, can be used to discern the modification and verify an image since it's too sensitive to possible changes. However, semi-fragile watermarks are very often applied in some special cases of verification and tamper detection. These arguments may consider lossy image compression as legitimate modifications while highlighting distortions as premeditated attacks.

In addition to watermark robustness, digital watermark can be as well classified into either visible or invisible watermark types.

4.2 Digital Watermarking Classifications

4.2.1 Visible Watermarking

In the visible watermarking technique, the structure is observable in the image or video for the observer. In a characteristic manner, the information can be referred as either text or a logo which can then acknowledge the rightful owner of the multimedia item. In case that when television broadcasters add up their logo to the corner of broadcast video, this is considered as a visible watermark [3].

4.2.2 Invisible Watermarking

In the invisible watermarking approach, the information (watermark) is supplemented as digital data to the entities of multimedia such that video, audio or even still images; however, the inserted hidden information would not be distinguished as such. A significant application of this technique (invisible watermarking) is to copyright protection systems, which are designed for the reason of preventing or even deter such unauthorized copying of the digital multimedia. Furthermore, there happens to be also another type known as Steganography which is an application of digital watermarking, where two parties are capable of communicating together via a secret message that is intended to be embedded in the digital signal. Annotation of digital photographs with descriptive information is another application of invisible watermarking. We are to express some thoughts about this technique through the next issue.

4.2.3 Steganography

This classification can comprise distinctive methods for concealing the existence of the additional information in a signal. The difference between Steganography and watermarking does not seem to be regularly obvious. Essentially, in case of watermarking the additional information is used to protect the original image (e.g. in case of copyright management), while on the contrary in the Steganography the image is used to protect the additional information (e.g. secret message) [4].

4.3 Visible Image Watermarking Technique

Figure 3. An example of a visible digital watermarking where you can see the watermark "Abdullah Alzaid" being perceptible enough

- A visible watermark is considered to be ostensible enough in both color and monochrome images.
- The watermark ought to be scattered in a spacious or consequential region of the image in order to hinder its obliteration by trimming off.
- The watermark need to be perceptible and must not necessarily blear details of the image underneath it.
- The watermark must be rigid to amputate.
- Amputating the watermark should be pricier and require a lot of work than procuring the image from the one who owns it.
- The watermark should be adjusted accordingly to the human interference and exertion.

4.4 Invisible Image Watermarking Technique

In this project, we are to go through one widely known approach used to address this technique is to add an invisible structure into a digital image which then can be used for purpose of trading or marketing copyright protection. Watermark imperceptibility is a common requirement and independent of the application purpose. However, it should be noted that digital watermarking can be implemented by either embedding or detecting the structure in such distinctive types of domains.

The invisible image watermarking consists of two major domain techniques regarded as follow:

- The spatial-domain technique
- The transform-domain technique

Generally speaking, digital watermarks can be embedded in the spatial domain or as well within the frequency domain. The most forthright approach is the watermarking in the spatial-domain where the pixel values can be modified in order to encode the watermark signal

One vital public method that uses the spatial domain is the Patchwork technique, which takes a statistical approach. It is founded on a pseudorandom, statistical process. Pairs of image points are randomly chosen and the brightness at one point is increased while the brightness of the corresponding point is decreased. The expected value of the sum of the differences of the n pairs of points is then 2n. The Patchwork technique is considered to be secure, in that it is difficult to remove the Patchwork coding without degrading the picture beyond recognition, although this method has a limitation on the amount of information that can be embedded. Another method is the Pulse Embedding System, where the position sequence is used to generate a sequence of pixel-mapped locations where the code is then embedded. The code pulses are superimposed on the signal-selected locations and then the quantized data is decoded and inversely transformed to produce the labeled image data. Watermarking in the spatial domain has its downfalls in that the watermark can be destroyed because the domain deals with the image pixels or luminance information. Hence any translation, rotation, scaling, compression or any change to the image can cause the destruction of the watermark, depending on its robustness. Therefore over the years the research focus has shifted from the spatial domain to the frequency domain [5].

4.5 The Frequency (Transform) Domain

In contemplation of the fact that a digital watermarking technique to be efficient enough, then it should be imperceptible, and robust to all common image manipulations available today such that rotation, filtering, scaling, compression, cropping and collusion attacks among many other digital signal processing operations. As we have mentioned antecedently that current digital image watermarking techniques can be classified into two major groups determined as spatial-domain and frequency-domain watermarking techniques. Equated with spatial domain techniques, frequency-domain watermarking techniques have been proven to be much more effective with regard to accomplishing the imperceptibility and robustness requirements of digital watermarking algorithms.

The frequency-domains such as DCT transform and DWT transform are more specifically to be dealing with digital image watermarking system. However, the frequency domain has an advantage over the spatial domain in those frequency-based schemes which can scatter the watermark over the whole spatial domain of the image; thus, it would be less afflicted with cropping. The Discrete Cosine Transform (DCT) is widely deliberate due to the fact that watermarks embedded in the DCT domain are often more robust to JPEG and MPEG compression, but as in general both the DCT and DWT transforms have been extensively used in many digital signal processing applications. Within the next subsections, we will focus on the DCT and DWT transforms, and outline their relevance to the implementation of digital watermarking.

Chapter 5: Discrete Cosine Transform (DCT)

5.1 Comprehension of the (DCT) technique

In order to make it possible to come across the watermarking program that can be implemented properly and has enough robustness in the transform domain, we are to choose a transform function that could be performed in the frequency domain. The fact is that there are multitude transform functions available such that functions include the discrete Fourier transform (DFT), or the discrete wavelet transform as well as the discrete cosine transform and so as other transform functions. The first technique to be chosen will be based on the DCT, with the main avail being that it is utilized with the Joint Photographic Experts Group (JPEG) standard, which is a file format that is widely used on the Internet. This guarantees that the watermark is robust against JPEG compression.

The discrete cosine transform (DCT) is used to transform a signal from the spatial domain into the frequency domain. The reverse process, that of transforming a signal from the frequency domain into the spatial domain, is called the inverse discrete cosine transform (IDCT).

A signal in the frequency domain contains the same information as that in the spatial domain. The order of values obtained by applying the DCT is coincidentally from lowest to highest frequency.

This feature and the psychological observation that the human eye and ear are less sensitive to recognizing the higher-order frequencies leads to the possibility of compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones. When reconstructing the signal and transforming it back to the spatial domain, the results are remarkably similar to the original signal.

5.2 The DCT Transform Algorithm

5.2.1 General Methodology

The discrete cosine transform (DCT) is a function that has the ability to convert a signal into elementary frequency components. It represents an image as a sum of sinusoids of varying magnitudes and frequencies. With an input image, *x*, the DCT coefficients for the transformed output image, *y*, are computed according to equation (1) shown below.

$$y(u,v) = \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}\alpha_u\alpha_v\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}x(m,n) \cos\frac{(2m+1)u\Pi}{2M}\cos\frac{(2n+1)v\Pi}{2N} \quad (1)$$

In the equation, *x*, is the input image having *N x M* pixels, *x(m,n)* is the intensity of the pixel in row *m* and column *n* of the image, and *y(u,v)* is the DCT coefficient in row *u* and column *v* of the DCT matrix.

The image is reconstructed by applying inverse DCT operation according to equation (2) shown below.

$$x(m,n) = \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\alpha_u\alpha_v y(u,v) \cos\frac{(2m+1)u\Pi}{2M}\cos\frac{(2n+1)v\Pi}{2N} \quad (2)$$

The popular block-based DCT transform segments an image non-overlapping block and applies DCT to each block. This result in giving three frequency sub-bands: low frequency sub band, mid-frequency sub-band and high frequency sub-band. DCT-based watermarking is based on two facts. The first fact is that most of the signal energy lies at low-frequencies sub band which contains the most important visual parts of the image. The second fact is that high frequency components of the image are usually removed through compression and noise attacks. The watermark is therefore embedded by modifying the coefficients of the middle frequency sub band so that the visibility of the image will not be affected and the watermark will not be removed by compression [6].

5.2.2 Precise Methodology

The fundamental method is exploited to inlay and bring back the watermark that has been inserted, and it is as shown in Figure 4 above. This technique however is on the basis of embedding the watermark by using two DCT. Equation (3) was employed for the purpose of calculating the forward DCT and equation (4) has been exploited in order to calculate the reversed DCT coefficients.

$$F(k,l) = \frac{2}{N} c(k)c(l) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n.m) \cos\left[\frac{(2n+1)k\pi}{2N}\right] \cos\left[\frac{(2m+1)l\pi}{2M}\right] \quad (3)$$

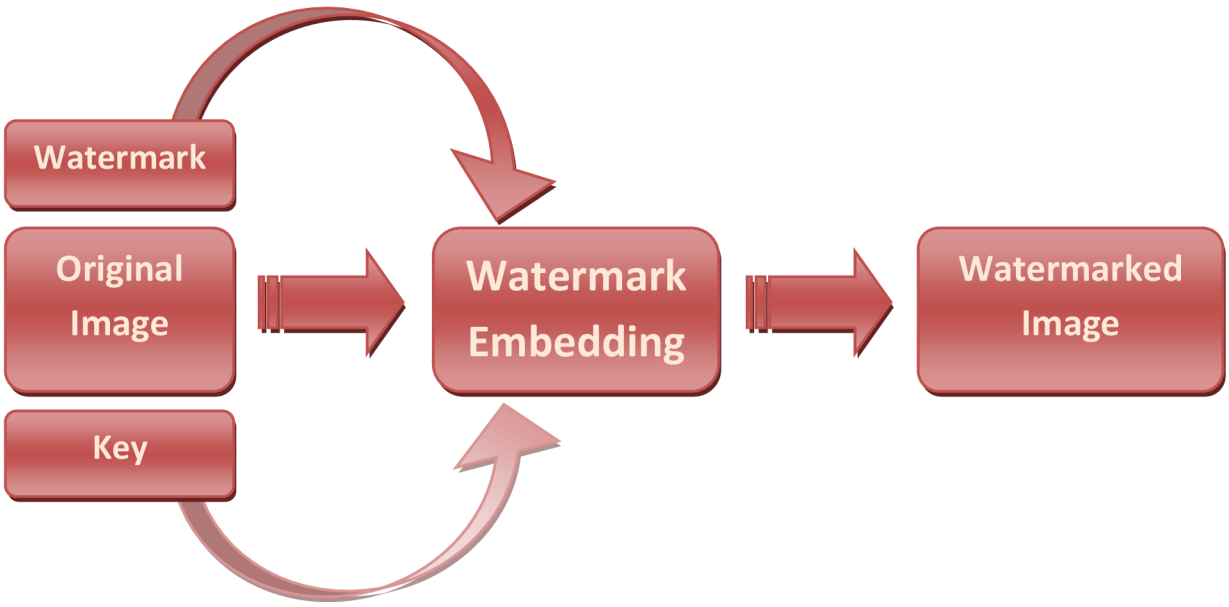


Figure 4. The generic scheme for embedding the watermark (DCT)

$$f(n,m) = \frac{2}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} c(k)c(l)F(k.l)\cos\left[\frac{(2n+1)k\pi}{2N}\right]\cos\left[\frac{(2m+1)l\pi}{2M}\right] \qquad (4)$$

With $k,l,n,m = 0,1,2,\ldots, N-1$

Where n,m = spatial coordinates in the pixel element domain

K,l = coordinates in the transform domain

$C(k), c(l) = \frac{1}{\sqrt{2}}$, for $k,l = 0$

= 1, for $k,l = 1,2,\ldots, N=1$

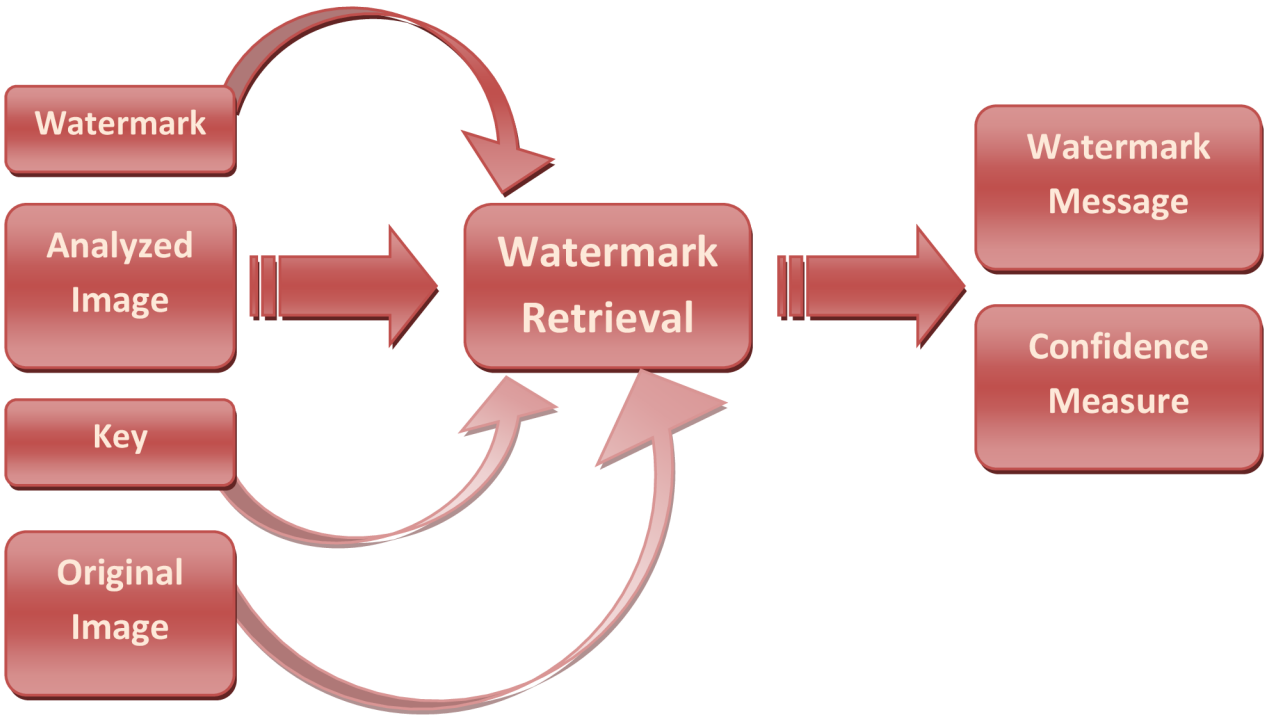


Figure 5. The generic scheme for extracting the watermark (IDCT)

5.3 Embedding and Extracting Systems

Within the next coming up figure 6, we can obviously observe the essential procedure that could insert the desired watermark into the DCT transform of the cover image. As well that the watermark can be extracted once again due to the procedure of executing the DCT transform on the watermarked image.

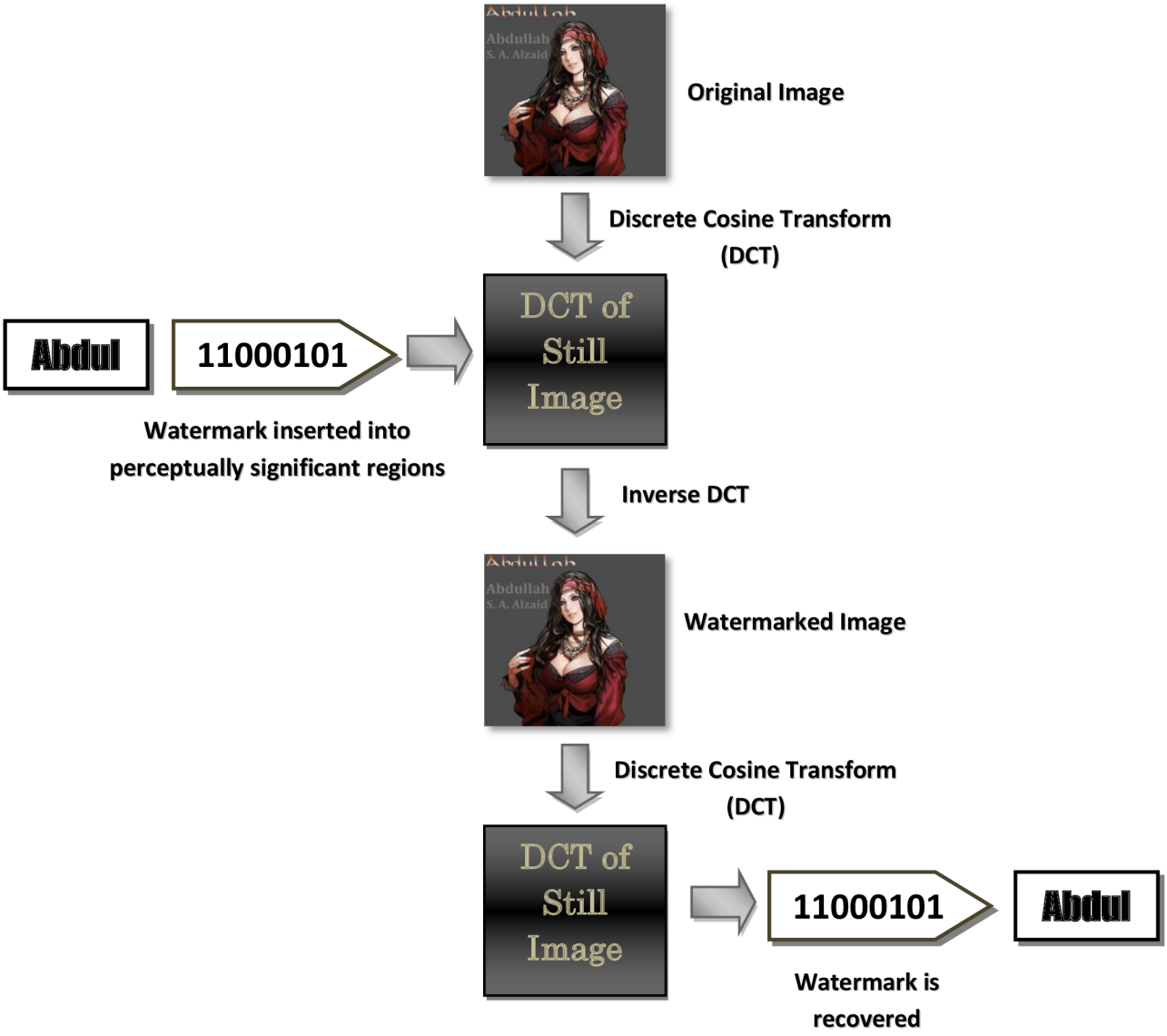


Figure 6. The fundamental technique that is used in order to inlay and extract the watermark

5.3.1 Embedding System Based on (DCT) Transform

To begin with processing the watermark embedding system, first of all, let's give some useful notations that can be beneficial and used later on as we go through the procedure. *X* is referred to the original image (cover image), and its size N_1 x N_2. Moreover, let us suppose *Y* as a notation for the watermark image and its size noted as M_1 *X* M_2. As long as the fact that the middle-frequency range of the original image (cover image) will be the only range to be processed during the whole watermark embedding system, the size of the watermark image that we donated as *Y* is supposed to be smaller than the size of the cover image *X*.

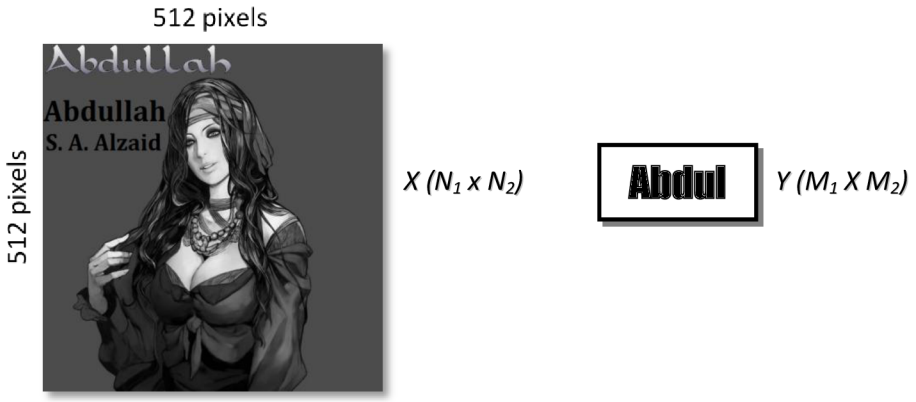


Figure 7. The cover image *X* and the watermark *Y* before splitting into blocks

Step 1: Both the cover image *X* and the watermark *Y* must be divided into blocks. With regard to the original image *X*, we can split it into 8 x 8 pixels for instance; and the watermark since it's assumed to be smaller, then it needs to be divided into the size of $(8x\frac{M_1}{N_1})$ x $(8x\frac{M_2}{N_2})$ pixels.

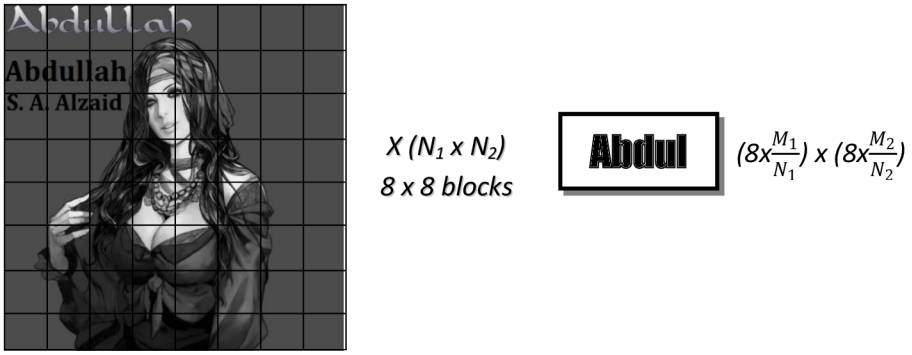


Figure 8. The original image is divided into (8x8) blocks

Step 2: For the purpose of enhancing the perceptual invisibility, we need to take within consideration the understanding of the original image characteristics. Thus, those divided blocks from the watermark which contain such more details or more complex contents should be in fact embedded into those blocks of the original image which also consist of more information.

Step 3: After breaking the original image *X* into blocks, the DCT transform is then applied and performed for each block of the image *X* so that each single block can be DCT transformed independently. This can be represented as in the following notation,

A = FDCT(X)

Where the abbreviation *FDCT* is noted as the operation of forward DCT of the image *X*.

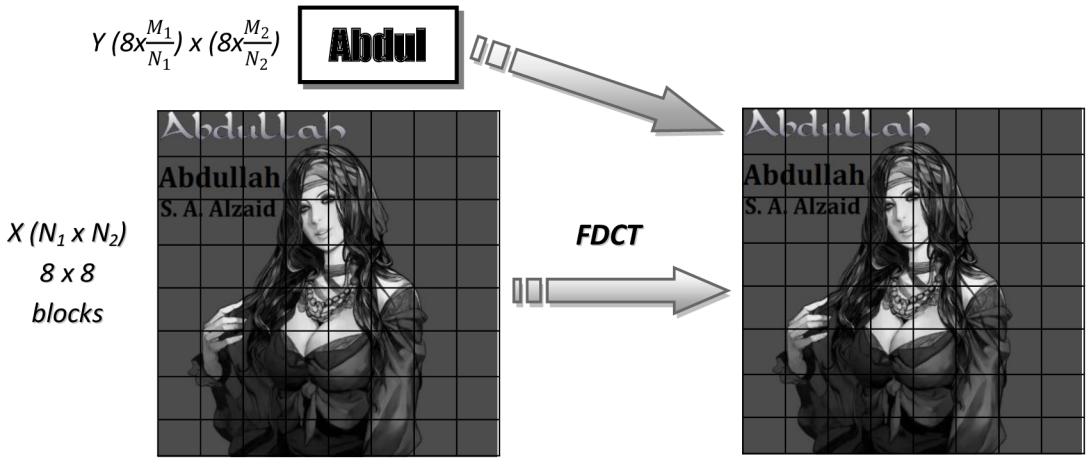


Figure 9. (The encoding process) obtaining the image *A* from processing the *FDCT* into the original image *X* and the watermark *Y* by converting the original image from the spacial domain into the frequency domain

Step 4: Since we have got the image *X* and the watermark *Y* converted into the frequency domain as in the new resultant image *A*, all we need to do next is extracting out the middle-frequency coefficients (F_M) from the computed image *A*. In fact, there is a significant cause for the reason of choosing the middle-frequency coefficients of *A* and it is due to two significant folded as follow:

- **The first basis** is that the human eye indeed is more sensitive to whatever noise existing within the frequency components of the low region (F_L), so it's more

sensitive to the noise in those lower frequency components than to the components of the higher frequencies. Therefore, we should not replace the watermark into those components where the low frequency region is placed; otherwise the watermark would arise then become somehow perceptible right after the process of the watermark embedding.

- **The second basis** however is the fact that since the components of the higher frequencies are being affected by means of the quantization operation for the JPEG lossy compression, replacing the watermark into this frequency band of the high region (F_H) may discard the watermark while performing the lossy compression. Consequently and in order to embed the watermark signature properly without the causes of being either perceptible or discarded, the appropriate domain to insert it is through the frequencies of the middle-region as it is shown in figure 10 below. Thus, the watermark can be then embedded properly.

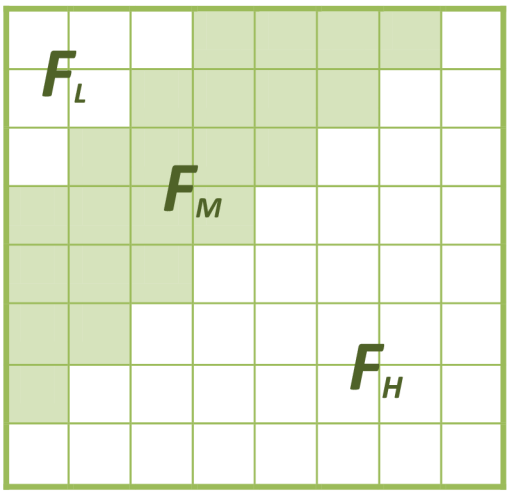


Figure 10. The common interpretation of DCT regions

Step 5: for any DCT transformed image that has been divided into the size of 8x8 block (e.g. image *A*), out of 64 coefficients there will be only two-middle coefficients chosen $(8x\frac{M_1}{N_1})$ *x* $(8x\frac{M_2}{N_2})$. Those coefficients are from the F_M (middle-band frequencies) and they are selected with respect to the operation of the quantization table for the JPEG lossy

compression, and those chosen coefficients can then compose the reduced image that has the size $M_1 X M_2$ which is the same size as the watermark.

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

Figure 11. Two middle frequency coefficients being selected based on the quantization table of the JPEG lossy compression

Step 6: Afterwards, we will attain the reduced image as well as obtaining the modified digital watermark which both have the size $M_1 \times M_2$, see figure 12. Each block of the reduced image and the watermark of the size $(8x\frac{M_1}{N_1}) \times (8x\frac{M_2}{N_2})$ at the corresponding spacial region will be then modified for the point of embedding the watermarked pixels.

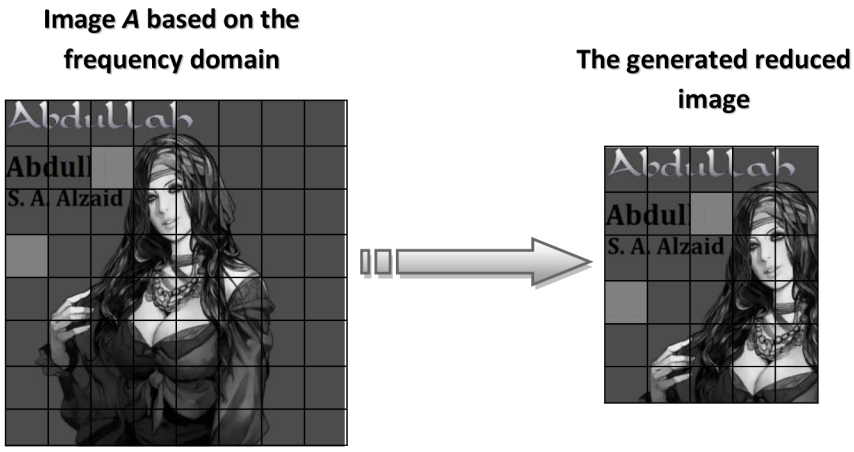


Figure 12. Attaining the reduced image that contains of the middle-band frequency coefficients only

Step 7: The final procedure can be performed by mapping the middle-frequency coefficients we achieved into the resulted image in the frequency domain *A* which leads us to come by the *A`*. The associated result we have got *A`* would then require the inverse function of the DCT for the purpose of accomplishing the watermarked image *X`*. *X` = FDCT(A`)*.

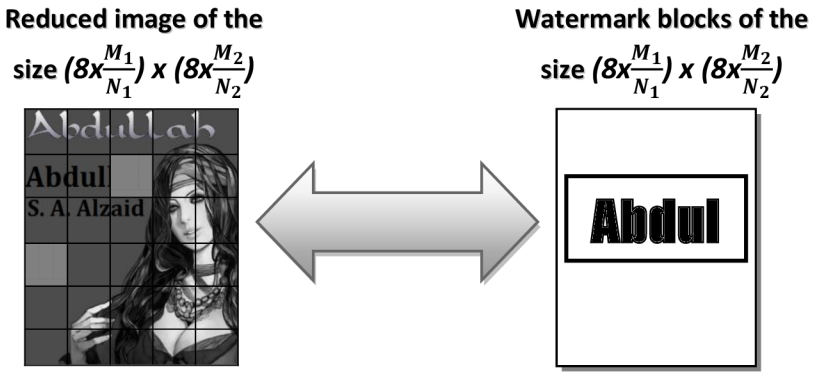


Figure 13. Modifying the reduced image and the watermark blocks for the case of embedding the watermark pixels

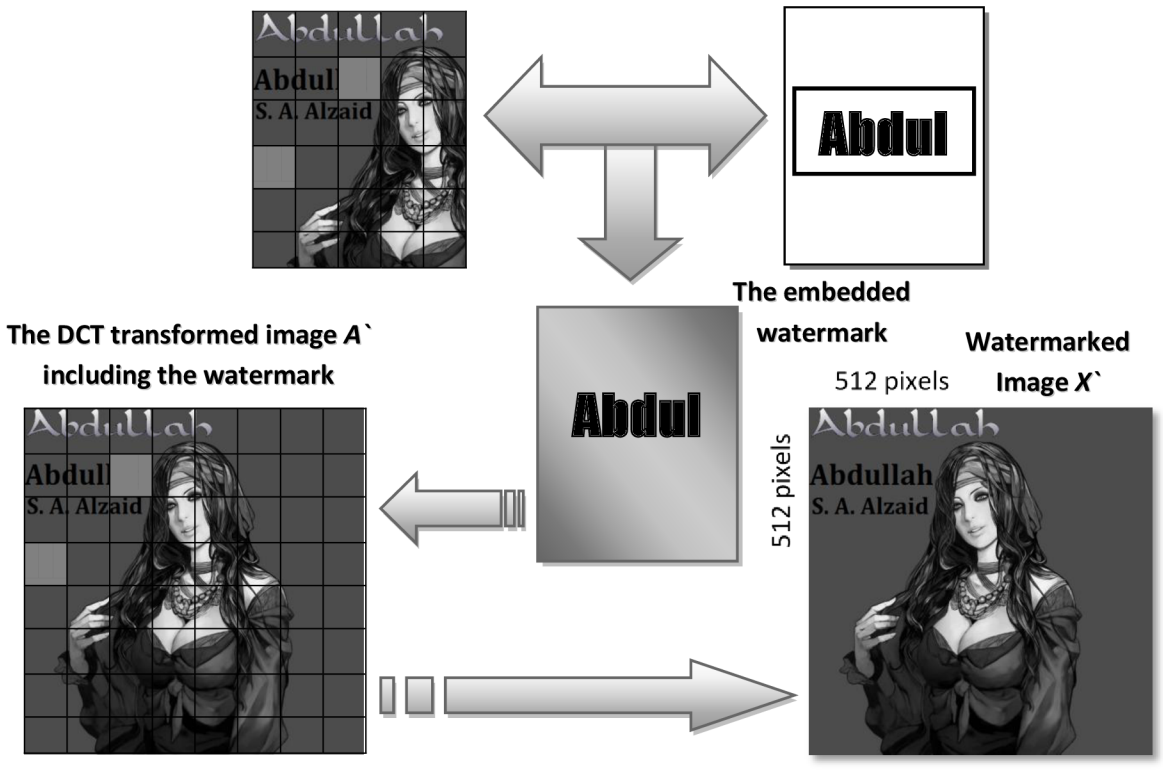


Figure 14. Embedding the watermark into the DCT transformed image *A`* then resulting out the watermarked image *X`*

5.3.2 Extraction System Based on (DCT) Transform

Throughout the above section, we have gone through the embedding system based on the discrete cosine transform (DCT) function by analyzing its procedure step by step and articulating its functionality. Now we are to express the proper technique for extracting back the invisible watermark from the watermarked image and analyzing this scheme by such following up stages.

Step 1: Before all else, the extracting system requires all of the original image *X* as well as the resultant watermarked image *X`* and either the digital watermark.

Step 2: The discrete cosine transform function should be applied to both of the watermarked image and the reference image through the decoding process, so that the images can be DCT transformed after being computed and converted into the frequency domain in consequence of the functionality of the DCT.

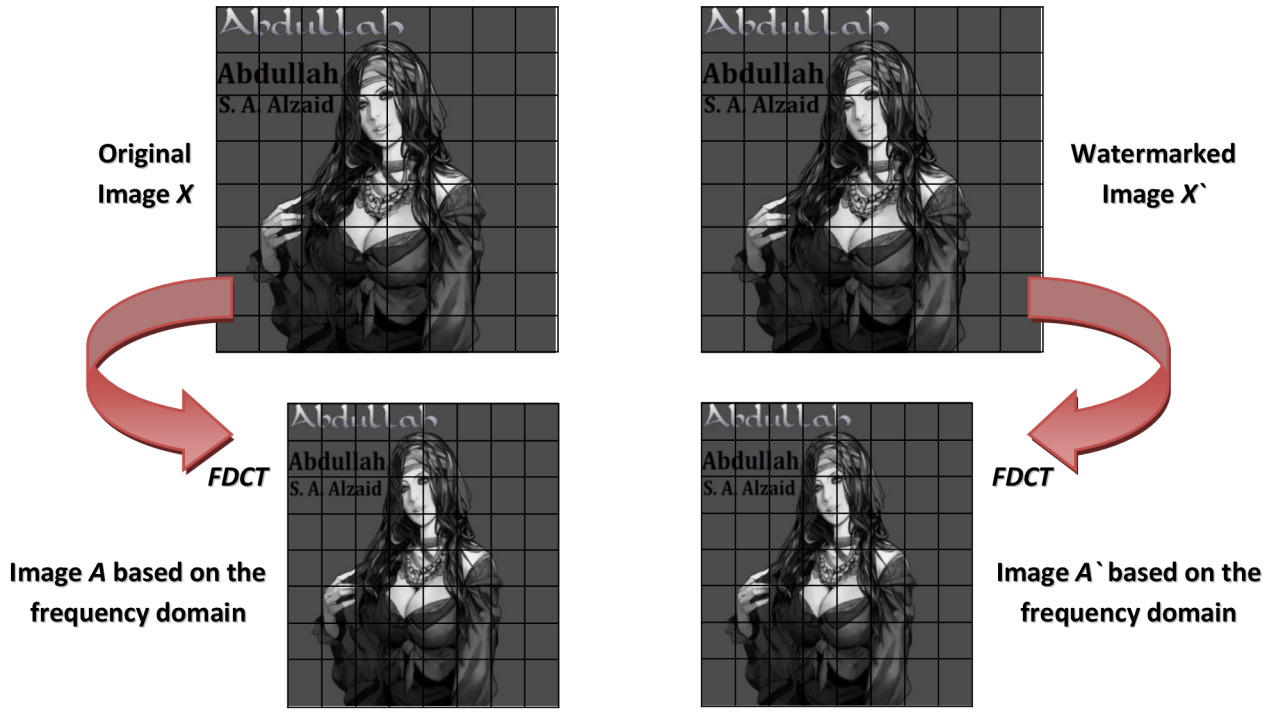


Figure 15. (Decoding process) the *FDCT* is applied to the original image as well as the watermarked image

Step 3: Now since we have already applied the DCT for the purpose of decoding the reference image as well as the watermarked image in virtue of the property (non-blind

detection of watermarking); what we should do thereafter is to generate the reduced image which will help us to get rid of the low frequency and high frequency DCT components as long as that reduced image contains only of the middle frequency coefficients.

Step 4: That generated image (reduced image) which is composed of the middle-band frequencies (F_M) will then give us the chance for cultivating the polarity pattern. So then we are almost in the last stage to retrieve back the embedded watermark signature.

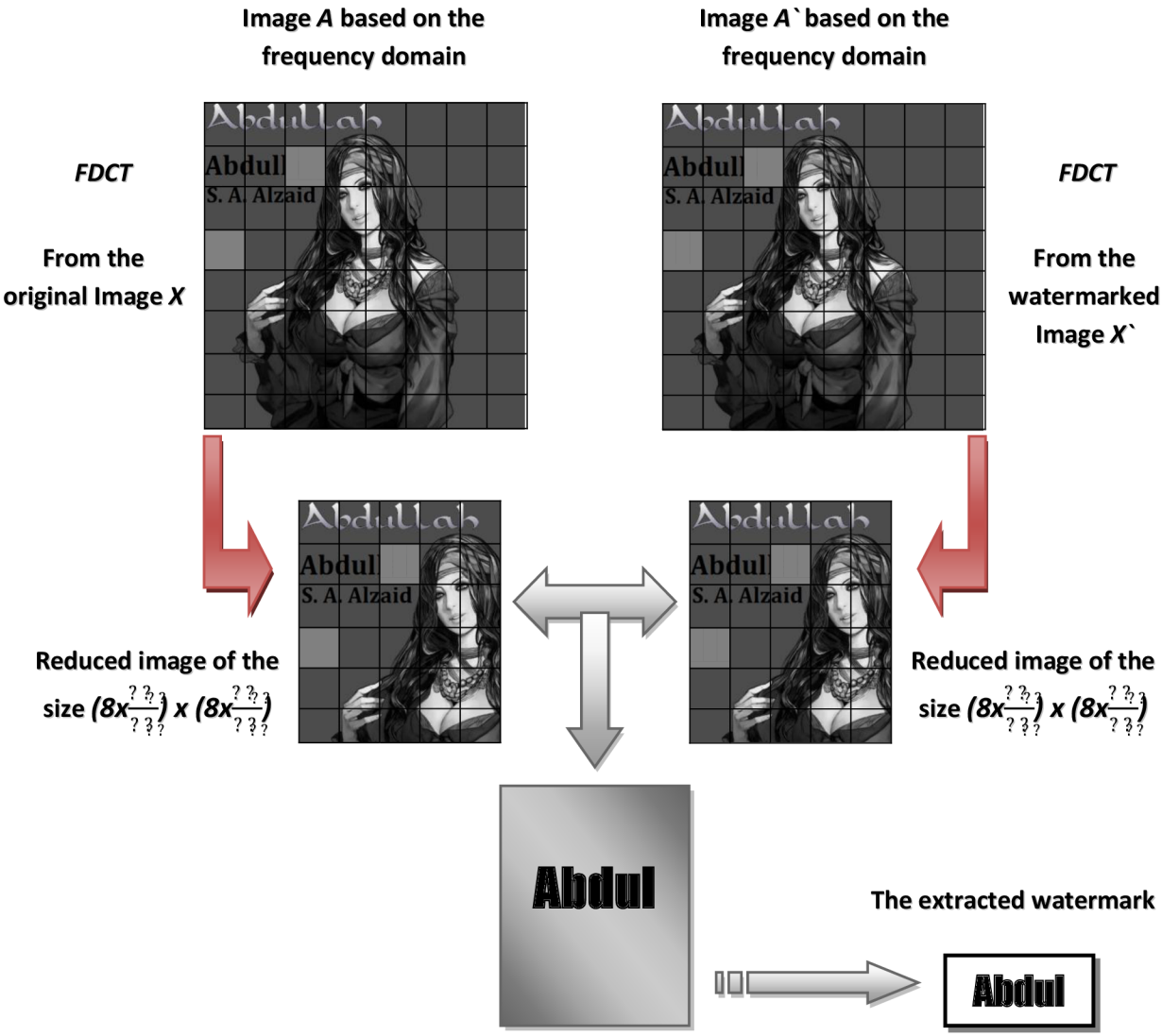


Figure 16. The final procedure for recovering the watermark from the digital watermarked image

5.3.3 An Additional Experiment Based on RGB image

This experimentation is to be considered as an additional research, not being an objective neither a part of the materials required to be covered throughout this project; however, we can go briefly through some aspects of how it would be implemented.

First of all, in case of an RGB image the image needs to be divided into 8x8 RGB blocks. And then right after, those blocks must be converted into components of YC_bC_r the color space and image compression, thus it is then possible to deal with the luminance values. After this procedure is over with, it will be the time when the DCT function should be performed and convert the YCC components we obtained from the RGB blocks into frequency components. As a result, two frequency coefficients will be selected based on the quantization table that is related to the JPEG lossy compression shown in figure 17 below. Those chosen coefficients are used in order to encode the digital watermark through the embedding process.

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

Figure 17. Quantization values used in JPEG compression scheme

Based on the quantization table, we can observe that coefficients (1,2) and (3,0) or (4,1) as well as (3,2) could possibly make such appropriate applicants thanks to the JPEG lossy comparison, as long as their quantization values are being identical.

The DCT block will encode a “1” if $a(a_1,a_2) > b(b_1,b_2)$ or simply put we can form it as $(a > b)$; in other respects it will encode a “0”. The coefficients will be then swapped in

case of being the relative size of each coefficient not having the future of being agreed with the bit which is desired to be encoded.

The watermark is encoded using the following up algorithm that will be shown within the next subsection, where "a" is determined as the first coefficient and "b" is supposed to be the second coefficient. The relationship between the two coefficients is altered so that they accomplish a certain necessity relying on the watermark bit itself [7].

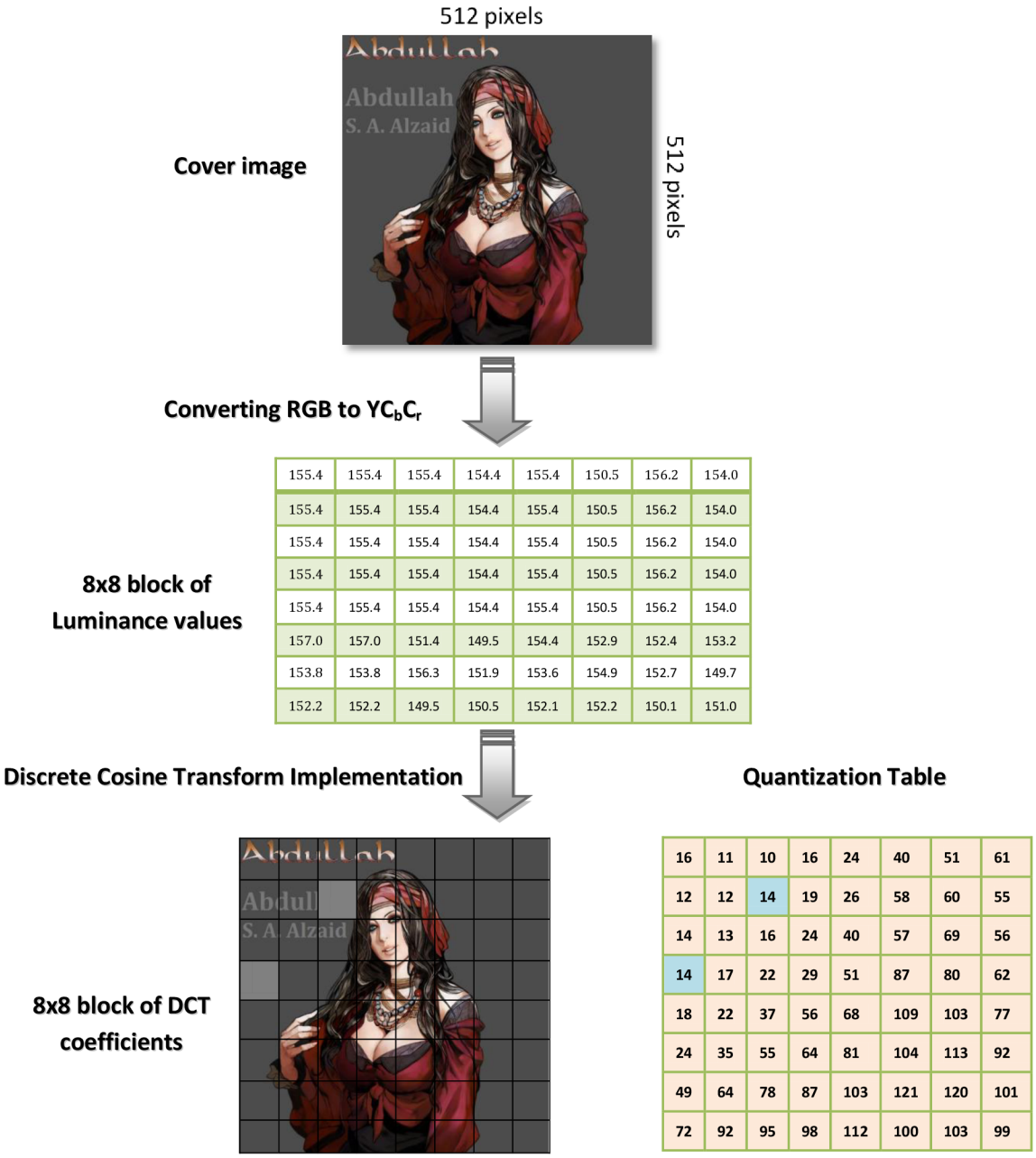


Figure 18. The digital watermark is being inserted using two DCT coefficients. It shows a detailed description of how the watermark is inserted

5.4 The Two-mid Coefficients Algorithm

5.4.1 Based on the Embedding method

The following code gives us a vital clue for comprehending how the embedding algorithm should function out properly in order to obtain the embedding system for in the invisible watermarking approach.

```
if (watermark bit is a 1)
{
        if (b < a)
        {
                swap (a, b)
        }
}
else watermark bit must be a 0
{
        if (b > a)
        {
                swap (a, b)
        }
}
both values are being adjusted
so that |a - b| > x
where x is a constant
```

(5)

5.4.2 Based on the Extracting method

The watermark is recovered by means of using the succeeding algorithm

```
if (b <= a)
{
        watermark bit is a 0
}
else
{
        watermark bit is a 1
}
```

(6)

For detection, the image is broken up into those same 8 x 8 blocks, and a DCT function is performed. The same PN sequence is then compared to the middle frequency values of the transformed block. If the correlation between the sequences exceeds some threshold T, a “1” is detected for that block; otherwise a “0” is detected. Again “k” denotes the strength of the watermarking, where increasing “k” increases the robustness of the watermark at the expense of quality.

Chapter 6: Discrete Wavelet Transform (DWT)

6.1 General Concepts of the Discrete Wavelet Transform

The DWT transform, Wavelets are special functions which, in a form analogous to sines and cosines in Fourier analysis, are used as basal functions for representing signals. For 2-D images, applying DWT corresponds to processing the image by 2-D filters in each dimension. The filters divide the input image into four non-overlapping multi-resolution sub-bands (LL_1), (LH_1), (HL_1) and (HH_1). The sub-band (LL_1) represents the coarse-scale DWT coefficients while the sub-bands (LH_1), (HL_1) and (HH_1) represent the fine scale of DWT coefficients. To obtain the next coarser scale of wavelet coefficients, the sub-band (LL_1) is further processed until some final scale "$_N$" is reached. When "$_N$" is reached we will have 3N+1 sub-bands consisting of the multi-resolution sub-bands (LL_N) and (LH_X), (HLx) and (HHx) where "$_X$" ranges from 1 until "$_N$". Due to its excellent spatio-frequency localization properties, the FDWT is very suitable to identify the areas in the host image where a watermark can be embedded effectively. In particular, this property allows the exploitation of the masking effect of the human visual system such that if a DWT coefficient is modified, only the region corresponding to that coefficient will be modified. In general most of the image energy is concentrated at the lower frequency sub-bands (LL_X) and therefore embedding watermarks in these sub bands may degrade the image significantly. Embedding in the low frequency sub-bands, however, could increase robustness significantly. On the other hand, the high frequency sub-bands (HH_X) include the edges and textures of the image and the human eye is not generally sensitive to changes in such sub-bands. This allows the watermark to be embedded without being perceived by the human eye. The compromise adopted by many DWT-based watermarking algorithm, is to embed the watermark in the middle frequency sub-bands (LH_X) and (HL_X) where acceptable performance of imperceptibility and robustness could be achieved [8].

6.2 The Implementation of the DWT

To a better way for understanding how the DWT transform function, first of all, we need to start off with its functionality with respect to a signal of one dimension then we will be expressing the way it deals with a two dimensional signal which is the part of our objective.

6.2.1 One-dimensional DWT

In the case of one dimensional signal, the signal itself is to be divided into two groups of frequency components as low frequency components and high frequency components which are mainly determined as the 1st pass of the low-pass and high-pass frequencies, see figure 19 below.

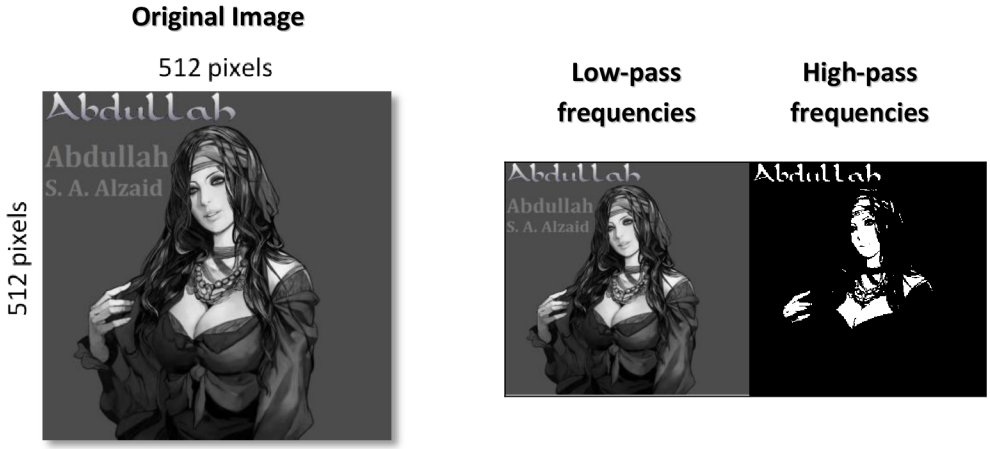


Figure 19. The 1st pass of DWT transform dividing the 1-dimensional image into two frequency groups, low-pass frequencies and high-pass frequencies

While the high-band frequency group would remain unchanged, the low-band frequency group will be then divided up into two other inner groups of frequencies causing the 2nd pass of the low-pass and high-pass frequencies. The same process is to be continued in such an arbitrary number of times making the next passes by dividing the low-band frequency blocks [9].

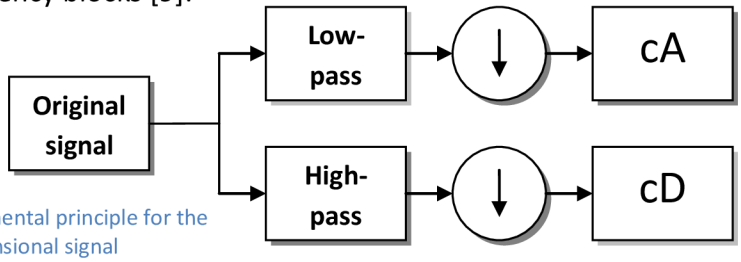


Figure 20. The fundamental principle for the 1st pass of a one-dimensional signal

The decomposition process can be simply constructed by applying the previous technique into the one-dimensional signal *X(N)* as shown next.

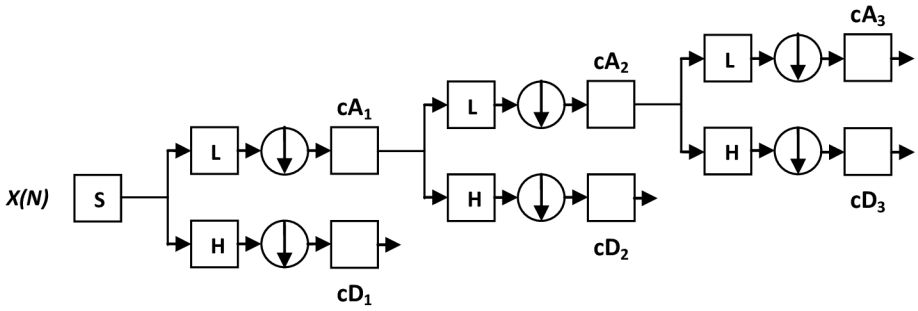


Figure 20. The decomposition process for a one-dimensional DWT being constructed by cD1, cD2, cD3, and cA3

In the figure above, we can analyze how the DWT of a one-dimensional signal *X(N)* can be constructed from cD_1, cD_2, cD_3, and cA_3 (frequency coefficients); this leads to result out three levels of decompositions which are enough to operate an 8^{th} length signal.

However, the original signal *X(N)* can be also reconstructed by using the same frequency coefficients which have been used through the decomposition process of the DWT. The reconstruction process is determined as the inverse DWT (IDWT) and can be demonstrated as it is appeared within the following figure 21.

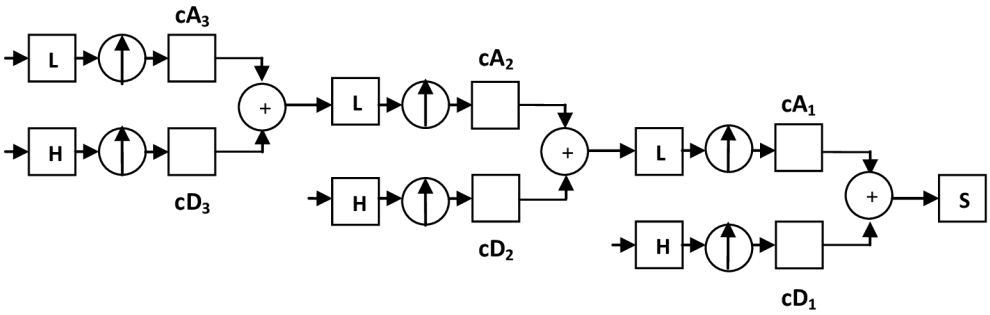


Figure 21. The reconstruction process for a one-dimensional (IDWT) *X`(N)* being constructed by cD1, cD2, cD3, and cA3

6.2.2 Two-dimensional DWT

What we have expressed in words above was the DWT as well as the IDWT for such one-dimensional signal *X(N)*, however, in the principle of experimenting both of the DWT and IDWT for two-dimensional signal; it would be most likely the same as multiplying a one-dimensional DWT by two. To be more exact, let's suppose our two-dimensional signal is specified as $X(N_1,N_2)$, for rows we have N_1 and for columns we have N_2; thus, the one-dimensional DWT will be then applied for N_1 and at the same time for N_2. The next figure 22 describes how in fact this process is being handled with respect to a two-dimensional signal as $DWT_{N1}[DWT_{N2}[X[N_1,N_2]]]$ (decomposition process) [10].

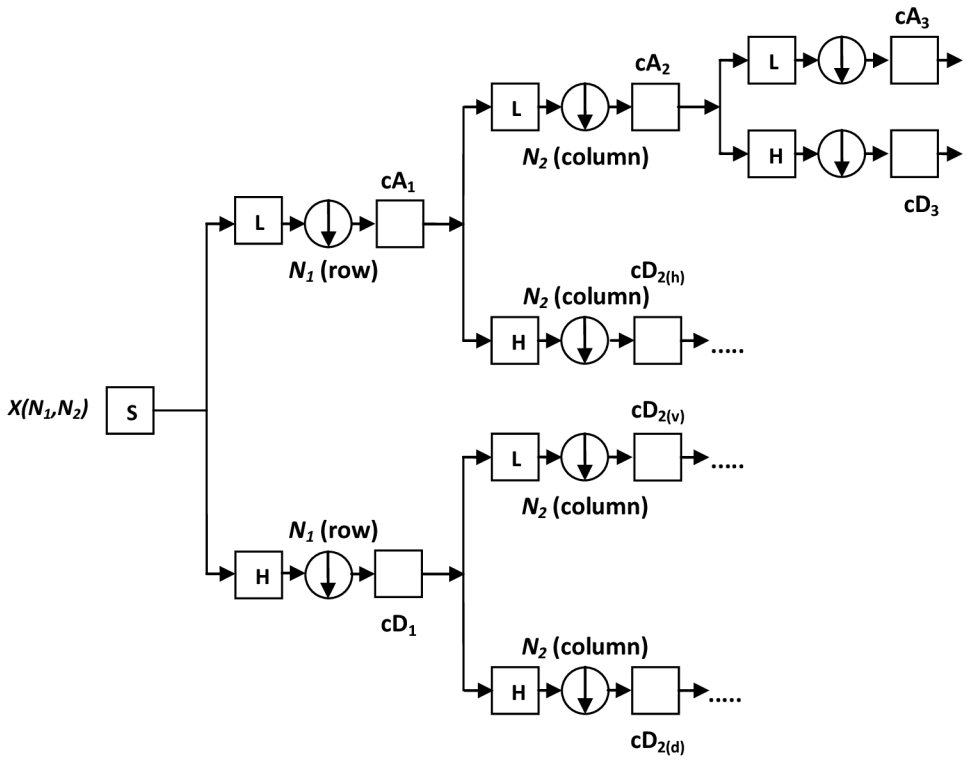


Figure 22. The decomposition process for a tow-dimensional DWT

With regard to a still image that consists of a two-dimensional signal, it's to be decomposed into DWT pyramid structure with various frequency bands such that low-low frequency band, also low-high frequency band, and high-low frequency band as well as high-high frequency components as exhibited in figure 23.

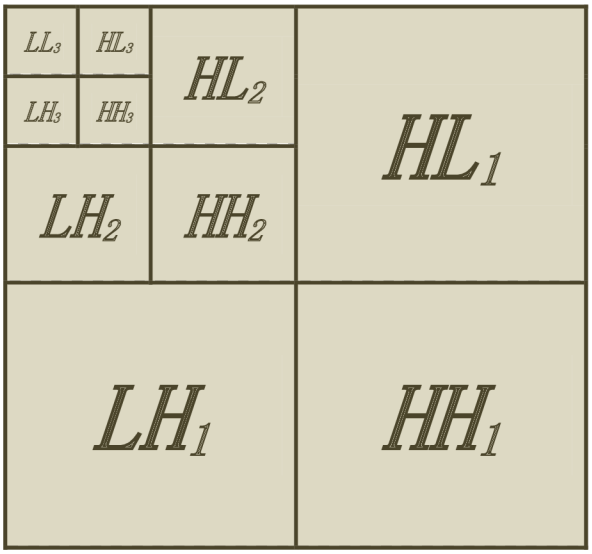


Figure 23. An image is split into various frequency bands with three passes during the DWT pyramid decomposition

Thereafter, the DWT function based on a two-dimensional signal can then divide the signal itself or let's say the image into a lower resolution approximation image (*LL*), also a vertical resolution approximation image (*LH*), plus a horizontal resolution (*HL*) and as well a diagonal (*HH*) detail frequency components. To demonstrate an illustration for an image going through the 2nd pass stage of the pyramid decomposition where the edges can appear in all the frequency bands except the band where it has the low-low frequency coefficients, see the afterwards figure.

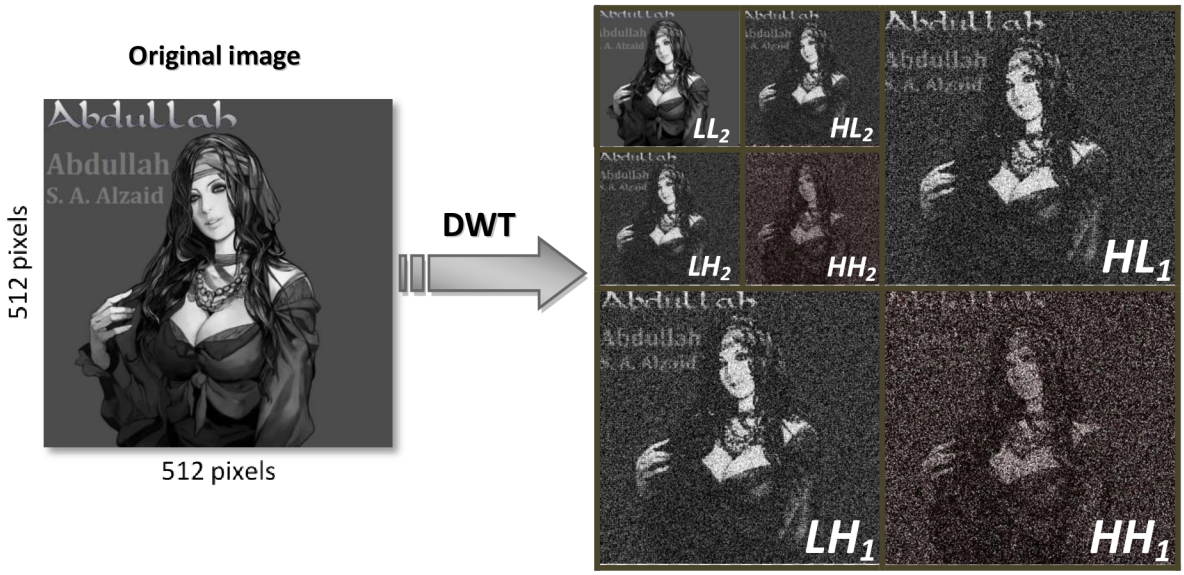


Figure 24. The DWT-transform pyramid decomposition of an image through the 2nd pass

6.3 Encoding and Decoding Procedures

The function of the DWT transform domain is based on the watermarking technology is composed of two major parts which are the encoding and decoding procedures. Those two procedures are plainly the key for implementing the embedding as well as the extracting system by way of the DWT transform.

6.3.1 DWT Encoding Procedure

Step 1: In the field of this process, by means of the DWT pyramid structure we are to decompose the image into various frequency bands as what we have done in figure 24.
Step 2: Then right after, we apply Gaussian Noise to the bands which include the high frequency components. Those large frequency coefficients are not located at the low-low frequency band, let us denote them as $M(m_1,m_2)$ and the original signal that's included in the lowest frequency band to be labeled as $X(N_1,N_2)$.

Let's assign Gaussian Noise sequence as $G(m_1,m_2)$, and then we add it to $M(m_1,m_2)$ by means of 1 and 0 variance as follows,

$$G(m_1, m_2)\, \alpha N^2(m_1, m_2) + M(m_1, m_2) = M`(m_1, m_2) \qquad (7)$$

Where α is determined as the parameter which controls the decomposition levels (passes).

The square $({}^2)$ signifies the amplification of the big frequency coefficients $M(m_1,m_2)$.

Step 3: The DWT of the lowest frequency bands $X(N_1,N_2)$ will not be moderated. The DWT of the large frequency bands $M(m_1,m_2)$ will be modified due to the process of applying the Gaussian Noise $G(m_1,m_2)$; as a result, we will obtain the modified DWT $M`(m_1,m_2)$. Then, the two-dimensional IDWT of $M`(m_1,m_2)$ must be taken beside to the not moderated DWT coefficients $X(N_1,N_2)$ and apply them together for the purpose of attaining $X`(N_1,N_2)$ which is referred to the IDWT coefficients.

However, this resultant image $X`(N_1,N_2)$ needs to have the same dynamic range that the cover image $X(N_1,N_2)$ has. As in consequent and in order to obtain that range, the resultant image must be calculated as follows,

$$X`(N_1,N_2)\ min(max\{X(N_1,N_2), max\{X`(N_1,N_2), min\{X(N_1,N_2)\}\}) \quad (8)$$

This resultant image $X`(N_1,N_2)$ is determined as the watermarked image, and the whole encoding process can be demonstrated as in the following figure 25.

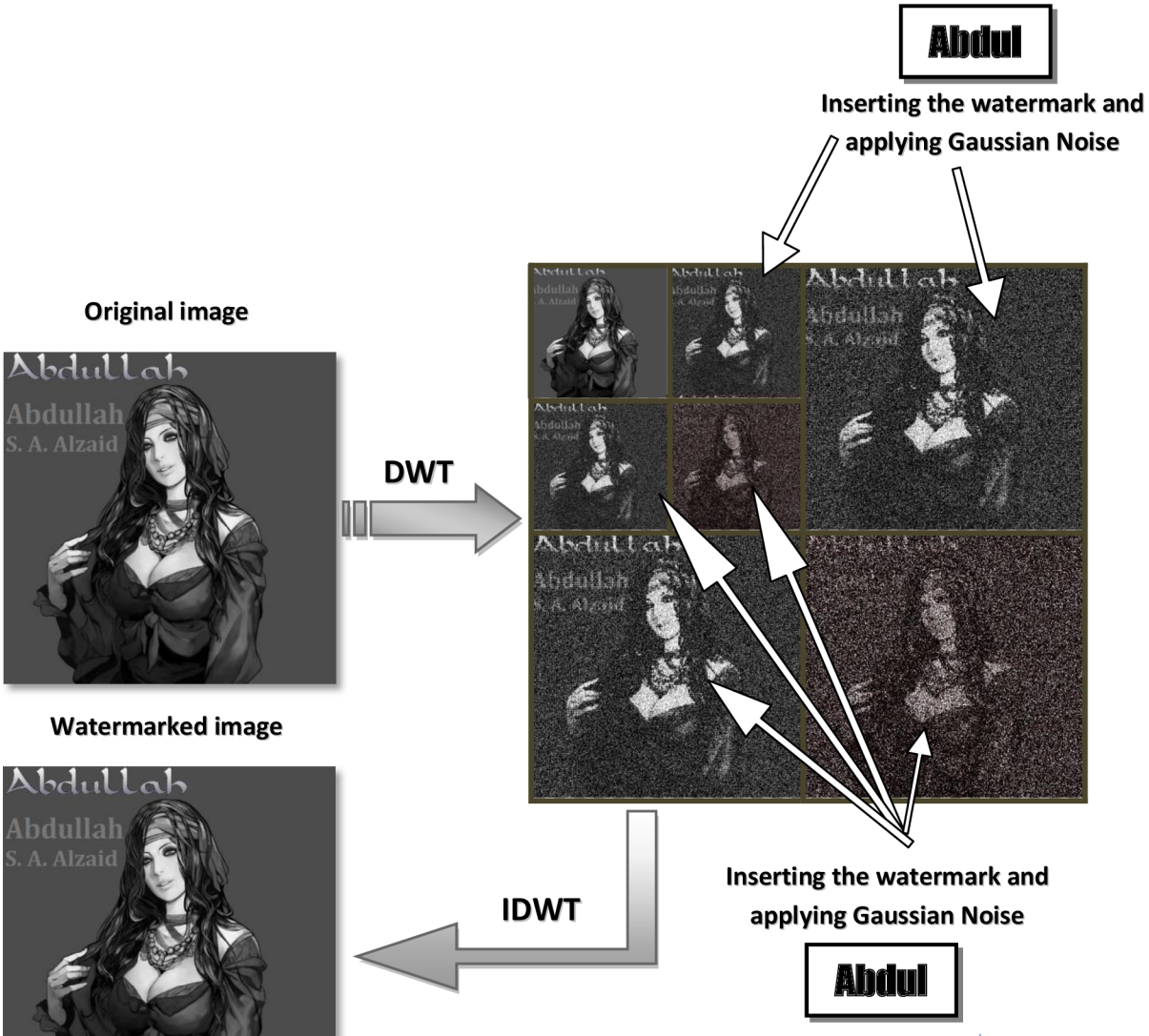


Figure 25. The encoding process of a two-dimensional DWT in the 2nd pass

6.3.2 DWT Decoding Procedure

Step 1: We first of them all need to have both of the cover image and the watermarked image (reference image) readily available and then decompose them both into four frequency bands, and let's say the decomposition is performed through the 1st pass as (LL_1), (LH_1), (HL_1) and (HH_1).

Figure 26. The 1st pass of the decomposition process on the DWT domain

Step 2: After decomposing the two images respectively and since we are dealing with the 1st pass of the IDWT initially, we will examine the signature which we inserted in the (HH_1) of the cover image and compare it to the difference between the current frequency band (HH_1) of the original image and the difference of the same band (HH_1) in the watermarked image (received image).

This process can be done by calculating the cross correlation of the (HH_1) from the original image and the cross correlation of the (HH_1) in the received image. The figure comes next will express the technique of detecting the watermark signature from the band (HH_1) of 1st pass by calculating the cross correlation as how it was explained above.

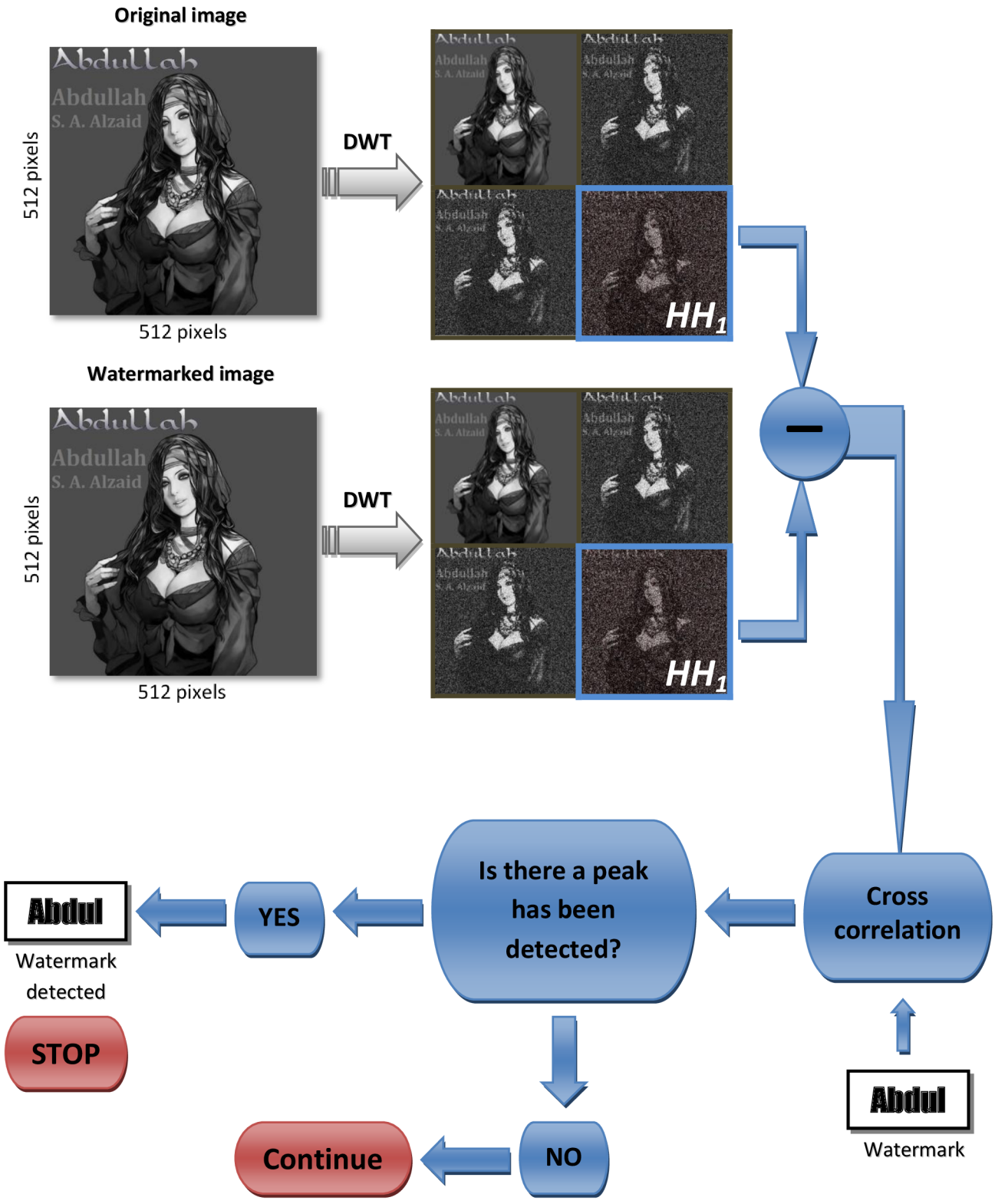


Figure 27. The scheme proposed for detecting the watermark signature in DWT domain in the 1st pass

Step 3: In case the watermark signature has been detected in the frequency band (HH_1), then the watermark will be extracted and the procedure will stop right after the detection; otherwise it will continue on checking out the other bands for the inserted watermark signature from the decomposed cover image and the decomposed watermarked image and comparing their differences with regard to their cross correlation. For instance, comparing the bands (HH_1) and (HL_1) in the decomposed cover image with the distinction of the DWT coefficients of the bands (HH_1) and (HL_1) in the decomposed watermarked image and examining their cross correlation if there is a peak being detected which means the watermark has been detected and extracted; or else, it would keep doing the same process with the other bands until it detects the watermark.

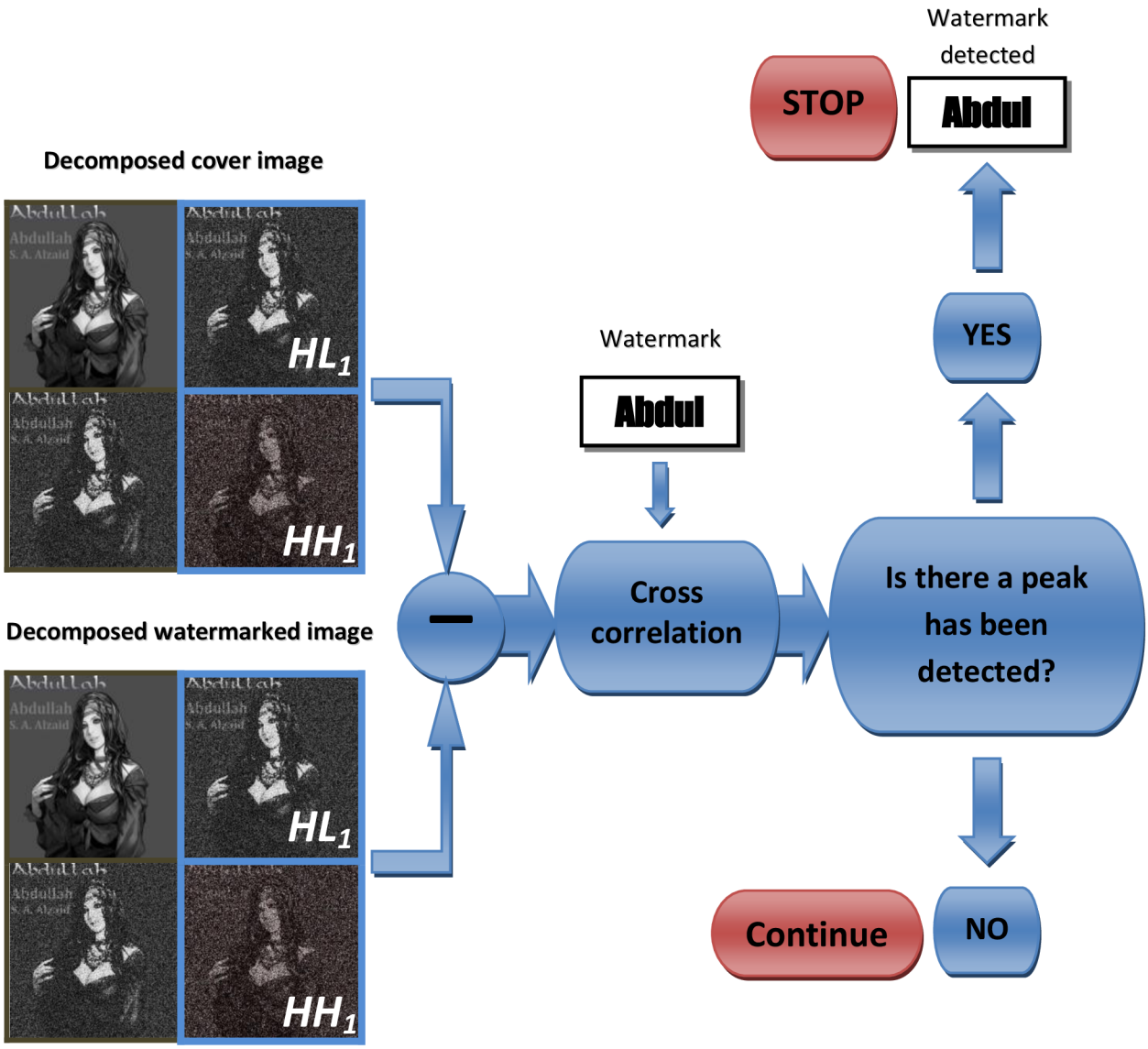


Figure 28. Examining the bands (HH_1) and (HL_1) in the decomposed cover image and comparing them to the difference of the DWT coefficients in the decomposed watermarked image to detect if their cross correlation has a peak or not.

Step 4: On the assumption that the watermark signature has not been yet detected within the frequency bands in the 1st pass of the decomposition for the decomposed cover image and watermarked, we will then go through the 2nd pass of the DWT decomposition and examine its frequency bands (HH_2), (HL_2) and (LH_2). The DWT detection procedure will keep investigating all the frequency bands and levels (passes) until the peak will be finally detected through the cross correlation, thus, the watermark is completely extracted then the detecting operation will stop.

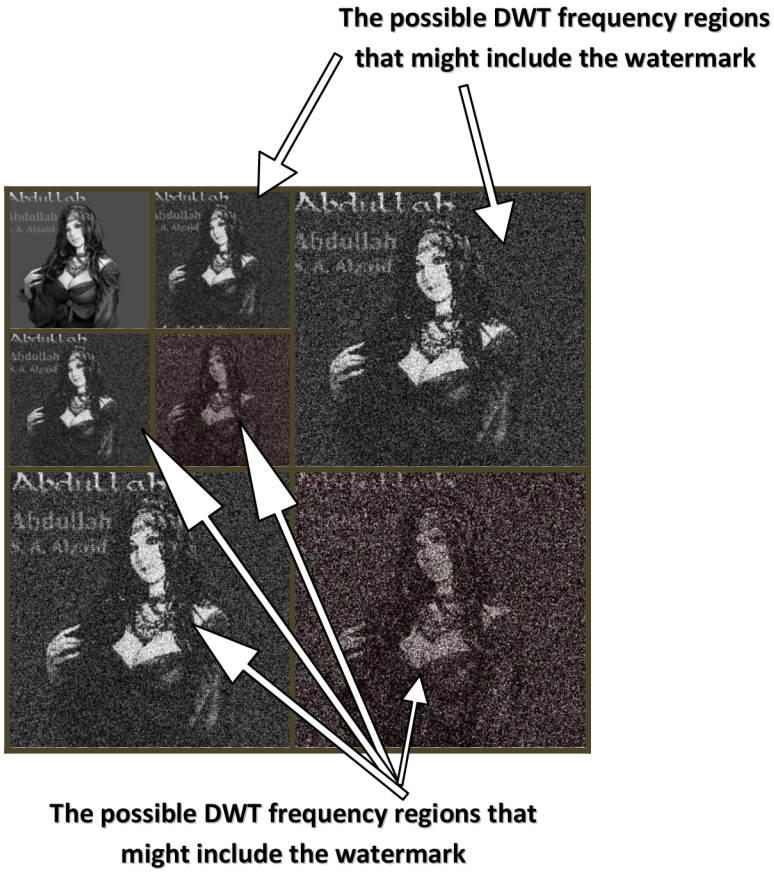


Figure 29. Possible DWT frequency bands which could include the watermark signature

6.4 Another scheme for the embedding and extraction techniques of the DWT

This further procedure was performed during the execution of the DWT experiment for the intention of applying the embedding and retrieving systems by means of using the Matlab software. We are to describe how this method was completed appropriately in such a detailed manner as in the following sub-sections [11].

6.4.1 The 2nd embedding technique for DWT

Step 1: As to start off, we first need to decompose the original image into four equal wavelets through what is called the 1st level (we will be dealing only with the 1st pass based on this particular procedure). Each wavelet is to be measured by 256 x 256 pixels.

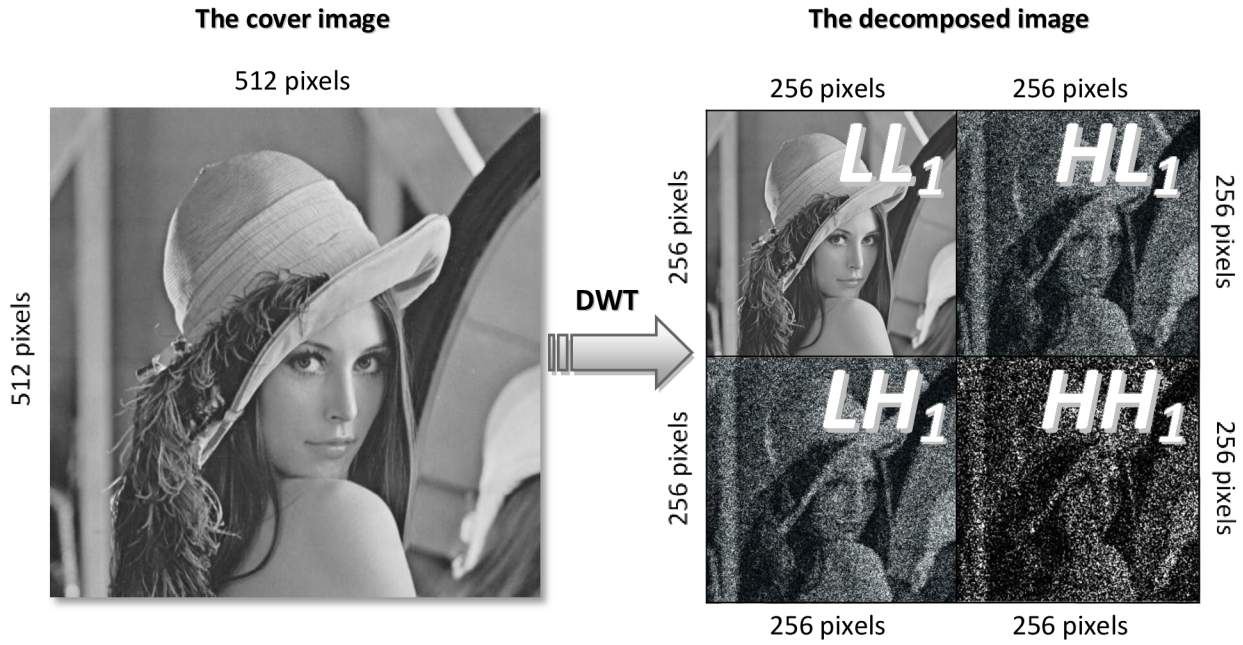


Figure 30. Decomposing the original image into equal wavelets throughout the 1st pass

Step 2: After decomposing the cover image into four equal wavelets through the 1st pass, we are then to select a frequency band of the wavelets where it is possible to interleave the watermark signature via the encoding and embedding scheme. Let the

chosen frequency band be the wavelet which is denoted as HH_1. Subsequent to the previous stage, we should bring up the watermark which we seek to hide in the course of the original image and divide it into the same size we applied for each wavelet from the decomposed image; therefore, the desired watermark signature will be measured up by 256 x 256 pixels.

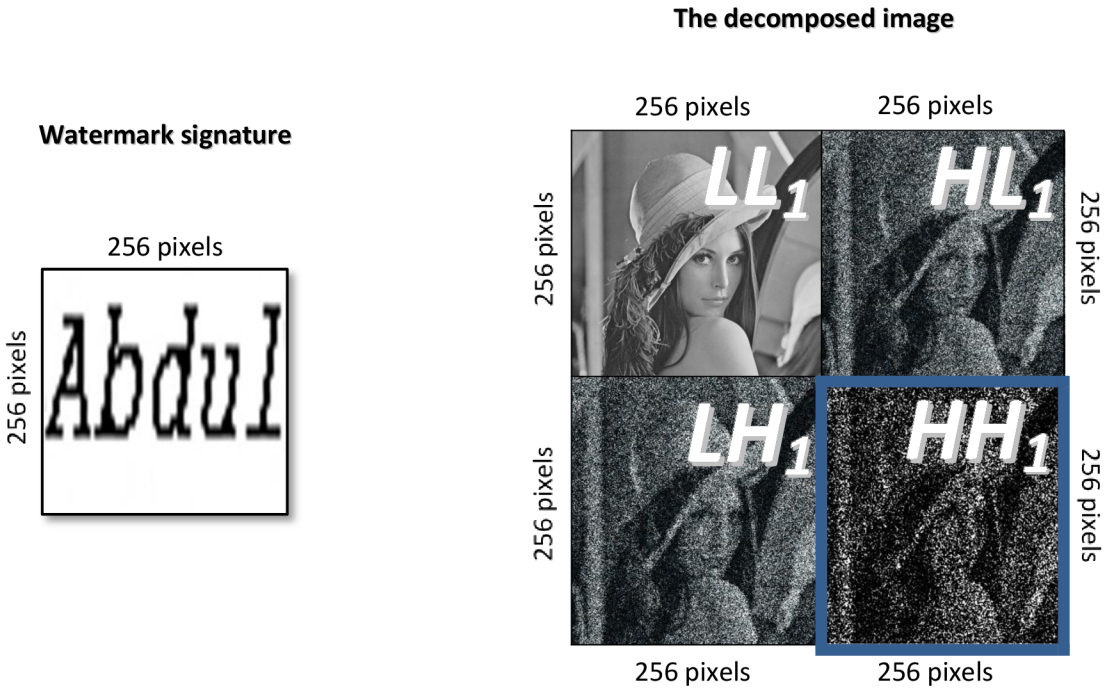


Figure 31. Matching the size of the desired watermark with respect to the chosen wavelet in the decomposed image

Step 3: At this instant, every single pixel from the desired watermark is calculated independently and thereafter it will be multiplied within the robustness factor that is designated as "*K*". Forwardly, the dedicated pixel from the watermark, e.g. (1,1) will be then compared with its matching pixel (1,1) from the chosen wavelet (HH_1); this stage is performed with the intention of duplication to be done in each pixels from the watermark as to be multiplied with the robustness factor as we described previously and then summing the resultant calculation with the rest of the opposite pixels in the selected wavelet so that the same performance is to be applied for the rest of the pixels by means of the following equation (9). Afterwards, the watermarked image will be

completely obtained so that the forward discrete wavelet transform FDWT will be applied with the intention of acquiring the watermarked image in due course.

$$watermark(1,1) * K + HH_1(1,1) \quad (9)$$

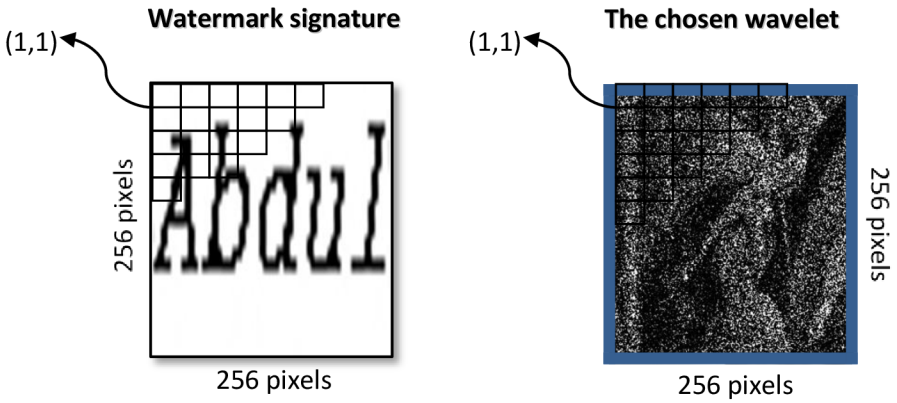


Figure 32. Applying the embedding stage by calculating equation (9) and obtaining the resultant watermarked image

6.4.2 The 2nd retrieving technique for DWT

Step 1: the resultant watermarked image from the previous scheme and the original image are both to be decomposed into four equal frequency coefficients for each of them. Ever since we have been the decomposition technique through the 1st level in the previous procedure, it means that we also need to decompose the watermarked image and the cover image into the same 1st pass as well.

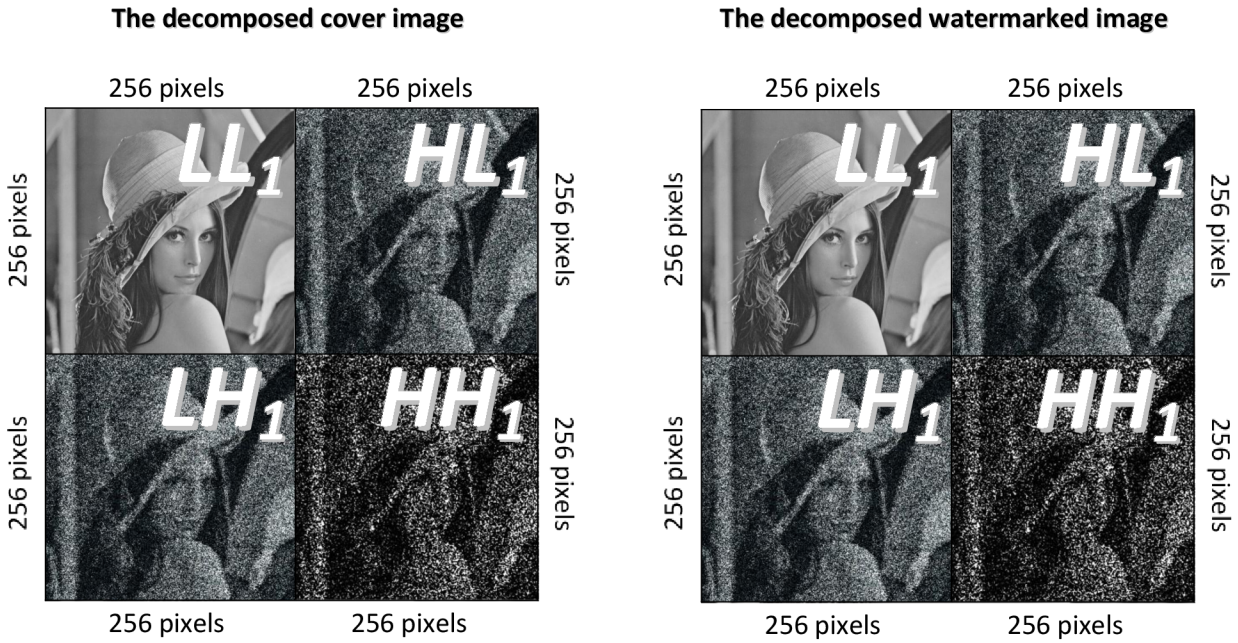


Figure 33. Decomposing the watermarked image as well as the original image into four wavelets through the 1st pass

Step 2: Subsequent to implementing the FDWT transform, the obtained wavelets from the decomposed images should be divided equivalently and deliberated by 256 x 256 pixels for each band independently. As a result, the same band where the watermark signature was embedded into will be then selected from both images; e.g. (HH_1) from the decomposed watermarked image and (HH_1) from the decomposed cover image.

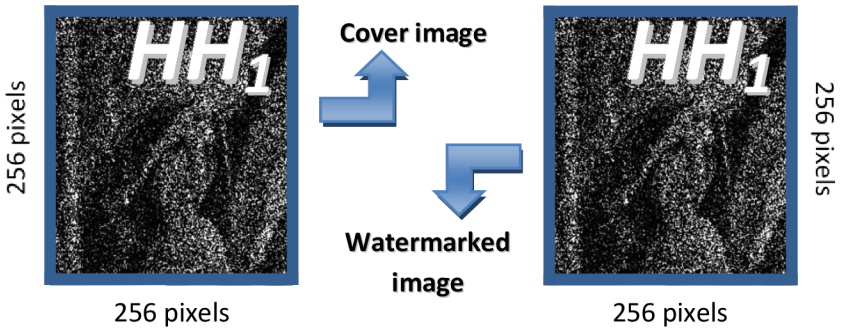


Figure 34. The selected bands from the two decomposed images

Step 3: Every single pixel from the dedicated frequency bands is to be computed by calculating the difference of the cross correlation when comparing the pixels from (HH_1) wavelet in the decomposed watermarked image and (HH_1) wavelet in the decomposed original image. This approach can be achieved by performing the following equation (10).

$$watermarked\ HH_1(1,1) - \ original\ HH_1\ (1,1)/\ K \tag{10}$$

For instance, the 1st pixel (1,1) from the selected band (HH_1) in the decomposed cover image will be compared with its matching pixel (1,1) from the opposite chosen wavelet (HH_1) in the other image (decomposed watermarked image). Consequently, the cross correlation between them will be defined by subtracting the two pixels, thus, the obtained value is to be divided up by the robustness factor "*K*"

Step 4: the entire procedure must be executed throughout the entire remaining pixels from those two selected frequency coefficients' bands, and once the process is performed appropriately the anticipated watermark signature can be then decoded and retrieved by completely the extracting and decoding stage.

Chapter 7: Results and Analysis of the Thesis

7.1 Gathering Results

The examining of the two required methods (DCT/DWT) have been performed by means of taking 512 x 512 grayscale image (cover image) and inserting an 8-bit length watermark signature into it. The function *PSNR* (peak signal to noise ratio) was computed and applied to the obtained watermarked images from both methods as well as applying it for the cover images. *PSNR* is a commonly used pixel-based visual distortion metric; it has been coded for the intention of computing and measuring the distinctive distortion between the cover image and the watermarked image. When the calculated results of the *PSNR* are rated in a low rank, which means the image has been distorted properly.

$$PSNR(dB) = 10 \,.\, Log_{10}(\frac{MAX_i^2}{MSE}) \qquad (11)$$

$$PSNR(dB) = 20 \,.\, Log_{10}(\frac{MAX_i}{\sqrt{MSE}})$$

Where, *PSNR* is peak signal to noise ratio.

MAX_i is the maximum possible pixel value of the image.

MSE is the mean squared error.

The peak signal to noise ratio is usually measured in decibels and converted using the above equation.

However, the percentage accuracy of the extracted watermark can be calculated by finding the frequency of the actual watermark signature and dividing it up by the total of all watermarks which have been recovered.

$$Accuracy = \frac{W_A}{W_T} x100 \qquad (12)$$

Where W_A is the frequency rate in which the actual watermark signature was found.

W_T is the total number of the discovered watermarks.

7.2 Results and analysis of the 1st technique (DCT)

With regard to this technique, we have computed the embedding code via Matlab software, and we calculated the *PSNR* value in order to measure the distortion between the watermarked image and the original image. Then right after, we applied the Checkmark software by setting up the (getconfig.m) file as a previous step of the upcoming procedure which was done by running the file (runcheckmark.m) so that we could be capable of testing the attacks based on the resultant watermarked image in the DCT domain. Consequently, we obtained 71 files of attacks were located into a separate folder. At long last, we applied the *NCC* function on the extraction code which we computed in contemplation of performing the DCT extracting method; forasmuch as to test the original watermark and the extracted watermark. A table of analysis was achieved by setting the type of the copyrights (attacks) with respect to the DCT in such three different levels or robustness as in the following table.

7.3 Table A, results of the 1st approach (DCT)

Copyright	(5,2) (4,3)	Bi_1	(8,8) (4,3)	Bi_2	(7,3) (5,1)	Bi_3	(1,8) (6,2)	Bi_4
Basic Collage Attack	0.4941	0	0.5074	0	0.5279	0	0.5060	0
Collage with rotated, scaled and cropped watermarked image	0.4875	0	0.5100	0	0.5292	0	0.4974	0
Watermark added to new image	0.9742	1	0.9039	1	0.5173	0	0.5213	0
Percent crop=10	0.4450	0	0.4476	0	0.5014	0	0.5067	0
Percent cropped=20	0.5074	0	0.5120	0	0.5034	0	0.5120	0
Percent cropped=50	0.5120	0	0.5027	0	0.4861	0	0.4914	0
Percent cropped=75	0.4324	0	0.4456	0	0.4622	0	0.4722	0
Dithered image	0.8389	1	0.6181	0	0.4981	0	0.4576	0
DPR attack 3x3 window	0.3276	0	0.6194	0	0.5054	0	0.4828	0
DPR attack 5x5 window	0.2414	0	0.5365	0	0.4994	0	0.4768	0
dprcorr attack 3x3 window	0.4324	0	0.6718	0	0.5272	0	0.5684	0
dprcorr attack 5x5 window	0.3429	0	0.5644	0	0.5418	0	0.5253	0
Gaussian filtered 3x3	1	1	0.9974	1	0.4775	0	0.4914	0
Gaussian filtered 5x5	1	1	0.9974	1	0.4729	0	0.4569	0
hardthresh 3x3 window	1	1	0.9921	1	0.5179	0	0.4623	0
hardthresh 5x5 window	0.9994	1	0.9862	1	0.4828	0	0.8303	1
JPEG equal=10	0.9265	1	0.9066	1	0.9265	1	0.4835	0
JPEG equal=15	1	1	0.9981	1	0.4921	0	0.7905	1
JPEG equal=25	0.9159	1	0.8821	1	0.8350	1	0.7090	1
JPEG equal=30	1	1	0.9749	1	0.7773	1	0.5299	0
JPEG equal=40	1	1	0.9822	1	0.5750	0	0.4987	0
JPEG equal=50	1	1	0.9941	1	0.7103	1	0.4463	0
JPEG equal=60	1	1	0.9941	1	0.6460	0	0.4961	0
JPEG equal=75	1	1	0.9968	1	0.5332	0	0.4881	0
JPEG equal=80	1	1	0.9981	1	0.5458	0	0.5054	0
JPEG equal=85	1	1	0.9961	1	0.5127	0	0.4967	0
JPEG equal=90	1	1	0.9988	1	0.5020	0	0.4994	0
JPEG equal=100	1	1	0.9981	1	0.5080	0	0.5186	0
T11=1.15; T12=-0.02; T21=-0.03; T22=0.90	0.5073	0	0.5034	0	0.5126	0	0.5604	0
T11=0.8; T12=-0.1; T21=0.05; T22=1.1	0.5485	0	0.5591	0	0.5677	0	0.5776	0
T11=-0.85; T12=-0.2; T21=-0.05; T22=1.3	0.5936	0	0.5962	0	0.5909	0	0.0604	0
median 2x2 window	0.9968	1	0.9736	1	0.1758	0	0.5796	0
median 3x3 window	0.9954	1	0.8754	1	0.4788	0	0.4875	0
median 4x4 window	0.8655	1	0.9901	1	0.6506	0	0.5067	0
xscale=0.8 yscale=1	0.9988	1	0.9391	1	0.4709	0	0.9749	1
xscale=1 yscale=0.8	0.9245	1	0.9477	1	0.6884	0	0.9762	1
xscale=1 yscale=0.9	0.9583	1	0.9954	1	0.8005	1	0.3018	0
xscale=1 yscale=1.1	1	1	0.9974	1	0.3787	0	0.4490	0
xscale=1 yscale=1.2	1	1	0.9928	1	0.4662	0	0.7468	1

Copyright	(5,2) (4,3)	Bi_1	(8,8) (4,3)	Bi_2	(7,3) (5,1)	Bi_3	(1,8) (6,2)	Bi_4
rowcol1: 1 row and column removed	0.9994	1	0.9842	1	0.7295	1	0.8244	1
rowcol2: 1 row and 5 columns removed	0.9941	1	0.9689	1	0.7050	1	0.8476	1
rowcol3: 17 rows and 5 columns removed	0.9848	1	0.9948	1	0.6115	0	0.7428	1
rowcol4: 5 rows and 1column removed	0.9994	1	0.9842	1	0.6771	0	0.7826	1
rowcol5: 5 rows and 17 columns removed	0.9968	1	0.9974	1	0.5040	0	0.5180	0
Downsample factor 0.75, Upsample factor 1.3	1	1	0.9921	1	0.4828	0	0.4888	0
Downsample factor 0.5, Upsample factor 1.9	0.9935	1	0.9875	1	0.3727	0	0.2686	0
Downsample factor 0.75, Upsample factor 1.333	0.9988	1	0.9802	1	0.5259	0	0.7089	1
Downsample factor 0.5, Upsample factor 2	0.9842	1	0.9895	1	0.4927	0	0.5478	0
Scaling 0.5	0.9862	1	0.9961	1	0.5100	0	0.5080	0
Scaling 0.75	1	1	0.9941	1	0.4689	0	0.4132	0
Scaling 0.9	1	1	0.9915	1	0.3767	0	0.2646	0
Scaling 1.1	0.9988	1	0.9974	1	0.4934	0	0.4914	0
Scaling 1.5	1	1	0.9968	1	0.4967	0	0.4861	0
Scaling 2.0	1	1	0.9988	1	0.5213	0	0.4788	0
sharpened	1	1	0.9901	1	0.4715	0	0.4609	0
softthresh 3x3 window	1	1	0.9862	1	0.4748	0	0.4576	0
softthresh 5x5 window	0.9988	1	0.5179	0	0.4841	0	0.5146	0
Image after template removal attack	0.3614	0	0.9570	1	0.8151	1	0.7965	1
Thresholded image	0.9855	1	0.6506	0	0.8708	1	0.2673	0
wavelet compression 8.0 bpp (uncompressed)	0.6314	0	0.9988	1	0.4729	0	0.5723	0
wavelet compression 3.5bpp	1	1	0.6778	0	0.7846	1	0.3429	0
wavelet compression 1.5bpp	0.6910	0	0.7136	1	0.7905	1	0.4052	0
wavelet compression 0.8 bpp	0.7540	1	0.9981	1	0.4536	0	0.4596	0
wavelet compression 0.6 bpp	1	1	0.8263	1	0.7023	1	0.6930	0
wavelet compression 0.5 bpp	0.9530	1	0.9404	1	0.6048	0	0.6725	0
wavelet compression 0.4 bpp	0.9881	1	0.9915	1	0.6387	0	0.8641	1
wavelet compression 0.3 bpp (uncompressed)	1	1	0.9961	1	0.6685	0	0.8389	1
wavelet compression 0.2 bpp (uncompressed)	1	1	0.9981	1	0.4921	0	0.4815	0
wavelet compression 0.1 bpp (uncompressed)	1	1	0.9908	1	0.4841	0	0.5160	0
wiener 3x3 window	1	1	0.6632	0	0.4702	0	0.4596	0
wiener 5x5 window	0.3449	0	-	-	-	-	-	-

When the *NCC* value is set as *NCC* > 0.7 (70%) that means detected as a 1, and when the *NCC* value is set to *NCC* < 0.7 (70%) then it will be detected as a 0.

7.4 Results and analysis of the 2nd technique (DWT)

With regard to this technique, we have computed the embedding code via Matlab software, and we calculated the *PSNR* value in order to measure the distortion between the watermarked image and the original image. Then right after, we applied the Checkmark software by setting up the (getconfig.m) file as a previous step of the upcoming procedure which was done by running the file (runcheckmark.m) so that we could be capable of testing the attacks based on the resultant watermarked image in the DWT domain. Consequently, we obtained 71 files of attacks were located into a separate folder. At long last, we applied the *NCC* function on the extraction code which we computed in contemplation of performing the DWT extracting method; forasmuch as to test the original watermark and the extracted watermark. A table of analysis was achieved by setting the type of the copyrights (attacks) with respect to the DWT in such three different levels or robustness as in the following table.

7.5 Table B, results of the 2nd approach (DWT)

Copyright	K=5dB	Bi_1	K=10dB	Bi_2	K=15dB	Bi_3	K=20dB	Bi_4
Basic Collage Attack	0.2495	0	0.1250	0	0.0707	0	0.0421	0
Collage with rotated, scaled and cropped watermarked image	0.2521	0	0.1271	0	0.0706	0	0.0429	0
Watermark added to new image	0.7892	1	0.6849	0	0.5342	0	0.4008	0
Percent crop=10	0.2677	0	0.1306	0	0.0612	0	0.0287	0
Percent cropped=20	0.2667	0	0.1272	0	0.0590	0	0.0266	0
Percent cropped=50	0.2451	0	0.1008	0	0.0393	0	0.0171	0
Percent cropped=75	0.1942	0	0.0652	0	0.0266	0	0.0140	0
Dithered image	0.5555	0	0.5330	0	0.5275	0	0.5252	0
DPR attack 3x3 window	0.1447	0	0.1010	0	0.0744	0	0.0561	0
DPR attack 5x5 window	0.1379	0	0.0916	0	0.0656	0	0.0471	0
dprcorr attack 3x3 window	0.1534	0	0.1008	0	0.0728	0	0.0521	0
dprcorr attack 5x5 window	0.1333	0	0.0838	0	0.0608	0	0.0474	0
Gaussian filtered 3x3	0.6910	0	0.1016	0	0.0097	0	0.0020	0
Gaussian filtered 5x5	0.6912	0	0.1016	0	0.0097	0	0.0020	0
hardthresh 3x3 window	0.1690	0	0.0933	0	0.0616	0	0.0416	0
hardthresh 5x5 window	0.1178	0	0.0491	0	0.0291	0	0.0201	0
JPEG equal=10	0.1627	0	0.0507	0	0.0222	0	0.0112	0
JPEG equal=15	0.8810	1	0.8810	1	0.8810	1	0.7767	1
JPEG equal=25	0.1626	0	0.0500	0	0.0209	0	0.0102	0
JPEG equal=30	0.1624	0	0.0491	0	0.0205	0	0.0095	0
JPEG equal=40	0.1624	0	0.0490	0	0.0201	0	0.0093	0
JPEG equal=50	0.1622	0	0.0475	0	0.0190	0	0.0086	0
JPEG equal=60	0.1615	0	0.0468	0	0.0185	0	0.0085	0
JPEG equal=75	0.1648	0	0.0479	0	0.0197	0	0.0101	0
JPEG equal=80	0.8136	1	0.6833	0	0.5386	0	0.3397	0
JPEG equal=85	0.7912	1	0.6268	0	0.4145	0	0.2715	0
JPEG equal=90	0.7840	1	0.5688	0	0.3596	0	0.2402	0
JPEG equal=100	0.8891	1	0.8399	1	0.7514	1	0.5765	0
T11=1.15; T12=-0.02; T21=-0.03; T22=0.90	0.1879	0	0.0675	0	0.0307	0	0.0186	0
T11=0.8; T12=-0.1; T21=0.05; T22=1.1	0.1956	0	0.0734	0	0.0343	0	0.0190	0
T11=-0.85; T12=-0.2; T21=-0.05; T22=1.3	0.2018	0	0.0777	0	0.0377	0	0.0204	0
median 2x2 window	0.1683	0	0.0550	0	0.0238	0	0.0112	0
median 3x3 window	0.5557	0	0.2878	0	0.0730	0	0.0148	0
median 4x4 window	0.1665	0	0.0515	0	0.0221	0	0.0116	0
xscale=0.8 yscale=1	0.2000	0	0.0557	0	0.0206	0	0.0105	0
xscale=1 yscale=0.8	0.3049	0	0.1383	0	0.0208	0	0.0032	0
xscale=1 yscale=0.9	0.3114	0	0.1704	0	0.0661	0	0.0036	0
xscale=1 yscale=1.1	0.3658	0	0.1461	0	0.0161	0	0.0031	0
xscale=1 yscale=1.2	0.5714	0	0.1203	0	0.0083	0	0.0018	0

Copyright	K=5dB	Bi_1	K=10dB	Bi_2	K=15dB	Bi_3	K=20dB	Bi_4
rowcol1: 1 row and column removed	0.2814	0	0.1276	0	0.0445	0	0.0058	0
rowcol2: 1 row and 5 columns removed	0.2674	0	0.1236	0	0.0448	0	0.0079	0
rowcol3: 17 rows and 5 columns removed	0.3113	0	0.1429	0	0.0466	0	0.0063	0
rowcol4: 5 rows and 1column removed	0.2814	0	0.1322	0	0.0502	0	0.0103	0
rowcol5: 5 rows and 17 columns removed	0.3190	0	0.1520	0	0.0483	0	0.0066	0
Downsample factor 0.75, Upsample factor 1.3	0.1239	0	0.0259	0	0.0085	0	0.0033	0
Downsample factor 0.5, Upsample factor 1.9	0.1584	0	0.0456	0	0.0191	0	0.0098	0
Downsample factor 0.75, Upsample factor 1.333	0.1346	0	0.0360	0	0.0138	0	0.0068	0
Downsample factor 0.5, Upsample factor 2	0.1452	0	0.0441	0	0.0190	0	0.0097	0
Scaling 0.5	0.1448	0	0.0439	0	0.0190	0	0.0100	0
Scaling 0.75	0.1211	0	0.0284	0	0.0106	0	0.0045	0
Scaling 0.9	0.1716	0	0.0284	0	0.0073	0	0.0030	0
Scaling 1.1	0.1485	0	0.0306	0	0.0106	0	0.0053	0
Scaling 1.5	0.2160	0	0.0244	0	0.0056	0	0.0018	0
Scaling 2.0	0.1810	0	0.0235	0	0.0057	0	0.0019	0
sharpened	0.9305	1	0.9187	1	0.9081	1	0.9002	1
softthresh 3x3 window	0.1227	0	0.0471	0	0.0201	0	0.0101	0
softthresh 5x5 window	0.1070	0	0.0358	0	0.0160	0	0.0081	0
Image after template removal attack	0.3361	0	0.1233	0	0.0451	0	0.0087	0
Thresholded image	0.2634	0	0.1631	0	0.1380	0	0.1286	0
wavelet compression 8.0 bpp (uncompressed)	0.2717	0	0.1686	0	0.1401	0	0.1154	0
wavelet compression 3.5bpp	0.8936	1	0.8813	1	0.8507	1	0.5323	0
wavelet compression 1.5bpp	0.6102	0	0.5392	0	0.5057	0	0.4285	0
wavelet compression 0.8 bpp	0.8394	1	0.7932	1	0.7535	1	0.6334	0
wavelet compression 0.6 bpp	0.8813	1	0.8810	1	0.8807	1	0.6326	0
wavelet compression 0.5 bpp	0.8783	1	0.8417	1	0.8002	1	0.6668	0
wavelet compression 0.4 bpp	0.8919	1	0.8620	1	0.8192	1	0.6749	0
wavelet compression 0.3 bpp (uncompressed)	0.8998	1	0.8754	1	0.8340	1	0.6844	0
wavelet compression 0.2 bpp (uncompressed)	0.8995	1	0.8795	1	0.8412	1	0.6733	0
wavelet compression 0.1 bpp (uncompressed)	0.8810	1	0.8810	1	0.8810	1	0.7736	1
wiener 3x3 window	0.3347	0	0.1234	0	0.0449	0	0.0087	0
wiener 5x5 window	0.2761	0	0.1280	0	0.0662	0	0.0141	0

When the *NCC* value is set as *NCC* > 0.7 (70%) that means detected as a 1, and when the NCC value is set to *NCC* < 0.7 (70%) then it will be detected as a 0.

Chapter 8: Summary & Comprehensive Overview of the Thesis

8.1 Discrete Cosine Transform (DCT)

This approach inserts the watermark (invisible structure) in the DCT coefficients by breaking the image into 8 x 8 blocks and then computing the DCT for each block. The affiliation intervening the two DCT coefficients and the alongside watermark bit is observed and if needed then such modifications are made in order to impose a specific relationship between the two coefficients. The determination for the two coefficients and how they can be both chosen is based on the JPEG quantization table, where the coefficients are quantized with the same value in the actuality that the relationship will remain the same after the inverse quantization. Therefore, The middle frequency bands are chosen such that they have minimize and evade the most visual important parts of the image (low frequency coefficients) without over-exposing themselves to removal through compression and noise attacks (high frequency coefficients).

The main advantage of using F_L frequency region is for the purpose of denoting the lowest frequency coefficients of the block; while on the other hand, the F_H frequency region is being used in order to indicate the higher frequency coefficients. The F_M frequency band however, has been chosen as the embedding region as well as to provide such additional resistance to JPEG lossy compression techniques, while avoiding significance of being perceptible to the human eye due to the sensitivity of the human eye for the meaning of detecting the existence of noises within the F_L region [12].

8.1.1 Embedding Technique for DCT

One such a vital technique utilizes the comparison of middle-band DCT coefficients to encode a single bit into a DCT block. There are two major processes, encoding and embedding processes. **To commence**, we first need to go through the initial method that is the encoding and break the image into 8 x 8 blocks then apply the DCT function and computing all the image blocks separately and converting them into frequency components based on the frequency domain and having the whole details and components of the cover image and the watermark converted from the spatial domain into the frequency domain. **Secondly**, the second procedure which is the embedding process, thus, we have to clarify the middle-band frequency components (F_M) of the

image that consists of 8 x 8 DCT block. Then, generating the reduced image which is composed of middle-band frequency components and applying it for the watermark blocks; and with regard to the quantization table, we can select the two coefficients where we can insert the bits needed to embed the watermark. As a result, the watermarked image will be attained.

8.1.2 Extraction Technique for DCT

This approach (watermark detection) could be attained by a blind detection either non-blind watermark detection. On assumption of the non-blind detection of a watermark, we have two operations, detecting and extracting process. Through the first conduct, we need to have the cover image, the watermarked image either the watermark signature and break them up into 8 x 8 blocks. Then we should apply the DCT function and transform them into the frequency domain. The extracting process can then start by identifying the middle frequency components (F_M region) for the images and selecting their two coefficients by means of the quantization table; so that we can generate the reduced images and applying them to the watermark blocks and start to detect the bits where the watermark was embedded into; therefore, the watermark signature will be extracted.

8.2 Discrete Wavelet Transform (DWT)

The DWT function of a two-dimensional signal has got the ability of implementing the embedding and extracting systems for the approach of the invisible watermarking technology based on two significant operations, encoding and decoding procedure.

8.2.1 Encoding Approach for DWT

This technique will decompose the cover image of the two dimensional DWT into four frequency bands through the first pass as (LL_1), (LH_1), (HL_1) and (HH_1) frequency coefficients. The frequency bands where it has the lowest resolution of the 1^{st} pass (LL_1) can be also decomposed into a 2^{nd} level (pass). Secondly, we are to apply the Gaussian Noise and can insert the watermark signature into the reset of the available frequency bands which include the high frequency coefficients without dealing with (*LL*) regions from all over the passes (levels). We must add the signal of the bands where the large frequency components to the signal of the Gaussian Noise and modifying them without moderating the original signal which resides in the (*LL*) band; thereafter, the watermarked image would be performed appropriately.

8.2.2 Decoding Approach for DWT

In contemplation of achieving this procedure, we should have the cover image and the watermarked image readily applicable. Consequently, the DWT decoding technique will decompose those two images into four frequency bands through the 1st pass as described previously. Afterward, we are to select for instance one of those bands where the large frequencies reside through one of the levels (passes) in the decomposed cover image and the decomposed watermark. Let's suppose the selected band from both decomposed images is (HH_1), we have then to compare the difference of the frequency coefficients in those bands of the decomposed images and examine their cross correlation. Subsequently, if the cross correlation has detected a peak, then the watermark signature will be extracted; if not, then the same operation will continue on comparing the rest of the other bands consist the high frequency components from both of the decomposed images and investigate their cross correlation until the peak is detected; correspondingly, the watermark signature will be latterly recovered.

Appendix A: (DCT-Embedding)

This is the performed code in Matlab software for the 1st method which is based on the embedding technique of DCT-transform function.

```
%Name:              Abdullah S. A. Alzaid
%Final Thesis:     Digital Watermarking For Still Images
%Main Objective:   By means of exploiting the comparison between the
two
%                  mid-bands of the DCT coefficients, we are to
implement
%                  the method of embedding the watermark

%-----------------------------------------------------------------------
% The vital values being used:
% Cover Image:     X(N1,N2), its rows and columns being donated as
(N1c) for row and (N2c) for column.
% Watermark:       Y(M1,M2), its rows and columns being donated as
(M1m) for row and (M2m) for column.
% Embedded bits:   a,b
%-----------------------------------------------------------------------


clear all;

% saving the start time
start_time=cputime;

k=50;           % setting the least possible amount of the coefficient
distinction
blocksize=8;    % setting the size of the block in the cover image for
the seek of using it for each bit in the digital watermark

% reading through the original object (cover image)
file_name='lena.jpg';
cover_object=double(imread(file_name));

% verifying the given size of the cover image such that in Column & Row
N1c=size(cover_object,1);           %Row
N2c=size(cover_object,2);           %Column

% verifying the maximum message size with respect to the cover object
as well as in the blocksize
max_message=N1c*N2c/(blocksize^2);


% reading through the obtained message image
file_name='abdul.jpg';
```

```
message=double(imread(file_name));
M1m=size(message,1);                    %Row
M2m=size(message,2);                    %Column

% converting the attained message into a vector
message=round(reshape(message,M1m*M2m,1)./256);

% appraise the length of the message in case it would become very large
for the cover object
if (length(message) > max_message)
    error('Message too large to fit in Cover Object')
end

% fill out the message to the atmost message size with ones
message_pad=ones(1,max_message);
message_pad(1:length(message))=message;

% creating the shell for the digital watermarked image
watermarked_image=cover_object;

% need to operate the image in the blocks
% the encoding process, it would encode for instance (5,2) > (4,3)
whenever the message is (kk)=0
% and would encode (5,2) < (4,3) in case the message is(kk)=1
a=1;
b=1;
for (kk = 1:length(message_pad))

    % transform block by means of using the DCT transform function
    dct_block=dct2(cover_object(b:b+blocksize-1,a:a+blocksize-1));

    % if the message bit is black, (5,2) > (4,3)
    if (message_pad(kk) == 0)

        % if (5,2) < (4,3) then it would be necessary to swap them
        if (dct_block(5,2) < dct_block(4,3))
                temp=dct_block(4,3);
                dct_block(4,3)=dct_block(5,2);
                dct_block(5,2)=temp;
        end

    % if the message bit is white, (5,2) < (4,3)
    elseif (message_pad(kk) == 1)

        % if (5,2) > (4,3) then it would be necessary to swap them
        if (dct_block(5,2) >= dct_block(4,3))
                temp=dct_block(4,3);
                dct_block(4,3)=dct_block(5,2);
                dct_block(5,2)=temp;
        end
    end

    % adjusting the two values such as their variance >= k
    if dct_block(5,2) > dct_block(4,3)
        if dct_block(5,2) - dct_block(4,3) < k
```

```
            dct_block(5,2)=dct_block(5,2)+(k/2);
            dct_block(4,3)=dct_block(4,3)-(k/2);
        end
    else
         if dct_block(4,3) - dct_block(5,2) < k
            dct_block(4,3)=dct_block(4,3)+(k/2);
            dct_block(5,2)=dct_block(5,2)-(k/2);
        end
    end

    % transforming back the block into spatial domain
    watermarked_image(b:b+blocksize-1,a:a+blocksize-
1)=idct2(dct_block);

    % moving ahead onto the afterwards block. in the case of the row,
it would also move onto the afterwards row
    if (a+blocksize) >= N1c
        a=1;
        b=b+blocksize;
    else
        a=a+blocksize;
    end
end

% transforing to uint8 and else writting the digital watermarked image
to a file
watermarked_image_int=uint8(watermarked_image);
imwrite(watermarked_image_int,'dct_watermarked_lena1.jpg','jpg',
'Quality',100);

% displaying the processing time
elapsed_time=cputime-start_time,


% demonstrating the resultant digital watermarked image
figure(1)
imshow(watermarked_image,[])
title('Watermarked Image')


% displaying PSNR of watermarked image
psnr=psnr(cover_object,watermarked_image, N1c, N2c),
```

Appendix B: (DCT-Extraction)

This is the performed code in Matlab software for the 2nd method which is based on the extracting technique of DCT-transform function.

```
%Name:              Abdullah S. A. Alzaid
%Final Thesis:     Digital Watermarking For Still Images
%Main Objective:   By means of exploiting the comparison between the
two
%                  mid-bands of the DCT coefficients, we are to
implement the method of
%                  retrieving the watermark


%--------------------------------------------------------------------
% The vital values being used:
% Watermarked Image:       X`(N1,N2), its rows and columns being
donated as (N1w) for row and (N2w) for column.
% Original Watermark:      Y (M1,M2), its rows and columns being
donated as (M1o) for row and (M2o) for column.
% Detected bits:           a,b
%--------------------------------------------------------------------


clear all;

fileList = dir('ATTACKED');
NumberOfFiles = size (fileList);
for j=3:NumberOfFiles(1)
    x=fileList(j).name
% reading through the obtained digital watermarked image (watermarked
% object)
    watermarked_image = double
(imread(['ATTACKED/',fileList(j).name]));
    sizeI = size(watermarked_image);
    sizeI(1,2);
    if sizeI(1,1) ~= 512 || sizeI(1,2) ~= 512
        watermarked_image = imresize (watermarked_image,[512 512],
'bilinear');
    end

    % saving the start time
    start_time=cputime;
    blocksize=8;    % setting the size of the block in the cover image
for the seek of using it for each bit in the digital watermark
```

```
% verifying the given size of the attained watermarked image such that
in row & column
    N1w=size(watermarked_image,1);          %Row
    N2w=size(watermarked_image,2);          %Column

    % verifying the maximum message size with respect to the cover
object as well as in the blocksize
    max_message=N1w*N2w/(blocksize^2);

    % reading through the original digital watermark
    file_name='abdul.jpg';
    orig_watermark=double(imread(file_name));

    % verifying the size of original digital watermark such that in row
& column
    M1o=size(orig_watermark,1);         %Row
    M2o=size(orig_watermark,2);         %Column

    % operating the image in the blocks
    a=1;
    b=1;
    for (kk = 1:max_message)

        % converting blocks by means of using the DCT function
        dct_block=dct2(watermarked_image(b:b+blocksize-1,a:a+blocksize-
1));

        % if the dct_block(5,2) > dct_block(4,3) then the message will
be(kk)=0
        % or else the message will be(kk)=1
        if dct_block(5,2) > dct_block(4,3)
            message_vector(kk)=0;
        else
            message_vector(kk)=1;
        end

        % moving ahead onto the afterwards block. in the case of the
row, it would also move onto the afterwards row
        if (a+blocksize) >= N1w
            a=1;
            b=b+blocksize;
        else
            a=a+blocksize;
        end
    end

    % regulating the message that has been embedded
    message=reshape(message_vector(1:M1o*M2o),M1o,M2o);

    % viewing the processing time
    elapsed_time=cputime-start_time;
```

```
 % viewing the recovered message
    imshow(message,[]);
    message(find(message>=0.5))=255;
    message(find(message<0.5))=0;

    NCC(j-2) = ncc(orig_watermark, message);
end
NCC (find(NCC>1))=1;
```

Appendix C: (DWT-Embedding)

This is the performed code in Matlab software for the 3rd method which is based on the embedding technique of DWT-transform function.

```
%Name:              Abdullah S. A. Alzaid
%Final Thesis:     Digital Watermarking For Still Images
%Main Objective:   By means of exploiting embedding of CDMA watermark
into H1,V1,D1 components of a 1-scale DWT
%                  by means of the method of embedding the watermark

%-------------------------------------------------------------------------
% The vital values being used:
% Cover Image:     X(N1,N2), its rows and columns being donated as
(N1c) for row and (N2c) for column.
% Watermark:       Y(M1,M2), its rows and columns being donated as
(M1m) for row and (M2m) for column.
% Frequency Bands in the 1st pass:     A1 referred to (LL1), H1
referred to (HL1), V1 (LH1), and D1 referred to (HH1.
%Coefficients in the 1st pass :        cA1 from (LL1), cH1 from (HL1),
cV1 from (LH1), and cD1 from (HH1) band.
%-------------------------------------------------------------------------


clear all;

% saving start time
start_time=cputime;

k=10;            % set the gain factor for embeding

% reading through the original object (cover image)

file_name='lena.jpg';
cover_object=double(imread(file_name));

% verifying the given size of the cover image such that in Column & Row
N1c=size(cover_object,1);           %Row
N2c=size(cover_object,2);           %Column


% reading through the obtained message image
file_name='abdul.jpg';
message=double(imread(file_name));
```

```
M1c=size(message,1);            %Row
M2c=size(message,2);            %Column

 for i=1:M1c
        for j=1:M2c
            if message(i,j) > 128
                message(i,j) = 1;
            else
                message(i,j) = 0;
            end
        end
    end


[cA1,cH1,cV1,cD1] = dwt2(cover_object,'haar');   %the frequency
coefficients through the bands of the 1st pass (level)

%reshape cH1, cV1

cD1 = cD1 + k*message



%perform IDWT
watermarked_image = idwt2(cA1,cH1,cV1,cD1,'haar');

% convert back to uint8
watermarked_image_uint8=uint8(watermarked_image);

% writting the digital watermarked image to a file
imwrite(watermarked_image_uint8,'dwt_watermarked_lena1.jpg', 'Quality',
100);

% displaying the processing time
elapsed_time=cputime-start_time,


% displaying the resultant digital watermarked image
figure(1)
imshow(watermarked_image_uint8,[])
title('Watermarked Image')


% displaying PSNR of watermarked image
psnr=psnr(cover_object,watermarked_image, N1c, N2c),
```

Appendix D: (DWT-Extraction)

This is the performed code in Matlab software for the 4th method which is based on the extracting technique of DWT-transform function.

```
%Name:              Abdullah S. A. Alzaid
%Final Thesis:     Digital Watermarking For Still Images
%Main Objective:   By means of exploiting extracting of CDMA watermark
into H1,V1,D1 components of a 1-scale DWT
%                  by means of the method of retrieving the watermark


%--------------------------------------------------------------------
% The vital values being used:
% Watermarked Image:       X`(N1,N2), its rows and columns being
donated as (N1w) for row and (N2w) for column.
% Original Watermark:      Y (M1,M2), its rows and columns being
donated as (M1o) for row and (M2o) for column.
% Frequency Bands in the 1st pass:      A1 referred to (LL1), H1
referred to (HL1), V1 (LH1), and D1 referred to (HH1.
%Coefficients in the 1st pass :         cA1 from (LL1), cH1 from (HL1),
cV1 from (LH1), and cD1 from (HH1) band.
%--------------------------------------------------------------------

clear all;


k=10
file_name = 'lena.jpg';
cover_image=double(imread(file_name));
file_name='abdul.jpg';
orig_watermark=double(imread(file_name));


fileList = dir('ATTACKED');
NumberOfFiles = size (fileList);
for j=3:NumberOfFiles(1)
    x=fileList(j).name
    % reading through the obtained digital watermarked image
(watermarked object)
    watermarked_image = double
(imread(['ATTACKED/',fileList(j).name]));
    sizeI = size(watermarked_image);
    sizeI(1,2);
    if sizeI(1,1) ~= 512 || sizeI(1,2) ~= 512
        watermarked_image = imresize (watermarked_image,[512 512],
'bilinear');
    end
```

```
    % verifying the given size of the attained watermarked image such
that in row & column
    N1w=size(watermarked_image,2);          %Row
    N2w=size(watermarked_image,1);          %Column

    [cA1o,cH1o,cV1o,cD1o] = dwt2(cover_image,'haar'); %2-D DWT
decomposition
    [cA1w,cH1w,cV1w,cD1w] = dwt2(watermarked_image,'haar'); %2-D DWT
decomposition

    extractedWatermark = ((cD1w-cD1o)/k);

    extractedWatermark(find(extractedWatermark>=0.5))=1;
    extractedWatermark(find(extractedWatermark<0.5))=0;


    orig_watermark(find(orig_watermark>=0.5))=1;
    orig_watermark(find(orig_watermark<0.5))=0;

    NCC(j-2) = ncc(orig_watermark, extractedWatermark)
end
NCC (find(NCC>1))=1;
```

Appendix E: (PSNR Function)

This is the method of computing the PSNR function (peak signal to noise ratio) in the Matlab software platform for the reason of measuring the distinctive distortion between the cover image and the watermarked image.

```
%Name:              Abdullah S. A. Alzaid
%Final Thesis:     Digital Watermarking For Still Images
%Main Project:     Calculating the PSNR (Peak Signal to Noise Ratio)
%                  of images X and X', both of size N1xN2


function [X] = psnr(image,image_prime,N1,N2)

    % converting to doubles
    image=double(image);
    image_prime=double(image_prime);

    % avoiding dividing by zero nastiness
    if ((sum(sum(image-image_prime))) == 0)
        error('Input vectors must not be identical')
    else
        psnr_num=N1*N2*max(max(image.^2));                % calculating
the numerator
        psnr_den=sum(sum((image-image_prime).^2));        % calculating
the denominator
        X=psnr_num/psnr_den;                              % calculating
the PSNR
    end

return
```

Appendix F: List of Files (Attacks) of DCT/DWT

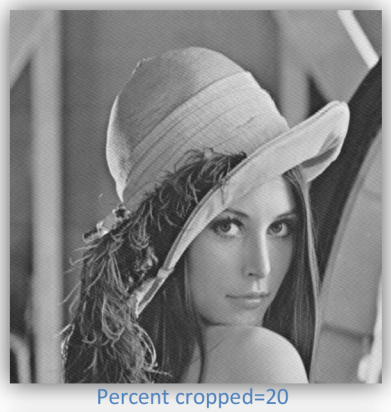
The following files are the attacks obtained by applying the checkmark for DCT/DWT:

Basic Collage Attack

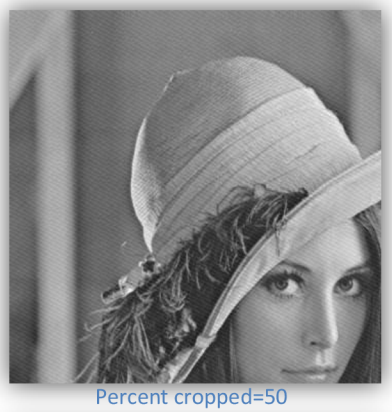
Collage with rotated, scaled and cropped watermarked image

Watermark added to new image

Percent crop=10



Percent cropped=20

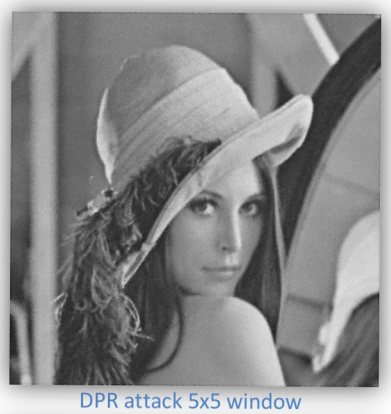


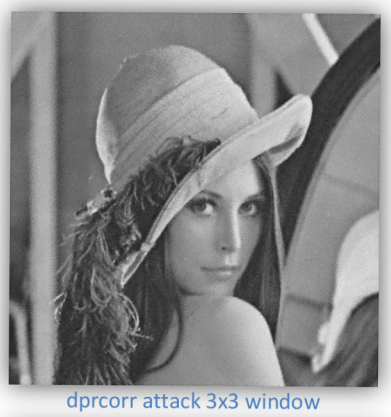
Percent cropped=50

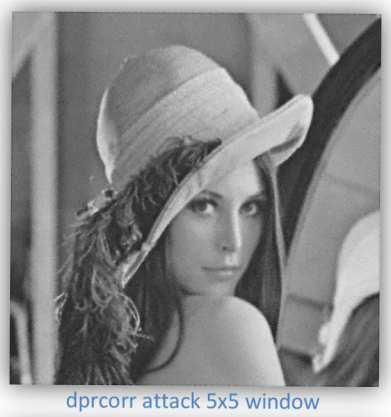
Percent cropped=75

Dithered image

DPR attack 3x3 window


DPR attack 5x5 window


dprcorr attack 3x3 window


dprcorr attack 5x5 window


Gaussiian filtered 3x3


Gaussiian filtered 5x5


hardthresh 3x3 window


hardthresh 5x5 window


JPEG 10


JPEG 15


JPEG 25


JPEG 30


JPEG 40

JPEG 50

JPEG 60

JPEG 75

JPEG 80

JPEG 85

JPEG 90

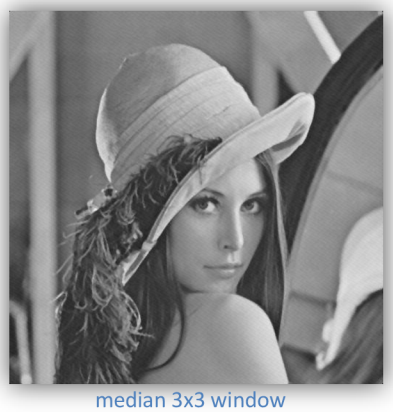
JPEG 100

T11=1.15; T12=-0.02; T21=-0.03; T22=0.90

T11=0.8; T12=-0.1; T21=0.05; T22=1.1

T11=-0.85; T12=-0.2; T21=-0.05; T22=1.3

median 2x2 window



median 3x3 window


median 4x4 window

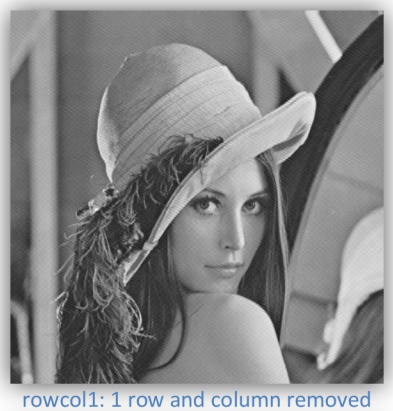

xscale=0.8 yscale=1

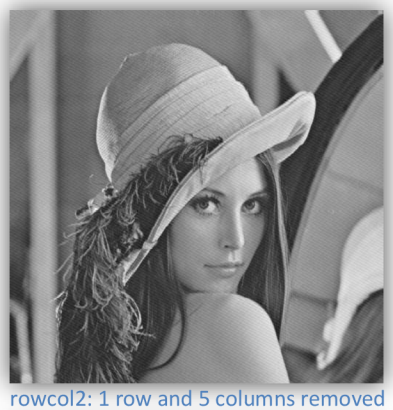

xscale=1 yscale=0.8

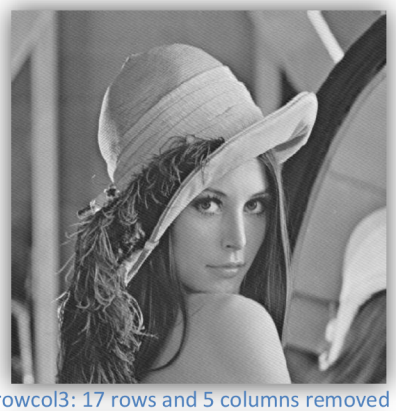

xscale=1 yscale=0.9

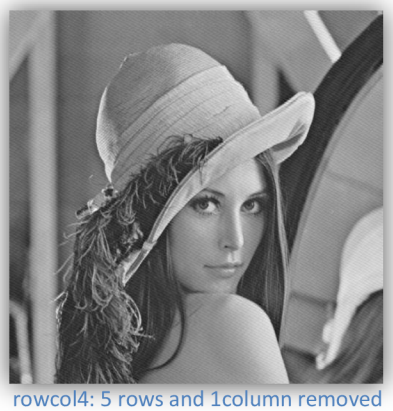

xscale=1 yscale=1.1

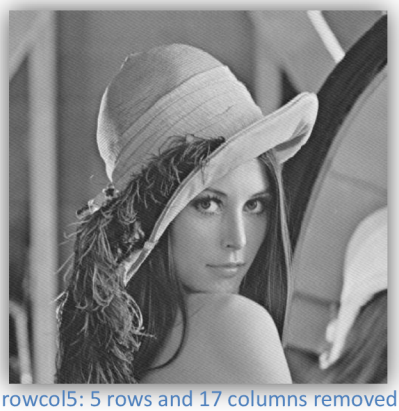

xscale=1 yscale=1.2


rowcol1: 1 row and column removed


rowcol2: 1 row and 5 columns removed


rowcol3: 17 rows and 5 columns removed


rowcol4: 5 rows and 1column removed


rowcol5: 5 rows and 17 columns removed


Downsample factor 0.75, Upsample factor 1.3

Downsample factor 0.5, Upsample factor 1.9

Downsample factor 0.75, Upsample factor 1.333

Downsample factor 0.5, Upsample factor 2

Scaling 0.5

Scaling 0.75

Scaling 0.9

Scaling 1.1

Scaling 1.5

Scaling 2.0

sharpened

softthresh 3x3 window

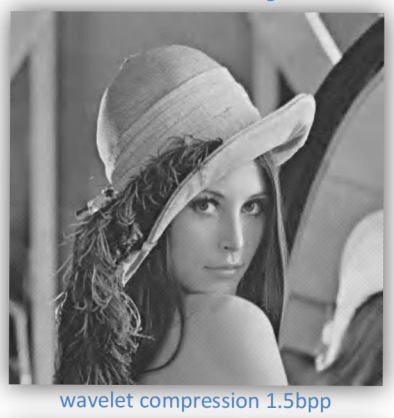
softthresh 5x5 window

Image after template removal attack

Thresholded image

wavelet compression 8.0 bpp (uncompressed)

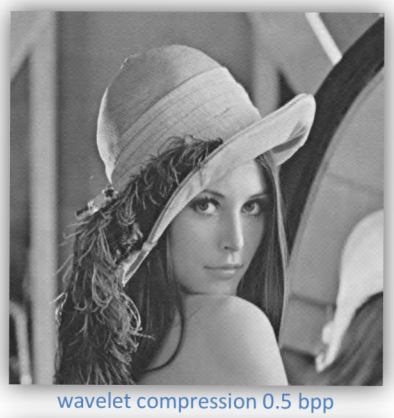
wavelet compression 3.5bpp



wavelet compression 1.5bpp

wavelet compression 0.8 bpp

wavelet compression 0.6 bpp



wavelet compression 0.5 bpp

wavelet compression 0.4 bpp

wavelet compression 0.3 bpp (uncompressed)

wavelet compression 0.2 bpp (uncompressed)

wavelet compression 0.1 bpp (uncompressed)

wiener 3x3 window

wiener 5x5 window

References

[1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, vol. 6, pp. 1673 1687, December 1997.

[2] W. Zeng and B. Liu, "A Statistical Watermark Detection Technique Without Using Original Images for Resolving Rightful Ownerships of Digital Images". *1EEE Transactions on Image Processing,* 8(11), 1534-1548.W. Zeng and B. Liu, "A Statistical Watermark Detection

[3] Kankanhalli M.S., et al., "Adaptive Visible Watermarking of Images", appeared in *Proc. of ICMCS'99*, Florence, Italy, June 1999.

[4] N.F. Johnson and S. Jajodia, “Exploring Steganography: Seeing the Unseen,” Computer, vol. 31, no. 2, 1998, pp. 26–34.

[5] Amin, P., Lue, N. and Subbalakshmi, K., “Statistically secure digital image data hiding”, in IEEE Multimedia Signal Processing MMSP05, China, Oct. 2005, pp. 1-4.

[6] Deng, F. and B. Wang, 2003. “A novel technique for robust image watermarking in the DCT domain,” in Proc. of the IEEE 2003 Int. Conf. on Neural Networks and Signal Processing, vol. 2, pp: 1525-1528.

[7] A. J. Ahumada and H. A. Peterson, “Luminance-modelbased DCT quantization for color image compression,” in *Human Vision, Visual Processing, and Digital Display III (Proc. of the SPIE)* (B. E. Rogowitz, ed.), 1992.

[8] Hsieh, M., D. Tseng, and Y. Huang, 2001. “Hiding Digital Watermarks Using Multiresolution Wavelet Transform,” IEEE Trans. on Industrial Electronics, 48(5): 875-882.

[9] R. Kronland-Martinet, J. Morlet, and A. Grossmann, “Analysis of Sound Patterms Through Wavelet Transforms,” Int. J. Pattern Recognition and Artificial Intelligence, Vol. 1, No.2, pp. 273-302, pp.97-126, 1987.

[10] Burrus,C.S., Gopinath, R.A., Guo, H., *Introduction to Wavelet and Wavelet Transforms: A Primer* (Prentice Hall, 1998).

[11] Misiti, M., Miairi, T., Oppenheim, G., Poggi, J.M., *Wavelet Toolbox User's Guide* (the MathWorks, Inc., 2000).

[12] J. R. Hern´andez, F. P´erez-Gonz´alez, J. M. Rodr´ıguez, and G. Nieto, "Performance analysis of a 2d-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 510–524, May 1998.