Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Design of an Efficient Robust Digital Image Watermarking Scheme Based on the DCT and Human Visual Model

Pieter Barnard

14312157

April 2019

A dissertation submitted in partial fulfilment of the degree of
MAI (Electronic & Computer Engineering)

Declaration

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

_____

Pieter Barnard

Abstract

The ever-growing sophistication of embedded devices such as laptops, tablets, smart phones and digital cameras, which are capable of not only producing large amounts of multimedia content on-the-fly but also capable of editing, uploading and sharing such data effortlessly over the Internet, has meant that content providers are increasingly presented with new challenges with regards to providing protection against unlawful copyright infringements, fraudulent tampering or any other form of authentication required by the potentially sensitive content being produced by these devices.

In an effort to address these issues, digital watermarking has emerged as an effective technique for deterring against the misappropriation of digital data, as well as providing a legal means for original content owners to identify and subsequently prosecute the unauthorised distributors of the copyrighted material in question.

In the following project, digital watermarking techniques suited towards the application of digital image content are investigated. Particularly, robust digital watermarking techniques which are based on the Discrete Cosine Transform (DCT) and Human Visual System (HVS) are explored and state-of-the-art watermarking schemes are investigated for their strengths and weaknesses. The outcome of this investigation has led to the design and implementation of a novel robust digital image watermarking scheme which is capable of withstanding the harsh attacks and lossy compression schemes that these watermarking schemes are expected to be subjected to.

Acknowledgments

I would like to thank my supervisor Dr. Liam Dowling for his invaluable support and guidance throughout the course of this project.

I would also like to express my profound gratitude to my parents and my brother who have supported me throughout all my years of study and have always encouraged me to do my best.

Contents

Abstract iii
Acknowledgments iv
Contents v
List of Tables viii
List of Figures ix
Abbreviations and Acronyms xi
1 Introduction 1
1.1 Problem Outline 1
1.2 Introduction to Digital Watermarking 1
1.3 Applications of Digital Watermarking 2
1.3.1 Copyright Protection 2
1.3.2 Broadcast Monitoring 2
1.3.3 Content Integrity 2
1.3.4 Content Filtering 2
1.4 Watermark Classification 3
1.4.1 Visible Watermarking 3
1.4.2 Invisible Watermarking 3
1.5 Project Scope & Objectives 4
2 A General Watermarking System 5
2.1 Overview 5
2.2 Watermark Embedding 5
2.2.1 A General Embedding Function 5
2.2.2 Spatial vs. Frequency Domain 6
2.2.3 Embedding within the DCT Domain 8
2.2.4 The Watermark 11
2.2.5 Properties of the Embedding Stage 12
2.3 Channel Stage 13
2.3.1 Average Filter 14
2.3.2 Median Filter 14
2.3.3 Wiener Filter 15
2.3.4 Gaussian Low Pass Filter 15
2.3.5 Additive Gaussian White Noise 15
2.3.6 Salt-and-Pepper Noise 15

2.3.7 Poisson Noise 16
2.3.8 Speckle Noise 16
2.3.9 Image Cropping 16
2.3.10 Affine Transformation 16
2.3.11 JPEG Compression 17
2.3.12 JPEG2000 18
2.4 Watermark Extraction 19
2.4.1 A General Extraction Function 19
2.4.2 Extraction within the DCT domain 19
2.4.3 Detection within the DCT Domain 20
2.4.4 Properties of the Extraction Stage: 21
3 Perceptual Modelling 24
3.1 Overview 24
3.2 Human Visual System Models 25
3.2.1 Spatial Frequency Sensitivity (CSF) 25
3.2.2 Luminance adaption 28
3.2.3 Contrast Masking 29
3.3 Common Image Quality Assessment Metrics 30
3.3.1 Mean Square Error (MSE) 30
3.3.2 Peak Signal-to-Noise Ratio (PSNR) 30
3.3.3 Absolute Reconstruction Error (ARE) 31
3.3.4 SSIM 31
4 Related Work 33
5 Experimental Work 35
5.1 Implementation of Ernawan's Scheme 35
5.2 Implementation of Levicky's HVS model 37
5.2.1 Frequency Sensitivity Model 38
5.2.2 Luminance Adaption 39
5.2.3 Contrast Masking 39
5.2.4 Proposed Design 41
6 Experimental Results & Evaluation 44
6.1 Implementation of Ernawan's Watermarking System 44
6.1.1 Comparison between SSIM & NC trade-off graphs 44
6.1.2 Comparison between PSNR 45
6.1.3 Comparison of Attacks (1) 45

6.1.4 Comparison of Attacks (2) 46
6.2 Comparison of Attacks (3) 47
6.3 Comparison of SSIM and ARE 48
6.4 Implementation of the proposed scheme 48
6.4.1 Comparison between SSIM & NC trade-off graphs 48
6.4.2 Comparison between PSNR 49
6.4.3 Comparison of Attacks (1) 50
6.4.4 Comparison of Attacks (2) 50
6.4.5 Comparison of Attacks (3) 51
6.4.6 Comparison of SSIM and ARE 51
6.4.7 Additional Testing 52
7 Conclusion 54
8 References 55
9 Appendix 57

List of Tables

Table 1: Abbreviations and Acronyms used in this report xi
Table 2: Comparison between ARE and SSIM for Ernawan's published scheme and Ernawan's implemented scheme..... 48
Table 3: Comparison between ARE and SSIM for proposed scheme and Ernawan's implemented scheme 51

List of Figures

Figure 1: Main stages of a generalized watermarking system. 5
Figure 2: Bit planes 1,3,6,8 of the Lena image (top row) and corresponding LSB watermarked images (bottom row). 6
Figure 3: Lena image (left) and binary watermark image (right)....7
Figure 4: Example of watermark embedding within DCT of a 4 x 4 image segment....9
Figure 5: Zig-Zag pattern used to vectorise DCT matrix. 11
Figure 6: Original watermark (left), scrambled watermark using Arnold chaotic map with 39462 iterations (right). 12
Figure 7: Average filter using the kernel concept. 14
Figure 8: Outline of the JPEG Compression Scheme. 17
Figure 9: Lena image under varying JPEG qualities, where left leftmost image has quality 5, centre image has quality 15 and rightmost image has quality 50. 17
Figure 10: Comparison between JPEG and JPEG2000. 18
Figure 11 DCT of unwatermarked image on the left and DCT of watermarked image (post quantization) on the right. 19
Figure 12: Graph of the JPEG baseline quantization table 24
Figure 13: Example CSF using Ahumada and Peterson's model [15] 27
Figure 14: Example of Ahumada's JND & CSF models 28
Figure 15: Relative Contrast masking threshold on Lena image. 30
Figure 16: Graph from Ernawan's scheme showing intersection point of SSIM and NC the optimal embedding strength [5]. 33
Figure 17: Selected embedding locations (in black) for Lena image using Ernawan's scheme (left image), Lai's scheme (centre image) and Maity's scheme (right image). 36
Figure 18: Plot of Frequency Sensitivity model used by Levický et *al*. [27] 38
Figure 19: Relative luminance adaption thresholds on Lena image 39
Figure 20: NVF map of Lena image. 40
Figure 21: Relative contrast visibility thresholds for Lena image. 41
Figure 22 : Comparison between Ernawan's SSIM and NC trade off vs. Implemented SSIM and NC trade off 44
Figure 23: Comparison of PSNR values between Ernawan's published scheme and implemented scheme 45
Figure 24: Comparison of attacks (1) between Ernawan's published scheme and implemented scheme 45
Figure 25: Comparison of compression attacks between Ernawan's published scheme and implemented scheme 46
Figure 26: Process used in estimating JPEG performance of Ernawan's scheme for qualities below 30 47
Figure 27: Comparison of attacks (3) between Ernawan's published scheme and implemented scheme 47
Figure 28: SSIM and NC trade-off curve of proposed scheme for the Lena image 48
Figure 29: Comparison of PSNR values between Ernawan's published scheme and implemented scheme 49

Figure 30: Comparison of attacks (1) between proposed scheme and Ernawan's implemented scheme. 50
Figure 31: Comparison of compression attacks between proposed scheme and Ernawan's implemented scheme 50
Figure 32: Comparison of attacks (3) between proposed scheme and Ernawan's implemented scheme 51
Figure 33: Rotational testing on proposed scheme and Ernawan's implemented scheme (1). 52
Figure 34: Rotational testing on proposed scheme and Ernawan's implemented scheme (2). 53

Abbreviations and Acronyms

AGWN	Additive Gaussian White Noise
HVS	Human Visual System
IQA	Image Quality Assessment (metric)
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MSE	Mean Square Error
NC	Normalized Cross-Correlation
NVF	Noise Visibility Function
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Metric

Table 1: Abbreviations and Acronyms used in this report

1 Introduction

1.1 Problem Outline

This decade has experienced a tremendous growth in the amount of multimedia content being published and distributed across the Internet and digital consumer market.

At one end of the scale, the increasing sophistication of embedded technology means that even the most inexpensive electronic devices available on the consumer market today tend to have the processing capabilities to allow multimedia data such as video, image and audio to be rapidly captured and effortlessly uploaded onto the Internet where it becomes easily accessible by other users who may copy, edit and share this data further down the networking stream.

At the other end of the scale, the Internet has quite recently become a massive hub for large content providers, such as Netflix and YouTube, to distribute their material both cheaply and quickly to the wider public.

Overall, the immense growth in digital content and ease in which it may be distributed over the Internet has made it easier than ever for people to illegally copy and distribute this content to others. Conversely, the challenges faced by content providers and in protecting against the illegal copying or tampering of their content is becoming increasingly more difficult. In an effort to address these issues, digital watermarking has emerged as an effective technique for deterring against the misappropriation of digital data, as well as providing a legal means for content owners to identify and possibly prosecute any unauthorised distributors of their copyrighted material.

1.2 Introduction to Digital Watermarking

In the most general sense, watermarking refers to a process in which an information-bearing signal is embedded within a host signal for the purpose of providing some form of authenticity to the underlying host signal. Undoubtedly, one of the most widely known examples of watermarking can be realized on banknotes, in which the presence of a watermark can be used to distinguish a counterfeit banknote from a genuine one. In a similar fashion, the concept of watermarking may also be applied to digital material, in which a digital watermark is embedded within a digital host such as an image, video, audio or text. This notion of applying the watermarking concept within the digital domain is commonly referred to as digital watermarking.

It is difficult to determine when and by whom the concept of digital watermarking was first conceived by, however, according to [1], the first paper to be published in this field was in 1993, when Caronni et al. [2, 3] presented the first implementation of a watermarking scheme for copyright protection of digital images. Since then, the field of digital watermarking has received tremendous interest from both the scientific community and cryptographic community, whom have been successfully able to extend the concept of digital watermarking to both a wide range of applications, as well as to various other forms of digital material such as video, audio and text documents.

1.3 Applications of Digital Watermarking

Today, some of the most common applications of digital watermarking include copyright protection, broadcast monitoring, content integrity and content filtering:

1.3.1 Copyright Protection

In copyright applications, a content owner creates multiples copies of their work and embeds a unique watermark within each copy. To provide the necessary protection against copyright infringement, the owner distributes a single watermarked copy to each of the respective recipients. Thus, if a copy of the content is later found to be leaked, the owner can extract the watermark from that copy and determine through association which recipient has leaked their version of the content.

Furthermore, because this type of application inherently deals with adversaries who may attempt to foil the underlying protection mechanism by destroying the watermark, it is essential that these watermarking schemes be highly imperceptible as well as resilient to any signal processing attacks, channel noise and lossy compression processes which the content may be subjected to. For this reason, the watermarking schemes used in such circumstances are commonly referred to as robust watermarking schemes.

1.3.2 Broadcast Monitoring

Watermarking is often employed by advertising companies as a means for tracking the duration and frequency of their advertisements as they are aired by television stations. According to [4], the need for this first emerged when a scandal broke out in Japan in 1997 after it was discovered some television stations were routinely overbooking air time with advertising agencies, and therefore failing to air thousands of advertisements that were paid for. In this context, an advertising agency can embed a watermark into their advertisements and use an automated method to continuously scan and extract the watermark as their advertisements are aired.

1.3.3 Content Integrity

In a content integrity application, a content user wishes to ensure that the content they are using has not been tampered with or altered in any way. To achieve this, a watermarking scheme may be designed so that the underlying watermark is extremely sensitive to any changes in the watermarked content. Thus, if the extracted watermark is found to be different from the original watermark in any way, the user will know that the content has been modified.

Furthermore, because the watermarking schemes used in these applications are designed to degrade at the early onset of any modifications to the content, these schemes are typically termed as fragile watermarking schemes.

1.3.4 Content Filtering

In content filtering, descriptors, known as metadata, are inserted into the header sections of content files to give an indication as to what that content contains or describes. This concept is particularly important in the case of multimedia content, as such content could not otherwise be included in a search engine's results, as the search engine can only perform a

textual scan of each file during a search operation. However, because the metadata is inserted within the header section of each file, a security flaw exits whereby it is possible for a hacker to corrupt or delete the metadata if they gain access to the respective content.

To tackle this issue, recent approaches have been to insert the metadata directly into the respective content by embedding this information as a watermark. In doing so, it not only becomes extremely difficult for a hacker to extract the metadata but can also be used as an additional form of copyright protection.

1.4 Watermark Classification

To date there have been hundreds to thousands of schemes published within the watermarking community. Unsurprisingly, it has become a challenge in some contexts as to how these schemes should be classified under a standard set of criteria. Most often, classification schemes are proposed in ad hoc manners according to some combination of criteria such as:

- Application
- Embedding domain
- Digital form
- Extraction information
- Robust/Fragile
- Watermark visibility etc.

For brevity, this report will consider only the case of watermark visibility to be of fundamental importance to the classification problem. However, it is assumed that further distinctions may be drawn where necessary, by considering the extent to which certain schemes may agree or disagree to the remaining criteria mentioned, as well as to some of the watermarking properties outlined in chapter 2 of this report.

1.4.1 Visible Watermarking

In visible watermarking, the watermark is directly embedded into the host content without any attempt in masking its existence from the user. An example of a visible watermark includes the '©' symbol which is commonly displayed on digital images to signify a copyright notice on the respective content. However, as discussed in [4] a fatal flaw with most visible watermarks is that they can be easily removed from the host content by means of simple photoshopping or cropping techniques. For this reason, visible watermarking will be considered outside the scope of this project.

1.4.2 Invisible Watermarking

Invisible watermarking considers the process of embedding the watermark so that its presence within the host content is extremely difficult for a user to visually perceive. This essential characteristic means that invisible watermarking schemes have an inherent security gain over their visible counterparts, as well as making their use preferable within multimedia applications where perceptual quality or fidelity of the rendered content is vital for the user's experience.

However, designing a watermarking scheme to be invisible to the human senses has proven itself to be an extremely challenging and complex task. Over the years, numerous schemes have been proposed by authors attempting to find optimal ways for embedding the watermark in an imperceptible manner. Some of the most successful schemes have utilised techniques such as:

- Embedding within the frequency domain of the image.
- Varying the strength and density of watermarking bits according to the local image content (i.e. texture regions, edges and flat regions etc.).
- Incorporating Human Visual System (HVS) models to determine optimal embedding parameters within the frequency and spatial domain of the content.

1.5 Project Scope & Objectives

The following project focuses on the design and evaluation of a novel digital watermarking scheme for copyright applications involving natural still images. In particular, the proposed scheme aims to build upon the existing work within this field by developing a novel framework which combines both the strengths across a multitude of existing and state-of-the-art research, as well adapting to overcome their individual weaknesses as a whole. To achieve this goal, a number of objectives have been set out over the course of the project and include,

- Any necessary prior study related to general digital watermarking theory and practices.
- Implementation of the state-of-the-art scheme presented by Ernawan et *al*. [5].
- Evaluation of the strengths and weaknesses of Ernawan's scheme.
- Implementation and Evaluation of various Human Visual System models adapted for watermarking applications and general image processing.
- Design and evaluation of an improved watermarking scheme by incorporating the techniques found in the former schemes, as well as in various other state-of-the-art watermarking literature encounter over the course of the project.

2 A General Watermarking System

2.1 Overview

A watermarking system encapsulates the start to end processes involved during the lifetime of a watermarked material. Essentially, this includes the time at which the watermark is first embedded, distributed to a recipient via a communication channel and consequently validated to identify the presence of the watermark. In the following project, a general watermarking system is considered to consist of three main components, the embedding stage, channel stage and extraction stage:

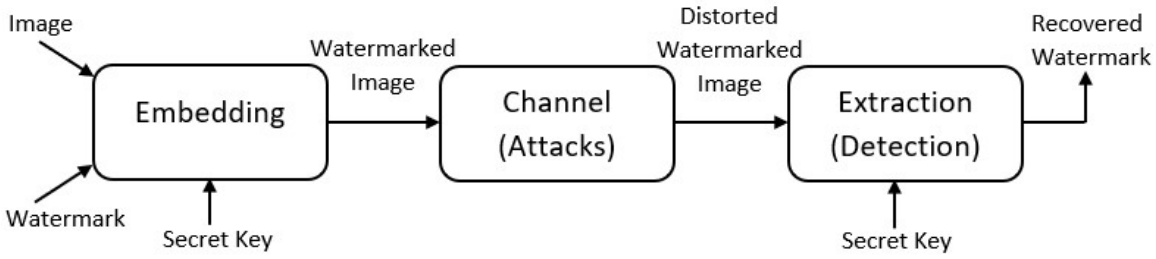


Figure 1: Main stages of a generalized watermarking system.

2.2 Watermark Embedding

2.2.1 A General Embedding Function

A watermarked image is created by embedding the watermark sequence into the host image according to some mathematical rule or function. In practice, there are numerous different functions which may be implemented at this stage, with each offering its own set of advantages and disadvantages for a given application. Nevertheless, the majority of existing functions may in fact be summarised according to one of the following general forms [6],

$$\boldsymbol{I}_W = \boldsymbol{I} + a\boldsymbol{W} \tag{1}$$

$$\boldsymbol{I}_W = \boldsymbol{I}(\mathbf{1} + a\boldsymbol{W}), \quad for\ \boldsymbol{I} \neq 0 \tag{2}$$

$$\boldsymbol{I}_W = \boldsymbol{I}(e^{a\boldsymbol{W}}), \quad for\ \boldsymbol{I} \neq 0 \tag{3}$$

where $\boldsymbol{I}_W$ refers to the watermarked image, $\boldsymbol{I}$ refers to the original host image, $\boldsymbol{W}$ refers to the watermark sequence and a refers to a scaling factor used in varying the strength of the watermark's presence within the image.

More importantly, equations 1-3 should be understood as illustrating the mere relationship between the respective parameters, $\boldsymbol{I}_W, \boldsymbol{I}$ and $\boldsymbol{W}$, and that there is in fact no implicit

restriction imposed on these parameters with regards to the domain in which they may be used in, i.e. the spatial domain or frequency domain.

2.2.2 Spatial vs. Frequency Domain

In spatial embedding, the watermark is embedded directly in to the pixel space of the image.

In one of the earliest ground-breaking techniques, spatial embedding could be achieved by simply replacing bits from a bit plane of the image, with that of the watermark bits. To demonstrate the general effectiveness of this technique, figure 2 illustrates the 1,3,6,8 bit planes of the Lena image, as well as the resulting watermarked versions that can be created by replacing each of these bit planes with a similar-sized binary watermark image. For reference, the Lena image and binary watermark are also shown in figure 3.

In figure 2, the 1,3,6,8 bit planes are shown on the top row, from left to right, whilst the bottom row shows the corresponding watermarked images, created in each case by replacing one of the respective bit planes with the binary watermark image. By visually inspecting the change in quality between each watermarked copy, it is clear to see that for high imperceptibility, the watermark must be embedded within the lower bit planes of the image. For this reason, schemes which adopt this technique, or a variation of it, are commonly referred to as least significant bit (LSB) schemes [6].

As further noted in [6], most schemes using the LSB technique tend suffer from a crucial flaw in that it is extremely easy for an attacker to destroy the embedded watermark through lowpass filtering or by randomly flipping bits within the lower bit planes of the image.

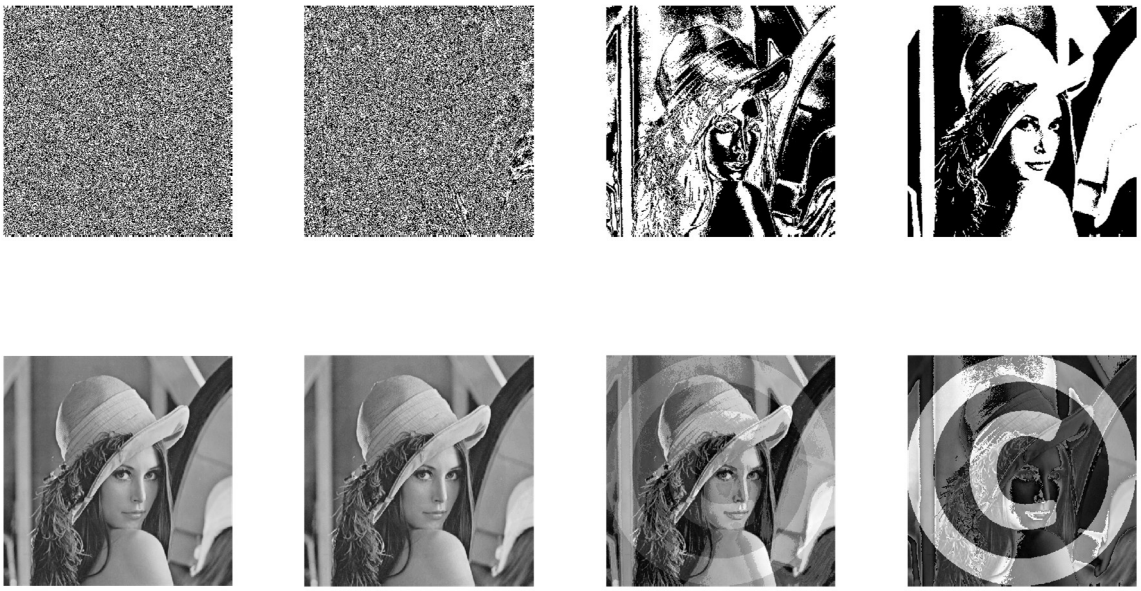


Figure 2: Bit planes 1,3,6,8 of the Lena image (top row) and corresponding LSB watermarked images (bottom row).

Figure 3: Lena image (left) and binary watermark image (right)

In contrast to spatial embedding, frequency embedding involves the process of embedding the watermark into the frequency components of an image, rather than directly into the pixel space.

In this technique, an image is first transformed into its frequency representation via a suitable frequency transformation, such the Fourier Transform, Discrete Cosine Transform, Wavelet Transform [1] etc. Watermark embedding is then achieved by modifying the frequency coefficients according to some mathematical rule or function. For example, the frequency interpretation of equations 1-3 may be as follows,

$$F(\boldsymbol{I}_W) = F(\boldsymbol{I}) + a\boldsymbol{W} \tag{4}$$

$$F(\boldsymbol{I}_W) = F(\boldsymbol{I})(\mathbf{1} + a\boldsymbol{W}) \tag{5}$$

$$F(\boldsymbol{I}_W) = F(\boldsymbol{I})\left(e^{a\boldsymbol{W}}\right) \tag{6}$$

where $F(\,.\,)$ denotes the frequency transformation used and $\boldsymbol{I}$, $\boldsymbol{I}_W$ denote the spatial domains of the host and watermarked image, respectively.

Finally, the watermarked image can then be obtained by applying the inverse transformation as

$$\boldsymbol{I}_W = F^{-1}(F(\boldsymbol{I}_W)) \tag{7}$$

In general, embedding within the frequency domain offers greater performance with regards to robustness, imperceptibility and payload capacity, than in the case of spatial embedding. However, one major limitation to this technique is that it also requires tremendous memory resources and computational power in its implementation. For this reason, it often the case that frequency-based schemes will adopt a forefront segmentation process, whereby the host

[1] In this context, it is implied that the 2-dimensional forms of the respective frequency transformations are used.

image is initially segmented into smaller portions and the embedding function applied to each portion separately.

In fact, this approach has been widely adopted within a large range of image processing applications as it not only lessens the memory and computational burdens of the overall embedding process but also allows the image to be processed in parallel [7]. However, this suboptimal approach does come with its drawbacks, which will typically manifest itself in the form of blocking artefacts within the rendered image. An example of this blocking effect may be seen in figure 9, which demonstrates the Lena image following JPEG compression under various qualities. It can be noted that this compression scheme uses a similar segmentation process to that of a vast majority of frequency-based watermarking schemes whereby the image is initially segmented into non-overlapping blocks of size 8x8 pixels.

2.2.3 Embedding within the DCT Domain

The discrete cosine transformation (DCT) is perhaps one of the most widely used transformations within the general field of image processing. Its use can be realized most notably within compression applications, in which schemes such as JPEG (image compression), MPEG (video compression) and MP3 (Audio compression) rely on the DCT's superior energy compaction characteristics to efficiently quantize redundant frequencies within a signal during compression. In image watermarking, the DCT has proven itself to be an effective frequency transform for yielding both high robustness and fidelity in watermarked images. The following section highlights the mathematical formulae related to the DCT and how it can be used within watermarking schemes, such as those covered over the course of this project.

For an image of size $M \times N$ and pixels, $i_{j,k}$, the DCT transformation coefficients, $f_{m,n}$, are given by:

$$f_{m,n} = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} i_{j,k}\, c_{j,m}\, c_{k,n}\ , \qquad for\ m,n = 0, \dots, N-1, \tag{8}$$

where,

$$c_{j,m} = \sqrt{\frac{1}{N}}, \qquad for\ m = 0, \tag{9}$$

$$c_{j,m} = \sqrt{\frac{2}{N}} \cos\left(\frac{\pi m}{2N} (2j+1) \right), \qquad for\ m > 0 \tag{10}$$

and the inverse transformation for reconstructing the image given by

$$i_{j,k} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n}\, c_{j,m}\, c_{k,n} \ , \quad for\ j,k = 0, \dots, N-1. \qquad (11)$$

Moreover, it is possible to write the DCT expressions in matrix form,

$$\boldsymbol{F} = \boldsymbol{CIC}' \qquad (12)$$

$$\boldsymbol{I} = \boldsymbol{C}'\boldsymbol{FC} \qquad (13)$$

where $\boldsymbol{F}$ refers to the DCT coefficient matrix, $\boldsymbol{I}$ refers to the spatial image, and $\boldsymbol{C}$ refers to the cosine matrix formed by expanding equations 9 and 10 across the image size.

As previously mentioned, many watermarking schemes implement the DCT independently within smaller segments of the image, as opposed to the entire image. In this project, a similar approach is used whereby the DCT is applied to segments of size 8 x 8 pixels, i.e. $M, N = 8$ for the above equations. In circumstances such as this, where the segment size is constant throughout the embedding process, equations 12 and 13 illustrate how the cosine matrix, $\boldsymbol{C}$, need only be computed and stored once throughout the entire embedding procedure. This effectively provides a more efficient method than if equations 8 and 11 were to be implemented for each watermarked segment. The following steps demonstrate a practical example of how a typical watermarking scheme might embed a watermark within the DCT domain of a 4 x 4 image segment:

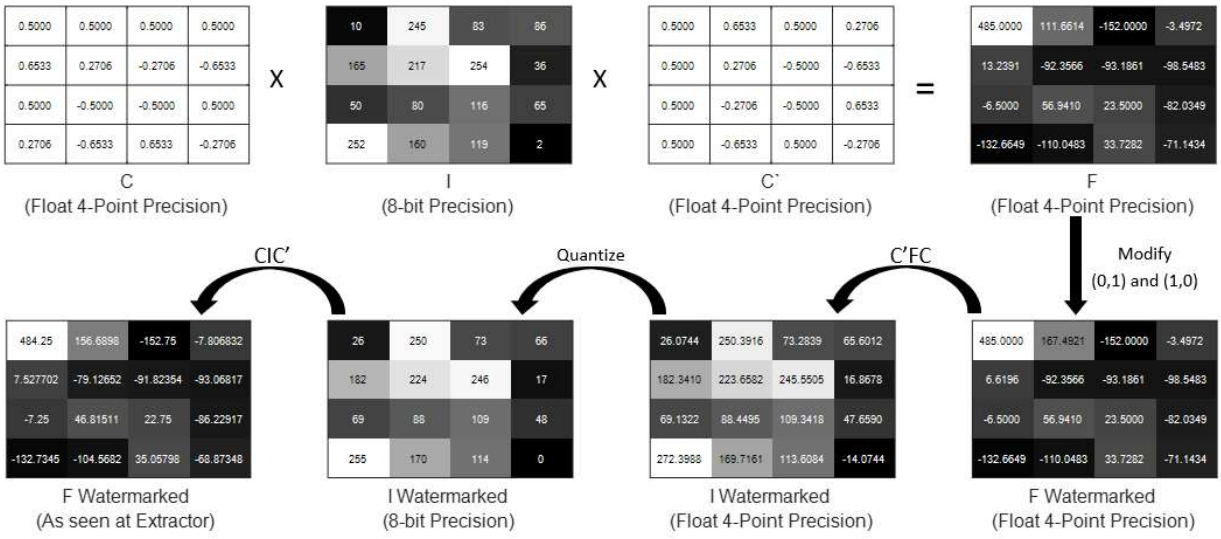


Figure 4: Example of watermark embedding within DCT of a 4 x 4 image segment

The top half of figure 4 shows how the DCT matrix, $\boldsymbol{F}$, is computed from an image segment, $\boldsymbol{I}$, and corresponding cosine matrices, $\boldsymbol{C}$, $\boldsymbol{C}'$, using equation 12. Within the DCT matrix, a two-bit watermark, consisting of $\boldsymbol{W} = [1, -1]$, is to be embedded. In this example, the embedding function is represented by equation 5 and the embedding strength, a, is taken to be 0.5. Furthermore, the embedding locations for this scheme have been chosen to be the (0,1) and

(1,0) elements of $\boldsymbol{F}$, respectively. That is, the (0,1) element of $\boldsymbol{F}$ will embed the watermark bit 1, and the (1,0) element of $\boldsymbol{F}$ will embed the watermark bit -1. This embedding order is particularly important whenever more than 1 bit is to be embedded within a single image segment and will have to be known at the extractor stage in order to properly reconstruct the watermark sequence.

The resulting watermarked DCT matrix can be seen in the bottom right corner of figure 4, after which the inverse DCT transformation from equation 13 is used to bring the image back in to the spatial domain. An important observation that can be noticed at this point is that the inverse DCT will typically result in a spatial image that requires double or float precision. Therefore, it is important that the image be re-quantized back to its original 8-bit format before being sent to the recipient. This re-quantization will effectively present itself as noise being added to the watermarked image and will be inherent to most if not all schemes operating within the frequency domain. To visualize the effect of this noise, the bottom left image of figure 4 shows the resultant DCT matrix (as seen at the extractor stage) following quantization of the watermarked image. By comparing this to the original DCT matrix (bottom right corner of figure 4) , it can be seen that the quantization has resulted in some minor changes to each element within the matrix. Most importantly however, a visual comparison between the original and watermarked images, within the spatial domain, illustrates two important characteristics associated with frequency-based embedding:

1. The embedding procedure modified only two elements within the DCT domain, however, each pixel element has subsequently been changed (by varying extent) within the spatial representation.
2. The (0,1) element within the DCT domain has been altered by ~ 45 (post quantization) units yet the watermarked image still appears perceptually similar to the original image.

Three final notes regarding the DCT may be made:

1. The (0,0) element of the DCT is commonly referred to as the DC term. This comes directly from the underlying equation, in which it can be seen that the DC term is essentially proportional to the average intensity values of the image segment, i.e.

$$DC = \boldsymbol{F}(0,0) = \frac{1}{8}\sum_{i=0}^{7}\sum_{j=0}^{7}\boldsymbol{I}(i,j), \qquad for\ an\ 8\ x\ 8\ image \tag{14}$$

2. The remaining elements of the DCT are commonly referred to as the AC terms, where the frequency of each term increases diagonally in going from the top left corner of the DCT matrix down to the bottom right corner.

3. It is common for image processing techniques to create a single vector of the 2-dimansional DCT by scanning across the matrix in a diagonal 'zig-zag' pattern as shown in figure 5 below.

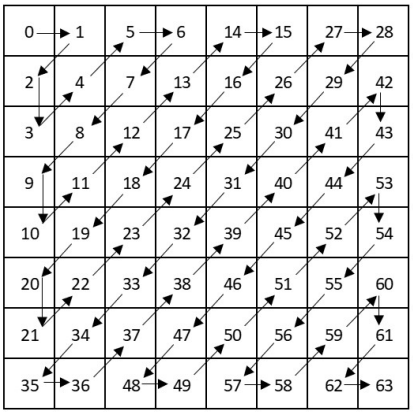


Figure 5: Zig-Zag pattern used to vectorise DCT matrix.

2.2.4 The Watermark

The watermark forms an essential aspect of the overall system and in general, should reflect the intended application of the watermarking scheme. In particular, for copyright applications the watermark must be chosen to convey information which is uniquely relatable to the recipient of that watermarked copy.

In this context, it is possible to distinguish between two types of watermarks [8]. In the first case, the watermark is chosen to contain information which is directly addressable to the recipient, such as their ID or logo whilst in the second case, the watermark is generated as a unique pseudo random sequence drawn from a probability distribution such as a gaussian or binomial distribution etc.

An important distinction can be made between both cases by considering that, in the former case, the extracted watermark can be identified subjectively through visual inspection whilst in the latter case, the original watermark must be known in order to identify the presence of the extracted watermark, i.e. following some type of correlation analysis. Furthermore, this distinction raises a possible security flaw in the overall system, in that, if an attacker were able to knowingly extract the embedded watermark, they could potentially frame the honest recipient by embedding that same watermark within another piece of work before leaking it [9].

Interestingly, it appears questionable as to why not all schemes adopt the more secure pseudo random watermark. In fact, according to [4], one possible explanation for this could be due to the extra cost associated with this option. That is, in order for a randomly-generated watermark to uphold any validity during a legal dispute, it is necessary that the watermark first be registered with a trusted third party entity (for a fee of course) before being embedded within the respective content. In this

case, a third party entity, such as the United States Copyright Office, would be able to provide convincing evidence in a court of law to prove that the extracted watermark is in fact genuine, even if its contents appears to be random. In the case of choosing a direct watermark, however, it is possible in some circumstances for the original owner to provide sufficient validity in their claim without the extra security of a third party. This is particularly true if the original owner has embedded the watermark using their own privately known embedding function and can further demonstrate the extraction process on the leaked material.

In fact, it is possible to improve the security of a direct watermark in the first instance by encrypting its contents before it is embedded. In this process, the watermark can be encrypted at the embedder (and decrypted at the extractor) using a chaotic mapping sequence, such as the Arnold cat map, Skew Tent map, PWLCM map etc. To demonstrate the effectiveness of this approach, figure 6 illustrates an original binary watermark (leftmost image) and its encrypted version (rightmost image) following 39462 iterations of the Arnold cat map. From this figure it can be seen that the encrypted output demonstrates good noise-like properties and would, in theory, require the attacker to test across an average of $(2^N-1)/2$ different secret keys before being able to visually determine the original watermark, where N equals the number of bits in the watermark (1024 in this case) and the secret key referring to the number of iterations used during the encryption/decryption process.

2.2.5 Properties of the Embedding Stage

Figure 6: Original watermark (left), scrambled watermark using Arnold chaotic map with 39462 iterations (right).

According to [4], a generalized embedding stage may be summarised according to the following essential properties:

2.2.5.1 Fidelity and Quality

When a watermark is embedded within digital content, the resulting content is altered in some shape or form. In this context, two subtly different definitions may be used to describe the perceptual degradation experienced by the watermarked content. The first, known as fidelity, refers to the perceptual difference that can be noticed when comparing the watermarked material to the original material, where it can be understood that a high fidelity implies little to no noticeable difference between both materials and a low fidelity implies that there is clear dissimilarity between both materials. In the second definition, known as quality, there is no comparison made between the watermarked and original copies but instead, the term quality is simply used to describe the perceptual appeal of the watermarked content.

In many practical applications, high fidelity and quality are simultaneously required and a distinction between both definitions may be challenging. On the hand, there are some circumstances where it may be possible to exploit one for the other. For example, in a scenario where the quality of the rendered content is inherently poor, such as during a noisy broad cast, it may in fact be advantageous to lessen the constraint on high fidelity in favour of increasing the watermark's strength within the content, and therefore its overall robustness, if it known that the quality that the user will experience will remain practically unchanged [4].

2.2.5.2 Embedding Effectiveness

In designing a watermarking system, it is generally the case that the embedding stage must be able to adaptively adjust its embedding parameters (i.e. embedding strength, embedding locations etc.) for each piece of input material so that an overall constraint is met within the watermarked output. In [4], an example is given in which a watermarking system may essentially fail to embed a watermark into a piece of content if the embedding stage is unable to adjust its parameters to achieve a fixed constraint set on the fidelity of the output content. In this context, embedding effectiveness refers to the probability that an embedding stage will embed a watermark that can in fact be extracted afterwards. This fact can be further realized by considering how a frequency-based scheme will always have to embed above a certain strength in order to ensure that the subsequent quantization noise will not destroy the watermark.

2.2.5.3 Data Payload

In still image applications, data payload simply refers the number of bits encoded by the watermark within a single image. Thus, if a watermark is of size N bits, the data payload will also be N bits. Related to this, it also important to note that a watermark comprising of N bits may be generated from a total of 2^N possible combinations.

2.3 Channel Stage

In communications theory, a communication channel is typically used as means for modelling the impact of noise on the information content of a signal during its transmission from one point to another. In a similar manner, the following project adopts a communication channel into its design as a means of modelling the effects of noise on the watermarked content. However, in contrast to the conventional model, a subtle difference may be drawn in that the proposed model will not only consider conventional channel noise effects but will also assume that the channel is open to a possible snoop attack, whereby an adversary may be able to capture the watermarked content during its transmission from the sender to the recipient. Under this assumption, it is further assumed that the adversary may decide to leak or pirate this information to the general public but will first attempt to foil the underlying copyright protection by destroying the watermark using additional signal processing, geometric or compression attacks.

In practice, this model may not be entirely sufficient as any threats from a snoop attack could in theory, be mitigated by encrypting the content before it is transmitted over the channel. However, even if this is the case, a possible threat still remains in that the rightful recipient themselves may try to pirate the watermarked content that they have just decrypted. Thus, an important aspect that may be realized, is that encryption can no longer guarantee

copyright protection once the content has been decrypted by the receiving party. On other hand however, a watermark forms a permanent bond with the marked content and may provide the necessary protection where encryption cannot [4].

The following section provides a brief overview into some of the common noise and signal attacks which a watermarking scheme for copyright applications is expected to be robust against.

2.3.1 Average Filter

In average filtering, the image is smoothed by replacing each pixel value with the mean value of itself and its surrounding neighbours. In a practical implementation, this is achieved by passing a kernel across each pixel within the image, where the kernel can be though of as a window centred around the current pixel and specifies which pixels are included within the neighbourhood (inside the kernel) and which are not (outside the kernel).

To demonstrate this idea, figure 7 illustrates how a kernel of size 3 x 3 is used to process two consecutive pixels within an image. In this figure, the current pixel is shown in blue, its neighbours are shown in green and the remaining pixels are shown in grey. In going from left to right, top to bottom, the kernel computes the average intensity value at each location and stores the result for that location within a separate output image. For boundary locations the kernel may either extrapolate values or assign a constant value to the outside pixels (in MATLAB the kernel will assign default values of 0 outside the image).

In MATLAB, an average filter of size 3 x 3 may be implemented for an image, I, as follows:

```
imfilter(I,ones(3,3)/9)
```

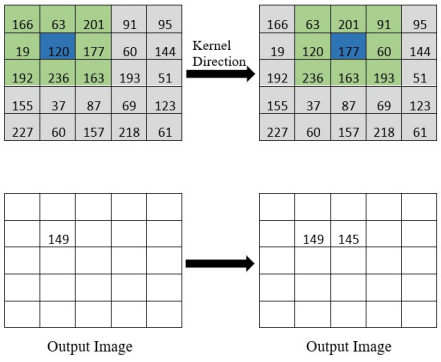


Figure 7: Average filter using the kernel concept.

2.3.2 Median Filter

In median filtering, a similar kernel process is used to update each pixel value within the image. However, in contrast to average filtering, median filtering involves updating each pixel according to the median value which appears within the kernel at each location. In MATLAB, a median filter may be implemented on an image, I, as follows:

```
medfilt2(I)
```

2.3.3 Wiener Filter

In image processing, a Wiener filter is commonly used to reduce the effects of additive noise or blurriness within an image. In this process, the Wiener filter attempts to statistically remove the observed noise within a signal by assuming that the noise and 'true' underlying signal can be modelled as stationary processes and that the spectral properties of each signal are known a priori. Under these assumptions, the Wiener filter then attempts to reconstruct the true signal by minimizing the Square Mean Error between the known signal and the reconstructed estimate. However, the Wiener filter may also be used as a source of noise by applying it adaptively within local regions of an image. In this case, the end result may be considered similar to a low-pass smoothing effect. In MATLAB an adaptive Wiener filter of size 3 x 3 can be applied to an image, I, as follows:

```
wiener2(I,[3,3])
```

2.3.4 Gaussian Low Pass Filter

A Gaussian low pass filter may be thought of as being closely identical to the averaging filter with the exception that each pixel within the kernel is initially weighted according to the 2-dimensional Gaussian distribution before being used in the averaging calculation. In this case, it should be understood that the current pixel will have the highest weighting and that this weighting will gradually decreases with increasing distance from the kernel centre. In MATLAB, a 3 x 3 gaussian filter can be applied to an image, I, as follows:

```
imfilter(I,fspecial('gaussian',[3 3]),'same')
```

2.3.5 Additive Gaussian White Noise

Additive Gaussian White Noise (AGWN) is commonly used in signal processing to model the noise which occurs in a natural environment. Characteristically, AWGN is modelled using a Gaussian distribution with mean 0 within the spatial domain and a uniform power distribution within the frequency domain. In MATLAB, AWGN with a variance of 0.001 and mean of 0, may be added to an image, I, as follows:

```
imnoise(I,'gaussian',0,0.001)
```

2.3.6 Salt-and-Pepper Noise

Salt-and-Pepper noise is an impulsive form of noise which manifests itself throughout the image as sharp spikes with either high or low intensity values. This type of noise is considered to be independent of the image content and will generally spread itself in a coarse fashion throughout the image. In MATLAB, Salt-and-Pepper noise with a 'strength' of 0.01, may be generated on an image, I, as follows:

```
imnoise(I,'salt & pepper',0.01)
```

2.3.7 Poisson Noise

In image processing, Poisson noise is a type of noise which is commonly associated with the variation of photons which pass through an image sensor during a given exposure period [10]. In MATLAB, this type of noise may be generated in an image, I, using the following command:

```
imnoise(I,'poisson')
```

2.3.8 Speckle Noise

In image processing, speckle noise is a type of noise which is commonly associated with ultrasound imaging such as that used in medical applications [11]. In this type of imaging, a transducer generates an ultrasonic waveform which travels through a body. Whenever a density change occurs within the body, this waveform is reflected back towards the transducer where it can be transformed into electrical pulses and processed into an image. In this context, speckle noise is thought to result due to interference of the reflected waveform as it travels back towards the transducer. This is often due to an improper contact between the transducer probe and the body, where an air gap in between will result in another density change at the transducer and in return cause part of the waveform to reflect and interfere with itself. In MATLAB, speckle noise of 'strength' 0.003 may be generated on an image, I, using the following command:

```
imnoise(I,'speckle',0.003)
```

2.3.9 Image Cropping

Image cropping is type of geometrical attack whereby an attacker will crop and remove whole parts of the image in the hope that the watermark has been embedded within those parts. In MATLAB, a cropping attack can be performed by setting the regions to be cropped within the image to a constant value such as 0 or 255, i.e. the following command can be used to crop the first 256 columns of the image, I.

```
I(1:end,1:256) = 0;
```

2.3.10 Affine Transformation

An affine transformation is generally defined as a transformation which preserves collinearity and ratios between points. In a practical sense, this includes transformations such as scaling, rotation, translation, shear etc. as well as any combination of these. In this project, rotation will be the main affine transformation type to be examined and can be implemented on an image, I, in MATLAB using the following command,

```
imrotate(I, 70);
```

where, in this case the image is rotated clockwise by 70°.

2.3.11 JPEG Compression

JPEG refers to an image compression standard which was first introduced in 1992. In this standard, a lossy compression scheme is defined based on the quantization of DCT coefficients taken from 8 x 8 pixel-sized blocks of the input image (within the YUV colour space). The main steps involved with the JPEG compression scheme may be summarised as shown in figure 8 below:

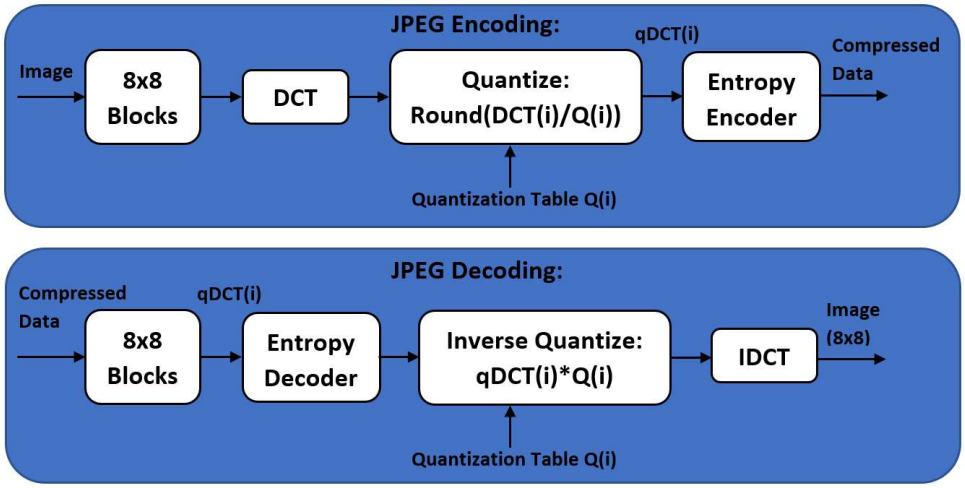


Figure 8: Outline of the JPEG Compression Scheme.

An essential drawback of the widely-popular JPEG standard is that the standard allows only one quantization table to be used across all blocks of the input image. However, it does allow for this table to be uniquely defined by the user and further allows its values to be extrapolated across different 'quality' scales (0-100) to produce a better compression ratio, at the expense of visual quality. By default, the standard provides a baseline quantization table (at quality = 50) for both the luminance channel and chrominance channels of the input image. For reference, figure 9 illustrates the Lena image following JPEG compression under qualities 5, 15 and 50. In MATLAB, an image, I, may be compressed using JPEG with quality of 50 as follows:

```
imwrite(I,'JPEG.jpg','jpg','Quality',50);
```

Figure 9: Lena image under varying JPEG qualities, where left leftmost image has quality 5, centre image has quality 15 and rightmost image has quality 50.

2.3.12 JPEG2000

JPEG2000 is another lossy compression scheme which was first introduced in 2000 and aimed to build upon the limitations of the original JPEG standard as well as offering a better trade-off between perceptual quality and compression ratio. JPEG2000 differs primarily from the original JPEG standard in that it uses the wavelet domain rather than the DCT domain to achieve its quantization. To demonstrate the overall difference between both schemes, figure 10 shows the Lena image under JPEG compression with quality 5 (left image) and JPEG2000 using a compression ratio of 45 (right image), where the overall file sizes of both images are almost identical. By comparing both images it can be seen that the JPEG2000 version has maintained much better quality than the JPEG version. The JPEG2000 version also appears to show little no blocking artefacts, however ringing noise can be seen around the edge regions of the image.

In MATLAB, an image, I, may be compressed using JPEG2000 with a compression ratio of 10 using the following command:

```
imwrite(I,'output.jp2','jp2','CompressionRatio',10);
```

Figure 10: Comparison between JPEG and JPEG2000.

2.4 Watermark Extraction

2.4.1 A General Extraction Function

Similar to the embedding process, a watermark may be extracted from a watermarked content in a mathematical manner by applying the inverse embedding function on that content. For example, equations 15-17 show the inverted equivalents of equations 1-3, where as before, there is no assumption made regarding the domain to which these equations may be applied within, so long as it is consistent with the domain used during the embedding stage.

$$\boldsymbol{W} = \frac{\boldsymbol{I}_W - \boldsymbol{I}}{a} \tag{15}$$

$$\boldsymbol{W} = \frac{\boldsymbol{I}_W - \boldsymbol{I}}{a\boldsymbol{I}}, \qquad for\ \boldsymbol{I} \neq 0 \tag{16}$$

$$\boldsymbol{W} = \frac{1}{a}\ln\left(\frac{\boldsymbol{I}_W}{\boldsymbol{I}}\right), \qquad for\ \boldsymbol{I} \neq 0 \tag{17}$$

2.4.2 Extraction within the DCT domain

The following section demonstrates how a typical watermarking system may extract a watermark by further considering the previous embedding example discussed in section 2.2.3. From this example, the DCT domains of the unwatermarked image, $\boldsymbol{F}$, and the watermarked image after quantization, $\boldsymbol{F}_W$, are shown in figure 11, respectively.

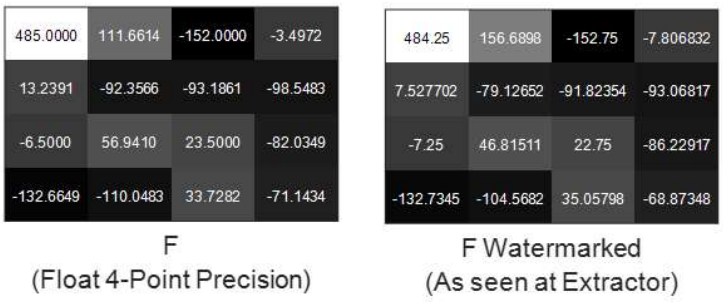


Figure 11 DCT of unwatermarked image on the left and DCT of watermarked image (post quantization) on the right.

Furthermore, the embedding function for this example was adapted from equation 2 and can be given in its frequency form as

$$F(\boldsymbol{I}_W) = F(\boldsymbol{I})(\boldsymbol{1} + a\boldsymbol{W}) \tag{5}$$

To determine the watermark, $\boldsymbol{W}$, the corresponding inverted form will be similar to equation 16 and can be given in its frequency form as,

$$\boldsymbol{W} = \frac{F(\boldsymbol{I}_W) - F(\boldsymbol{I})}{aF(\boldsymbol{I})}\,, \tag{18}$$

where $F(\boldsymbol{I}_W)$ corresponds to $\boldsymbol{F}_W$, and $F(\boldsymbol{I})$ corresponds to $\boldsymbol{F}$.

By considering equation 18, it can be seen that the extractor requires additional information regarding each of the $F(\boldsymbol{I}_W), F(\boldsymbol{I})$ and a terms in order to be able to solve for the watermark bits. In fact, the amount information which the extractor requires to solve for the watermark bits depends highly on the assumptions that can be made at this stage and will generally impact the usefulness of the overall watermarking system with respect to certain applications. This aspect is discussed in greater detail in section 2.4.4.1. However, for the purpose of this example it will be assumed that the extraction stage has been given any necessary information to be able to solve for the watermark bits.

In this case, the watermark bits can be extracted as follows,

$$\boldsymbol{W}_E(1) = \frac{\boldsymbol{F}_W(0,1) - \boldsymbol{F}(0,1)}{a\boldsymbol{F}(0,1)} = \frac{156.6898 - 111.6614}{(0.5)(111.6614)} = \ 0.8065$$

$$\boldsymbol{W}_E(2) = \frac{\boldsymbol{F}_W(1,0) - \boldsymbol{F}(1,0)}{a\boldsymbol{F}(1,0)} = \frac{7.5277 - 13.2391}{(0.5)(13.2391)} = -0.8628$$

where the extraction information that is required includes, $\boldsymbol{F}_W(0,1)$, $\boldsymbol{F}_W(1,0)$, $\boldsymbol{F}(0,1)$, $\boldsymbol{F}(1,0), a$, and the embedding order. Additionally, it can be observed that the quantization process alone has resulted in a discrepancy of roughly 14%-20% in the extracted values.

2.4.3 Detection within the DCT Domain

Watermark detection follows from the extraction stage and involves the process of determining whether an extracted watermark is in fact valid or not. To determine this, most detection stages perform a correlation-based analysis between the extracted watermark and the original watermark in order to determine an objective value that is indicative of the similarity between both watermarks. The following project considers a correlation method used in [5] and is referred to by the authors as the Normalized Cross-Correlation (NC), given by

$$NC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}_o(i,j)\boldsymbol{W}_E(i,j)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}_o(i,j)^2 \sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}_E(i,j)^2}} \tag{19}$$

where M, N refer to the dimensions of the watermarks, $\boldsymbol{W}_o$ refers to the original watermark, $\boldsymbol{W}_E$ refers to the extracted watermark.

This NC expression will yield a maximum value of 1.0 whenever both watermarks are completely similar, a value of centred around 0.5 whenever both watermarks are truly

dissimilar (random) to one another and a value of 0.0 if one watermark happens to be the exact inverted form of the other watermark.

Furthermore, in the previous section the extracted watermark was found to be

$\boldsymbol{W}_E = [0.8065, -0.8628]$**,**

whereas the original embedded watermark that was given by

$$\boldsymbol{W}_o = [1, -1].$$

At this point, 2 important distinctions can be made:

1. The NC value may be computed directly on $\boldsymbol{W}_E$ and $\boldsymbol{W}_o$ to give $NC = 0.9994$. In this case, the extracted watermark has been allowed to contain floating-point elements, as opposed to the fundamental notion that it must contain only binary numbers (i.e. a single bit per element). In fact, treating the watermark in this nature does not violate the underlying functions used during the embedding and extraction stages of the system and the technique of both generating and extracting the watermark as a continuous variable and has previously been exploited in [6] to generate a watermark using a Gaussian distribution, which due to having the highest entropy among all other distributions, is considered to be optimal in terms of its use in a spread spectrum embedding scheme [12].
2. If the watermark is assumed to be binary in nature, the extractor could exploit prior knowledge regarding the distribution that was used in generating the watermark in order to modify and adjust the float-point values that have been extracted. For instance, if the extractor knows that the watermark will contain only values of ± 1, the extractor could simply round each extracted value to the nearest ± 1 and in this case obtain a perfect NC score of 1.

In addition to the NC value, some watermarking systems may also implement the Bit Error Rate (BER) as an alternative or complimentary measure for detecting the watermark, where the BER is given by

$$BER = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}_o(i,j) \oplus \boldsymbol{W}_E(i,j)}{MN} \tag{20}$$

and $\oplus$ refers to the exclusive OR operator.

Here the BER will return a value of 0 if the extracted watermark is exactly the same as the original, a value centred around 0.5 if both watermarks are truly dissimilar and a value of 1 if one watermarks is the inverted form of the other.

2.4.4 Properties of the Extraction Stage:

According to [4], a generalized embedded stage may be summarised according to the following essential properties:

2.4.4.1 Blind or Informed Detection

Conceptually, an extraction stage may be classified according to whether or not it requires any additional side information related to the original content in order to extract the watermark. In this context, an extraction process is said to perform blind detection if it does not require any additional side information whereas an informed extraction requires either the original unwatermarked content or some information related to that content (i.e. embedding locations, decryption key etc.).

Moreover, informed detection will generally enhance the effectiveness of the extraction stage however, the drawback to this method is that is requires the additional information to both available as well as to be stored within the memory of the extraction stage. This may be feasible in a scenario where an original content owner may want to check against a leaked copy of their work and would therefore have all the necessary side information at hand. However, in a large-scale scenario where the extraction stage may be required to process across a wide range of unpredictable content (i.e. extraction device embedded within a DVD player), informed detection may no longer become feasible. For this reason, it also common to consider a watermarking system which performs informed detection as a private watermarking system, and those which perform blind detection as public watermarking systems.

2.4.4.2 Robustness

Robustness refers to a watermarking system's ability to withstand any distortions which may be imposed on the watermarked content, so that a scheme which is considered to be robust is capable of experiencing a high degree of degradation of the watermarked content before the watermark can no longer be identified within the content [6]. This typically includes distortions arising from channel noise (i.e. Gaussian, salt and pepper etc), common signal processing attacks (i.e. low pass filtering, median filtering etc), geometrical attacks (i.e. rotation, scaling, cropping), and lossy compression schemes.

From an applications point-of-view, watermarking schemes which are to be used in copyright and owner authentication scenarios must exhibit a high degree of robustness to prevent any attackers from gaining access to potentially sensitive information (authentication system) or from distributing copyrighted material without the possibility of the owner being able to identify them as the leaked source. On the other hand, some applications of digital watermarking, such as content integrity, rely on the watermarking scheme to fail at the early onset of any modifications, so as to indicate that the corresponding content is no longer authentic. Schemes of this nature are commonly referred to as fragile watermarking systems.

2.4.4.3 False Positive Rate

The false positive rate measures the frequency of occurrence of the detector incorrectly detecting a watermark within an image, when in fact that watermark is not present within the image. Similarly, the false negative rate measures the frequency of occurrence of detector failing to detect a watermark in an image, when in fact that watermark is present within the image. Furthermore, according to [9], it is possible to define two subtle variations of this definition depending on whether the watermark or the image are considered to a random variable of interest:

In the first instance, known as the random-watermark false positive probability, the image is assumed to be constant for all detector runs, whilst the watermark is treated as a random variable drawn from a representative distribution (i.e. Gaussian distribution, uniform distribution, binomial distribution etc.) at each run. Additionally [9], claims that under these conditions, the false positive rate will, in most cases, be independent of the input image and will depend solely on the distribution used to generate the watermark at each run. This definition of the false positive rate is most appropriate when considering authentication or transaction applications such as in a fingerprint system, where a falsely detected watermark can lead to a false authorization of sensitive information [13].

In the second instance, known as the random-Work false positive probability, the watermark is kept constant throughout all runs of the detector, whilst the image is randomly selected at each run from a distribution of content which appropriately reflects the nature of the application (i.e. a database of natural images, aerial images, people etc.). This definition of the false positive rate is considered more applicable than the first and is typically adopted in public copyright applications, where a large volume of digital content must be examined for their corresponding copyright permissions.

The false positive and false negative rates of a given watermarking system are essential characteristics which allow the overall reliability of the system to be measured. This is particularly important within a legal context wherein the watermarking system may be used as evidence in a copyright infringement scenario.

3 Perceptual Modelling

3.1 Overview

In chapter 2, a general overview of a watermarking system was introduced where the primary concern was to identify and demonstrate the basic mechanisms underpinning the embedding stage and extraction stage of a DCT-based watermarking system as well as providing a brief overview into some of the noise attacks which such a system may be subjected to. The following chapter aims to build upon this general framework by presenting an essential aspect required of any robust watermarking system, namely, the imperceptibility of the watermark's presence as it is perceived by the content user.

Designing imperceptibility into a watermarking system is often a difficult and complex task. For the main reason this is because imperceptibility requires a deep understanding into the Human Visual System (HVS) and how humans perceive noise within an image under various conditions. Over the years, numerous techniques have been proposed in an effort to determine a mathematical framework for modelling this phenomenon. These models are often referred to as HVS models and are typically based on psychophysical experiments in which human interaction is employed to subjectively score the impact of noise under various different stimuli and viewing conditions in an effort to determine the degree of noise required before a difference can be detected between a distorted image and its original.

Generally, the minimum noise threshold which results in over 75% of the participants detecting a change in distortion is referred to as the Just Noticeable Difference (JND) and can be used to give a direct estimate of the noise tolerance of a signal. More importantly, a psychovisual experiment of this nature may be performed in both the spatial domain [14, 15] and frequency domain [16, 17], where the former leads to a pixel-wise model and the latter leads to a subband model [18]. Although a pixel-wise model may offer a more direct and simplistic view of the of the noise tolerance throughout an image, the subband alternative is widely considered to be more realistic and better suited towards the true nature of the HVS. In fact, a subband model can be used to determine a unique JND for each frequency component of a signal and has previously formed the basis for determining the baseline quantization table used within the DCT domain of the JPEG standard. For reference, a graph of this table is shown in figure 12, in which it can be seen that the underlying experiment for its construction has found the HVS to be most sensitive towards the low to mid-range frequencies and highly insensitive to the high-range frequencies.

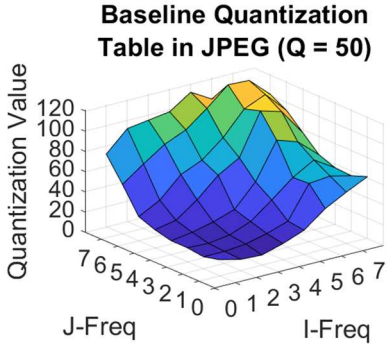


Figure 12: Graph of the JPEG baseline quantization table

As an HVS model can be used to estimate the noise tolerance of an image, or image segment, this would in turn suggest that it can also be used during the embedding stage to determine an optimal trade-off between embedding strength and imperceptibility of the resultant watermark. However, it also important to be able to measure imperceptibility in terms of the overall loss of quality (or fidelity) that the image as a whole has experienced following the addition of noise, or in this case the watermark. Techniques that allows this to be done are commonly referred to as Image Quality Assessment (IQA) metrics and will typically allow a single objective measurement to be made in relation to the total degradation that the image has experienced. In addition to this, it was previously mentioned in section 2.2.5.1 that image degradation may be viewed in two subtly different manners, fidelity and quality. Where the main distinction is that fidelity measurements require the original image to be compared against the watermarked image while quality measurements do not. In a similar manner, IQA metrics can be classified as being either full-reference metrics or no-reference metrics, where the former case requires both the original image and watermarked image and the latter case requires only the watermarked image. Conceptually, a good watermarking system will take both metric types into account. However, it is possible to argue that a watermarked image which has high fidelity must also have high quality (but not necessarily true in the reverse case). For this reason, many watermarking systems will tend to only consider the case of full-reference IQA's in their designs.

3.2 Human Visual System Models

A basic HVS model will typically incorporate three fundamental phenomena related to the visual system: spatial frequency sensitivity, luminance adaption and a contrast masking:

3.2.1 Spatial Frequency Sensitivity (CSF)

In the HVS model, spatial frequency refers to the way in which humans perceive distortion as a function of luminance contrast within an image. In a most basic experiment, this type of visual response can be subjectively estimated by generating sinusoidal gratings with varying amplitudes on a screen and observing how the minimum detection threshold varies across different spatial frequencies. The resulting graph from such an experiment is commonly referred to as the visibility threshold and can be used to provide an image-independent estimate for the JND of each frequency component or basis function of an image within the frequency domain. Additionally, it is often useful to interpret the behaviour of the visibility threshold in terms the Contrast Sensitivity Function (CSF), where the CSF may be computed by taking the inverse of the visibility threshold and can be used to demonstrate the relative sensitivity or response of the HVS to each frequency component.

One of the earliest and most widely adopted CSF models is that proposed by Ahumada and Peterson [16] in which the authors not only model the effects of varying amplitude of the sinusoidal gratings, but also model additional effects such as viewing angle, veiling luminance, display luminance, varying pixel spacing and varying spatial frequencies. In this model, the absolute visibility threshold, T, can be calculated for the $(i,j)^{th}$ DCT basis function as[2]

[2] The overall expression shown in equation 20 is a derived form of the original equation given in [16]. For reference, the corresponding derivation can be found in section A1.1 in the Appendix chapter.

$$T_{i,j} = A \cdot \frac{T_{min} \cdot {f_{i,j}}^4}{C(i) \cdot C(j) \cdot \left({f_{i,j}}^4 - 4(1-r){f_{i,0}}^2 {f_{0,j}}^2\right)} \cdot 10^{K\left(\log\sqrt{{f_{i,0}}^2 + {f_{0,j}}^2} - \log(f_{min})\right)^2} \quad (21)$$

where,

$$A = \frac{M}{(L_{max} - L_{min})}, \quad (22)$$

with M being the number of bits representing a pixel (i.e. 256 for an 8-bit image), L_{max} corresponding to the maximum display luminance (i.e. 255 for an 8-bit image), and L_{min} corresponding to the minimum display luminance (i.e. 0 for an 8-bit image),

$$C(i) = \begin{cases} \sqrt{\frac{1}{N}} & i = 0 \\ \sqrt{\frac{2}{N}} & i > 0 \end{cases} \quad (23)$$

with N denoting the size of the Nx N DCT matrix used and $C(i)$ denoting the DCT normalization factors. Furthermore,

$$T_{min} = \begin{cases} \frac{L}{S_o}\left(\frac{L_T}{L}\right)^{1-a_T} & L \leq L_T \\ \frac{L}{S_o} & L > L_T \end{cases} \quad (24)$$

where $L_T = 13.45\ cd/m^2$, $S_o = 94.7$ and $a_T = 0.649$,

$$f_{min} = \begin{cases} f_0\left(\frac{L}{L_f}\right)^{a_f} & L \leq L_f \\ \frac{L}{S_o} & L > L_f \end{cases} \quad (25)$$

where $f_0 = 6.78\ cyles/deg$, $L_f = 300\ cd/m^2$ and $a_F = 0.182$,

$$K = \begin{cases} K_0\left(\frac{L}{L_K}\right)^{a_K} & L \leq L_K \\ K_0 & L > L_K \end{cases} \quad (26)$$

with $K_0 = 3.125$, $L_K = 300\ cd/m^2$, $a_K = 0.0706$, and L being the sum of the veiling luminance and image luminance. In addition, the spatial frequency, f, of the $(i,j)^{th}$ basis function can be computed as

$$f_{i,j} = \frac{1}{2N}\sqrt{\left(\frac{i}{W_x}\right)^2 + \left(\frac{j}{W_y}\right)^2} \quad (27)$$

where W_x refers to the horizontal width of the pixel in degrees of visual angle and W_y refers to the vertical height of a pixel in degrees of visual angle and can be approximated as

$$W_n = 2\tan^{-1}\left(\frac{\vartheta}{2V}\right) \tag{28}$$

where ϑ is the image length in the n-direction and V is the viewing distance.

In particular, the visibility threshold $T_{i,j,}$ can be associated with two distinctive experiments:

In the first type of experiment, the sinusoidal gratings are generated in either a vertical direction or horizontal direction. Essentially, this means that the threshold visibility, $T_{i,j}$ is modelled for either i or j being fixed to zero throughout the entire experiment and means that the resulting CSF is a one-dimensional curve with spatial frequencies that correspond to the basis functions of either the first row or first column of the corresponding DCT transformation. As an example, figure 13 illustrates Ahumada's model using input parameters of $j = 0,\ W_x, W_y = 32$ pixels/degree, $L = 65\ cd/m^2$, $N = 8$. By inspecting this CSF curve, it can be seen that the HVS demonstrates a bandpass behaviour whereby it is most sensitive to spatial frequencies between 3-8 cycles/deg and less sensitive to the frequencies outside this range.

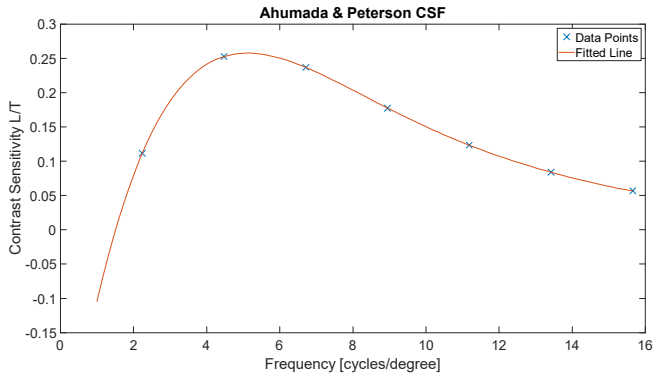


Figure 13: Example CSF using Ahumada and Peterson's model [15] .

In the second type of experiment, the sinusoidal gratings are generated in various different orientations such that all spatial frequencies resulting from the possible combinations of i and j are modelled. In this case, the resulting CSF is a 2-dimensional surface for which an example can be seen in figure 14 (B). By inspecting this graph, another anomaly of the HVS system can be observed by considering how the frequency sensitivity appears to be highest around the vertical and horizontal orientations ($i\ or\ j = 0$) and lowest along the diagonal ($i = j$).

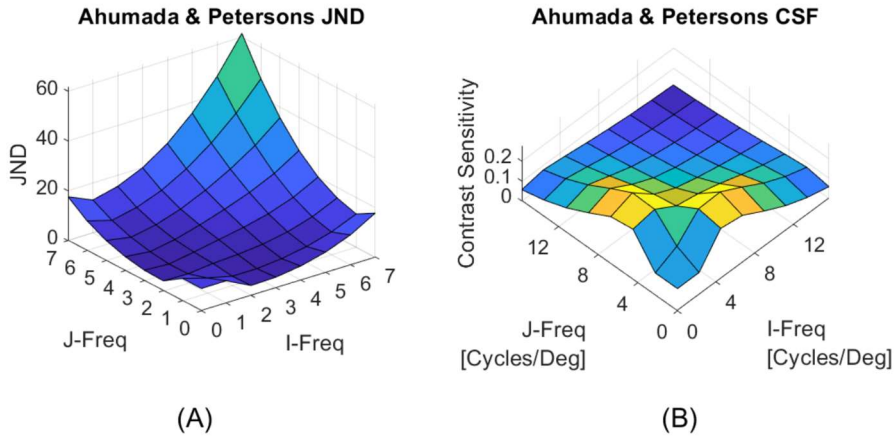


Figure 14: Example of Ahumada's JND & CSF models

A comparison may be made between the resulting JND table for Ahumada's frequency sensitivity model shown in figure 14 (A) and the previously discussed JPEG quantization table shown in figure 12. In this case, it should be noted that the relationship between the individual coefficients of a quantization table, Q, and a JND table are given by equation 29 [19] and implies that the JND table should conceptually be scaled by a factor of 2 in order to make a fair comparison between the relative values. In doing so it can be observed that Ahumada's model yields a slightly more conservative threshold than that use in the JPEG standard as well as producing symmetrical values around the $i = j$ diagonal, which the JPEG table does not.

$$Q_{ij} = 2 * JND_{ij} \tag{29}$$

where i, j are the individual coefficients of each respective table/matrix.

3.2.2 Luminance adaption

In the HVS model, luminance adaption is typically required in order to overcome some of the limitations or oversights of the frequency sensitivity model. In particular, the frequency sensitivity model proposed by Ahumada and Peterson assumes that visibility threshold can be adequately captured by only considering the mean luminance of the display. However, it is widely known that the true visibility threshold can demonstrate significant dependence on the local luminance within the image as well as on other luminance-related sources such as gamma correction and ambient lighting conditions [17]. In an effort to overcome this limitation, Watson [19] proposed a correction factor expression that could be adaptively applied to individual segment blocks of the image in order to take local luminance effects into account, where the relevant expression is given by

$$L_{ijk} = T_{i,j,k}(f_{0,0,k}/\bar{f}_{0,0})^{\,a_T} \tag{30}$$

where $T_{i,j,k}$ refers to the original $(i,j)^{th}$ threshold visibility coefficient of the k^{th} image block, L_{ijk} refers to the modified threshold visibility coefficient, $f_{0,0,k}$ refers to the DC coefficient (DCT(0,0)) of the k^{th} image block, a_T refers to a masking strength parameter and is suggested as being 0.649 and $\bar{f}_{0,0}$ refers to the DC coefficient corresponding to the mean luminance of the display and is suggested as being 1024 for an 8-bit image but can be defined as given in [20] as,

$$\bar{f}_{0,0} = \frac{1}{N_b}\sum_{m=1}^{k} f_{0,0,m} \tag{31}$$

where N_b refers to the total number of sub-blocks within the image and it can be proven that

$$\bar{f}_{0,0} = 8 * \bar{I} \tag{32}$$

for an image that is segmented into non-overlapping blocks of 8 x 8 and $\bar{I}$ = average pixel value of image.

3.2.3 Contrast Masking

In the HVS model, contrast masking refers to a well-known HVS property which says that signals can be perceptually supressed in the presence of other signals which are in close proximity and particularly of similar phase. In the image processing side of things, contrast masking is typically associated with the relative noise tolerance of three main image components, namely, a plain region, an edge region and a texture region. In this context, a plain region is generally regarded as having the least tolerance to noise while texture regions are considered to have the most tolerance and edges regions considered to have the second highest [21]. Furthermore, contrast masking is typically implemented in a watermarking system as a means for distinguishing between each of these region types and assigning an overall weight or elevation factor to the visibility thresholds of each frequency component of a particular image block or segment.

One popular contrast masking model was proposed by Watson [19] and is based on the seminal work done by Legge and Foley [22, 23]. In this model, contrast masking is given by

$$M_{ijk} = max\left[T_{i,j,k}, \left|f_{i,j,k}\right|^{w_{i,j}} \cdot {T_{i,j,k}}^{1-w_{i,j}}\right] \tag{33}$$

where $T_{i,j,k}$ refers to the original $(i,j)^{th}$ threshold visibility coefficient of the k^{th} image block, M_{ijk} refers to the modified threshold visibility coefficient, $f_{i,j,k}$ refers to the $(i,j)^{th}$ DCT coefficient of the k^{th} image block and w refers to a strength parameter that can be defined separately for each $(i,j)^{th}$ coefficient and can take values between 0 and 1, where 0 supresses all effects of the contrast masking model and 1 results in the "Weber Law " behaviour whereby the resulting $M_{i,j,k}$ is constant for $f_{i,j,k} > T_{ijk}$. Furthermore, Watson suggests using a value $w = 0.7$ for all coefficients apart from the DC coefficient, which he further suggests should have a weight of 0.

To demonstrate the effect of contrast masking, figure 15 shows the relative visibility thresholds after applying equation 33 to each of the 8 x 8 segments within the Lena image, where brighter regions indicate higher threshold values.

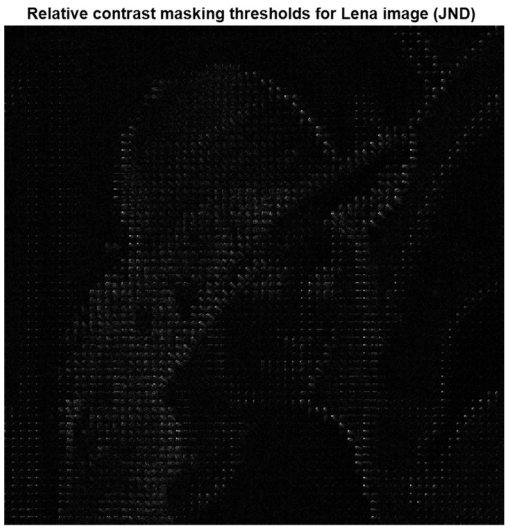


Figure 15: Relative Contrast masking threshold on Lena image.

3.3 Common Image Quality Assessment Metrics

3.3.1 Mean Square Error (MSE)

The Mean Square Error (MSE) between a watermarked image, I_w, and the original image, I_o, is given by,

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I_w - I_o)^2 \tag{35}$$

Where M, N refers to the size of the image.

The MSE metric is commonly used in image processing as a simple test for estimating the fidelity of a watermarked image. However, its usefulness is typically regarded as being highly limited because it does not take into account any of the known HVS characteristics of noise perception, nor does it provide an easy means for interpreting its value across different bit-depths. For example, an MSE value of 100 for a 10-bit image may appear to indicate good quality while for an 8-bit image this would indicate poor quality.

3.3.2 Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) between an 8-bit watermarked image, I_w, and the original image, I_o, is given by,

$$PSNR = 10\log_{10}\frac{{MAX_I}^2}{MSE}\ (dB) \tag{36}$$

Where M, N refers to the image size and $MAX_I = 255$ for an 8-bit image.

The PSNR can be considered closely related to the MSE but with the advantage that it does take the bit depth of the image into account and means that a PSNR value for one bit depth (i.e. 8-bit image) will be more easily comparable to the PSNR value of another bit depth (i.e. 10-bit image). For example, it is widely accepted that a PSNR of below 30 dB indicates low fidelity while a PSNR above 30 dB indicates acceptable fidelity. Similar to the MSE however, the PSNR also does not exploit any of the known HVS properties.

3.3.3 Absolute Reconstruction Error (ARE)

The Absolute Reconstruction Error (ARE) between an 8-bit watermarked image, I_w, and the original image, I_o, is given by,

$$ARE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|I_w - I_o| \tag{37}$$

Where M, N refers to the size of the image.

The ARE metric is closely related to the MSE and also suffers due to its limited interpretability between images of various bit depths.

3.3.4 SSIM

The Structural Similarity Index Measurement (SSIM) [24] is an IQA metric which attempts to model the HVS properties associated with local luminance, l, contrast masking, c, and structural similarity, s, between the distorted image, I_w, and the original image, I_o, where, if given two corresponding image segments, $\boldsymbol{x}$ and $\boldsymbol{y}$ from each of the respective images, the corresponding models may be separately defined as

$$l(\boldsymbol{x}, \boldsymbol{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{38}$$

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{39}$$

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{40}$$

where $\mu_x, \sigma_x, \sigma_{xy}$ refers to the mean, standard deviation and cross-correlation for segment $\boldsymbol{x}$, respectively and C_1, C_2, C_3 refer to small-valued regularization constants used to avoid possible instabilities for the cases in which the respective mean, standard deviation and cross-correlation may equate to zero.

More specifically, when equations 38-40 are multiplied together, the resulting measurement is known as the local SSIM, $SSIM$ and provides a distortion measurement between the two corresponding image segments, where a value of 1 indicates that no distortion is present

between the two segments and a value of 0 indicates that both image segments are completely different to one another. In general, however, a value of less than 0.8 may be regarded as demonstrating poor image quality.

The local SSIM provides a single distortion measurement corresponding to the local pixel region centred around a given pixel location within the image. If the local SSIM is applied to the entire image using a sliding window approach, the resulting output is considered to be the SSIM quality map and can subsequently be averaged to produce a more meaningful quality score which is reflective of the distortion within the entire image, and is referred to as the mean SSIM score, $MSSIM$.

4 Related Work

Ernawan et al. [5] proposed a robust watermarking scheme based on the DCT domain and a psychovisual threshold technique to embed a binary watermark into a grayscale image with minimal loss of perceptual fidelity in the output image. During the watermark embedding process, the host image is initially sub-divided into non-overlapping blocks of size 8 x 8 and an entropy expression used to determine the N-most suitable blocks to embed each bit of the watermark into, where N refers to the number of bits comprising the watermark itself. Each bit of the watermark is then embedded within the DCT domain of one of the N selected blocks and recombined with the remaining image to produce the watermarked image.

To obtain an optimal balance between the embedding strength and perceptual quality of the output image, Ernawan proposes a novel approach whereby the SSIM is computed for the watermarked image and subsequently compared against the NC value of the extracted watermark following a JPEG compression process. Furthermore, Ernawan argues that if the SSIM is found to be larger than the NC value, then the embedding strength could be increased further without compromising the imperceptibility of the watermark and that, therefore, the embedding process should repeated until this is no longer found to be true, i.e. the optimal balance between embedding strength and perceptual quality is assumed to be at the intersection point between the SSIM of the watermark image and the NC of the extracted watermark.

To demonstrate this idea, figure 16 illustrates the resultant curve presented in Ernawan's paper, for which the optimal embedding strength, or threshold in this case, has been found to be approximately 20 for that particular image. That is, Ernawan's embedding scheme proposes an image-dependant method for embedding the watermark. However, it should be noted that this optimization scheme is based on a global optimization process and that once the optimal embedding strength has been determined, it remains constant throughout all embedding regions within the image .

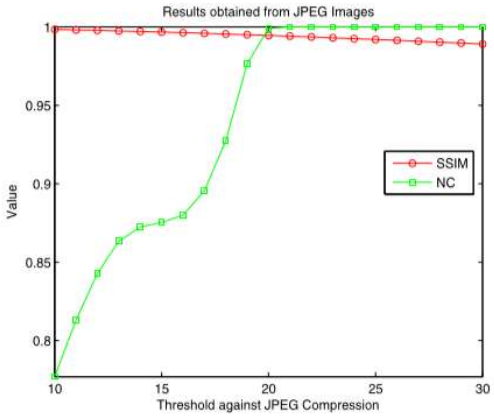


Figure 16: Graph from Ernawan's scheme showing intersection point of SSIM and NC the optimal embedding strength [5].

Cox et al. [6] proposed a watermarking scheme in which the DCT is computed globally over the entire host image and the N-largest coefficients (excluding the DC term) chosen to embed each of the N bits of the watermark in a spread-spectrum fashion. In contrast to Ernawan's scheme, which uses a binary image watermark and a fixed threshold value throughout the entire embedding procedure, Cox proposes that the watermark should be constructed as a zero-mean, independent and identically distributed (i.i.d.) Gaussian vector and an elevation factor or perceptual mask be used to vary the weight of the threshold across the frequency bands of the DCT matrix.

During the watermark extraction process, Cox is then able to exploit the known characteristics of the i.i.d Gaussian-generated watermark by removing the nonzero mean of the extracted watermark before performing the detection process. In doing so, Cox finds both experimentally and theoretically, that the overall effectiveness of the detection stage is essentially doubled, particularly if the channel stage is assumed to undergo an additive Gaussian white noise process.

The watermarking scheme proposed by Das et al. [25] uses an inter-block correlation method on the DCT matrices of two adjacent 8 x 8 sub-blocks to embed the relevant 1 or 0 watermark bits. This approach is based on the reasonable believe that small adjacent sub-blocks (taken from the spatial domain) will yield similar DCT matrices. If this is found to be true, then the DCT coefficients between both blocks can be altered to uphold an inequality expression depending on whether a binary 1 or 0 is encoded, i.e. the coefficients of one matrix may be made larger than the second to signify that a 1 has been encoded, or the vice versa in the case of encoding a binary 0. Furthermore, the amount by which each coefficient is altered differs amongst each adjacent block-pair and is a function of a scaling factor (used to control the balance between robustness and imperceptibility) the DC term and Median term of the first 11 AC coefficients of the DCT matrix being altered, relative to the other. Overall, this scheme appears to demonstrate satisfactory robustness to attacks such as JPEG compression, rotational attacks (up to 5°) and cropping attacks.

Levický and Foriš [26, 27] introduce a Human Visual System (HVS) model based in the DCT domain and demonstrate how this can be applied within the context of digital watermarking to identify both the perceptually significant segments within the image as well the acceptable ranges (JND thresholds) in which the corresponding DCT coefficients of each segment may be modified during the embedding stage without compromising the perceptual fidelity of the resulting image.

The proposed HVS model from Levický et *al*. consists of a frequency sensitivity function based from the seminal research done by Ahumada et *al*. [28], a Region of Interest (ROI) model from [29] for capturing the eye's eccentricity effect, a luminance adaption model and contrast model presented by Watson [19] as well as a Noise Visibility Function (NVF) based from the work of [12] for identifying the flat, edge and textured regions of the image and therefore allowing the best embedding locations throughout the image to be determined (from within the spatial domain). Overall, their experimental results indicate that their HVS model shows strong resistance against JPEG compression, cropping, gamma correction, brightness change and contrast change.

5 Experimental Work

The framework for the proposed watermarking scheme has been built following the implementation and evaluation of various watermarking schemes and algorithms using MATLAB. The following chapter aims to provide a practical outline into each of these schemes, their role within the overall design, as well to address any additional assumptions that were necessary during this stage for successful implementation of the relevant work.

5.1 Implementation of Ernawan's Scheme

As discussed in the previous chapter, Ernawan et al. [5] proposed a state-of-the-art robust watermarking scheme based on the discrete cosine transform (DCT) and incorporating a psychovisual threshold technique based on the intersection point between the SSIM of the watermarked image and NC of the extracted watermark, in order to embed a binary watermark into a grayscale image with minimal loss of perceptual fidelity. Furthermore, published results for this scheme indicated great promise during the early stages of the project and motivated further investigation into its implementation and evaluation using MATLAB. For reference, Ernawan's embedding stage and extraction stage can be summarised according to the pseudocode found in sections A1.2 and A1.3 of the appendix section.

During the implementation of Ernawan's embedding procedure, a number of discrepancies were uncovered. The first of these included the entropy expression which Ernawan uses to determine the most suitable embedding blocks and can be summarised according to expression shown in equation 41. In his paper, Ernawan merely refers to this expression as being a combination between edge entropy and visual entropy and gives no additional information concerning the legitimacy or perceptual significance of this expression. However, a reference is provided for this expression and is given by Lai's scheme [30]. By further investigating Lai's scheme, a number of observations were obtained and can be summarised as follows:

1. In Lai's scheme, visual entropy is defined by Shannon's entropy and is given by equation 42, while edge entropy is referenced from Maity et al. [31] and is given by equation 43.
2. In Lai's scheme, the embedding blocks are determined in a similar fashion as with Ernawan's scheme whereby the edge entropy and visual entropy are summed for each block and the N-lowest entropy blocks selected.
3. In Maity's scheme, the embedding blocks are determined by applying the visual entropy (Shannon entropy) on the edge map of the image and summing this with the edge entropy applied to the original grayscale image. Here, it understood that the edge map is calculated by applying a gradient operator on the image before extracting each block. Furthermore, Maity embeds the watermark twice and considers taking the lowest entropy blocks as well as the blocks with mid-range entropy values for embedding the watermark.
4. In Maity's scheme, the expression for edge entropy is referenced from Pal et al. [32]. Pal et al. appear to be the original authors of the edge entropy expression. However, their paper is based on object-background segmentation and does not explicitly consider the use of edge entropy within watermarking applications - In fact, their definition of the edge entropy also appears to be different from all previous definitions considered by Maity, Lai

and Ernawan, although Maity does appears to address this in their paper by counter arguing that their edge map approach is equivalent to Pals method.

$$E_{Ernawan} = -\frac{1}{2}\sum_{i=1}^{M} p_i \exp(1-p_i) + p_i \log_2(p_i) \tag{41}$$

$$E_{visual} = -\sum_{i=1}^{M} p_i \log_2(p_i) \tag{42}$$

$$E_{edge} = \sum_{i=1}^{M} p_i \exp(1-p_i) \tag{43}$$

$$E_{Lai} = E_{visual} + E_{edge} = -\sum_{i=1}^{M} -p_i \exp(1-p_i) + p_i \log_2(p_i) \tag{44}$$

In summary, the above points lead to three varying methods for which the most suitable embedding blocks may be determined from. To gain insight into the differences between these methods, each method was simulated in MATLAB and subsequently compared against one other. The results of this can be seen in figure 17, which illustrates the selected blocks in black for the Lena image using Ernawan's scheme (left image), Lai's scheme (centre image) and Maity's scheme (right image). By observing each of these methods, it can be observed that all three appear to give considerably similar results, although none are exactly the same. Furthermore, by comparing Ernawan's entropy expression to Lai's entropy expression, given by equation 44, it can be observed that Ernawan's entropy expression differs from Lai's only in that it does not negate the visual entropy term. This fact was subsequently treated as being indicative of a typo in Ernawan's paper, and the entropy which has been used to implemented Ernawan's scheme in MATLAB has been taken to be that of Lai's scheme.

Figure 17: Selected embedding locations (in black) for Lena image using Ernawan's scheme (left image), Lai's scheme (centre image) and Maity's scheme (right image).

Another discrepancy which was discovered during this stage included that of Ernawan's embedding function. Unlike most schemes which implement one of the general embedding functions given by equations 1-3, Ernawan's embedding function takes an open-form solution and embeds each bit of the watermark by modifying the relationship between two DCT coefficients, referred to here as a pair, and repeating this process in total across 3 pairs within the DCT domain of the selected embedding blocks. To gain a more meaningful insight how this is achieved, pseudocode of Ernawan's embedding function for a single pair can be seen in section A1.4 of the appendix. By examining this embedding function it can be observed that the embedding of a binary bit of 1 will always result in the first coefficient having a larger absolute value than that of the second coefficient and intuitively, it should be expected that the embedding of a binary bit of 0 would yield the reverse result, i.e. that the absolute value of the second coefficient will always be larger than the first coefficient. However, by closely inspecting Ernawan's algorithm it can be determined that this is not always guaranteed to be true. In fact, by mapping out Ernawan's embedding function onto the real line for each possible case (i.e. bit to embed = 0 or 1, coefficient 1 less than coefficient 2 etc.) it can be shown that there exists no determinable way to guarantee which bit has been embedded. In an effort to resolve this issue, it was subsequently assumed that the most likely candidate for this discrepancy was due to a possible typo in Ernawan's presented algorithm, that, when fixed would result in the previously expected behaviour of a deterministic embedding function. To demonstrate the root of this typo and how it may be fixed, the embedding algorithm in section A1.5 shows the actual embedding function that was subsequently used in implementing Ernawan's scheme.

In addition to Ernawan's embedding function, a discrepancy was also found in his proposed extraction function. To demonstrate this typo, section A1.6 shows the pseudocode for Ernawan's original extracting function, from which it can be seen that this function fails to consider the absolute values of the respective coefficients and therefore would not be able to deterministically extract the watermark bits. To overcome this problem, section A1.7 shows the pseudocode corresponding to the actual embedding function that was subsequently implemented, of which a deterministic extraction of the watermarks bits is possible.

5.2 Implementation of Levicky's HVS model

As discussed in the previous chapter, Levický and Foriš [26, 27] introduced a HVS model and demonstrated how this could be applied to a DCT-based watermarking system. Their published results for this scheme demonstrated high imperceptibility as well as good robustness to a wide range of attacks and has encouraged further investigation into implementing their scheme using MATLAB.

In this scheme, the HVS model presented by Levický and Foriš includes a frequency sensitivity model, a foveation model, a luminance adaption model and a contrast sensitivity model. However, the use of foveation models are typically better suited towards video applications and it was subsequently decided to be omitted from the overall design of this project, for which still image applications is the primary concern.

5.2.1 Frequency Sensitivity Model

The frequency sensitivity model used by Levický et *al*. is based from the model proposed by Ahumada et *al*. [33] and is essentially an improved model to their earlier one, which was previously discussed in section 3.2.1. In this model, the JND threshold matrix, $T_{i,j}$, can be computed as,

$$T_{i,j} = \frac{T_{min} \cdot {f_{i,j}}^4}{C(i) \cdot C(j) \cdot \left({f_{i,j}}^4 - 4(1-r){f_{i,0}}^2 {f_{0,j}}^2\right)} \cdot 10^{K\left(\log\sqrt{{f_{i,0}}^2 + {f_{0,j}}^2} - \log(f_{min})\right)^2} \quad (45)$$

where, $f_{i,j}$ refers to the $(i,j)^{th}$ spatial frequencies and can be computed using equation 27, T_{min} refers to the minimum threshold occurring at the spatial frequency, f_{min}, and are given as $T_{min} = 0.25,\ f_{min} = 3.1$ in [33] and K refers to a steepness parameter and is also given to be $K = 1.34$ in [33].

Considering similar conditions as used by Levický et *al*. [27], with an 8 x 8 image segment and a viewing distance of 8 image heights, the corresponding JND threshold matrix, T, can be calculated and plotted as shown in figure 18.

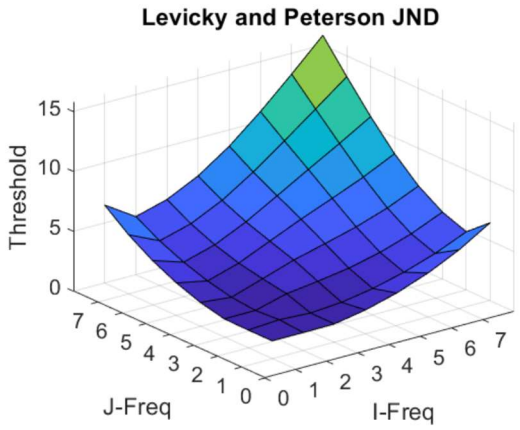


Figure 18: Plot of Frequency Sensitivity model used by Levický et *al*. [27].

5.2.2 Luminance Adaption

The luminance adaption model, L_{ijk}, used by Levický et *al*. is the same as that proposed by Watson [19] and can be calculated using equation 30. Moreover, it is possible to gauge the effect of this adaptation by applying it to the Lena image and plotting the relative values of each threshold, the result of which can be seen in figure 19. In this figure it can be seen that the threshold values appear highest around the bright regions of the original Lena image and is indicative of the known HVS property that more noise can be tolerated at brighter regions.

Figure 19: Relative luminance adaption thresholds on Lena

5.2.3 Contrast Masking

The contrast masking model used by Levický et *al*. [26] is given by

$$M_{ijk} = L_{ijk} \cdot max\left[1, \left(e^{\frac{-\pi\left((i-i_m)^2+(j-j_m)^2\right)}{\varphi \cdot \max(1,\sqrt{i^2+j^2})} \cdot \left|\frac{f_{i_m,j_m,k}}{L_{ijk}}\right|}\right)^{w_{i,j,k}}\right] \tag{46}$$

where $f_{i_m,j_m,k}$ is the DCT coefficient which acts as a mask to the $f_{i,j,k}$ DCT coefficient, φ is a model parameter and $w_{i,j,k}$ is a parameter which can take values between 0 and 1 (used to control the masking effect similarly to Watsons expression from equation 33) and is defined by Levický et *al. as*

$$w_{i,j,k} = \begin{cases} \frac{\gamma}{3M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(1 - NVF_{i,j,k}\right) & if \sum_{i=1}^{M}\sum_{j=1}^{M} NVF_{i,j,k,p} < M^2 \\ \frac{\gamma}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(1 - NVF_{i,j,k}\right) & if \sum_{i=1}^{M}\sum_{j=1}^{M} NVF_{i,j,k,p} \geq M^2 \end{cases} \quad (47)$$

where M is the block size (i.e. $M = 8$), γ can be used to vary the strength or robustness of the embedded watermark, and $NVF_{i,j}$ refers to the Noise Visibility Function [12] which yields values between 0 and 1 and is defined by

$$NVF_{i,j} = \frac{1}{1 + \mu\sigma_{i,j,x}^2} \quad (48)$$

where $\sigma_{i,j,x}$ denotes the local variance centred around the pixel at location i, j and μ denotes a tuning parameter given by

$$\mu = \frac{D}{\sigma_x^{max}} \quad (49)$$

where σ_x^{max} is the maximum variance across all windowed regions within the image, and D can take values between 50 and 100.

In particular, the NVF corresponds to a pixel-wise HVS model and be used within the spatial domain to identify the regions within an image which are plain, edge, or texture based, where high values indicate plain regions, medium values indicate edges and low values indicate texture regions. For example, assuming $D = 100$, and using a spherical sliding window of 13 pixels, the corresponding NVF map of the Lena image can be computed as shown in figure 20:

Figure 20: NVF map of Lena

Furthermore, assuming $\gamma = 1$, $\varphi = 1$, and modelling only for self-contrasting (i.e. $i_m = i$, $j_m = j$) the corresponding relative contrast visibility thresholds for the Lena image can be computed as shown in figure 21.

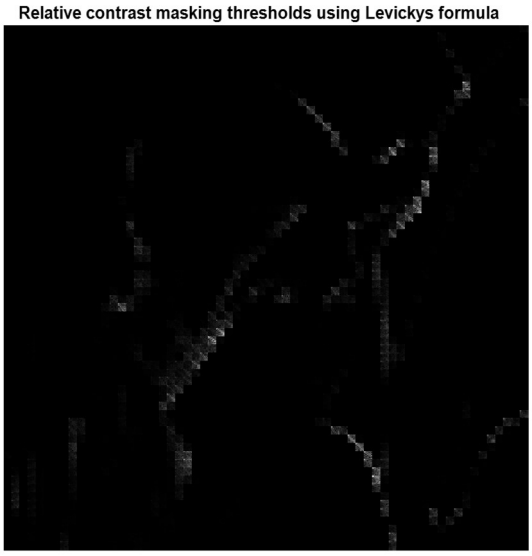


Figure 21: Relative contrast visibility thresholds for Lena image.

5.2.4 Proposed Design

The proposed design aims to build upon Ernawan's previous work by incorporating the above-mentioned HVS models into Ernawan's original design in order to overcome some of the weakness that have been identified with this scheme. In particular, two essential weakness have been identified and include the global embedding strength and entropy equation:

- Ernawan's Original scheme relies on a global embedding strength parameter in order to embed each bit of the watermark. Although, the method used in determining this parameter may be regarded as optimal from a global optimization sense, this approach does not properly exploit the full potential that can be realized by incorporating an HVS model, which will instead allow the embedding strength to be adaptively varied throughout the image.
- As previously discussed, a discrepancy was discovered with the use of Ernawan's entropy expression and subsequently lead to an in-depth investigation into its origin and legitimacy within a watermarking application. The result of which found that part of this expression, namely the edge entropy, was widely misused by the previous watermarking schemes referenced by Ernawan and that, in fact, the original use of this expression was intended for applications such as background segmentation with no further indication given with regards to its use or reliability within watermarking.

In addition to considering the weaknesses of Ernawan's scheme, the proposed scheme also aims to consider some of the strengths found in Ernawan's scheme. Perhaps the best of which can be considered to be Ernawan's novel approach for determining the embedding strength as a function of the SSIM of the watermarked image and NC of the extracted watermark. This approach is particularly interesting as it known that the use of a HVS model as a sole means for estimating the impact of distortion within an image can be quite misleading when

considering the overall impact of image quality and not just on the individual segments considered by the HVS model [19]. Therefore, by incorporating the SSIM into the embedding strength function, a more complete estimate of the perceptual degradation of the image should be achieved. Meanwhile, the use of the NC value being compared against the SSIM means that the quality constraint must contend with the overall robustness of the system and while it uncertain as to whether such approach is in fact optimal, it does appear to be one possible solution for addressing the embedding effectiveness dilemma previously mentioned in section 2.2.5.2.

The following pseudocode highlights the algorithm for the embedding stage of the proposed design, where it should be further note that the extraction stage has remained unchanged from the initial design that was implemented for Ernawan's scheme.

Embedding Stage of Proposed Design

1. **Adopt an initial value for the global embedding strength, i.e. α_{global}**
2. **Segment the input image into non-overlapping blocks of size 8 x 8 pixels.**
3. **Calculate the NVF map of the image and select the N-lowest average NVF blocks as corresponding to the embedding locations.**
4. **Store the positions of each selected block.**
5. **Encrypt the binary watermark image.**
6. **FOR each selected block,**
   compute its JND mask using the HVS models and embed a single bit of the watermark within the DCT domain according to the proposed embedding function.
7. **Perform the inverse DCT on each selected block and recombine with the remaining blocks to create the watermarked image.**
8. **Compute the SSIM of the watermarked image (SSIM).**
9. **Create a compressed copy of the watermarked image using JPEG with a quality factor of 50.**
10. **Extract the watermark from the compressed copy and compute its normalized cross-correlation value (NC).**
11. **WHILE (SSIM > NC),**
    increment the embedding strength, i.e. $a \rightarrow a + 1$ and repeat steps 6 – 11.

Proposed Embedding Function

FOR each coefficient pair, (X,Y) and its corresponding JND pair (X_{JND},Y_{JND})

$\alpha_{total} = ((X_{JND} + Y_{JND})/2) + \alpha_{global}$

IF (watermark bit == 1)

IF (abs(X) < abs(Y))

Z = X

X = Y + sign(Y)* α_{total}

Y = Z

ELSE

X = X + sign(X)* α_{total}

Y = Y

END

ELSE (watermark bit == 0)

IF (abs(X) < abs(Y))

X = X

Y = Y+ sign(Y)* α_{total}

ELSE

Z = X

X = Y

Y = Z + sign(X)* α_{total}

END

END

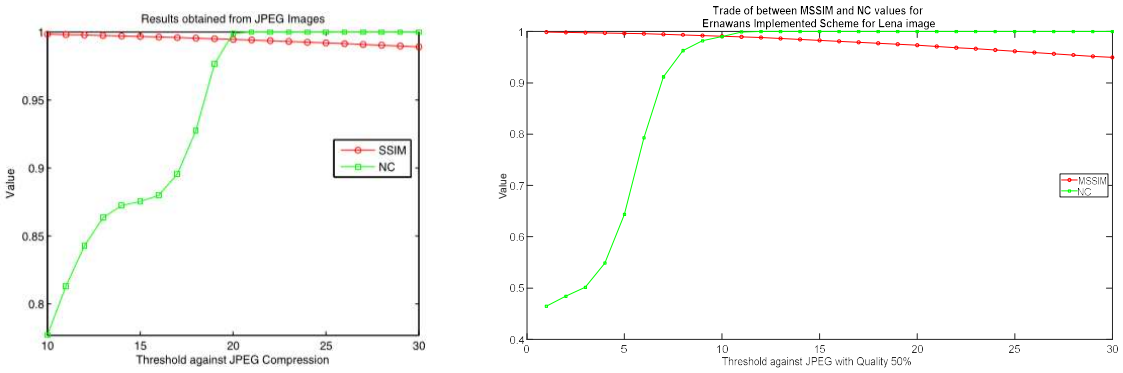
6 Experimental Results & Evaluation

The following chapter highlight the results that were obtained following the implementation of Ernawan's scheme and the subsequent proposed scheme. For each of these experiments (unless otherwise stated) the corresponding set of results have been obtained by averaging across a set of 9 test images.[3]

6.1 Implementation of Ernawan's Watermarking System

The following section compares Ernawan's presented results with those achieved from his implemented scheme:

6.1.1 Comparison between SSIM & NC trade-off graphs



(A) (B)

Figure 22 : Comparison between Ernawan's SSIM and NC trade off vs. Implemented SSIM and NC trade off

Figure 22 (A) illustrates the SSIM-NC trade-off curve given in Ernawan's paper while Figure 22 (B) illustrates the corresponding curve obtained for the implemented scheme for the Lena image. In comparing both images it can be observed that the MSSIM curves (red line) between both graphs appear to agree quite well while the NC curves appears to disagree to some extent. In particular, the NC curve from Ernawan's paper appears to reach its maximum value of 1 (or close to it) at a threshold of approximately 20 while in the case of the implemented scheme, the corresponding NC curve reaches this range at a threshold of approximately 10.

One possible reason for this discrepancy could be due to an incorrect assumption that was made during the implementation stages of the project, however, it also unknown as to which image Ernawan's graph corresponds to and could also explain part of this discrepancy.

[3] These images can be accessed at:
https://github.com/barnardp/5E1-Digital-Image-Watermarking/tree/master/images/images

6.1.2 Comparison between PSNR

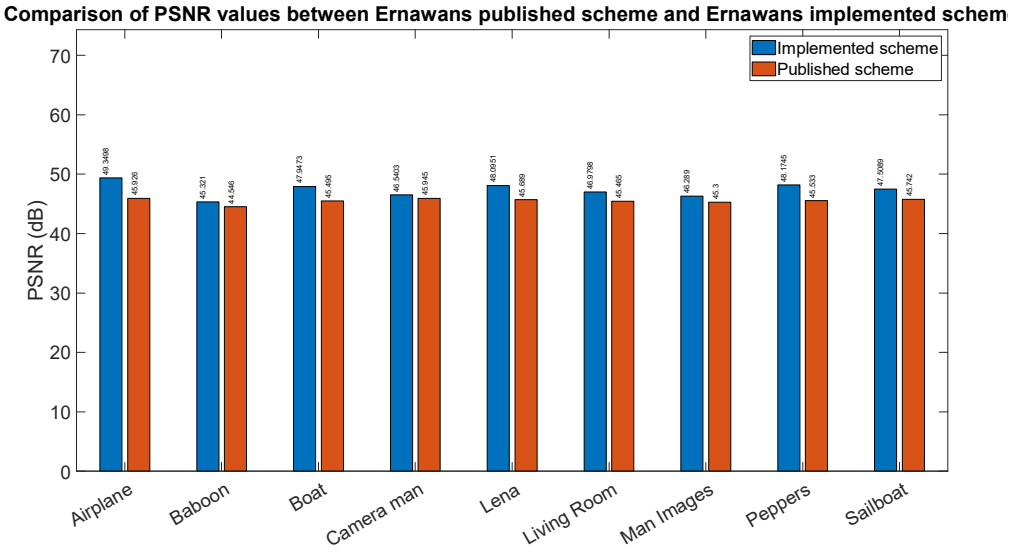


Figure 23: Comparison of PSNR values between Ernawan's published scheme and implemented scheme

In figure 23, the PSNR values of Ernawan's published results are shown by the red bars while the corresponding blue bars indicate the implemented results. A comparison between each scheme shows that the implemented scheme achieves a slightly better PSNR value in all the test cases examined.

6.1.3 Comparison of Attacks (1)

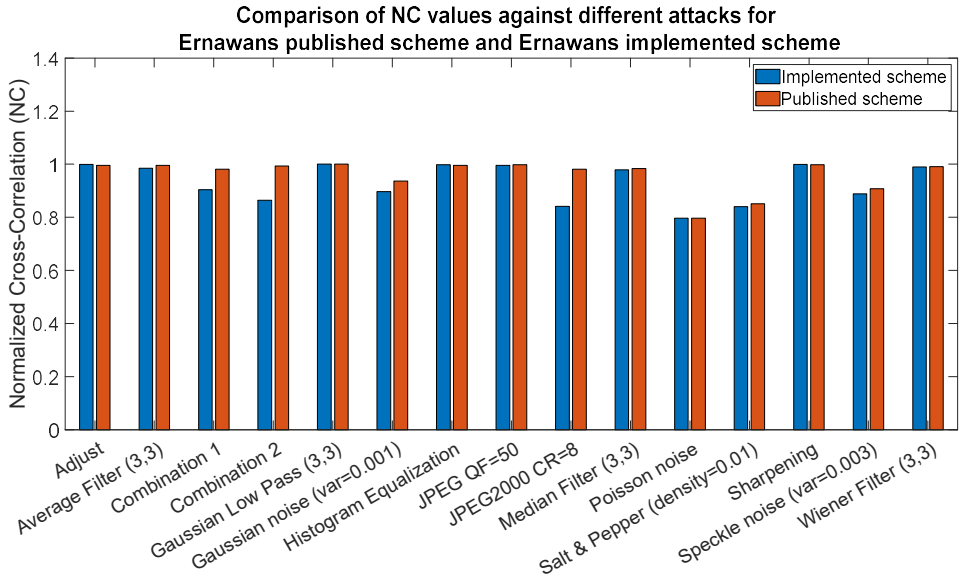


Figure 24: Comparison of attacks (1) between Ernawan's published scheme and implemented scheme

Figure 24 illustrates the performance of both schemes against common noise attacks, compression attacks and two combinational attacks, where the combinational attack 1 corresponds to a 3 x 3 median filter, followed by a salt and pepper attack of strength 0.003, and the combinational attack 2 corresponds to JPEG compression with quality factor of 50, followed by a centre crop of 256 x 256 pixels. In comparison it can be seen that the Ernawan's published scheme demonstrates slightly better robustness to the combinational attack 1 and Gaussian low pass filter while the implemented scheme demonstrates superior robustness to JPEG2000 compression. Nonetheless, there appears to be good agreement between both schemes for the remaining attacks.

6.1.4 Comparison of Attacks (2)

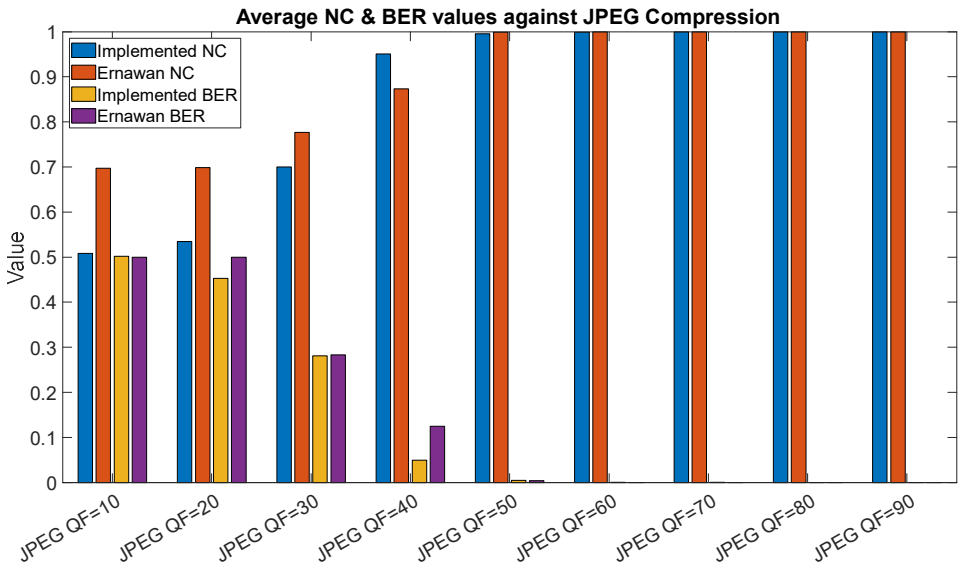


Figure 25: Comparison of compression attacks between Ernawan's published scheme and implemented scheme

Figure 25 illustrates the performance of both schemes under JPEG compression for quality factors between 10 and 90. From this figure it can be seen that the implemented scheme demonstrates slightly higher robustness at a quality factor of 40 while Ernawan's published scheme demonstrates far greater robustness at low quality levels of 10 and 20.

However, it should be noted that Ernawan does not explicitly provide the raw performance values for his scheme at JPEG compression qualities of below 30 and that the presented results for qualities 10 and 20 have been estimated from a graph within the paper using the imtool function in MATLAB, and a process as demonstrated in figure 26. As seen in this figure, the relevant values could be estimated by measuring the relative lengths of each point within the graph where, for example, the NC value for quality of 30 could be estimated as $\frac{534}{743} = 0.719$. In fact, Ernawan tubulates the NC value for JPEG of quality 30

as being 0.7769 in his paper. This may suggest that either there is a typo in the paper or the error in the graph estimating method may be as high as 7%.

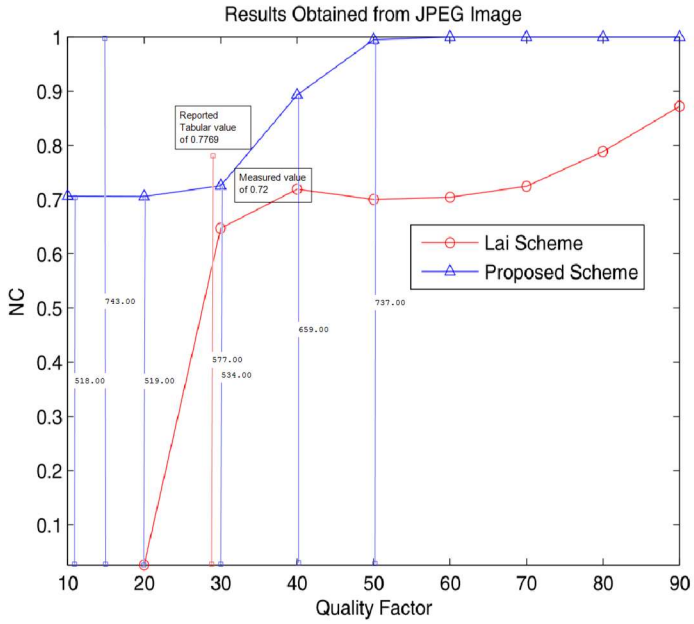


Figure 26: Process used in estimating JPEG performance of Ernawan's scheme for qualities below 30.

6.2 Comparison of Attacks (3)

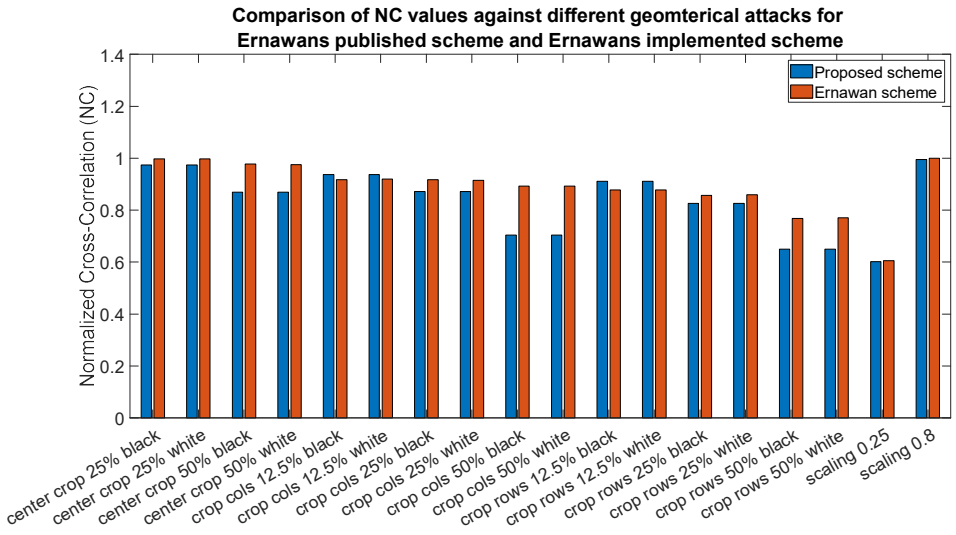


Figure 27: Comparison of attacks (3) between Ernawan's published scheme and implemented scheme

Figure 27 demonstrates the performance of both schemes under various geometrical attacks. In particular, the schemes are tested against a wide range of cropping attacks as well as two scaling attacks. A comparison between both schemes indicate that Ernawan's published scheme appears to be slightly more robust to cropping attacks while both schemes appear to show highly similar results for the scaling attacks.

6.3 Comparison of SSIM and ARE

	Implemented ARE	Ernawan ARE	Implemented SSIM	Ernawan SSIM
Airplane	0.2363	0.3030	0.9922	0.9930
Baboon	0.3618	0.3510	0.9907	0.9950
Boat	0.2823	0.3180	0.9911	0.9950
Camera man	0.3323	0.3030	0.9841	0.9940
Lena	0.2823	0.3120	0.9895	0.9940
Living Room	0.3087	0.3190	0.9917	0.9960
Man Images	0.3276	0.3210	0.9881	0.9950
Peppers	0.2765	0.3170	0.9906	0.9940
Sailboat	0.2950	0.3100	0.9897	0.9940
Average	0.3003	0.3171	0.9897	0.9944

Table 2: Comparison between ARE and SSIM for Ernawan's published scheme and Ernawan's implemented scheme

Table 2 demonstrates the objective measurements on perceptual quality of the watermarked images for schemes. In particular, the ARE and SSIM are examined in this case, in which it can be observed that Ernawan's scheme demonstrates slightly higher SSIM scores than the implemented scheme, while the implemented scheme demonstrates marginally better performance with regards to the ARE.

6.4 Implementation of the proposed scheme

6.4.1 Comparison between SSIM & NC trade-off graphs

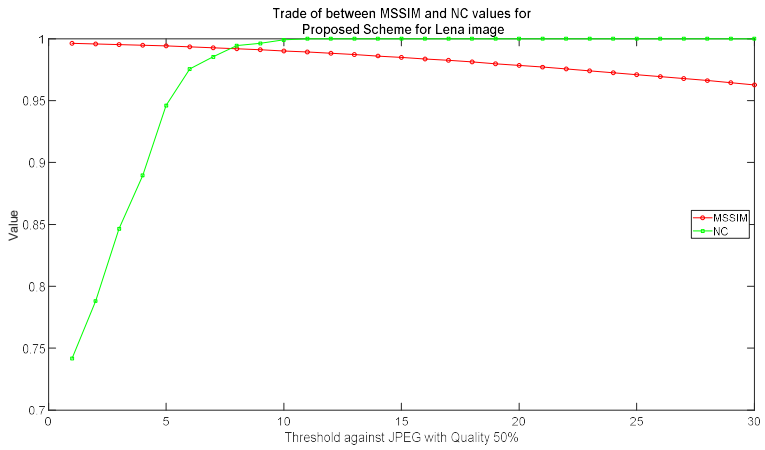


Figure 28: SSIM and NC trade-off curve of proposed scheme for the Lena image

Figure 28 illustrates the SSIM and NC trade-off curve of the proposed scheme for the Lena image, and may be compared to Ernawan's implemented scheme, for which corresponding trade-off curve can be seen in figure 22 (B). In comparison, it can be observed that the SSIM value of the proposed scheme appears to fall at a slightly higher rate than that of Ernawan's implemented SSIM curve while the NC value of the proposed scheme appears to demonstrate more linearity within a threshold of 1-10. However, it should be noted that the threshold shown in figure 28 corresponds to the 'global' portion of the overall threshold used in the proposed scheme as the JND portion is not included in this curve.

6.4.2 Comparison between PSNR

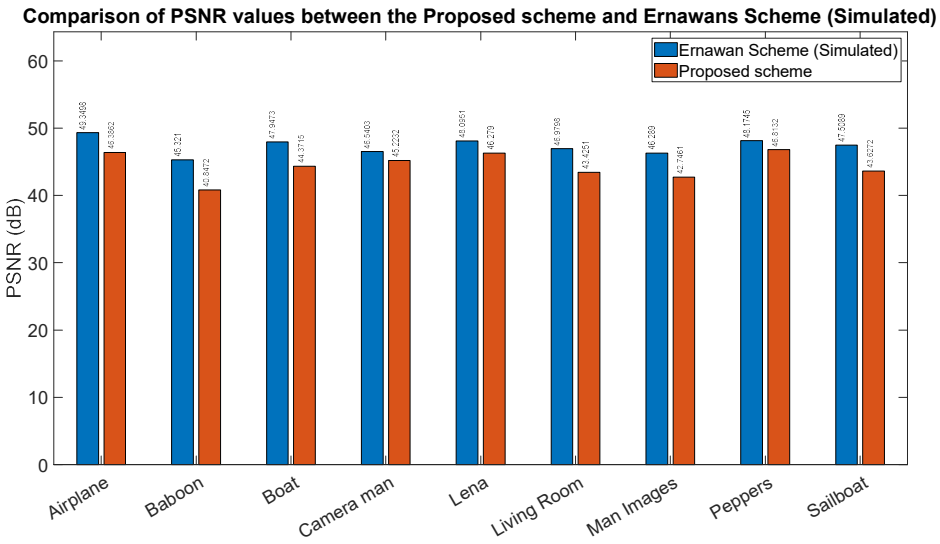


Figure 29: Comparison of PSNR values between Ernawan's published scheme and implemented scheme

Figure 29: illustrates the corresponding PSNR values between the proposed scheme and Ernawan's implemented scheme. From this figure, it can be seen that Ernawan's implemented scheme demonstrates much better performance than the proposed scheme for the test images examined. However, the performance of the proposed scheme is still above 40 dB is all cases, which is generally considered to be indicative of good perceptual quality.

6.4.3 Comparison of Attacks (1)

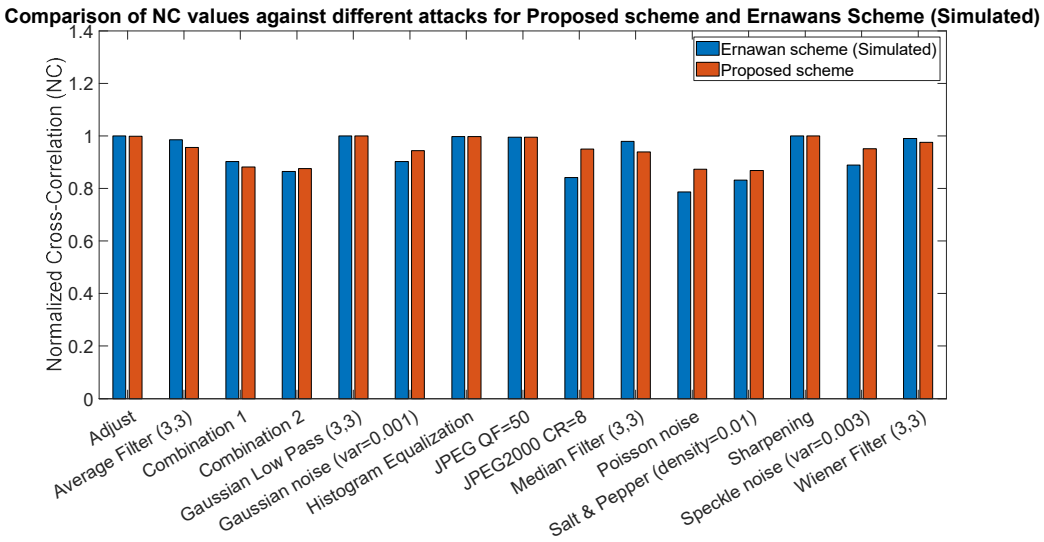


Figure 30: Comparison of attacks (1) between proposed scheme and Ernawan's implemented scheme.

Figure 30 demonstrates the performance of the both schemes against a wide range of attacks, including noise attacks, compression attacks and combinational attacks. In comparison, it can be observed that both the proposed scheme and Ernawan's implemented scheme appear to indicate similar robustness in most cases. However, the proposed scheme demonstrates slightly better performance under JPEG2000 compression, speckle noise and Poisson noise.

6.4.4 Comparison of Attacks (2)

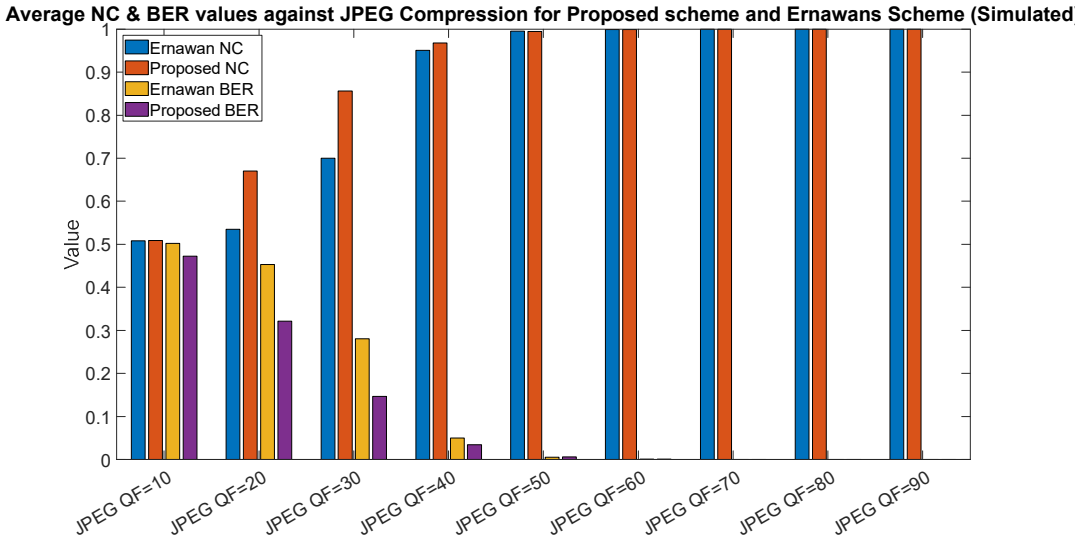


Figure 31: Comparison of compression attacks between proposed scheme and Ernawan's implemented scheme

Figure 31 shows the performances of both schemes against a wide range of JPEG compression qualities. In comparison, it can be observed that the proposed scheme outperforms Ernawan's scheme at qualities of 20 and 30 but appear reasonably similar for the remaining qualities. Neither schemes appear to be reliable at a compression quality of below 20 (i.e. NC values fall to approximately 0.5, indicating a random extracted watermark).

6.4.5 Comparison of Attacks (3)

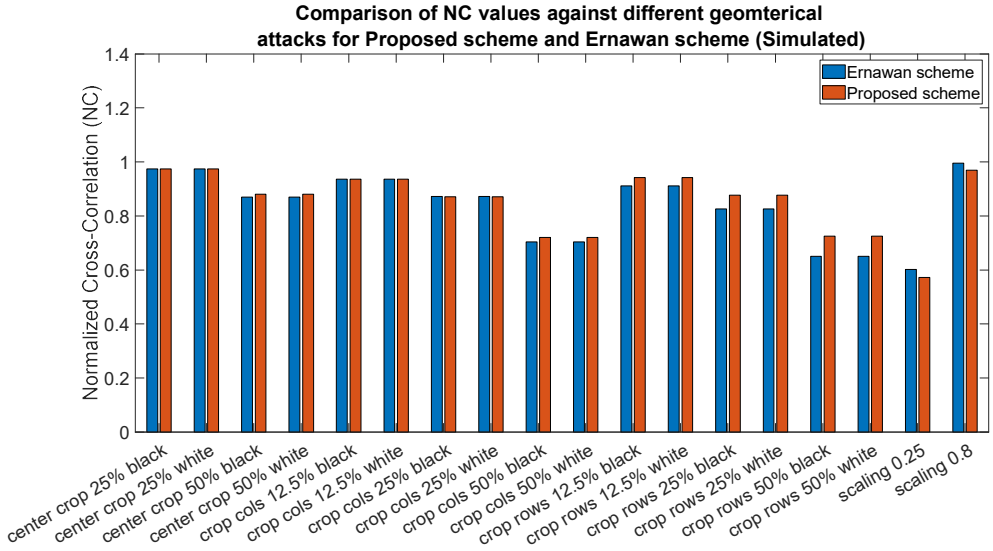


Figure 32: Comparison of attacks (3) between proposed scheme and Ernawan's implemented scheme

Figure 32 shows the performance of both schemes under various cropping and scaling attacks, in which it can be seen that the performance of both schemes are reasonably similar.

6.4.6 Comparison of SSIM and ARE

	Proposed ARE	Ernawan ARE	Proposed SSIM	Ernawan SSIM
Airplane	0.3281	0.2363	0.9917	0.9922
Baboon	0.6387	0.3618	0.9886	0.9907
Boat	0.4055	0.2823	0.9912	0.9911
Camera man	0.3777	0.3323	0.9851	0.9841
Lena	0.3318	0.2823	0.9920	0.9895
Living Room	0.4539	0.3087	0.9911	0.9917
Man Images	0.4920	0.3276	0.9905	0.9881
Peppers	0.3164	0.2765	0.9933	0.9906
Sailboat	0.4396	0.2950	0.9927	0.9897
Average	0.4204	0.3003	0.9907	0.9897

Table 3: Comparison between ARE and SSIM for proposed scheme and Ernawan's implemented scheme

Table 3 illustrates the corresponding ARE and SSIM values of each scheme. From this table it can be observed that the proposed scheme indicates a relatively poor ARE in comparison to Ernawan's implemented scheme while the SSIM for the proposed scheme is slightly higher than Ernawan's implemented scheme and is given by an average value of 0.9907. This would suggest that the proposed scheme demonstrates good perceptual quality.

6.4.7 Additional Testing

Additional test was also done to test the performance of the proposed scheme and Ernawan's implemented scheme against rotation. In the first test, both schemes were tested against rotational attacks between 0-360° and intervals of 1° on the Lena image, during which no attempt was made to correct the input images at the extractor stage. The corresponding results for this test can be seen in figure 33, in which it can be seen that the proposed scheme demonstrates slightly better robustness than Ernawan's implemented scheme. However the overall performance of both schemes is still regarded as poor due to the fact that the NC value never appears to surpass the 0.6 range for angles other than 0° and 360°.

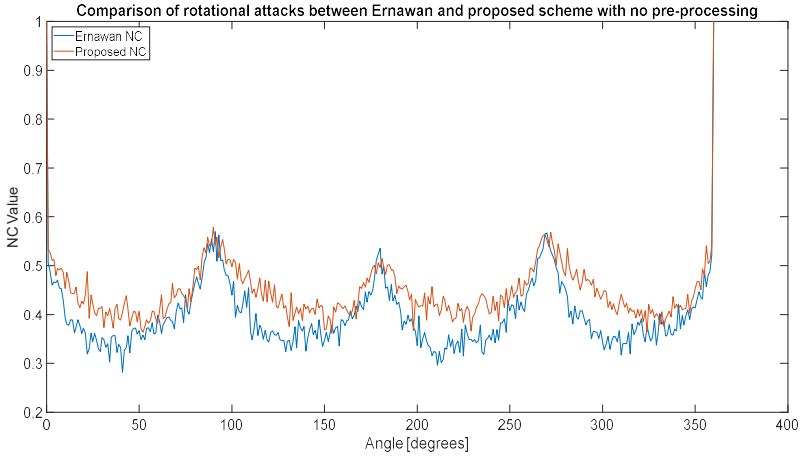


Figure 33: Rotational testing on proposed scheme and Ernawan's implemented scheme (1).

In the second experiment, the same test was performed but with the exception of using the SURF feature detector [34] in order to identify the possible rotation angle of the input images before they are processed by the extractor stage. The corresponding results of which can be seen in figure 34. From this figure it can be seen that the use of the SURF feature detector has significantly improved the robustness of both schemes against rotational attacks, where the average NC values across this range has ben found to be 0.9218 and 0.8877 for the proposed scheme and Ernawan's implemented scheme, respectively.

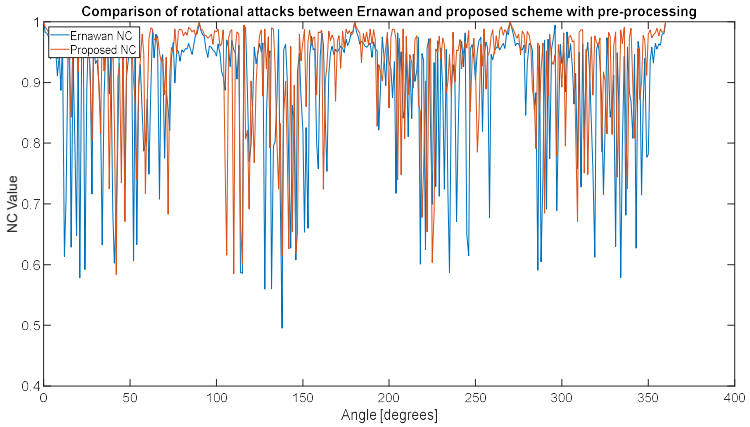


Figure 34: Rotational testing on proposed scheme and Ernawan's implemented scheme (2).

7 Conclusion

In this project, a novel robust watermarking scheme based within the DCT domain has been developed for use within copyright applications of digital still images. In particular, the framework for this design has been constructed following a careful investigation into the various state-of-the-art watermarking schemes within this field and has found the recent work of Ernawan et *al*. [5] to be of great promise.

In addition to this, the proposed scheme has also investigated the use of Human Visual System models in order to achieve an optimal trade-off between the imperceptibility and robustness criteria required by a robust watermarking scheme.

The experimental results of this investigation have found that an HVS model can be combined with a global quality measurement to adaptively determine optimal embedding parameters to yield a strong balance between robustness and imperceptibility. Furthermore, the use of the SURF feature detector has also been implemented as means for tackling the rotational vulnerability which most watermarking schemes exhibit. The results of which have proven a significant increase in the systems resilience to this type of attack

8 References

[1] M. Kutter and F. A. Petitcolas, "A fair benchmark for image watermarking systems," in *Security and Watermarking of Multimedia Contents*, 1999, vol. 3657: International Society for Optics and Photonics, pp. 226-240.

[2] G. Caronni, "Ermitteln unauthorisierter Verteiler von maschinenlesbaren Daten," *ETH, Zürich, Switzerland, Tech. Rep,* 1993.

[3] G. Caronni, "Assuring ownership rights for digital images," in *Verläßliche IT-Systeme*: Springer, 1995, pp. 251-263.

[4] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., 2008, p. 624.

[5] F. Ernawan and M. N. Kabir, "A Robust Image Watermarking Technique With an Optimal DCT-Psychovisual Threshold," *IEEE Access,* vol. 6, pp. 20464-20480, 2018.

[6] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans Image Process,* vol. 6, no. 12, pp. 1673-87, 1997.

[7] A. Obukhov and A. Kharlamov, "Discrete cosine transform for 8x8 blocks with CUDA," *NVIDIA white paper,* 2008.

[8] V. Potdar, S. Han, E. Chang, and C. Wu, "Subjective and Objective Watermark Detection Using a Novel Approach–Barcode Watermarking," in *International Conference on Computational and Information Science*, 2006: Springer, pp. 576-586.

[9] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 11, pp. 2594-2608, 2016.

[10] S. W. Hasinoff, "Photon, poisson noise," *Computer Vision: A Reference Guide,* pp. 608-610, 2014.

[11] F. Benzarti and H. Amiri, "Speckle noise reduction in medical ultrasound images," *arXiv preprint arXiv:1305.1344,* 2013.

[12] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun, "A stochastic approach to content adaptive digital image watermarking," in *International Workshop on Information Hiding*, 1999: Springer, pp. 211-236.

[13] I. J. Cox, M. L. Miller, and J. A. Bloom, "Watermarking applications and their properties," in *Proceedings International Conference on Information Technology: Coding and Computing (Cat. No. PR00540)*, 2000: IEEE, pp. 6-10.

[14] C.-H. Chou and C.-W. Chen, "A perceptually optimized 3-D subband codec for video communication over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, no. 2, pp. 143-156, 1996.

[15] X. Yang, W. Lin, Z. Lu, E. Ong, and S. Yao, "Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, no. 6, pp. 742-752, 2005.

[16] A. J. Ahumada and H. A. Peterson, "Luminance-model-based DCT quantization for color image compression," in *Human vision, visual processing, and digital display III*, 1992, vol. 1666: International Society for Optics and Photonics, pp. 365-375.

[17] X. Zhang, W. Lin, and P. Xue, "Improved estimation for just-noticeable visual distortion," *Signal Processing,* vol. 85, no. 4, pp. 795-808, 2005.

[18] Z. Wei and K. N. Ngan, "Spatial just noticeable distortion profile for image in DCT domain," in *2008 IEEE International Conference on Multimedia and Expo*, 2008: IEEE, pp. 925-928.

[19] A. B. Watson, "DCT quantization matrices visually optimized for individual images," in *Human vision, visual processing, and digital display IV*, 1993, vol. 1913: International Society for Optics and Photonics, pp. 202-217.

[20] G. Dimauro, "A new image quality metric based on human visual system," in *2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS) Proceedings*, 2012: IEEE, pp. 69-73.
[21] M. Barni and F. Bartolini, *Watermarking systems engineering: enabling digital assets security and other applications*. Crc Press, 2004.
[22] G. E. Legge and J. M. Foley, "Contrast masking in human vision," *Josa,* vol. 70, no. 12, pp. 1458-1471, 1980.
[23] G. E. Legge, "A power law for contrast discrimination," *Vision research,* vol. 21, no. 4, pp. 457-467, 1981.
[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing,* vol. 13, no. 4, pp. 600-612, 2004.
[25] C. Das, S. Panigrahi, V. K. Sharma, and K. K. Mahapatra, "A novel blind robust image watermarking in DCT domain using inter-block coefficient correlation," *AEU - International Journal of Electronics and Communications,* vol. 68, no. 3, pp. 244-253, Mar 2014.
[26] D. Levicky and P. Foris, "Implementations of HVS Models in Digital Image Watermarking," vol. 16, no. 1, 16, pp. 45-50, Apr 2007.
[27] P. Foris and D. Levicky, "Human Visual System Models in Digital Image Watermarking," vol. 13, no. 4, 13, pp. 38-43, Dec 2004.
[28] H. A. Peterson, A. Ahumada, and A. Watson, *An Improved Detection Model for DCT Coefficient Quantization*. 1993.
[29] Z. Wang, L. Lu, and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Transactions on Image Processing,* vol. 12, no. 2, pp. 243-254, 2003.
[30] C.-C. Lai, "An improved SVD-based watermarking scheme using human visual characteristics," *Optics Communications,* vol. 284, no. 4, pp. 938-944, 2011.
[31] S. P. Maity and M. K. Kundu, "DHT domain digital watermarking with low loss in image informations," *AEU-International Journal of Electronics and Communications,* vol. 64, no. 3, pp. 243-257, 2010.
[32] N. R. Pal and S. K. Pal, "Object-background segmentation using new definitions of entropy," *IEE Proceedings E (Computers and Digital Techniques),* vol. 136, no. 4, pp. 284-295, 1989.
[33] A. Ahumada, H. Peterson, and A. Watson, "An improved detection model for DCT coefficients quantization," *Human Vision, Visual Processing, and Digital Display IV. Allebach ed., SPIE,* 1993.
[34] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006: Springer, pp. 404-417.

9 Appendix

Derivation of Ahumada and Peterson equation for determining threshold visibility, $T_{u,v}$, and corresponding JND matrix, $Q_{u,v}$:

Given,

$$\log T_{u,v} = \log\left(\frac{T_{min}}{r + (1-r)\cdot\cos^2\theta}\right) + K\left(\log f_{u,v} - \log f_{min}\right)^2, \qquad m,n \in [0, N-1]$$

⇒

$$T_{u,v} = 10^{\log(T_{min}) - \log(r+(1-r)\cdot\cos^2\theta)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

⇒

$$T_{u,v} = \frac{10^{\log(T_{min})}}{10^{\log(r+(1-r)\cdot\cos^2\theta)}} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

⇒

$$T_{u,v} = \frac{T_{min}}{(r + (1-r)\cdot\cos^2\theta)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

And JND matrix, $Q_{u,v}$ can be computed as,

$$Q_{u,v} = \frac{T_{u,v}}{C(u)\cdot C(v)} = \frac{T_{min}}{C(u)\cdot C(v)\cdot(r + (1-r)\cdot\cos^2\theta)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

and,

$$\theta = \arcsin\left(\frac{2 f_{u,0} f_{0,v}}{{f_{u,v}}^2}\right), \cos^2\left(\arcsin\left(\frac{2\cdot f_{u,0}\cdot f_{0,v}}{{f_{u,v}}^2}\right)\right) = \left(\sqrt{1 - \frac{4\cdot {f_{u,0}}^2\cdot {f_{0,v}}^2}{{f_{u,v}}^4}}\right)^2 = \frac{{f_{u,v}}^4 - 4\cdot {f_{u,0}}^2\cdot {f_{0,v}}^2}{{f_{u,v}}^4}$$

⇒

$$Q_{u,v} = \frac{T_{min}}{C(u)\cdot C(v)\cdot\left(r + (1-r)\cdot\left(\frac{{f_{u,v}}^4 - 4\cdot {f_{u,0}}^2\cdot {f_{0,v}}^2}{{f_{u,v}}^4}\right)\right)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

⇒

$$(\ldots continued)$$

$$Q_{u,v} = \frac{T_{min} \cdot {f_{u,v}}^4}{C(u) \cdot C(v) \cdot \left((r \cdot {f_{u,v}}^4) + (1-r) \cdot \left({f_{u,v}}^4 - 4 \cdot {f_{u,0}}^2 \cdot {f_{0,v}}^2 \right) \right)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \mathrm{l} \quad (f_{min}))^2}$$

$\Rightarrow$

$$Q_{u,v} = \frac{T_{min} \cdot {f_{u,v}}^4}{C(u) \cdot C(v) \cdot \left({f_{u,v}}^4 - 4(1-r) {f_{u,0}}^2 {f_{0,v}}^2 \right)} \cdot 10^{\mathrm{K}(\log(f_{u,v}) - \log(f_{min}))^2}$$

and

$$f_{u,v} = \sqrt{{f_{u,0}}^2 + {f_{0,v}}^2}$$

$\Rightarrow$

$$Q_{u,v} = \frac{T_{min} \cdot {f_{u,v}}^4}{C(u) \cdot C(v) \cdot \left({f_{u,v}}^4 - 4(1-r) {f_{u,0}}^2 {f_{0,v}}^2 \right)} \cdot 10^{K\left(\log \sqrt{{f_{u,0}}^2 + {f_{0,v}}^2} - \log(f_{min}) \right)^2}$$

A1. 1: Derivation of equation 21

Ernawan's Embedding Stage

1. **Adopt an initial value for the embedding strength, i.e. $a = 1$.**
2. **Segment the input image into non-overlapping blocks of size 8 x 8 pixels.**
3. **FOR each block,**
   calculate and record its modified entropy according to equation 41, where M refers to the number of elements within the block and p_i denotes the relative occurrence probabilities of the i^{th}pixel within the block.
4. **Sort the entropy values of each block into ascending order and select the blocks with the N-lowest entropy scores, where N refers to the number of bits in the watermark.**
5. **Store the positions of each selected block.**
6. **Encrypt the binary watermark image.**
7. **FOR each selected block,**
   compute its DCT and embed a single bit of the watermark according to the embedding function in section A1.4.
8. **Perform the inverse DCT on each selected block and recombine with the remaining blocks to create the watermarked image.**
9. **Compute the SSIM of the watermarked image (SSIM).**
10. **Create a compressed copy of the watermarked image using JPEG with a quality factor of 50.**
11. **Extract the watermark from the compressed copy and compute its normalized cross-correlation value (NC).**
12. **WHILE (SSIM > NC),**
    increment the embedding strength, i.e. $a \rightarrow a + 1$ and repeat steps 7 – 11.

A1. 2: Ernawan's embedding stage pseudocode.

Ernawan's Extraction Stage

1. **Using the previously stored locations (step 5 of embedding stage), locate each selected block and transform into its DCT domain.**
2. **FOR each block,**
   perform the extraction function found in section A1.6 to determine whether a binary 1 or 0 is embedded.
3. **Combine each extracted bit to form the extracted watermark sequence.**
4. **Compute the normalized cross-correlation value (NC) of the extracted watermark and determine whether the true watermark has been detected.**

A1. 3: Ernawan's extracting stage pseudocode.

Ernawan's Original Embedding Function

```
FOR each coefficient pair, (X,Y)
        IF ( watermark bit == 1 )
                IF ( abs(X) < abs(Y) )
                        Z = X
                        X = Y + sign(Y)* α
                        Y = Z
                ELSE
                        X = X + sign(X)* α
                        Y = Y
                END
        ELSE ( watermark bit == 0 )
                IF ( abs(X) < abs(Y) )
                        X = X + sign(Y)* α
                        Y = Y
                ELSE
                        Z = X
                        X = Y
                        Y = Z + sign(X)* α
                END
        END
END
```

A1. 4: Ernawan's published embedding function algorithm.

Ernawan's Implemented Embedding Function

```
FOR each coefficient pair, (X,Y)
        IF ( watermark bit == 1 )
                IF ( abs(X) < abs(Y) )
                        Z = X
                        X = Y + sign(Y)* α
                        Y = Z
                ELSE
                        X = X + sign(X)* α
                        Y = Y
                END
        ELSE ( watermark bit == 0 )
                IF ( abs(X) < abs(Y) )
                        X = X
                        Y = Y+ sign(Y)* α
                ELSE
                        Z = X
                        X = Y
                        Y = Z + sign(X)* α
                END
        END
END
```

A1. 5: Ernawan's implemented embedding function.

Ernawan's Original Extracting Function

Within DCT domain of current block
Extract all 3 coefficient pairs, (X1,Y1), (X2,Y2), (X3,Y3),
Create sequences: A_K = [X1, X2, X3] and A_{K+1} = [Y1, Y2, Y3]
IF ($A_K < A_{K+1}$)
Watermark bit = 1
ELSE
Watermark bit = 0
END
END

A1. 6: Ernawan's published extracting function.

Ernawan's Implemented Extracting Function

Within DCT domain of current block
$C_1 = 0$
$C_0 = 0$
FOR each coefficient pair, (X, Y)
IF (|X| > |Y|)
C_1 ++
ELSE
C_0 ++
END
END
IF ($C_1 > C_0$)
Watermark bit = 1
ELSE
Watermark bit = 0
END
END

A1. 7: Ernawan's implemented extracting function.