Nat.Lab. Report rep 7254

Date of issue: November 2002

Analysis of Quantization based Watermarking

M. Staring

Authors' address data: M. Staring WY71; staringm@natlab.research.philips.com

Report:	rep 7254
Title:	Analysis of Quantization based Watermarking
Author(s):	M. Staring

Part of project:	VSP: 1996 - 355
Customer:	Digital Networks

Keywords:	Digital watermarking, quantization watermarks, QIM, DC-QIM, distortion compensation, SCS, adaptive quantization stepsize, error correcting codes
Abstract:	see page v - vi

Conclusions:	see page 95 - 97

This is the graduation report of

M. Staring,

student Applied Mathematics at the University of Twente (UT), at the chair Stochastic System and Signal Theory of the Systems, Signals and Control group.

This graduation project was carried out at Philips Research Laboratories in Eindhoven (Nat.Lab.) in the cluster Watermarking and Fingerprinting, from the group Processing and Architectures for Content Management (PACMan), from the sector Access and Interaction Systems (AIS). The project was done from December 2001 till November 2002.

The graduation committee consists of the following people:

Prof. dr. A. Bagchi, UT
Prof. dr. P.H. Hartel, UT
Dr. Ir. J.C. Oostveen, Philips Research
Dr. J.W. Polderman, UT

Philips Research Laboratories
Prof. Holstlaan 4
5656 AA Eindhoven
The Netherlands

University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

Summary

In order to protect (copy)rights of digital content, means are sought to stop piracy. Several methods are known to be instrumental for achieving this goal. This report considers one such method: digital watermarking, more specific quantization based watermarking methods. A general watermarking scheme consist of a watermark embedder, a channel representing some sort of processing on the watermarked signal, and a watermark detector. The problems related to any watermarking method are the perceptual quality of the watermarked signal, and the possibility to retrieve the embedded information at the detector.

From current quantization based watermarking algorithms, like QIM, DC-QIM, SCS, etc., it is known that the achievable rates are promising, but that it is hard to meet the required robustness demands. Therefore improvements of current algorithms are sought that are more robust against normal processing. This report focusses on two possible improvements, namely the use of error correcting codes (ECC) and the use of adaptive quantization.

Watermarking can be seen as a form of communication. Therefore, the robustness demand for watermarking is equivalent with the demand of reliable communication for communication models. Therefore, the use of ECC gives certainly an improvement in robustness. This is confirmed by experiments. Repetition codes are simple to implement and already gives a gain in robustness. The concatenation of convolutional codes with repetition codes gives an improvement only in the case of mild degradations due to the above mentioned processing.

In this report watermarking of signals with a luminance component are considered, like digital images and video data. Adaptive quantization refers to the use of a larger quantization step size for high luminance values, and a lower quantization step size for low luminance values. It is known from Weber's law that the human eye is less sensitive for brightness changes in higher luminance values, than it is in lower luminance values. Therefore, using adaptive quantization does not come at the cost of a loss of perceptual quality of the host signal. Adaptive quantization gives a large robustness gain for brightness scaling attacks. However, the adaptive quantization step size must be estimated at the detector, which potentially introduces an additional source of errors in the retrieved message. By means of experiments it is shown that this is not such a big problem. Therefore, adaptive quantization improves the robustness of the watermarking scheme.

It is valuable to know the performance of the watermarking scheme with the two improvements. The used performance measure is the bit error probability. The total bit error probability is build up from two components: One estimates the bit error probability for the case of fixed quantization, with an Additive White Gaussian Noise (AWGN) or uniform noise attack; The other estimates the bit error probability for the case of adaptive quantization, without any attack. Models for these two bit error probabilities are developed.

At the embedder the distortion compensation parameter α has to be set. The optimal value for this parameter is derived for the case of a Gaussian host signal and an AWGN channel. The value of

this optimal parameter α^* is compared with an earlier result of Eggers [17] and is shown to be identical. But whereas Eggers found a numerical function, which he numerically optimizes, our result leads to an analytical function, which can be optimized numerically.

So, we use two methods to improve robustness, namely the use of error correcting codes and an adaptive quantization step size. These two methods are shown to be improvements. Also an analytical model for the performance is derived, which can be used to verify analytically the robustness improvement.

Acknowledgements

”Everything should be as simple as it is, but not simpler.”
Albert Einstein (1879 - 1955)

First of all I would like to express my gratitude to Philips Research for the opportunity to work on an interesting and challenging subject like digital watermarking. I have enjoyed working at the Nat.Lab., for it is a stimulating environment where people can grow to their highest potential. In writing this report, I have tried to keep things as simple as possible; Einstein’s quote is very applicable when writing a report!

I would like to express my gratitude to my supervisors, who helped me reach the project goals: Job Oostveen and Ton Kalker from Philips Research and Jan Willem Polderman from the University of Twente. Very valuable where the many discussions I had with them, other members of the group and with fellow students; I really appreciated those discussions. Also, I would like to thank my parents, other family and friends for their support over the years.

Marius Staring
Eindhoven, November 2002

Contents

Chapter 1

Preface

In today's modern world, digital products are seen everywhere. The digitalization of all kinds of media, like digital audio and video, has a lot of benefits for the consumer. Copying and distributing digital content has become relatively simple with the introduction of computers and (fast) internet. But there are also some drawbacks. Analog video, for example, has a build in copy protection system. An analog copy results in quality degradation, and a copy of a copy of an analog video has degraded so much, that it is not a pleasure to look at. Digital video however doesn't have this build in copy protection system, because a copy of a digital video is identical to the original; There is no quality loss. Therefore it has become difficult for people to protect their intellectual properties, like the intellectual rights on a digital video.

Methods to prevent the illegal copying of digital video has been sought since. One way of preventing illegal copying is the use of digital watermarks. This is the subject of the report.

Watermarking for example digital video is changing the video sequence slightly, in such a way that at a later time, this slight change can be detected. For example, if a DVD recorder is equipped with such a detection device, it can search for a watermark in all video offered for copying. If a watermark is found, copying is denied. In this way digital watermarking can provide a copy protection mechanism.

Two of the most important demands that have to be met for watermarking digital content are the following. The slight change made to the content, has to be slight indeed; People should not be able to see or hear (in the case of audio) that an image, a video sequence, an audio stream, etc., is watermarked. The second demand is that the watermark should not get lost very easily. For example: modern televisions are utilized with image format buttons in order to switch between screen ratio's of 4:3 and 16:9. The watermark should still be detectable after such a switch. Other things a watermark should be able to resist are, for example, the broadcasting of video over satellites, conversions from one format to another (for example `wav` to `mp3` for audio), etc.

This report focusses on improvements of an existing watermarking algorithm in order to get better results for the above mentioned two demands. In Chapter 2 and 3 a general introduction to digital watermarking and to a specific existing watermark algorithm are given, respectively. Only in Chapter 4 a more detailed problem description is given, because at that point the necessary terminology has been derived. In later chapters, possible solutions to the problems stated in Chapter 4 are derived.

In this report some abbreviations are used which the reader might not be familiar with. These abbreviations, together with the notation used in this report, are listed in Appendix A and the

Index, see page 99.

Chapter 2

Introduction

In this chapter digital watermarking is defined (see Section 2.1) and some of its applications are given in Section 2.2. In Section 2.3 criteria are given to judge the quality of a watermarking process. Some characteristics of the Human Visual System (HVS) are given in Section 2.4 and in Section 2.5 some of the possible attacks on digital watermarks are described. Finally, in Section 2.6, possible methods to put a watermark in digital content are considered.

For the general principles of digital watermarking the reader is referred to the excellent book of Cox, Miller and Bloom, see [14].

2.1 What is digital watermarking?

In order to understand what digital watermarking is, it is good to understand the principle of hiding information, because digital watermarking can be seen as a part of Information Hiding. In Subsection 2.1.1 Information Hiding is treated. The general principles of watermarking are discussed in Subsection 2.1.2. In Subsection 2.1.3 an analogy with banknotes is drawn.

2.1.1 Information hiding

Information Hiding is the process of sending information from one party to another, such that either the message or the identity of the sender and the receiver is 'hidden'. Hidden means that the message can not be read or removed by a third party, either because they don't know the existence of the hidden message or they are not capable of doing this.

Already in the early days of history, in the time of the ancient Greek, people used information hiding techniques. One example (see [27]) tells about Histiaeus who informed his allies about the exact moment to revolt against the Persians, by shaving the head of a trusted slave and tattooing a message on his head. The hair of the slave grew back on and the slave was send through enemy territory, looking like an innocent traveller. On his arrival, the slave just told the allied leader to shave his head an read the message hidden there.

According to [27] Information Hiding can be divided into four categories:

Covert channels: Covert channels are communication channels that can be exploited by a process to transfer information in a manner that violates the systems security policy. So, for

example, information is carried over unintentionally by a computer program, to a user of the system.

Steganography: Steganography comes from the Greek words *steganos* and *graphie*, meaning ”covered” and ”writing”. It comprehends the hiding of a secret message into a host message, such that nobody suspects its existence.

Anonymity: Anonymity is the communication of a message in which not the message itself is the secret, but the sender or the receiver or both.

Watermarking: Watermarking is steganography plus a robustness (see Section 2.3) requirement; Watermarking refers to the hiding of a message in a host message in such a way that if this signal is altered, the hidden message still survives if the host survives.

So watermarking can be seen as a form of hiding information. Steganography and watermarking are somewhat alike, because both methods have the goal to hide a message in a host signal. There are however three differences between watermarking and steganography:

1. Watermarking considers information about the object, the host signal, where for steganography the hidden message can be anything.

2. The robustness criteria are different. Steganography has the objective that information must remain hidden, where watermarking has the objective that the information must not be removed, altered or damaged, even if the watermarking algorithm is known to the adversary.

3. Usually steganography is one-to-one and watermarking one-to-many; For steganography there is usually one sender and one receiver, and for watermarking there is usually one sender and many receivers.

2.1.2 Digital watermarking

Digital watermarking is the watermarking of digital content, like digital images, digital audio or digital video. This digital content is often referred to as the *host signal* or the *host data*.

Digital watermarking is the process of making small adjustments to the host signal, in such a way that these adjustments cannot be perceived by humans. These small changes represent in some way the information one wishes to hide. The changes are called the *watermark*. The host signal together with the watermark adjustments is called the *watermarked signal*. If the host signal is represented by x and the watermarked signal by y, then the watermark, represented by w, is defined as $w \triangleq y - x$, i.e., the difference between the watermarked and the host signal. Unless stated otherwise the signal are supposed to be N-dimensional vectors in the real space. So $x = (x_1, \cdots, x_N)$, $y = (y_1, \cdots, y_N)$, with $x_i, y_i \in \mathbb{R}$, $i \in \{1, 2, \cdots, N\}$.

The process of watermarking a host signal is usually referred to as *embedding* a message in the host signal. The process of reading the message at the receiver side is usually referred to as *detecting* a message from the received signal. The devices that embed or detect a watermark are called the *embedder* and the *detector*, respectively.

Watermarking can be seen as a form of communication. A general communication model is depicted in 2.1. After a message m is embedded into a host signal x, the watermarked signal y is sent over some channel to the receiver. On the path from the sender to the receiver processing

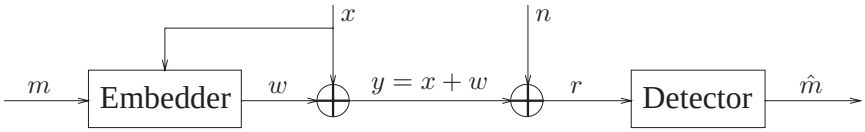


Figure 2.1: A general watermarking process. A message m is embedded in a host signal x. The embedder uses the host signal to make a watermark w. The watermarked signal is given by $y = x + w$. After the embedding, attacks take place on the watermarked image, here in the form of additive noise n. At the detector the signal $r = y + n = x + w + n$ is received. The watermark is extracted from r and then decoded to get a message $\hat{m}$, which is hopefully the same as the embedded message m.

of the watermarked signal will take place. This processing can be anything, ranging from normal processing to deliberate attacks. In Figure 2.1 this processing is depicted with additive noise n. Processing alters the watermarked signal such that at the detector a signal r is received.

Normal processing is considered to be all the processing that is regularly done on the signal, for instance compression or perceptual improvement. Two other examples of this 'normal' processing are:

- Modern tv's are equipped with a scaling button to go from one format to another. This scaling is supposed to be possible without destroying the watermark.
- If a movie is broadcasted using satellites, the film is first compressed, then sent to a satellite and back to earth, and then decompressed. The compression - decompression combination introduces a distortion to the movie signal. The watermark is supposed to withstands these processing steps.

A *deliberate attack* is a form of processing that has the goal to damage the watermark such that the sent message can not be retrieved at the detector. However, in this report both kinds of processing are denoted by the word attack.

So, digital watermarking enables one to send information together with the host signal, the digital content, and to extract this information afterwards. It can be seen from Figure 2.1 that a digital watermark is subjected to the same attack as the host signal.

2.1.3 Analogy with watermarks in banknotes

From everyday live it is known that watermarks are used in bank notes. This watermark is only visible under the special operation of holding it to the light; This is the way to detect the watermark. Seeing the watermark tells us that this bank note is trustworthy. There is an analogy with digital watermarks. The presence of a digital watermark is only supposed to be detectable by using the appropriate watermark detector.

For most applications it should be difficult to remove the watermark. Removal of the watermark from a bank note is likely to destroy the bill in the process. The same is required for digital watermarks. Depending on the application it is desirable for the digital watermark to survive common signal processing or sometimes even deliberate attacks, similar to an attempt to remove a watermark from a bank note.

For a bank note it should not be possible to recreate the watermark and in doing so being able to create trustworthy bank notes. This is also required for digital watermarks. It should not be possible for unauthorized persons to detect, extract or insert a digital watermark from or in audio-visual content. These aspects are referred to as the *security* of a digital watermark.

These, and more, requirements for digital watermarks are discussed in more detail in Section 2.3.

2.2 Applications

Digital watermarking can be used for a variety of applications. In [13, 22, 27] a number of them are given:

Proof of ownership: It is possible to embed a watermark in digital content representing copyright information. In this case the information might prove who is the copyright owner of the digital content. It should be a hard task to remove the watermark.

Fingerprinting: It is possible to embed a watermark representing a customer identifier. If an illegal copy of the digital content is found, the copyright owner is able to trace it back to the customer by reading the watermark on the illegal copy.

Copy Protection: It is possible to embed a watermark in audio-visual content representing a copy status (e.g. copy once, free to copy, copy never). A watermark detector reads this copy status message and depending on the nature of this message the digital recorder can decide whether or not to copy the content.

Broadcast Monitoring: In any broadcasted media, like commercials or television programs, it is possible to embed a watermark. A watermark detector can automatically check if this media item was actually broadcasted. ”Other applications include verification of commercial transmissions, assessment of sponsorship effectiveness, protection against illegal transmission, statistical data collection, and analysis of broadcast content [27].”

Data Authentication or Tamper Proofing: It is possible to use a fragile watermark, see Subsection 2.3.2, to determine if digital content is altered, by checking whether or not the watermark is altered. In this case the watermark should be robust against small modifications, but not against larger modifications. An example is the removal or insertion of a person into an image of a crime scene: the watermark should not be robust against this kind of modifications, where it should be robust against minor modifications like lossy compression.

Indexing or Feature Tagging: The watermark can also represent additional information about the content, like comments, captions or information useful for search engines (keywords). The security requirements are not relevant in this case, because indexing is considered a service, so it is unlikely that someone wants to remove the watermark.

Medical safety: In order to not accidently combine an x-ray image with the wrong patient, it is possible to embed a patients name on the x-ray image. In this case high demands have to be met with respect to the imperceptibility of the watermark, because a doctor does not want any uncertainty whether something on his x-ray image is a tumor or just some watermark artefact.

Data hiding: A non-perceptible message embedded in digital content can contain secret or just hidden information. It is for example possible for terrorists to communicate secretly using digital images on an internet-site.

The first two applications together are usually referred to as *copyright protection*.

2.3 Performance criteria and characteristics

In order to judge the quality of a watermark embedding algorithm, some criteria are given. There are several criteria, some of them already mentioned. Depending on the application some properties are more important than others. For example for medical safety applications (see Section 2.2) it is important that the watermark is absolutely imperceptible, but not necessarily robust against signal processing. For copy protection the imperceptibility properties are still high, but not that absolute. The robustness demands are very high in this case.

The following characteristics of watermarking algorithms are considered: imperceptibility, robust and fragile watermarks, false positive and false negative probabilities, payload and capacity, security, computational cost, blind and non-blind detection. These criteria are described in Subsection 2.3.1 - 2.3.7. In Subsection 2.3.8 the relations between those criteria are given.

2.3.1 Imperceptibility

A first criterion is that of the already mentioned *imperceptibility* of the watermark. A watermark is truly imperceptible if humans cannot distinguish the original from the watermarked data if they are laid side by side. Sometimes this is relaxed to the condition that one cannot see the watermark if the original data is not available for comparison. Imperceptibility is also referred to as *perceptual transparency*.

In practical situations it is sometimes necessary to allow some amount of perceptibility of the watermark. But how to measure this amount? The best way would be to use the human senses to determine the perceptibility, because this is the ultimate criterion for deciding whether or not a watermark is perceptible for humans. Unfortunately, this is not convenient for practical purposes. Therefore, some sort of objective measure of perceptibility is used. The best objective measures are those that imitate the human senses. Unfortunately, these measures are really complex, because the models of the Human Auditory System (HAS), for audio, or the Human Visual System (HVS), for still images or video, are complex. Therefore three rather simple measures to establish the perceptibility of a watermark are used: the Signal-to-Noise Ratio (SNR, the watermark is seen as noise), the *Mean Squared Error* (MSE) and for still images the Watson-metric. Because this report is focused on still images, a description of the HVS is given in Section 2.4.

The idea behind the use of the *Signal-to-Noise Ratio* is that a watermark is less perceptible if its energy is low compared to that of the host signal. The SNR is defined as:

$$\mathrm{SNR} \triangleq 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_w^2} \right), \tag{2.1}$$

where x is the host signal (the digital content), w the watermark and σ^2 the variance, which represents the energy. The variance of x, σ_x^2, is calculated as usual as:

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \tag{2.2}$$

where $\bar{x}$ is the mean value of x. σ_w^2 is calculated in the same way. Another distortion measure is the MSE, which is defined as

$$\text{MSE} \triangleq \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2 = \frac{1}{N}\sum_{i=1}^{N} w_i^2. \tag{2.3}$$

The SNR and the MSE are more a measure of distance than of perceptual distance.

The Watson-metric is more like a measure of the perceptual distance. This distance is calculated using the Watson model which takes into account three factors: contrast sensitivity, luminance masking and contrast masking. This metric is a little more sophisticated than just using the SNR, because it takes into account some aspects of the Human Visual System, see Section 2.4. For a description of the Watson model see Section 2.3 of [26].

Sometimes also visible (but not disturbing) watermarks are used, but we will not consider those kind of watermarks. We will focus on imperceptible watermarks.

2.3.2 Robustness

Another important criterion already mentioned is that of the *robustness* of a certain watermarking algorithm. According to [9], a truly robust watermark is a watermark that survives signal processing whenever the host signal does. In other words, robustness is the ability to withstand normal processing of the watermarked signal.

Two criteria for robustness are used in this study, robustness against noise and against lossy compression. Robustness against noise is measured as follows. White Gaussian noise is added to the watermarked image and it is checked if it is still possible to detect the embedded information. The maximum amount of noise added, after which it is still possible to detect the information, can be considered as a measure for robustness against noise. Also the bit error rate at certain values of noise power can be considered as robustness measure. In order to check robustness against lossy compression techniques, JPEG compression on the watermarked image is performed. The minimum JPEG quality factor used, after which it is still possible to correctly detect the watermark, is considered as a measure for robustness against lossy compression.

It is possible to determine the probability that after a noise attack a detected message bit is different from the embedded message bit. This bit error probability as a function of the noise energy can also be seen as a measure for robustness against noise.

There are also other types of digital watermarks, like fragile ones. A *fragile watermark* is the opposite of a robust one. If the host data is somehow altered, this should be visible in the detection result. Sometimes it is important to prove that a photograph of e.g. a crime scene is not altered. This could be proved in court by showing that the fragile watermark has not been changed.

Because in most applications the watermark should be robust, we do not consider fragile watermarks, but robust ones.

2.3.3 False positive and false negative probabilities

A *false positive* is a detection of a watermark while no watermark is embedded. The *false positive probability* is the probability that a false positive occurs. A *false negative* is no detection while a watermark is embedded. The *false negative probability* is the probability of such a missed detection. The relevance of these probabilities depends on the application. For the case of DVD video copy protection it should never happen that a DVD recorder refuses to copy something the user has made himself (and is therefore not watermarked, at least not with the correct watermark). Therefore the false positive probability should be very low in this case. The false negative probability is in this case not as important as the false positive probability; It is not a really big problem if a user makes an illegal copy every now and then, as long as this is within limits. In general the false negative probability is important for most applications, because this probability represents the robustness of a watermark.

2.3.4 Payload and Capacity

The *payload* is the amount of information that can be embedded in the host data. The *capacity* is defined as the amount of information that can be embedded and detected without errors. These amounts depend on the host data and on the watermarking algorithm. Directly related to capacity is the *rate*. The rate is defined as the capacity divided by the total length N of the host signal.

2.3.5 Security

The *security* of a watermark relates to the (in)ability to detect, remove or insert a watermark from or in the host signal. Using *Kerckhoffs' principle* [21], a watermark should still be secure if the watermarking algorithm is known to the adversary. Security must lie in the use of a secret key. Security is the ability to withstand active (deliberate) attempts to disable the communication through the watermark channel. Security also relates to the total watermarking system. If it is possible for example to bypass the hardware in a DVD recorder that detects the watermark in a DVD, then the system is not secure, although the watermarked secret has not been compromised.

2.3.6 Computational cost

The *computational cost* is the effort it takes to embed or detect a watermark. This can be measured in time, like clock cycles of a computer, but also in the need of extra memory capacity (thus making an application more expensive). The importance of this criterion varies per application. For example for broadcast monitoring the detection must be done in real-time, so the speed is important, but for copyright protection the speed is not important.

2.3.7 Blind detection

At the detector the original data may or may not be available for comparison with the received data. The absence of the original data at the detector is referred to as *blind detection*. It is clear that with non-blind detection it is easier to check for a watermark. In practice, however, the original data is usually not known at the detector. We will therefore focus on blind detection.

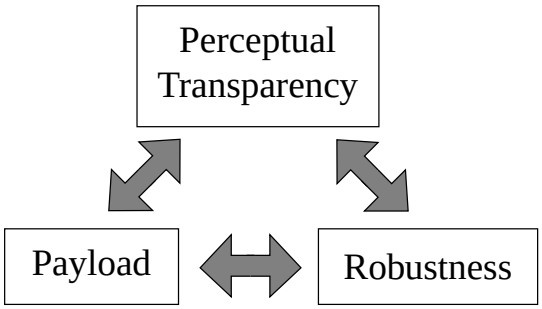


Figure 2.2: Mutual dependencies between the performance criteria. Partially reproduced from [22].

2.3.8 The trade-off between performance criteria

In order to make a watermark very robust, very large modifications to the host data are needed. But, in doing so the watermark is made perceptible. This is an important observation, because it is telling us that for an optimal watermark a trade-off has to be made between the performance criteria. A large payload will come at the cost of reduced robustness and perceptual transparency. The relationships between the criteria are shown in Figure 2.2.

2.4 The Human Visual System

In order to measure the impact of a watermark in terms of perceptibility, some knowledge of the *Human Visual System* (HVS) is needed. Another reason to consider the HVS is that it gives information about the regions in which the human eye is insensitive. Those regions are a perfect option to embed the information in, because larger modifications are possible, without the human eye detecting them. In this section some basic characteristics of the HVS are treated. See [10, 14, 29, 30] for more on this subject.

The sensitivity of the human eye for certain stimuli is treated in Subsection 2.4.1. The fact that certain stimuli can be masked with others is considered in Subsection 2.4.2. Pooling is treated in Subsection 2.4.3.

2.4.1 Sensitivity

The two most important sensitivity characteristics for the human eye are brightness and frequency sensitivity. *Brightness sensitivity* refers to the fact that the human eye is less sensitive to brighter signals, i.e., it will take a larger difference in bright signals than in signals with low luminance in order for humans to perceive it. Therefore it is possible to embed more of the watermark in regions with higher brightness. This principle is depicted in Figure 2.3.

A just noticeable difference is the amount two stimuli need to differ by in order for the difference to be perceived. *Weber's law* states that these just noticeable differences are a function of the percentage change in stimulus intensity not the absolute change in stimulus intensity. This means for example that the perceived difference between the light intensity of 10 and 11 candles in a room is not the same as the perceived difference of 100 and 101 candles, but rather the difference of 100 and 110 candles. So the bigger the stimulus the bigger a change required for it to seem

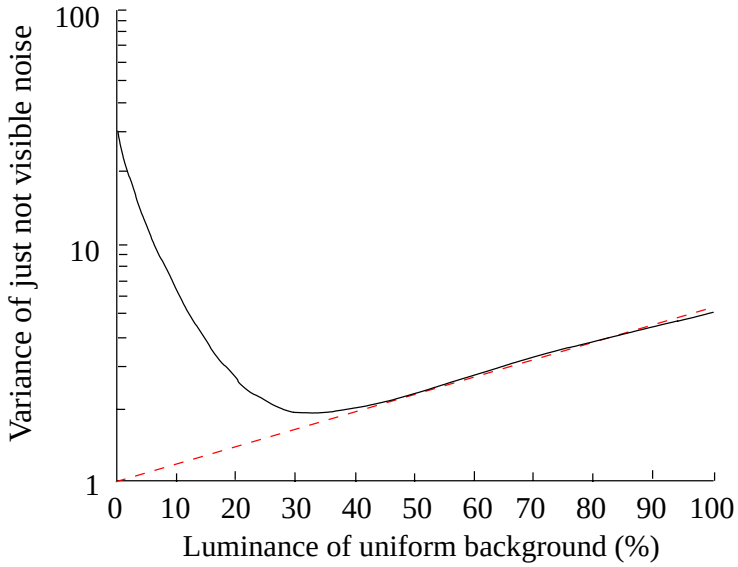


Figure 2.3: Reproduced from [29]. The effect of background luminance on the maximum contrast of a just-not-visible noise pattern. Zero luminance corresponds to a black background on a CRT display; a luminance level of 100 % corresponds to a white background on a CRT display. At luminance levels around 30 % the human eye is most sensitive to noise patterns. The straight dashed line through the origin represents Weber's law.

different. In order to perceive light differences the percentage two stimuli should differ is around 6 %. A mass difference can be detected by humans if it is more than 2 % and a sound frequency difference if it is more than 0.3 %. So the human ear is much more sensitive than the human eye, therefore it is more difficult to embed information in audio than in video. As can be seen from Figure 2.3, Weber's law holds for luminance levels over 30 %.

Frequency sensitivity can be divided into three forms for vision: spatial, spectral and temporal frequency sensitivities. Spatial frequencies are perceived as patterns or textures, spectral frequencies as colors and temporal frequencies as motion or flicker.

Spatial sensitivity is usually described by the sensitivity to contrast in luminance as a function of spatial frequency, the result is the contrast sensitivity function, see Figure 2.4. It can be seen that the human eye is most sensitive to luminance differences in the mid-frequencies and less sensitive for the higher and lower frequencies.

The *frequency sensitivity* is illustrated with Figure 2.5. The normal responses to the low, the middle and the high frequencies, often called the blue, the green and the red channel, respectively are depicted. The human eye is significantly less sensitive for blue frequencies than it is for the red and green frequencies. Therefore, some watermarking methods embed a large proportion of the watermark in the blue channel of an RGB image.

Figure 2.6 shows the *temporal sensitivity* for the human eye. It can be seen that for frequencies over 30 Hz, temporal sensitivity falls rapidly. This is why television and cinema frame rates are not necessarily to be more than 60 frames per second. The total theory for temporal sensitivity is very difficult, because the human eye follows moving objects. Temporal and spatial frequencies are therefore partially converted into each other. A static watermark that is imperceptible in a single video frame, can therefore be perceptible if more frames are shown after each other, like in

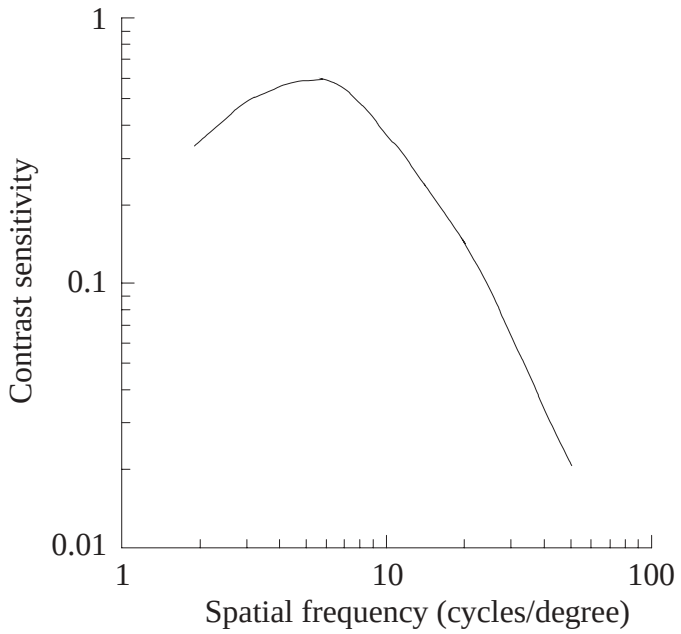


Figure 2.4: Reproduced from [29]. The contrast sensitivity function in order to illustrate the spatial sensitivity of the HVS.

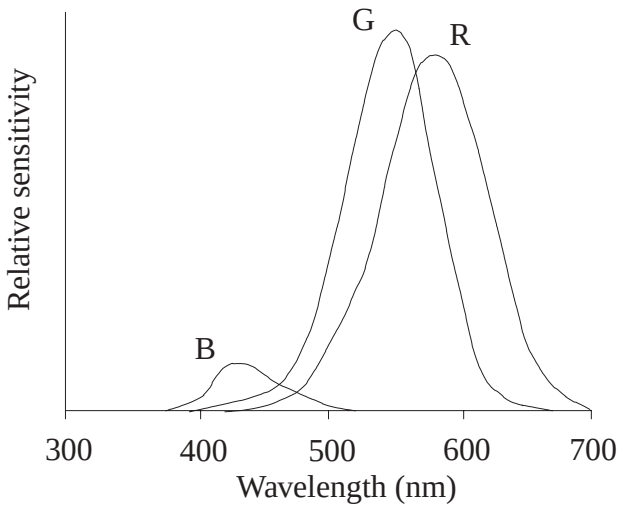


Figure 2.5: Reproduced from [14]. The sensitivity of the human eye for color, for the red (R), green (G) and blue (B) channel.

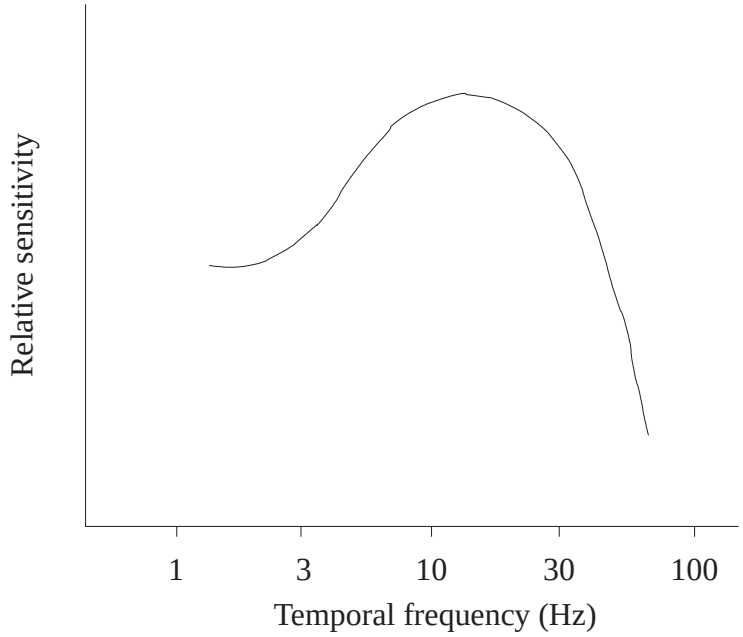


Figure 2.6: Reproduced from [14]. The sensitivity of the human eye for temporal frequencies.

video. This effect is called the dirty window effect.

2.4.2 Masking

Signals can be *masked* by other signals. This means that the presence of one signal, may render another signal imperceptible. For example, we may hear a single tone with some intensity at 1000 Hz, but we may not hear that same tone if at the same time there is a tone at 1050 Hz with a higher intensity. Another example is that it is easy to see noise in a part of an image where everything is smooth and flat, but the same amount of noise in a highly textured area of an image might be imperceptible for the human eye.

Three of the most important masking principles are brightness or contrast masking, frequency masking and temporal masking. *Brightness masking* refers to the fact that local brightness is able to mask contrast changes. *Frequency masking* is the fact that one frequency can mask another. The perception of a sound can be masked with a previous or even a future sound, this is called *temporal masking*.

A good area to embed a watermark in an image are highly textured areas, because the watermark noise is masked by the patterns of the image.

2.4.3 Pooling

Once one has developed models for estimating the effect on perceptibility of changing certain characteristics (like changing pixel values, frequencies, etc.), it is necessary to combine all these characteristics in order to come to an overall indication for the perceptibility of a watermark in an image. Combining all the separate perceptibility changes in one estimate is called *pooling*.

2.5 Possible attacks on digital watermarks

As already mentioned, watermarks need to be robust against all kinds of attacks. A distinction has to be made between attacks directly on the watermark and targeted against other components of the watermarking system. For example, if it is relatively simple to remove or disconnect a watermark detection chip in a DVD-player, an attacker will focus his attention to this part of the system in order to reach his goal (in this case: to copy a DVD). In this section those kinds of attacks are not considered, but the attention is focused on direct attacks on the watermarked data. An arrangement of attacks is given in Subsection 2.5.1 and some of the most common attacks are mentioned in Subsection 2.5.2 - 2.5.4.

Some of the attacks mentioned in this section are considered to be normal processing, where others are deliberate attacks.

2.5.1 Arrangement of attacks

Setyawan [32] gives a description and classification of different attacks. An overview can be extracted from his article and is given in Figure 2.7.

From this figure it can be seen that a distinction is made between attacks aimed at the watermark itself and attacks aimed at the host signal alone. The latter one can for example be used by an adversary wanting to change a watermarked image of a crime scene, because this image identifies him as the perpetrator. His goal will be to change the watermarked image in such a way that he can no longer be identified as the guilty person, but that it is not visible in the watermark that the image has been altered.

Attacks on watermarks can be divided into three cases, namely attacks with the goal to remove the watermark, synchronization attacks and ambiguity attacks, see Subsection 2.5.2, 2.5.3 and 2.5.4 respectively.

2.5.2 Removal attacks

Removal attacks have the goal to completely remove a watermark or damage a watermark to an extent that the watermark detector cannot detect it anymore. Generally, there are two types of removal attacks: one that processes the watermarked signal without analyzing it, and one that analyzes the watermarked signal in order to remove the watermark.

The first category, simple removal attacks, works both on the host signal and the watermark, and it affects the quality of both. Because the watermark energy is usually much lower than the host signal energy, the watermark is faster degraded than the host signal.

The second category, the analysis removal attacks, tries to analyze or estimate the watermark. After this analysis it is attempted to remove the watermark. These kinds of attacks are a lot smarter than simple attacks, and they usually affect the watermark, but keep the host signal more or less intact.

Examples of simple removal attacks are

Lossy compression: Examples of lossy compression techniques are JPEG (Joint Photographic Experts Group: named after the committee that defined it) and MPEG (Motion Pictures Experts Group: data compression standard for motion-video and audio) compression. These

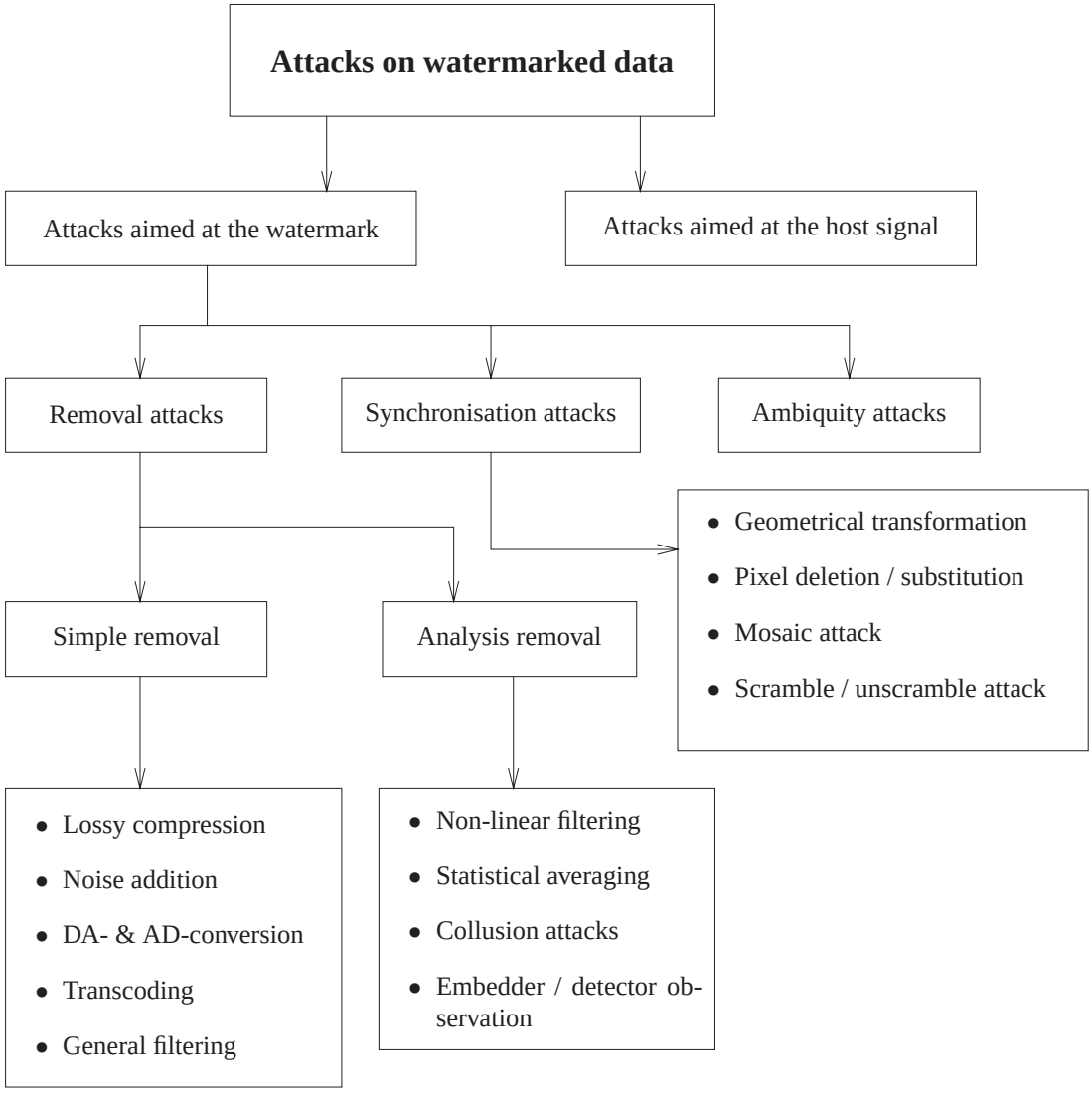


Figure 2.7: An overview of possible attacks on a watermarked signal. This overview is based on [32]. Some of the attacks are normal processing and other deliberate attacks.

techniques usually work on the parts of the data that are less important for the perceptual quality of the signal. Watermarking techniques usually also work in these parts, therefore it is likely that the watermark gets damaged due to lossy compression techniques.

Noise addition: The added noise is usually random and uncorrelated with the host signal and, more importantly, the watermark. Therefore it is unlikely that the noise affects the detection of the watermark, unless the power of the noise is very high. In this case the perceptual quality of the watermarked signal is highly degraded, which is usually also unwanted by an attacker.

DA- & AD-conversion: Digital-to-Analog and Analog-to-Digital conversion takes place for example if a digital image is printed and then scanned again, or when a digital movie is recorded on an analog videotape. In these cases the watermarked signal, and also the watermark, is not completely the same as it was in the digital case. Some embedding techniques, like LSB modification (see Subsection 2.6.2), are not robust against these degradations.

Transcoding: The process of going from one representation of data to another is called transcoding. Examples are going from a bitmap (BMP) image file to a GIF file (Graphics Interchange Format) and re-encoding an MPEG-stream into a higher or lower bit rate. This affects the watermarked signal and the watermark could be damaged or even lost.

General filtering: General filtering techniques can be used to remove a watermark. Low pass filtering for example, might be able to remove a pseudo-random noise watermark, since the watermark is essentially a high frequency noise [32].

Examples of analysis removal attacks are

Non-linear filtering: Using these techniques watermarks can be estimated and a watermark can be removed by subtracting this estimate from the watermark data. In this way an estimate from the unwatermarked host signal is obtained.

Statistical averaging: If an attacker has a number of signals (images, video-frames), all watermarked with the same watermark, it is possible to average all images and extract an estimate of the watermark from this average. This only works if the watermarks are not dependent of the host signal.

Collusion attacks: In this case there are a number of copies of one host data, watermarked with different watermarks. Combining all these copies, it is possible to estimate the original host signal by statistical averaging.

Embedder / detector analysis: An attacker possessing a watermark detector is able to look for the smallest modifications made on the watermarked signal, which renders the watermark undetectable. An attacker possessing a watermark embedder is able to watermark unwatermarked material and obtaining the watermark by taking the difference. After this it is possible to subtract the watermark from the original signal. If this modified host signal is then watermarked, the result is approximately the same as the original unwatermarked material.

2.5.3 Synchronization attacks

In the case of *synchronization attacks*, an attacker will not try to remove the watermark, but just to remove the synchronization with the detector, and in this way making a watermark undetectable. The quality of the image is to a large extent unchanged.

Examples of synchronization attacks are

Geometrical transformation: By simply shifting an image a few pixels to some direction (translation), the watermark detector is unable to find the watermark (although it is still there), while the image remains largely unchanged. Other examples of geometrical transformations are scaling, zooming, cropping and rotating.

Pixel deletion / substitution: It is possible to render a watermark undetectable by removing a complete column of an image, or replacing it by another column. The image is still largely the same after this operation.

Mosaic attack: With this attack an image is divided into smaller blocks. The detector will usually fail to detect a watermark in this smaller portion. The total image can still be represented by holding the smaller blocks together, like a mosaic.

Scramble / unscramble attack: It is used to bypass copy protection schemes in for example DVD-players. First the watermarked and illegal copy of a video is scrambled (mixed), this scrambled copy is played on a DVD-player, which doesn't recognize the watermark. Then, the copy is unscrambled in order to obtain the original copied video.

2.5.4 Ambiguity attacks

In the case of *ambiguity attacks* the attacker simply tries to embed another watermark in the already watermarked image. This way it is difficult for the detector to detect the original watermark.

2.6 Different watermarking techniques

Embedding a watermark can be done in a number of different ways. First of all it is possible to change directly the pixel values of an image or a video frame, this is called embedding in the spatial domain. It is also possible to find a representation of a host signal in another domain and perform modifications on the coefficients in that domain. Examples of those domains are the Discrete Fourier Transform (DFT) domain, the Discrete Cosine Transform (DCT) domain and the wavelet domain, see Subsection 2.6.1.

When embedding a watermark in the DCT-domain, first the Discrete-Cosine-Transform is taken from the host signal, and then modifications are mode on the DCT-coefficients. The same holds for the other domains.

Making these modifications can also be done in a number of different ways. Well known techniques are Least Significant Bit (LSB) modification, noise-addition, Spread Spectrum techniques (for example the Patchwork algorithm), the reordering or deletion of coefficients, warping or morphing data parts, etc., see Subsection 2.6.2 - 2.6.4. Other watermarking processes are quantization techniques like Quantization Index Modulation (QIM), see Chapter 3.

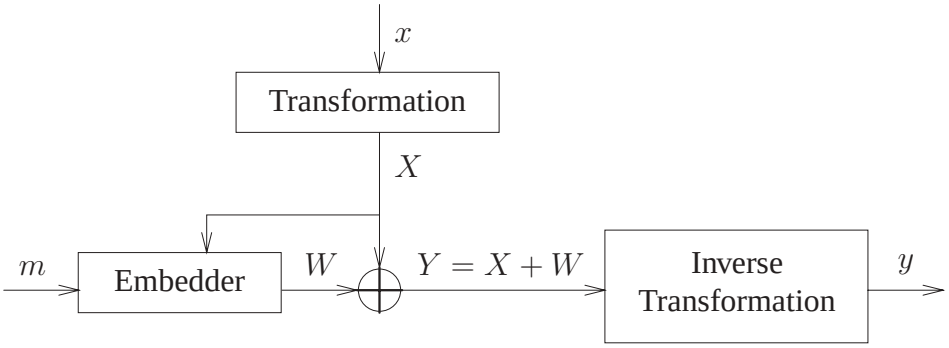


Figure 2.8: The embedding of a watermark in a transform domain. Representations in the transform domain are denoted with capitals.

2.6.1 Domain of embedding

To embed a watermark in another domain than the spatial one, it is necessary to first make a transform to that other domain. Then the watermarking is done in this transform domain, after which an inverse transform is done in order to get back to the spatial domain. This is after all the domain humans can see things in. This watermarking scheme is illustrated with Figure 2.8.

Some of the most important transformations are the Discrete Fourier Transform (DFT), the Discrete Cosine Transform (DCT) and the Mellin-Fourier Transform. All these transforms only give information about the frequency of a signal. A quite new concept is that of the Discrete Wavelet Transform (DWT). In the DWT-domain there is not only information available about the frequency aspects of a signal, but also about the time or spatial aspects. Using the DWT it is possible to control both aspects of a signal, while watermarking it. From [27] we have:

Discrete Fourier Transform (DFT): The Discrete Fourier Transform is very well known in the area of signal processing. It is useful for controlling the frequency aspects of a host signal and the watermark. For a host signal $f(x, y)$ of size $N_1 \times N_2$, the DFT $F(u, v)$ is given by:

$$F(u,v) \triangleq \beta \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) \exp\left(\frac{-2\pi i x u}{N_1} - \frac{-2\pi i y v}{N_2}\right), \tag{2.4}$$

with $\beta = (N_1 \cdot N_2)^{-1/2}$. The inverse DFT (IDFT) is given by

$$f(x,y) \triangleq \beta \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} F(u,v) \exp\left(\frac{2\pi i x u}{N_1} + \frac{-2\pi i y v}{N_2}\right). \tag{2.5}$$

Discrete Cosine Transform (DCT): The Discrete Cosine Transform is used in lossy compression techniques, like JPEG and MPEG. It is also used in studies on visual distortions. Because of this, using the DCT gives more robustness to JPEG and MPEG compression, the perceptual quality of a watermarked signal is more easily calculated, and it is possible to directly embed a watermark in the compressed domain (which saves computation time and thus costs). The DCT for a signal $f(x, y)$ of size $N_1 \times N_2$ is given by:

$$F(u,v) \triangleq \frac{2}{\sqrt{N_1 N_2}} C(u) C(v) \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) \cos\left(\frac{\pi u (2x+1)}{2N_1}\right) \cos\left(\frac{\pi v (2y+1)}{2N_2}\right). \tag{2.6}$$

where $C(u) = \frac{1}{2}$ for $u = 0$ and $C(u) = \frac{1}{2}\sqrt{2}$ otherwise. The inverse DCT (IDCT) is given by

$$f(x,y) \triangleq \frac{2}{\sqrt{N_1 N_2}} \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} C(u)C(v)F(u,v) \cos\left(\frac{\pi u(2x+1)}{2N_1}\right) \cos\left(\frac{\pi v(2y+1)}{2N_2}\right). \tag{2.7}$$

Fourier-Mellin Transform: Most watermarking algorithm are not (very) robust against geometrical attacks, see Subsection 2.5.3. The *Fourier-Mellin Transform* is based on the translation invariance property of the Fourier transform:

$$f(x_1 + a, x_2 + b) \leftrightarrow F(u,v) \exp\left(-i(au + bv)\right). \tag{2.8}$$

Use of this property, makes the watermark robust against a spatial shift. Robustness to rotation and scaling (zoom) can be achieved by using a log-polar mapping:

$$(x,y) \mapsto \begin{cases} x = \exp(\rho)\cos(\theta) \\ y = \exp(\rho)\sin(\theta) \end{cases} \quad \text{with } \rho \in R \text{ and } \theta \in [0, 2\pi], \tag{2.9}$$

where R is the set of possible scale-factors. This way, a rotation results in a translation in the logarithmic coordinate system and a zoom results in a translation in the polar coordinate system.

2.6.2 Least Significant Bit (LSB) modification

Pixels in an image or a video frame are usually represented by an 8-bit number (0 - 255 in $\mathbb{Z}_{255}$). Modifying the last bit from a pixel results in the addition of ± 1 in $\mathbb{Z}_{255}$. This modification is so small, that it is very difficult to see in the image or video frame. Because modifying this last bit introduces the smallest possible distortion, it is called the Least Significant Bit (LSB). A very simple way of embedding a binary message in an image is by modifying the LSB such that it is equal to the message bit. This way it is possible to embed a message bit in every pixel, so the payload of this method is very high. But the drawback is that it is not very robust: changing the LSB's by a random pattern, completely removes the watermark. Therefore the capacity is very low.

2.6.3 Spread Spectrum techniques

The embedding of a watermark using *Spread Spectrum* (SS) techniques, consist of the adding of a pseudo-random noise pattern p to a host signal:

$$y = x + p. \tag{2.10}$$

The detection is based on correlation. The correlation of the watermarked signal y with the known pattern p is determined. If the correlation is above some threshold T, it is assumed that the watermarked signal was indeed watermarked with pattern p:

$$\langle y, p \rangle = \langle x + \tilde{p}, p \rangle = \langle x, p \rangle + \langle \tilde{p}, p \rangle \approx \langle \tilde{p}, p \rangle \tag{2.11}$$

$$\approx \begin{cases} 1 & \text{if } \tilde{p} = p \\ \ll 1 & \text{otherwise} \end{cases}, \tag{2.12}$$

where the first approximation comes from the fact that the pseudo-random noise pattern p and the host signal x are highly uncorrelated, and the second approximation follows from the definition of the correlation coefficient.

This Spread Spectrum technique is highly robust against simple attacks, but it is only possible to embed one bit: watermark or no watermark. It is possible to embed more than one bit into a host signal by dividing the host signal into smaller subsamples and embed one bit in each part. This will come at the cost of some robustness. If an image is cropped, bits will get lost.

Another way to embed more than one bit into a host signal is by using a technique called Direct Sequence Code Division Multiple Access (DS-CDMA) Spread Spectrum communication. Where as normal Spread Spectrum adds only one pattern, DS-CDMA Spread Spectrum uses more than one (N) patterns. The embedding is then:

$$y = x + \sum_{i=1}^{N} \beta_i p_i, \tag{2.13}$$

where

$$\beta_i = \begin{cases} +1 & \text{if message bit} = 0 \\ -1 & \text{if message bit} = 1 \end{cases}. \tag{2.14}$$

Detection is again done by correlation. The decision rule for pattern i is:

$$\mathrm{E}\left[(p_i - \mathrm{E}p_i)(y - \mathrm{E}y)\right] = \begin{cases} > 0 & \text{then a 0 is detected} \\ < 0 & \text{then a 1 is detected} \end{cases}. \tag{2.15}$$

This DS-CDMA Spread Spectrum technique is (to some amount) robust against cropping, but there is some interference between the watermark patterns.

2.6.4 Patchwork Algorithm

A very simple example of a Spread Spectrum watermarking method is the *Patchwork algorithm*. The pseudo-random pattern used here consists of ± 1's.

The Patchwork algorithm divides an image into two sets A and B. To embed 0, the values of the pixels in set A are raised by 1 and the values of the pixels in set B are lowered by 1. In order to embed 1, the values of the pixels in set A are lowered instead of raised and that of B raised instead of lowered. Let's say that the sets A and B both have the size N. Detection is done by comparing the difference D of the averages $\overline{y_A}$ and $\overline{y_B}$ of set A and B. If a watermark is embedded (in this case a 0), this detection looks like:

$$\begin{aligned} D &= \overline{y_A} - \overline{y_B} = \frac{1}{N}\sum_{y\in A} y_i - \frac{1}{N}\sum_{y\in B} y_i = \frac{1}{N}\sum_{x\in A}(x_i + 1) - \frac{1}{N}\sum_{x\in B}(x_i - 1) \\ &= \overline{x_A} - \overline{x_B} + 2 \approx 2, \end{aligned} \tag{2.16}$$

where it is assumed that the average of pixel values in set A equals that of set B. If a 1 is embedded $\overline{y_A} - \overline{y_B} \approx -2$ and if nothing is embedded $\overline{y_A} - \overline{y_B} \approx 0$.

A threshold $T \in]0, 2[$ can be set, and the decision rule for the detected message bit $\hat{m}$ is:

$$\hat{m} = \begin{cases} 1 & \text{if } D < -T \\ \text{nothing} & \text{if } -T \leq D \leq T \\ 0 & \text{if } D > T \end{cases} . \tag{2.17}$$

The precise form of the sets A and B is the secret of the watermark embedders.

Chapter 3

Quantization Watermarking

As seen in Section 2.6, watermarking can be done using several techniques. The focus of this report is on quantization watermarking. In this chapter we will explain the basics of quantization based watermarking methods. As seen before, a watermarking process consists of the embedding of a watermark in a host signal, the transmission of the watermarked signal over a channel modelling the attacks, and the subsequent detection. Because in practice the host signal is not available for comparison at the detector, only blind detection is considered (see Subsection 2.3.7). The detection of the watermark at the detector is influenced by the host signal, this is called *host signal interference*. At the embedder the host signal is known. It is possible to use this information in order to reduce the host signal interference on detection. The principle of using the host signal at the embedder is known as using *side information at the encoder*, a concept of Claude Shannon [34].

Costa [11] considered this principle for an Additive White Gaussian Noise (AWGN) channel and an i.i.d. Gaussian host signal. See Figure 2.1, where the noise n is i.i.d. Gaussian noise. Costa proposed a blind scheme that performs as well as a non-blind scheme, i.e., the detection performance cannot be improved by giving the detector access to the original data. Host interference at the decoder is completely absent. Chen and Wornell rediscovered the paper of Costa [11] and introduced the principle of using side information at the encoder into the world of watermarking. The so-called Ideal Costa Scheme (ICS) is discussed in Section 3.1.

It was shown by Chen and Wornell that their previously proposed watermarking scheme [3, 4, 5, 6] based on the so-called Quantization Index Modulation (QIM) scheme, can be explained in terms of Costa's scheme. An extended version of QIM, distortion compensated QIM (DC-QIM), performs as well as the Costa scheme. In Section 3.2 QIM and DC-QIM are discussed.

The Costa scheme is not practical, as will be seen in Section 3.1, and therefore attention is paid to some practical implementation of the ICS, the so-called Scalar Costa Scheme (SCS), developed by Eggers et al. [15, 17, 18]. This scheme is considered in Section 3.3.

This report is focussed on binary dithered quantization, which is a simple case of the SCS or DC-QIM. The resulting scheme is explained in simple terms in Section 3.4.

In Section 3.5 and 3.6 the principles of embedding and detecting a message are treated.

The performance judgement of the different schemes is based on information theoretic notions of capacity, rate and distortion. Good books on this subject are [12, 25]. Basic knowledge on information theory is assumed.

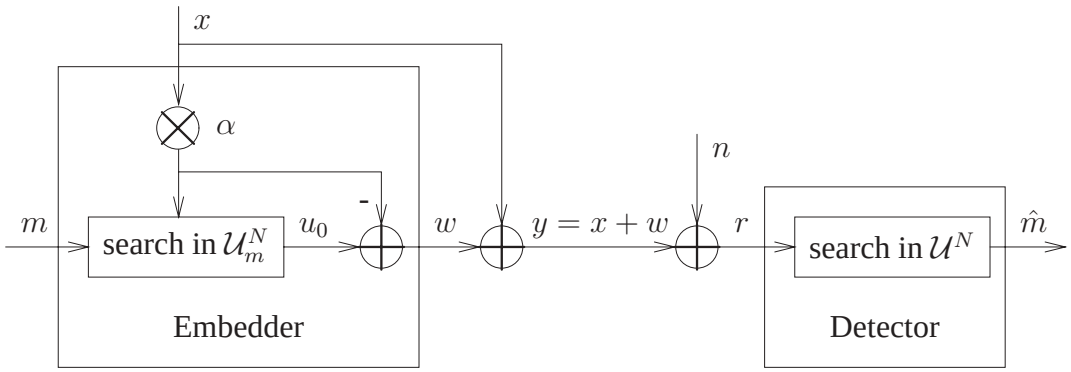


Figure 3.1: Costa's scheme. Partially reproduced from [18]. Compare with Figure 2.1.

3.1 The Ideal Costa Scheme

The communications problem, as depicted in Figure 2.1, is to retrieve the message m embedded in a host signal x at the detector. Costa assumed that the host signal x and the noise signal n are of length N and i.i.d. Gaussian distributed, i.e., $x \sim N(0, \sigma_x^2 I_N)$ and $n \sim N(0, \sigma_n^2 I_N)$. The interfering Gaussian noises x and n are not known at the decoder. The encoder, however, knows x.

At the encoder, the watermarked signal is chosen depending on the message m, the side information x and realizations u of an auxiliary random variable U. Appropriate realizations u of U for all possible messages m and all possible side information x are listed in a codebook $\mathcal{U}$. The watermark w has a length N equal to the length of the host signal.

In order to solve this communications problem, Costa introduced an N-dimensional random codebook $\mathcal{U}^N$ with codewords u

$$\mathcal{U}^N \triangleq \left\{ u^{(p)} = \omega^{(p)} + \alpha\chi^{(p)} \mid p \in \{1, 2, \cdots, P\}, \right. \\ \left. W \sim N\left(0, \sigma_W^2 I_N\right), X \sim N\left(0, \sigma_X^2 I_N\right) \right\}, \tag{3.1}$$

where $\omega^{(p)}$ and $\chi^{(p)}$ are realizations of two N-dimensional independent random processes W and X with Gaussian pdf, and α a real codebook parameter, with $0 \leq \alpha \leq 1$. The size of the codebook, i.e., the number of codewords, is given by P, typically a large number. So the codebook $\mathcal{U}^N$ is really a random codebook, containing P codewords of length N.

In the limit as $N \to \infty$, Costa's codebook $\mathcal{U}^N$ achieves the capacity of communication with i.i.d. Gaussian side information x both at the encoder and decoder and an AWGN channel. See [11, 18] for more details on the Ideal Costa Scheme.

The codebook is partitioned into M disjoint sub-codebooks, with M the number of possible watermark messages. Partitioning is done such that each sub-codebook $\mathcal{U}_m^N$ contains about the same number of codeword sequences. So $\mathcal{U}^N = \mathcal{U}_1^N \cup \mathcal{U}_2^N \cup \cdots \cup \mathcal{U}_M^N$. This codebook is available both at the encoder and decoder. See also Figure 3.1.

According to [18] the embedding of a message m into the host signal x is equivalent to finding a sequence u_0 in the set $\mathcal{U}_m^N$ such that $w = u_0 - \alpha x$ is nearly orthogonal in the Euclidean sense to

x. If the length of the codewords N goes to infinity the probability that no such sequence u_0 exist, goes to zero exponentially.

Subsequently, the watermarked signal $y = x + w = (1-\alpha)x + u_0$ is transmitted over the AWGN channel and the received signal is $r = y + n$.

Decoding is done by finding a sequence u in the entire codebook $\mathcal{U}^N$ such that $u - \alpha r$ is nearly orthogonal to r. The probability that there is only one such sequence u_0 is high if N goes to infinity. The index $\hat{m}$ of the sub-codebook $\mathcal{U}_{\hat{m}}^N$ containing u is the decoded message.

Costa showed that for the codebook (3.1), for $N \to \infty$, and with optimal parameter α^*, with

$$\alpha^* = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_n^2} = \frac{1}{1 + 10^{-\mathrm{WNR}/10}}, \tag{3.2}$$

the capacity is

$$C_{\mathrm{ICS}} = \frac{1}{2} \log \left(1 + \frac{\sigma_w^2}{\sigma_n^2} \right), \tag{3.3}$$

which is equal to the capacity of the transmission scenario where the host signal x is known to the decoder [18]. So the Costa scheme is optimal in the sense that it is capacity achieving. Here the WNR is the *Watermark-to-Noise Ratio* which is defined as

$$\mathrm{WNR} \triangleq 10 \log_{10} \left(\frac{\sigma_w^2}{\sigma_n^2} \right). \tag{3.4}$$

So, not knowing the host signal at the decoder does not decrease capacity. Note that the capacity depends only on the WNR and is independent from x. Also note that this capacity is only achieved for a huge random codebook size, therefore this is not a practical way of achieving full capacity.

3.2 Quantization Index Modulation

Chen and Wornell proposed a watermarking scheme which they called Quantization Index Modulation. They showed that this scheme is a special case of the Ideal Costa Scheme [6, 7]; QIM is ICS with $\alpha = 1$. They also showed that QIM is not capacity achieving compared to the ICS, but in fact close. Therefore they proposed an improvement of QIM, which they called Distortion Compensated QIM (DC-QIM). This DC-QIM scheme is in fact equal to the ICS and therefore optimal. The principle of QIM is explained in Subsection 3.2.1 and the extension of this scheme (DC-QIM) is treated in Subsection 3.2.2.

3.2.1 Quantization Index Modulation

Chen and Wornell view the embedding function $y = s(x; m)$ as an ensemble of functions of the host signal x, indexed by the message m. For the distortion to be small it is required that y and x are close in distance, so $s(x; m) \approx x$, $\forall m$. For this system to be robust it is required that the embedding functions are far away in some sense for different messages, so $\mathrm{d}(s(x; m_i), s(x, m_j)) \gg 0, \forall i, j, i \neq j$, with $\mathrm{d}(u, v)$ some distance measure. At least the ranges should be non-intersecting, because else there will be some values of s from which it is not possible to uniquely determine m.

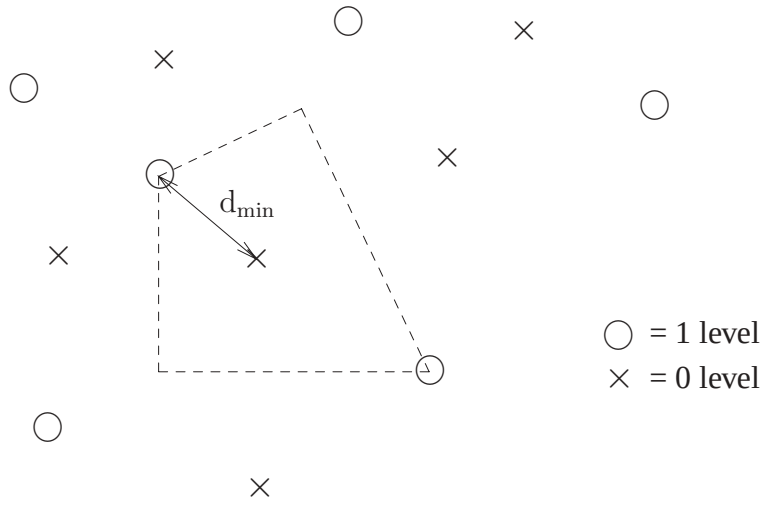


Figure 3.2: Reproduced from [9]. QIM for information embedding. The points marked with $\times$'s and $\bigcirc$'s belong to two different quantizers, each with its associated index. The minimum distance $\mathrm{d}_{\min}$ measures the robustness to perturbations, and the sizes of the quantization cells, one of which is shown in the figure, determine the distortion. If $m = 0$, the host signal is quantized to the nearest $\times$. If $m = 1$, the host signal is quantized to the nearest $\bigcirc$.

According to [9] these two requirements suggest that s should be discontinuous and quantizers are a class of discontinuous, approximate-identity functions. Chen and Wornell use quantizers for the embedding.

For a message m of length l and alphabet-size D, i.e., $m \in \{1, 2, \cdots, D\}^l$, there are D^l quantizers. In order to embed a message $m^{(j)}$, the host signal is quantized with the j^{th} quantizer, $j \in \{1, 2, \cdots, D^l\}$, i.e., $y = s(x; m^{(j)})$ is the point in the j^{th} quantizer-set closest to x.

For example let the message length $l = 1$ and $D = 2$, so $m \in \{0, 1\}$, see Figure 3.2 for illustration. Now two quantizers are needed, and their corresponding set of reconstruction points are shown in Figure 3.2 with $\times$'s and $\bigcirc$'s.

Detection is done using a minimum distance decoder. The received signal r is quantized with all quantizers, i.e., $s(r; m)$ is calculated $\forall m$. The quantizer for which the distance with the received signal $\mathrm{d}(r, s(r; m))$ is minimal, is assumed to be the original embedded message. So the decoding problem is [9]:

$$\hat{m} = \arg\min_{m} \ \mathrm{d}(r, s(r; m)) \tag{3.5}$$

Embedding and detection procedures are illustrated in Figure 3.3 for $l = 1$ and $D = 2$.

3.2.2 Distortion Compensated Quantization Index Modulation

Watermarking is always a matter of trading off the rate and the induced distortion of the embedding against the robustness. In order to improve this tradeoff Chen and Wornell introduced distortion compensation. They scale all the quantizers by a factor $1/\alpha$, where $\alpha \leq 1$, which increases the squared minimum distance $\mathrm{d}_{\min}$ by a factor $1/\alpha^2$, thereby increasing robustness, but also the distortion is increased by this factor. Adding back a fraction $1 - \alpha$ of the quantization error to the quantized value, compensates for this additional distortion. This last step is called

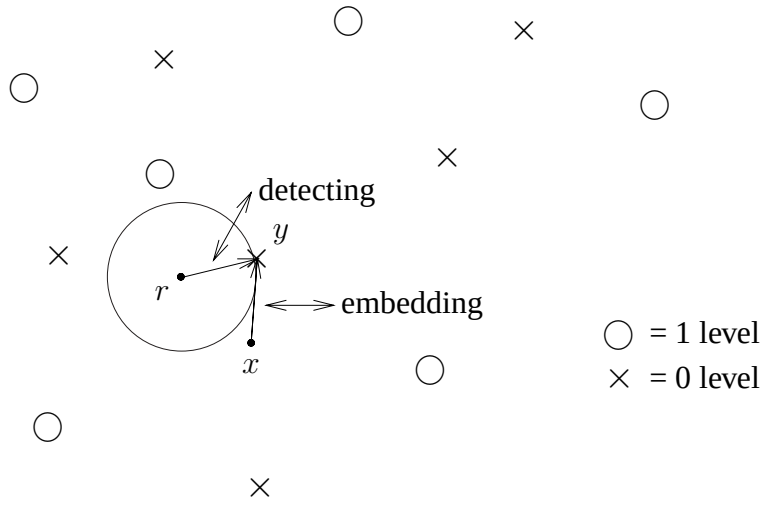


Figure 3.3: Embedding is shown for $m = 0$. The host signal x is quantized to the nearest $\times$ in order to get a watermarked signal y. At the detector a signal r is received, which is not equal to y due to a noise attack. The nearest reconstruction point is again an $\times$ and therefore a message $m = 0$ is detected.

distortion compensation. The minimum distance using distortion compensation is at least equal to the original minimum distance, but possibly larger. In that case the robustness is improved, whereas the embedding induced distortion is equal for both cases. The embedding formula for DC-QIM is now

$$s(x;m) = \mathcal{Q}_{\Delta/\alpha}(x;m) + (1-\alpha)\left(x - \mathcal{Q}_{\Delta/\alpha}(x;m)\right), \tag{3.6}$$

where $\mathcal{Q}_{\Delta/\alpha}(x;m)$ is the m^{th} quantizer of an ensemble whose reconstruction points are separated by a distance Δ before scaling are separated by a distance Δ/α after scaling. The first term of Equation (3.6) is normal QIM embedding and the second term is the distortion compensation term. The quantizer is possibly high dimensional, which relates to the large codebook from the Ideal Costa Scheme.

Chen and Wornell found that the optimal scaling parameter α^* which "optimizes the rate distortion - robustness tradeoffs" is equal to α^* of the ICS (3.2). This is not surprising, because, as stated before, DC-QIM is an equivalent but different formulation of the ICS [18].

3.3 Practical watermarking schemes

As explained below, both ICS and DC-QIM are not practical. Therefore practical schemes were developed by several people. Chen and Wornell came with a practical version of QIM, which they called Dither Modulation (DM). This scheme can easily be extended to the distortion compensated case. Eggers proposed a practical version of the Costa scheme [15, 17, 18], which he called the Scalar Costa Scheme (SCS). SCS is discussed in Subsection 3.3.1 and DM in Subsection 3.3.2. In Section 3.4 attention is focussed on the simple case where there are only two quantizers: in fact a binary SCS scheme, or also a binary DC-QIM scheme.

3.3.1 The Scalar Costa Scheme

The reason why the Costa scheme is not practical, is the used huge random codebook. Searching this codebook is comprehensive because of the size and because there is no structure on this codebook. Therefore Eggers proposed to use a suboptimal, structured codebook, without changing the main concept of Costa's idea. Eggers called this scheme the Scalar Costa Scheme (SCS). Like Costa's scheme, this scheme is also independent from the host signal, because a random key sequence d is used.

The codebook $\mathcal{U}^N$ of Costa, see Equation (3.1), is structured by Eggers as a product codebook $\mathcal{U}^N = \mathcal{U}^1 \otimes \mathcal{U}^1 \otimes \cdots \otimes \mathcal{U}^1$ of N identical one-dimensional component codebooks. Because of the use of the one-dimensional (scalar) component codebooks, Eggers refers to this watermarking scheme as the Scalar Costa Scheme. The one-dimensional component codebooks $\mathcal{U}^1$ are separated into D disjoint parts, where D is the size of the alphabet $\mathcal{D} = \{0, 1, \cdots, D-1\}$. So $\mathcal{U}^1 = \mathcal{U}_0^1 \cup \mathcal{U}_1^1 \cup \cdots \cup \mathcal{U}_{D-1}^1$. Taking a product codebook is equivalent with sample wise quantization. Separating the one-dimensional component codebook $\mathcal{U}^1$ is equivalent with using quantizers that are shifted versions of each other. The one-dimensional codebook $\mathcal{U}^1$ with the use of a secret key d is chosen to be

$$\mathcal{U}^1(\alpha, \Delta, D, d) = \left\{ u = (k+d)\alpha\Delta + m\frac{\alpha\Delta}{D} \mid m \in \mathcal{D}, k \in \mathbb{Z} \right\} \tag{3.7}$$

and the m^{th} sub-codebook is given by

$$\mathcal{U}_m^1(\alpha, \Delta, D, d) = \left\{ u = (k+d)\alpha\Delta + m\frac{\alpha\Delta}{D} \mid k \in \mathbb{Z} \right\}, \tag{3.8}$$

where Δ is the step size used for quantization. Without knowing d it is impossible to reconstruct the codebook $\mathcal{U}^N$ used for the watermark embedding.

Using the product codebook is equivalent to calculating the quantization sample-wise, i.e., for each $n \in \{1, 2, \cdots, N\}$ calculate

$$q_n = \mathcal{Q}_\Delta\left\{x_n - \Delta\left(\frac{m_n}{D} + d_n\right)\right\} - \left(x_n - \Delta\left(\frac{m_n}{D} + d_n\right)\right), \tag{3.9}$$

where $\mathcal{Q}_\Delta\{\cdot\}$ denotes scalar uniform quantization with step size Δ, so $\mathcal{Q}_\Delta\{z\} = \left\lfloor \frac{z}{\Delta} \right\rceil \Delta$, with $\lfloor \cdot \rceil$ rounding to the nearest integer. The watermark is then given by $w = \alpha q$ and the watermarked signal is $y = x + w = x + \alpha q$.

It is shown in [20, 24, 31] that for a uniformly distributed key sequence d that q and w are statistically independent from x.

Eggers gives the following relation between α, the quantization step size Δ and the watermark power σ_w^2:

$$\alpha = \frac{\sigma_w\sqrt{12}}{\Delta}. \tag{3.10}$$

Because this scheme is suboptimal, the capacity will be lower than the Costa capacity, see Equation (3.3). Also the value of the optimal α^* will be different. Eggers found a numerical solution for this optimal α^*. However, no analytical solution has been found yet [8]. In Chapter 8 we will give an implicit analytical expression for the optimal α^*.

SCS decoding is similar to ICS decoding as described in Section 3.1, except that a different codebook is used. If the product codebook $\mathcal{U}^N$ is treated as a quantizer, than decoding is equivalent to quantizing the received value r to the closest codeword.

3.3.2 Dither Modulation

Embedding using quantization and a dither signal d gives an embedding function $s(x;m) = q(x+d)-d$. In [9] Chen and Wornell give some practical examples of Dither Modulation: Coded Binary Dither Modulation with Uniform Scalar Quantization and Spread-Transform Dither Modulation. See this article for more details.

3.4 Embedding by binary dithered quantization with uniform quantization step size

This report focuses on the case of binary dithered quantization with uniform quantization step size. Binary means that the alphabet $\mathcal{D}$ has size 2, so $\mathcal{D} = \{0, 1\}$. This means that the message m consists of 0's and 1's only; The message is considered to be a binary message. Dithered quantization means that the quantization cells and reconstruction points are shifted versions of each other, it is the same as saying that Dither Modulation is used. A uniform quantization step size refers to the fact that the reconstruction points are uniformly spread over the quantization space.

This section gives a simple explanation of the workings of a watermarking embedding algorithm that uses the principles of the SCS, or a practical implementation of DC-QIM. In Subsection 3.4.1 the basic watermarking scheme, without dither and distortion compensation, is given. The principle of dithering is further explained in Subsection 3.4.2 and distortion compensation is treated in Subsection 3.4.3.

In this report the watermarking of images is considered, so the host signal x is in fact an image. It is convenient to view an image as a matrix of pixel values, where the pixel values represent the intensity or the color of a pixel.

For the case of an 8 bit grey-scale image, every pixel consists of one value in the set $\mathbb{Z}_{255} \triangleq \{z \in \mathbb{Z} \mid 0 \leq z \leq 255\}$, representing the luminance. For an 8 bit color image, every pixel consists of three values, representing either red, green and blue color intensities (for an RGB image), or one luminance and two chrominance values (for a YUV image). Each of these values are again in the set $\mathbb{Z}_{255}$. From now on, when we talk about changing a pixel value, we mean changing the luminance value. An RGB image and a YUV image are linearly related and can be simply converted into each other.

3.4.1 The basic scheme

A message m is a sequence of bits of a certain length, say l (So, $m \in \{0,1\}^l$). It is possible to embed a message bit m_i, $i \in \{1, 2, \cdots, l\}$, in a pixel of choice with value x_j, $j \in \{1, 2, \cdots, N\}$. Using QIM, pixels are quantized to values in the set $\{z \in \mathbb{Z}_{255} \mid z = \frac{k}{2}\Delta, k \in \mathbb{Z}\}$. So the value of a pixel is no longer represented by the values in the set $\mathbb{Z}_{255}$, but by fewer values. The values in the new set are called the quantization levels. A level with value $k\Delta$, $k \in \mathbb{Z}$ is associated with message bit 0 and is defined to be an even level. The levels associated with 1 are shifted versions of the even levels, they are shifted by $\Delta/2$. These levels have a value $(k + \frac{1}{2})\Delta$, $k \in \mathbb{Z}$ and are defined to be odd levels. Here, Δ represents the minimal distance between two equal levels, so the minimal distance between two odd or two even levels, and is called the *quantization step size*. See Figure 3.4.

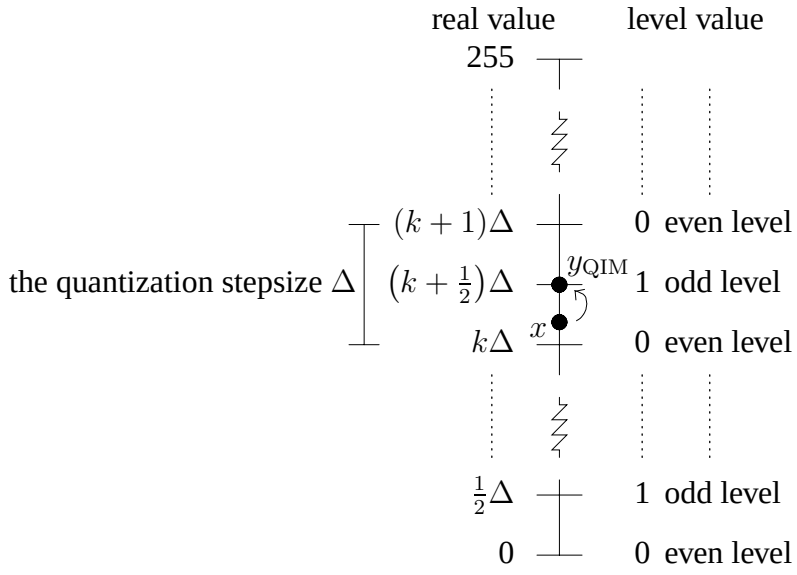


Figure 3.4: The basic idea behind QIM: the host signal value x is rounded to the nearest odd or even level, depending on the message bit one wants to embed. Here we show the embedding of a 1, so the watermarked signal y_{QIM} is the odd level above x. The quantization step size Δ is the minimal distance between two equal levels.

Watermarking is done by rounding a pixel value to a level, this is called *quantization*. Embedding 1 in a pixel is done by quantizing the value x to the value of the nearest odd level. Embedding 0 is done by quantizing the value of x to the value of the nearest even level. See Figure 3.4 for an illustration of this principle.

The process of embedding a message bit in a pixel is also illustrated with the following example.

Example 3.1. *Suppose a pixel has a value $x = 123$ and suppose a quantization step size is used of $\Delta = 20$. So, even levels are given by values in the set $\{0, 20, 40, \cdots, 120, 140, \cdots\}$ and odd levels are given by values in the set $\{10, 30, 50, \cdots, 110, 130, \cdots\}$. If the message bit corresponding to the pixel is 0, this message bit is embedded by altering the pixel value to $y_{\mathrm{QIM}} = 120$ (120 is the closest even level). 1 is embedded by altering the pixel value to $y_{\mathrm{QIM}} = 130$ (130 is the closest odd level).*

The quantization of a pixel value x to the nearest odd or even level is mathematically done in the following way:

The watermarked value using QIM y_{QIM} is chosen as close as possible to x, where y_{QIM} is an integer times the quantization step size Δ for embedding 0, i.e., $y_{\mathrm{QIM}} = k_0\Delta$, $k_0 \in \mathbb{Z}$, and an integer times the quantization step size plus an offset $\frac{1}{2}\Delta$ for embedding 1, i.e., $y_{\mathrm{QIM}} = (k_0 + \frac{1}{2})\Delta$. That is, choose $k_0 \in \mathbb{Z}$ such that $y_{\mathrm{QIM}} - x$ is minimal, or

$$k_0 = \arg\min_{k\in\mathbb{Z}} \left| x - \left(k + \frac{m}{2}\right)\Delta \right|, \tag{3.11}$$

where $m \in \{0, 1\}$ represents the message bit. The solution of (3.11) is

$$k_0 = \left\lfloor \frac{x - \frac{1}{2}m\Delta}{\Delta} \right\rceil. \tag{3.12}$$

So for each pixel we have the codebook

$$\mathcal{U}^1(\Delta) = \left\{ u = \left(k + \frac{m}{2}\right)\Delta \mid m \in \{0,1\}, k \in \mathbb{Z} \right\}. \tag{3.13}$$

Compare this with the component codebook of Eggers, with $\alpha = 1$, $\mathcal{D} = \{0,1\}$, so $D = 2$, and $d = 0$, see Equation (3.7).

Consequently, the embedding formula of the basic QIM scheme becomes

$$y_{\mathrm{QIM}} = \left(k_0 + \frac{m}{2}\right)\Delta = \left\lfloor \frac{x - \frac{1}{2}m\Delta}{\Delta} \right\rceil \Delta + \frac{m\Delta}{2}, \tag{3.14}$$

which can be compared with $y = x + \alpha q$, with q as in (3.9), for $\alpha = 1$, $D = 2$ and $d = 0$.

Example 3.2. *(Continued) The watermarked pixel value y_{QIM} using QIM can also be calculated using Equation (3.14). Then we have $y_{\mathrm{QIM}} = \left\lfloor \frac{123 - \frac{1}{2}m \cdot 20}{20} \right\rceil \cdot 20 + \frac{m \cdot 20}{2}$, which gives $y_{\mathrm{QIM}} = 120$ for $m = 0$ and $y_{\mathrm{QIM}} = 130$ for $m = 1$.*

3.4.2 Dithering

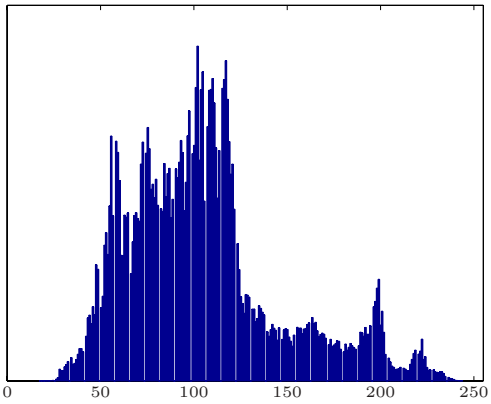
As said before, embedding information in every pixel of an image using QIM, causes every pixel to have a value a multiple of half the quantization step size. This way the statistics of an image are changed. This is undesirable, because a watermarked image should be difficult to distinguish from the original one. It is very easy to find out what the quantization levels are and what the quantization step size Δ is, just by drawing a histogram of the pixel values of the watermarked image; The values will be centered around the quantization levels. This is illustrated by Figure 3.5. Information about the quantization levels can be used to remove the watermark. For security reasons it is undesirable to reveal this information and for perceptibility reasons it is undesirable to change the image statistics.

Another disadvantage of the above described basic scheme is the appearance of contour lines in the watermarked image y_{QIM}. While in the original image a smooth change to another color can be seen, in the watermarked image these changes will go with jumps; The bigger the quantization step size Δ, the bigger the jumps. This is illustrated with Figure 3.6.

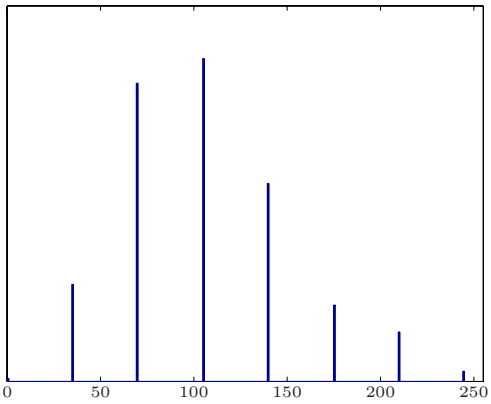
These disadvantages can be overcome by introducing *dither*. Dither shifts the levels by a value d, so now $y = k\Delta + d$, where the dither d is chosen differently for different pixels. The quantization procedure is now to first add a value d to the original pixel value x, then quantize this new value $x + d$ to the nearest odd or even level and finally subtract the dither d again. This kind of dither is called *subtractive dither*.

As said before, for every pixel a different dither value is chosen. This way, the contour lines disappear, giving the watermarked image better perceptual quality. This effect can be seen in Figure 3.6. We have watermarked the monarch image with parameter $\Delta = 35$. Because dither adds different values to different pixels, the values of the watermarked image will be no longer centered around the levels. It is not possible anymore to know what the quantization levels are by analyzing the watermarked image (See Figure 3.5).

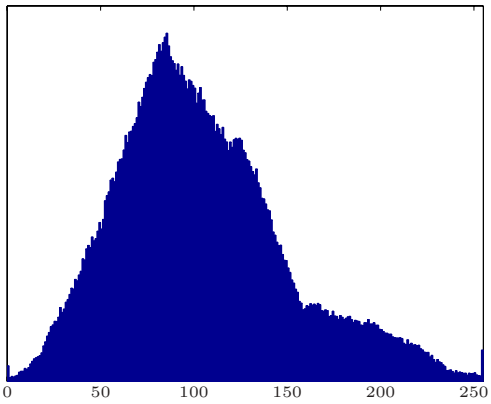
The new quantization formula using QIM with dithering is now (for comparison see Equation (3.14)):



(a) Histogram of original image `monarch.bmp`



(b) Histogram of the watermarked image `monarch.bmp` using the basic QIM scheme without dither modulation.



(c) Histogram of the watermarked image `monarch.bmp` using the basic QIM scheme with dither modulation.

Figure 3.5: The effect of dither: Using the basic QIM scheme quantization levels can be read off easily (Subfigure 3.5(b)); This is not possible anymore using dither modulation (Subfigure 3.5(c)). Watermarking is done here using QIM (with or without dither) with a quantization step size $\Delta = 35$. This is taken quite large, for illustrative reasons. Histogram 3.5(c) does not resemble histogram 3.5(a) in this case, because the large Δ causes a large distortion.

$$y_{\mathrm{QIM}+d} = \left\lfloor \frac{x + d - \frac{1}{2}m\Delta}{\Delta} \right\rceil \Delta - d + \frac{m\Delta}{2}. \tag{3.15}$$

Compare Equation (3.15) with the watermarking formula given in Equation (3.9), with $\alpha = 1$. For a uniformly distributed dither sequence the watermark w is statistically independent from the host signal x. The dither sequence must be known at the embedder and the detector. Therefore the dither sequence is determined in advance. Following Chen [8] the dither is chosen uniformly distributed over the range $[-\frac{1}{4}\Delta, \frac{1}{4}\Delta]$.

At the detector the dither sequence is first added to the received image before proceeding with the detection process. The detection procedure is explained in Section 3.6.

(a) Original image `monarch.bmp`

(b) Watermarked image using the basic QIM scheme without dither modulation.

(c) Watermarked image using the basic QIM scheme with dither modulation.

Figure 3.6: The effect of dither: Contour lines, introduced by the basic QIM scheme (Subfigure 3.6(b)), disappear with dither modulation (Subfigure 3.6(c)). Watermarking is done using a large quantization step size $\Delta = 35$.

3.4.3 Distortion compensation

In Subsection 3.2.2 distortion compensation is treated. For the case of the practical scheme this means that part of the watermarking error $\epsilon = y - x$ is added back:

$$y_{\mathrm{SCS}} = x + \alpha\epsilon = x + \alpha\left(y_{\mathrm{QIM}+d} - x\right), \tag{3.16}$$

where $0 \leq \alpha \leq 1$. The watermarking formula for QIM with dithering and distortion compensation is now:

$$y_{\mathrm{SCS}} = x + \alpha\left(\left\lfloor \frac{x + d - \frac{1}{2}m\Delta}{\Delta} \right\rceil \Delta - d + \frac{m\Delta}{2} - x\right). \tag{3.17}$$

This embedding formula can be compared to the codebook of the SCS, see Equation (3.7). It is clear that (3.17) equals the SCS of Eggers, where $y = x + \alpha q$, with q given in Equation (3.9).

This process of bringing down the quantization error is illustrated in Figure 3.7. Using it will reduce the watermark distortion, but decrease robustness, as smaller changes are more difficult to notice. Chen and Wornell [9] propose to use distortion compensation, as it can improve the achievable rate distortion-robustness tradeoffs of QIM methods.

Now there are two parameters to control the robustness and the perceptibility of the watermark, namely the quantization step size Δ and the distortion compensation parameter α. That is, the

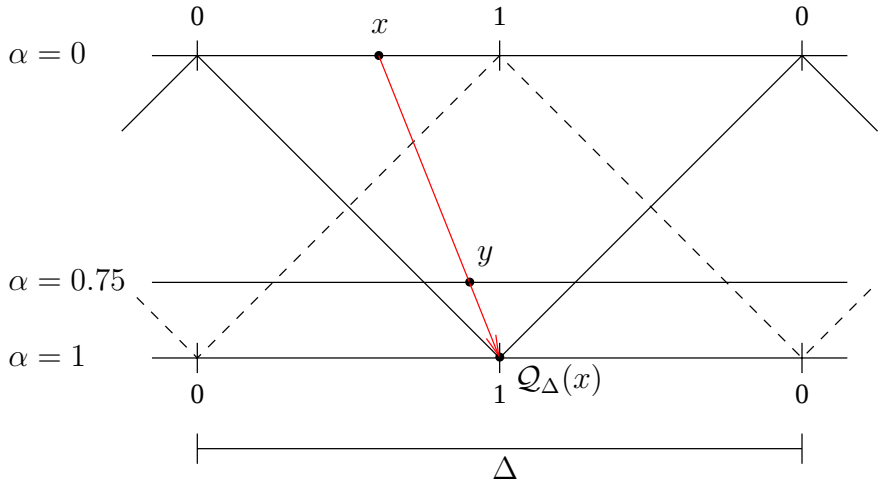


Figure 3.7: The process of distortion compensation. The upper line represents the value of the host signal x. Without distortion compensation ($\alpha = 1$) and dithering, all values are quantized to an odd or even level with a function $\mathcal{Q}_\Delta(x; m)$ and the watermarked signal equals $\mathcal{Q}_\Delta(x; m)$. With distortion compensation ($\alpha \in]0, 1[$) a part of the quantization error is returned and the watermarked signal equals y.

distortion $D(\alpha, \Delta)$ and the robustness are determined by α and Δ.

In order to begin embedding, one has to know the embedding parameter pair (α, Δ). The reader may wonder whether it is possible to simply choose a pair $(1, \Delta)$ and do just as well as the case $(\alpha, \Delta/\alpha)$, i.e., embedding using QIM versus embedding using DC-QIM.

In both cases the distortion introduced by the watermark will be $\frac{\alpha^2\Delta^2}{12}$. Is the robustness better for QIM? It is made clear intuitively by Figure 3.8 that for equal distortion, DC-QIM has a larger minimal distance, and thus a larger robustness. This can also be seen as follows, see Figure 3.7 for reference. Suppose x is uniformly distributed over the range of a quantization bin. For the case of QIM with parameters $(1, \Delta)$ x is quantized to a quantization level. The distance from this level to the nearest wrong level is given by $d_{\mathrm{QIM}} = \frac{1}{2}\Delta$. For the case of DC-QIM with parameters $(\alpha, \Delta/\alpha)$, the watermarked value y is within $(1-\alpha)\frac{1}{2}\frac{\Delta}{\alpha}$ of a quantization level. On average y is $\frac{1}{2}(1-\alpha)\frac{1}{2}\frac{\Delta}{\alpha}$ from this level, because y is also uniformly distributed. The distance to the nearest wrong level is given by $d_{\mathrm{DC-QIM}} = \frac{1}{2}\frac{\Delta}{\alpha} - \frac{1}{2}(1-\alpha)\frac{\Delta}{2\alpha} = \frac{1}{4}(\frac{\Delta}{\alpha} + \Delta)$. We have $d_{\mathrm{DC-QIM}} > d_{\mathrm{QIM}}$ for $\alpha < 1$, so DC-QIM performs better on average.

In Chapter 8 it is shown that DC-QIM with parameter $\alpha \in]0, 1]$ is usually better than QIM with $\alpha = 1$, by minimizing the bit error probability.

3.5 Spreading a message and repetition coding

Equation (3.17) defines a watermark embedding algorithm. In this section this algorithm is applied on digital images. It is assumed in this section that an image is a matrix of luminance-values of size $N \times M$, so the host x is $N \times M$. A message m is a series of bits of length l and is given beforehand.

If the size of a message l equals the number of image pixels $N \times M$, then each message bit m_i, $i \in \{0, 1, \cdots, l\}$ is associated with one pixel x_{ij}, $i \in \{0, 1, \cdots, N\}$, $j \in \{0, 1, \cdots, M\}$ in a

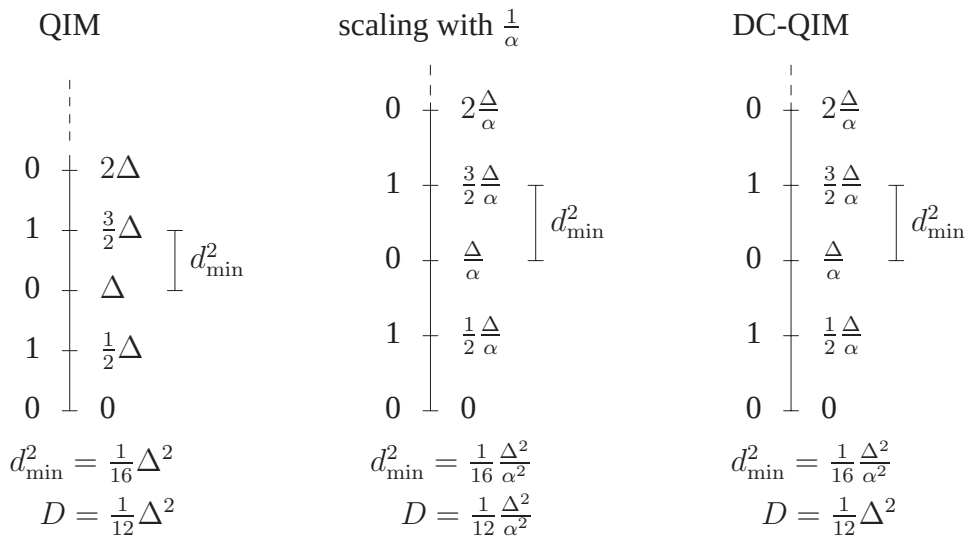
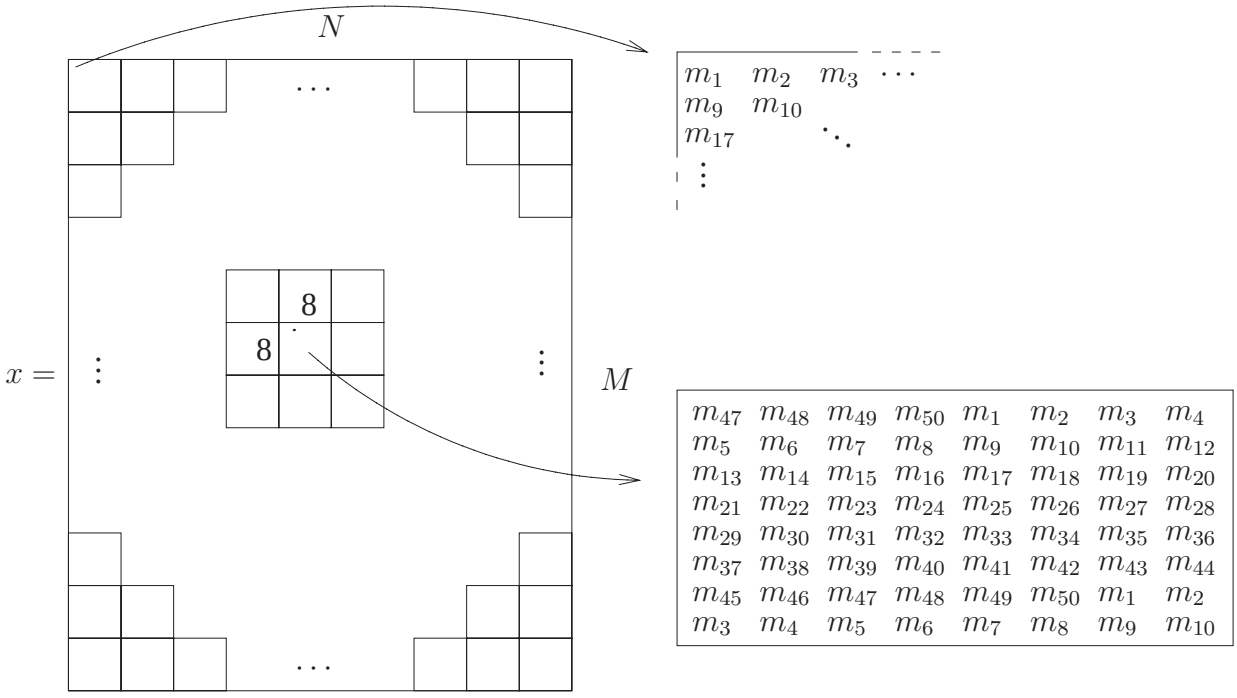


Figure 3.8: The left axis represents embedding with QIM with quantization step size Δ. If the quantizers are scaled with a factor $\frac{1}{\alpha}$, the squared minimum distance $d^2_{\min}$ and the distortion D are increased with a factor $\frac{1}{\alpha^2}$. This is depicted with the middle axis. Embedding using DC-QIM with parameters α and $\frac{\Delta}{\alpha}$ is depicted by the third axis. A part of the quantization error is added back, which causes the distortion to be equal to the case of QIM. The minimum distance however is still better than QIM, and therefore the robustness.

unique way. If $l < N \times M$, then after each message bit is associated with one pixel, there are some unwatermarked pixels left. Those pixels can be used to gain robustness. A very simple way of doing this is to embed the message m another time in the image. If for example $l = \frac{1}{3}(N \times M)$, then each message bit is associated with 3 pixels, i.e., each message bit is embedded three times in three different pixels. Using this repetition coding, robustness can be gained, because at the detector three estimates for the i^{th} message bit $\hat{m}_i$ are available instead of only one. For example, if 1 is embedded one time, and one error occurs, then 0 is detected. But if 1 is embedded three times, and one error occurs, then for instance the series $0, 1, 1$ is detected. Using a majority vote the detected message bit will be $\hat{m}_i = 1$, which equals the embedded message bit, so no error is made.

In order to associate message bits with image pixels, some rule must be chosen in order to do that. It is possible to pseudo randomly connect message bits with image pixels, but we have chosen another association rule. Because typically the message length l will be far less than the number of available pixels $N \times M$, a message m can be embedded several times in an image, say K times. This can be done in various ways. It is important that not all the pixels corresponding to the same message bit are in the same area of an image. If for example, all third message bits m_3 are embedded in a part of an image that gets damaged after an attack, it will not be possible to retrieve this third message bit. In order to minimize this possibility this third message bit should be somehow uniformly spread over the image. For practical reasons it is convenient to use a clear structure for associating the message bits with image pixels, and therefore a random function will not be used.

For these reasons we use *block-based embedding*. The message is embedded in the following way: The image is divided in 8×8 blocks and the message is embedded in these blocks. Embedding is started in the upper left corner of the image and works towards the lower right corner; First going



$$\begin{array}{cccc} m_1 & m_2 & m_3 & \cdots \\ m_9 & m_{10} & & \\ m_{17} & & \ddots & \\ \vdots & & & \end{array}$$

$$\begin{array}{cccccccc} m_{47} & m_{48} & m_{49} & m_{50} & m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 & m_9 & m_{10} & m_{11} & m_{12} \\ m_{13} & m_{14} & m_{15} & m_{16} & m_{17} & m_{18} & m_{19} & m_{20} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & m_{27} & m_{28} \\ m_{29} & m_{30} & m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{37} & m_{38} & m_{39} & m_{40} & m_{41} & m_{42} & m_{43} & m_{44} \\ m_{45} & m_{46} & m_{47} & m_{48} & m_{49} & m_{50} & m_1 & m_2 \\ m_3 & m_4 & m_5 & m_6 & m_7 & m_8 & m_9 & m_{10} \end{array}$$

Figure 3.9: The message m is spread through the $N \times M$ image x and embedded several times. The message is embedded in 8×8 blocks, from the upper left corner to the lower right. After embedding in the upper left block, embedding is continued in the block immediately right to that block, and do on. Here, the message length l is arbitrarily chosen as $l = 50$. After bit m_{50} is embedded, m is embedded another time.

to the right and then downwards. After embedding in one block, we proceed with the next. This process is illustrated with Figure 3.9.

3.6 Scalar Costa Scheme detection

For detection we have to decide if the received value r represents 0 or 1. Because 0's and 1's are associated with even and odd levels, the following easy decision rule is used: If the received value is closest to an even level, 0 is detected, but if the received value is closest to an odd level, 1 is detected. Because subtractive dither is used, first the dither is added to the received value before using this decision rule.

If each pixel is associated with only one message bit, then the detection is equal to the detected message bit $\hat{m}_i$ and the stated decision rule is equivalent to calculating

$$\hat{m} = \left\lfloor \frac{r+d}{\frac{1}{2}\Delta} \right\rceil \pmod 2. \tag{3.18}$$

Equation (3.18) can be easily associated with Equation (3.5), the decoder for QIM.

For the case of embedding a message with repetition, there is a sequence of detection values,

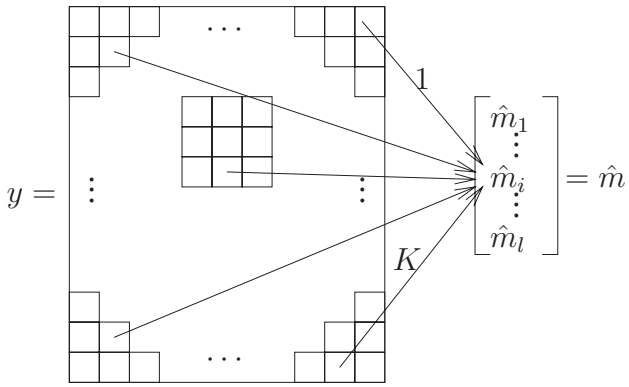


Figure 3.10: The message bits m_i are represented by several pixels in the watermarked image y of size $N \times M$. The detected message bits $\hat{m}_i$ are found, tracing back from the pixels. A message bit is embedded K times in the image.

calculated by Equation (3.18), associated with one message bit. The sequence first has to be traced back to the corresponding message bit. This is the inverse operation of the block based embedding procedure described in Section 3.5. This process is illustrated in Figure 3.10. A message bit is embedded in an image K times. At the detector the pixels representing one and the same message bit are grouped together in a set S_i.

At this point there are l sets of pixel values, where each set S_i, $i \in \{1, 2, \cdots, l\}$ is associated with one message bit m_i. If a message is embedded K times than each set S_i has size K. The question is now how to determine a detected message bit $\hat{m}_i$ from these values. This can be done using hard or soft decision decoding and is described in Subsection 3.6.1.

3.6.1 Hard versus soft decision decoding

In the sequel of this section a message bit m_i is not in the set $\{0, 1\}$, but in the set $\{-1, 1\}$. It is easy to go from one representation to another by using the relation $m_{-11} = 2m_{01} - 1$, where m_{-11} is a representation of the message m in the set $\{-1, 1\}$, and m_{01} a representation of the m in $\{0, 1\}$. This will show to be a convenient representation, because of the used detection by correlation.

Using *hard decision decoding* the values in the sets S_i, $i \in \{1, 2, \cdots, l\}$ are calculated using Equation (3.18) and then going to the representation in the set $\{-1, 1\}$. So the values in S_i are -1 or 1. The detected message bit $\hat{m}_i$ is determined using a so-called *majority vote*, i.e., if the majority of these values in S_i are -1, $\hat{m}_i = -1$ and if the majority is 1, $\hat{m}_i = 1$. In the case of a tie, it is undecided what $\hat{m}_i$ will be.

Because some of the bits are more reliable than others, it is desirable to give a larger weight to the reliable ones than to the unreliable ones. A measure of reliability is the distance from the received value plus the dither $r + d$ to the nearest level. If this distance is zero, the reliability is maximal; If this distance is $\frac{1}{4}\Delta$ (the maximal possible distance), the reliability of this bit is minimal. The values in the sets S_i are now K pairs (s, L), where s is this distance and L the value of the nearest odd or even level (-1 or 1).

The reliability of a bit can be taken into account by using these distances s as a weight. The detected message bit $\hat{m}_i$ is then calculated as the weighted average over all K values corresponding

to the message bit $\hat{m}_i$. This procedure of determining the detected message bit is called *soft decision decoding*. The weight s is given by

$$s = 1 - \frac{4}{\Delta}\left| r + d - \left\lfloor \frac{r+d}{\frac{1}{2}\Delta} \right\rceil \frac{1}{2}\Delta \right|. \tag{3.19}$$

It can be easily seen that if a received value plus the dither is close to a level, s is close to 1 and vice versa. Now the weighted average $\tilde{m}_i$ for each message bit $\hat{m}_i$, $i \in \{1, 2, \cdots, l\}$ can be calculated by

$$\tilde{m}_i = \frac{\sum_{k=1}^{K} s_{ik} L_{ik}}{\sum_{k=1}^{K} s_{ik}}, \tag{3.20}$$

where s_{ik} and L_{ik} are the soft decision value and the nearest level respectively for the i^{th} message bit and the k^{th} pair from the set S_i. The detected message bit $\hat{m}_i$ is determined by

$$\hat{m}_i = \begin{cases} 1 & \text{if } \tilde{m}_i \geq 0 \\ -1 & \text{if } \tilde{m}_i < 0 \end{cases}, \tag{3.21}$$

which can be converted into a binary message by calculating $\hat{m}_{01} = \frac{1}{2}(\hat{m}_{-11} + 1)$.

3.6.2 Detection by correlation

At this point the detected message $\hat{m}$ is known at the decoder. If also the original message m is known at the decoder then detection can be done by calculating the correlation between the original embedded message m and the detected one $\hat{m}$. For high correlations, the detected message $\hat{m}$ is (almost) the same as the embedded message m. For low correlations there will be too much difference between $\hat{m}$ and m, so that there will not be enough evidence in order to decide that an image is watermarked with a message m.

If the correlation between m and $\hat{m}$ is greater or equal to a certain threshold T, it is said that the watermark is detected; otherwise the watermark is not detected. The correlation is calculated by

$$\text{cor}(\hat{m}, m) = \frac{1}{l} \sum_{i=1}^{l} \hat{m}_i m_i. \tag{3.22}$$

The threshold T represents the amount of similarity between $\hat{m}$ and m that is wanted, before it is declared that an image is watermarked with a message m. If T is close to 1, almost every bit of $\hat{m}$ should equal m. If $\text{cor}(\hat{m}, m) = 0$, then $\hat{m}$ and m are only similar for 50 % of the bits, the same amount as one would expect from two random messages. For $\text{cor}(\hat{m}, m) = -1$ every detected message bit $\hat{m}_i$, $i \in \{1, 2, \cdots, l\}$ is the opposite of the corresponding message bit m_i. In the next subsection the question of setting this threshold is treated.

It is possible that the original message is not available for comparison at the decoder. In this case some structure on m can be used. For example, a message m can be constructed from two parts. The first part is known to the decoder and serves as a mean to determine whether or not a message

is embedded. This part can be detected by correlation. The second part is an arbitrary message. If the existence of a message is detected by means of the first part, then the second part is that message.

So, summarizing we have the following. A message $m = m^{(1)}m^{(2)}$ consisting of two parts is embedded in an image. The decision whether or not the received image r is watermarked is based on:

$$\begin{cases} \text{cor}(\hat{m}^{(1)}, m^{(1)}) > T & r \text{ is said to be watermarked} \\ \text{cor}(\hat{m}^{(1)}, m^{(1)}) \leq T & r \text{ is said to be unwatermarked} \end{cases}, \tag{3.23}$$

where $m^{(1)}$ is known at the detector. If r is said to be watermarked, it is said to be watermarked with the message $\hat{m}^{(2)}$.

Another possibility is to choose messages m from a code $\mathcal{C}$, so $m = c, c \in \mathcal{C}$. The detector knows the codebook corresponding to this code. At the detector a message $\hat{m}$ is estimated. If $\hat{m}$ is equal to or close enough to a codeword $\hat{c} \in \mathcal{C}$, then r is said to be watermarked with message $\hat{c}$.

3.6.3 The threshold setting

The *threshold setting* determines the trade-off between the false positive and the false negative probabilities. For a high threshold setting the false positive probability will be low; It is unlikely that in an unwatermarked image, the detected message will be almost identical to the reference message m. The probability that detection on an image watermarked with message m results in a detected message $\hat{m}$ almost identical to m, will be low for high thresholds, even for moderate attacks. For lower threshold settings the false negative probabilities are lower, but the false positives probabilities are higher. The false negative probability is hard to model, so we only look at the false positive probability in order to determine the threshold.

For an unwatermarked image it is also possible to extract a sequence of -1's and 1's. The false positive probability P_{fp} is now considered to be the probability that a detection will occur, i.e., the probability that the correlation with the reference message m is higher than the threshold T, that is

$$P_{\text{fp}} = P\left(\frac{1}{l}\sum_{i=1}^{l} n_i m_i \geq T\right) \tag{3.24}$$

Because the reference message m usually has an equal amount of -1's and 1's, it is assumed that m is uniformly distributed with $P(m_i = -1) = p$ and $P(m_i = +1) = 1 - p$, with $p = \frac{1}{2}$. We assume that a message n extracted from an unwatermarked image is also uniformly distributed, with $P(m_i = -1) = q$, $P(m_i = +1) = 1 - q$ and $q = \frac{1}{2}$. This assumption is valid, because the dither makes the pixel values uniformly distributed over a quantization bin [20, 24, 31], so the probability that a pixel value is closest to an even or an odd level is equal to $\frac{1}{2}$.

In order to derive the probability (3.24), first the distribution of $S_l = \frac{1}{l}\sum_{i=1}^{l} z_i$, with $z_i = n_i m_i$, has to be derived. We prove that the distribution of S_l can be approximated by a normal distribution:

Theorem 3.3.

$$\lim_{l\to\infty} P\left(\sqrt{l}\, S_l \leq x\right) = \frac{1}{2}\,\text{erf}\left(\frac{x}{\sqrt{2}}\right), \quad \forall x \in \mathbb{R},$$

where $\mathrm{erf}(x)$ *is the error function.*

Proof: It is easy to see that the z_i are also uniformly distributed with $P(z_i = -1) = 2pq + 1 - p - q = r$ and $P(z_i = +1) = -2pq + p + q = 1 - r$. The sum $T_l = \sum y_i$, with the y_i uniformly distributed with $P(y_i = 0) = r$ and $P(y_i = 1) = 1 - r$, is a binomial distribution. Note that y_i is over 0 and 1, where z_i is over -1 and 1. It is very well known (the Central Limit Theorem) that T_l can be approximated by a normal distribution with mean lr and variance $lr(1-r)$, for l sufficiently large:

$$\lim_{n\to\infty} P\left(\frac{T_l - lr}{\sqrt{lr(1-r)}\sqrt{l}} \leq x\right) = \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right), \quad \forall x \in \mathbb{R}, \tag{3.25}$$

where $\mathrm{erf}(x)$ is the error function. Because $z_i = 2y_i - 1$ is a linear function, the sum $\sum z_i$ can also be approached by a normal distribution with parameters

$$\mu_{\sum z_i} = \mathrm{E}\left[\sum z_i\right] = \mathrm{E}\left[2\sum y_i - l\right] = 2\mathrm{E}\left[\sum y_i\right] - l = (2r-1)l, \tag{3.26}$$

$$\sigma^2_{\sum z_i} = \mathrm{var}\left[\sum z_i\right] = \mathrm{var}\left[2\sum y_i - l\right] = 4\mathrm{var}\left[\sum y_i\right] = 4lr(1-r). \tag{3.27}$$

Now $S_l = \frac{1}{l}\sum_{i=1}^{l} z_i$ can also be approached by a normal distribution with parameters

$$\mu_{S_l} = \mathrm{E}\left[\frac{1}{l}\sum z_i\right] = 2r - 1, \tag{3.28}$$

$$\sigma^2_{S_l} = \mathrm{var}\left[\frac{1}{l}\sum z_i\right] = \frac{1}{l^2}\mathrm{var}\left[\sum z_i\right] = \frac{1}{l}\,4r(1-r). \tag{3.29}$$

So

$$\lim_{l\to\infty} P\left(\frac{S_l - (2r-1)}{\sqrt{\frac{1}{l}4r(1-r)}} \leq x\right) = \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right), \quad \forall x \in \mathbb{R}. \tag{3.30}$$

For our case of $p = q = \frac{1}{2}$, we have $r = \frac{1}{2}$, which gives the desired result. ■

The false positive probability is equal to (see Equation (3.24))

$$P_{\mathrm{fp}} = P\left(S_l \geq T\right) = \int_T^\infty \frac{\sqrt{l}}{\sqrt{2\pi}}\, e^{-l\frac{s^2}{2}}\,\mathrm{d}s = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{l}{2}}T\right), \tag{3.31}$$

with $\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}}\int_z^\infty e^{-t^2}\,\mathrm{d}t$ the complementary error function.

The desired false positive probability depends on the application. For a given false positive probability it is possible to calculate the corresponding threshold using Equation (3.31). Suppose the threshold belonging to a standard normal distribution given P_{fp} is T. The corresponding threshold T_X for a distribution $X \sim N(\mu, \sigma^2)$ is $T_X = \mu + \sigma T$. Because in our case $\mu = 0$ and $\sigma^2 = l$, the threshold $T_{S_l} = \frac{T}{\sqrt{l}}$.

l	$T_{10^{-3}}$	$T^{\exp}_{10^{-3}}$	$T_{10^{-5}}$	$T^{\exp}_{10^{-5}}$	$T_{10^{-7}}$	$T_{10^{-15}}$
10	0.977	0.999	1.350	0.999	1.644	2.510
50	0.437	0.439	0.604	0.600	0.735	1.123
100	0.309	0.301	0.427	0.419	0.520	0.794
500	0.138	0.137	0.191	0.188	0.233	0.355
1000	0.098	0.098	0.135	0.137	0.164	0.251

Table 3.1: Threshold settings for false positive probabilities $P_{\text{fp}} \in \{10^{-3}, 10^{-5}, 10^{-7}, 10^{-15}\}$. T is the theoretical derived threshold and $T^{\exp}$ the threshold resulting from experiments.

This threshold T_{S_l} is calculated for the false positive probabilities $\{10^{-3}, 10^{-5}, 10^{-7}$ and $10^{-15}\}$. For $P_{\text{fp}} = 10^{-3}$ and $P_{\text{fp}} = 10^{-5}$ this threshold is also determined by experiment in order to verify the model behind the theoretically calculated thresholds. For $P_{\text{fp}} = 10^{-3}$, 100.000 messages are pseudo randomly created and their correlation is calculated with an also pseudo randomly created reference message. Then it is investigated for which threshold setting T_{S_l} the number of false positives is smaller than or equal to 100 (a maximum of 100 false positives out of 100.000 gives $P_{\text{fp}} \leq 10^{-3}$). For $P_{\text{fp}} = 10^{-5}$ we have taken 10.000.000 sample messages. The results are stated in Table 3.1.

For low message lengths ($l = 10$) the theoretical results do not match with the experimental results. This is because the normal distribution is not a good approximation of the binomial distribution for low values of l. For higher values of l both results correspond, so we conclude that the model is correct for message lengths of at least 50. For lower message lengths l, the threshold T can be determined by experiments.

As said before, the threshold setting is a trade-off between the false positive and the false negative probability. If the threshold is chosen as $T = 1$, all bits must be correct, so it is very likely that an embedded message will not be detected and the false negative probability will be high. If the threshold is chosen as $T = 0$, the false positive probability will be $P_{\text{fp}} = P(S_l \geq 0) = \frac{1}{2}$ (see Equation (3.31)), which is very high.

3.7 Summary

We summarize the SCS embedding and detection methods, as they will be used throughout the remainder of this report.

- **Embedding** a message m in a host signal x is done by

$$y_{\text{SCS}} = x + \alpha \left(\left\lfloor \frac{x + d - \frac{1}{2} m\Delta}{\Delta} \right\rfloor \Delta - x - d + \frac{m\Delta}{2} \right), \tag{3.32}$$

  where the dither d, the quantization stepsize Δ and the distortion compensation parameter α are determined in advance. The dither d is a pseudo random uniformly distributed sequence and serves as a secret key. Δ and α are parameters to control robustness and perceptibility. The watermarked image y is sent over a channel, which models an attack.

- At the **detector** the image r is received. The message $\hat{m}$ is detected by

$$\hat{m} = \left\lfloor \frac{r+d}{\frac{1}{2}\Delta} \right\rceil \pmod 2. \tag{3.33}$$

In order to determine whether or not the received image r was watermarked, the correlation with the reference message m is calculated by

$$\mathrm{cor}(\hat{m}, m) = \frac{1}{l} \sum_{i=1}^{l} \hat{m}_i m_i. \tag{3.34}$$

For correlations higher than a threshold T, the image is said to be watermarked with message m.

Chapter 4

Problem description and Contributions

In Chapter 2 a general introduction to watermarking has been given. Because of the possible high payloads, attention has been focussed on the class of quantization watermark systems in Chapter 3, notably Quantization Index Modulation and the extension Distortion Compensated QIM, the Costa scheme and the Scalar Costa Scheme. For a Gaussian host signal and an AWGN channel the DC-QIM or Costa's scheme are optimal; They achieve the same capacity as non-blind detection. However, these schemes are not practical. The SCS is practical and sub-optimal under the condition that $x \sim N(0, \sigma_x^2 I_N)$ and the noise is AWGN.

In the real world these conditions may not be satisfied. Digital images are usually not Gaussian distributed and all the various attacks, discussed in Section 2.5, cannot be modelled by AWGN. For JPEG compression, for example, the noise n is highly correlated with the host signal x. For these reasons it is necessary to adapt the SCS, so that it can also be robust against other attacks, like, JPEG-compression and brightness scaling.

This reports focuses on two possible solutions to these deficiencies. The first solution is the use of Error Correcting Codes (ECC). Instead of embedding a message m into an image, it is encoded into the coded message m^c, which in turn is embedded. At the detector an estimate of this encoded message $\hat{m}^c$ is retrieved. Using an error correcting decoding mechanism an estimate of the original message $\hat{m}$ is found, with less errors than in the case of embedding m directly into the image. A short introduction to ECC's and an experimental evaluation of the performance increase by ECC methods is presented in Chapter 5.

Another solution is the use of an adaptive quantization step size. In standard SCS the quantization step size Δ is fixed and known at both the embedder and the detector. An adaptive quantization step size is dependent on the host signal x. Adaptive quantization introduces robustness against the attack of scaling the brightness values. At the same time it gives a more efficient allocation of watermark energy, because this allocation is conform a (simple) Human Visual Model: Weber's law, see Section 2.4. We will investigate several methods to implement adaptive quantization in Chapter 6. The performance of the several methods of adaptive quantization is compared with each other and with fixed quantization. The robustness gain with regard to brightness scaling compared to the use of a fixed quantization step size is also verified by experiment in this chapter.

We want to have a mathematical analysis of the performance of our adaptive quantization watermarking system. As a measure of performance the bit error probability is taken. This bit error probability is build up from two components: one which takes into account AWGN and uniform noise, with a fixed quantization step size, and the other an adaptive quantization step size, without

noise. Stochastic models for these two bit error probabilities are developed in Chapter 7.

The performance of the watermarking model depends on some system parameters, like the distortion compensation parameter α. An optimal value of this parameter should be applied at the encoder in order to get optimal performance. This optimization problem is treated in Chapter 8; The result is compared with a similar result of Joachim Eggers.

Summarizing, we have

- Improvements of current quantization based watermarking schemes have to be sought, in order to increase robustness;
- A first improvement is the use of ECC's. The increase of overall robustness is verified by experiment;
- A second improvement is the use of an adaptive quantization step size. A gain in robustness against brightness scaling is verified, both analytically and experimentally. The performance of adaptive quantization is compared with fixed quantization by experiment;
- An analytical performance analysis of the new watermarking system is made;
- The distortion compensation parameter α is optimized for a Gaussian signal and an AWGN channel.

Chapter 5

Enhancing robustness using Error Correcting Codes

In this chapter Error Correcting Codes (ECC) are discussed. These ECC provide a tool to enhance the robustness of a watermarking system, i.e., to minimize the probability that errors are being made when sending a message from one point to another using a watermarked signal. In Section 5.1 it is shown that indeed ECC can enhance robustness. Some well known ECC are mentioned in Section 5.2 and special attention is given to the Convolutional Codes (CC). Optimal decoding of CC is done with a Viterbi decoder, which is described in Section 5.3. Some results for the watermarking algorithm are given in Section 5.4.

5.1 The gain from using Error Correcting Codes

Coding theory started with the famous paper of Claude Shannon *"A mathematical theory of communication"* [33] in 1948. For a communications model where a message is sent over some channel that corrupts this message, Shannon identified a number called the capacity of the channel. He proved that for any rate below the capacity of the channel, arbitrarily reliable communication is possible. In order to receive a message, which is equal to the message that was sent, some redundancy is added to the message. This redundancy is used at the receiving side to detect and identify the errors introduced by the channel. Adding the redundancy is called *encoding* a message and retrieving an estimate of the sent message at the detector is called *decoding* a message.

This principle can be used in watermarking by encoding the message and embedding the encoded message into a host signal. Thus, a message m of length l is encoded to get the encoded message m^{c} of length $l^{\mathrm{c}} > l$. At the detector a distorted version of the encoded message m^c is retrieved, lets say $\hat{m}^{\mathrm{c}}$. Using a decoder, $\hat{m}^{\mathrm{c}}$ is decoded to get an estimate $\hat{m}$ of m. The sent and the received message will be equal if not too many errors have occurred. See Figure 5.1 for a watermarking model with error correction coding.

In fact, the robustness requirement in watermarking is a requirement of reliable communication. Hence Shannon's results, as well as other results of communications theory are relevant for robust watermarking. One of the consequences is that error correction can be used to improve robustness.

A very simple way of encoding a message is repetition coding; In this case the encoded message m^{c} is just a repeated version of m. Decoding can easily be done by a majority vote, see Section

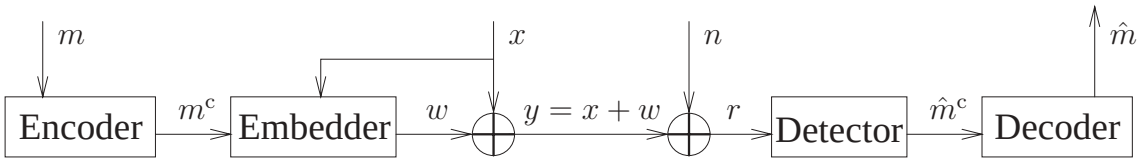


Figure 5.1: A general watermarking model using error correcting codes.

3.6.1. The larger the number of repetitions K, the more reliable is the decoding outcome. The price to be paid for this increased reliability, is the loss of rate. According to Shannon's theorem it is possible that communication at a certain rate is error free, provided the rate is below the channel capacity. In order to exploit this, a better code than repetition coding is needed. Shannon's proof is not constructive, so no code is given that satisfies his theorem. The search for practical, rate-achieving codes is still continuing. In the meanwhile codes are developed that approach the Shannon limit, some of them are listed in Section 5.2.

So, it is possible to use better codes than simply repetition codes. These better codes can correct more errors that occur when processing the image. But repetition coding is still used for two reasons: The first reason is that the watermark channel may have to operate at very high bit error rates. Under such severe conditions, some codes stop bringing in any advantage, while the repetition codes continue with their modest protection [2]. Concatenation of repetition codes and more advanced codes is a way of improving decoding performance in this critical range. The second reason is that repetition codes are very simple to implement, where better codes possibly add a lot of complexity to the watermarking algorithm, which is usually undesired.

5.2 Convolutional Codes

In practise often *linear codes* are used, because they are easier to describe, encode and decode than nonlinear codes. Let A^n denote the linear space of all n-tuples over the alphabet A, where A is a field, i.e., $A^n = \{(x_1, \cdots, x_n) \mid x_i \in A, i \in \{1, 2, \cdots, n\}\}$. An (n, k) linear code over the alphabet A is a code with the property that the set of codewords $\mathcal{C}$ is a k-dimensional linear subspace of A^n. Examples of linear codes are given in [25, 28]. We mention the Reed-Solomon (RS) codes, which are used in CD players, and Convolutional Codes (CC), which we use. A class of codes that performs close to the Shannon limit is the class of Turbo codes. Turbo codes are a clever combination of two or more convolutional codes.

Block codes are codes that encode a message of fixed length into a codeword with a larger, but also fixed length. For a *convolutional code* however, the codeword is not only derived from the present message bits, but also from a fixed amount of 'earlier' message bits (see [28]). The number of these earlier blocks is called the *memory* of the convolutional encoder.

For convolutional codes it is known that they have good performance and are easy to implement. Because these codes are the building blocks for the well performing Turbo codes, we use CC's.

There are different approaches to describe convolutional codes. A lot of them can be found in [25]. This report describes the shift-register approach, see Subsection 5.2.1, the convolutional representation, see Subsection 5.2.2, and the state representation, see Subsection 5.2.3, for explaining the encoding stage. For the Viterbi decoding of a CC, the state-diagram and the trellis approach are convenient, see Section 5.3.

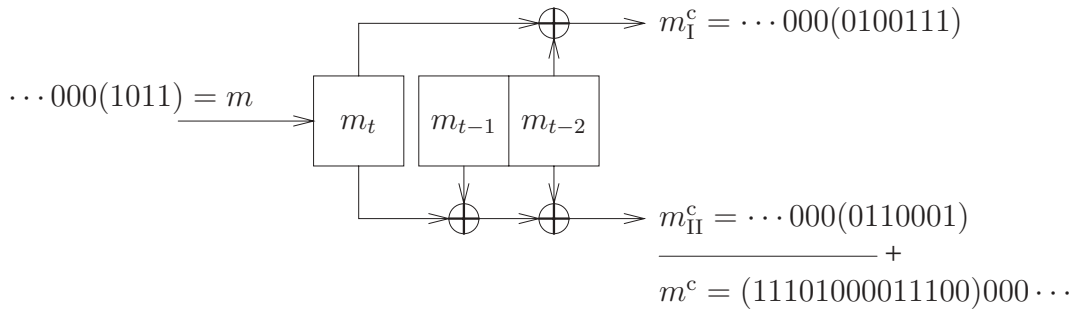


Figure 5.2: This is a shift register of memory size 2, the memory consists of the blocks m_{t-1} and m_{t-2}. Because there is 1 input and 2 outputs, this is a rate $\frac{1}{2}$ convolutional code. The encoding of the message $m = (1101)$ is illustrated. The encoding process starts and ends with state 00. At the start of the encoding process the first message bit $m_1 = 1$ is shifted into the memory and the output is calculated, and so on, until the last message bit is encoded. The $+$-operator represents the intertwining of $m_{\mathrm{I}}^{\mathrm{c}}$ and $m_{\mathrm{II}}^{\mathrm{c}}$

5.2.1 Shift-register approach

In order to encode a message using a convolutional code (CC) a device is needed that is capable of accepting an input stream of message bits and producing an encoded stream of codeword bits as output. One way to describe such a device is as a *shift-register encoder*. An example of a such an encoder is shown in Figure 5.2. Important properties of a CC are the rate and the memory size. The rate of a code is the quotient between the number of input bits and the corresponding number of output bits. A CC depends also on preceding message bits, $m_{t-1}, \cdots, m_{t-M}$, where M is the *memory size* (or the dimension of the state).

The *state* s_i of an encoder is defined as the contents of the memory blocks. For the CC shown in Figure 5.2, the state is the contents of m_{t-1} and m_{t-2}, so $s_1 = m_{t-1}$ and $s_2 = m_{t-2}$. In the case of binary messages there are 2^M states. For the CC of Figure 5.2 there are four different states: 00, 01, 10 and 11. It is possible to go from one state to another by an input, which has as result that the contents of the memory blocks are shifted to the right. It is for example possible for the encoder depicted in Figure 5.2 to go from state 01 to 10 with an input 1, but it is impossible to go from 00 to 11 with any input. The possible transitions from one state to another are described by a *state diagram*. The state diagram for the example encoder is depicted in Figure 5.3.

Encoding can be simulated by traversing the state diagram, but it is difficult to keep track of time. A diagram that combines the possible transitions to states and the factor time is a *trellis*. A trellis that describes this for a finite time is called a truncated trellis. The truncated trellis for the example encoder of Figure 5.2 is shown in Figure 5.4.

5.2.2 Convolutional representation

The *convolutional representation* of a convolutional code is given by

$$m^{\mathrm{c}}(t) = \sum_{j=-\infty}^{t} h(t-j)m(j),$$

where $h(t)$ is the impulse response or the convolutional kernel of the coding system. We illustrate this representation with an example.

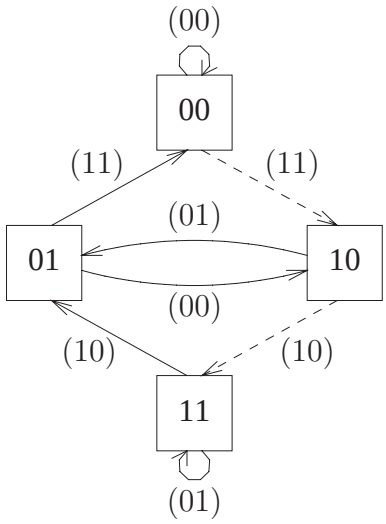


Figure 5.3: Reproduced from [25]. This is a state diagram for the encoder of Figure 5.2. The states are displayed in the squares. The arrows represent possible transitions: A dashed arrow when the encoder input is 1 and a solid arrow for an input 0. The numbers between brackets at the arrows are the encoder outputs corresponding to a transition from one state to another.

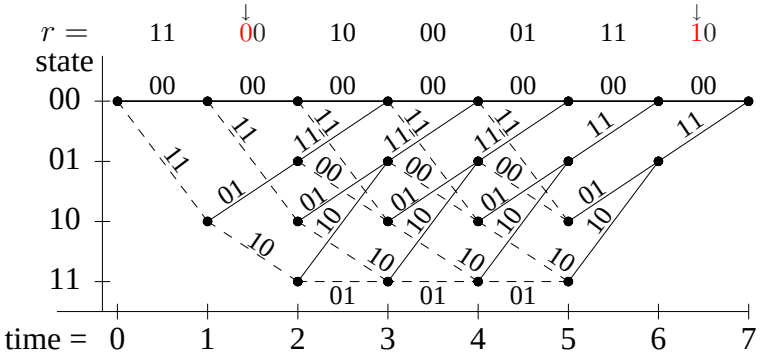


Figure 5.4: Reproduced from [25]. A truncated trellis for the encoder of Figure 5.2. In the top of the figure the received message r is shown. Two errors are made at the indicated positions. The transitions are labelled with the encoder outputs. A dashed line corresponds with an encoder input 1 and a solid line with encoder input 0.

Example 5.1. *For the convolutional encoder given in Figure 5.2 the impulse response $h(t)$ is given by*

$$\begin{aligned} h &= \begin{bmatrix} h(0) & h(1) & h(2) & \cdots \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & \cdots \\ 1 & 1 & 1 & 0 & \cdots & 0 & \cdots \end{bmatrix}, \end{aligned}$$

and $h(t) = 0$ for $t < 0$, and the message $m(t)$ is

$$m = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & \cdots & 0 & \cdots \end{bmatrix},$$

and $m(t) = 0$ for $t < 0$. Encoding m is now done as follows

$$\begin{aligned} m^{\mathrm{c}}(0) &= \sum_{j=-\infty}^{0} h(-j)m(j) = h(0)m(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\ m^{\mathrm{c}}(1) &= \sum_{j=-\infty}^{1} h(1-j)m(j) = h(1)m(0) + h(0)m(1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} 1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ m^{\mathrm{c}}(2) &= \sum_{j=0}^{2} h(2-j)m(j) = h(2)m(0) + h(1)m(1) + h(0)m(2) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} 0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ m^{\mathrm{c}}(3) &= \sum_{j=1}^{3} h(3-j)m(j) = h(2)m(1) + h(1)m(2) + h(0)m(3) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} 0 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} 1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ &\vdots \end{aligned}$$

which gives as encoded message

$$m^{\mathrm{c}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & \cdots \end{bmatrix},$$

or written down differently $m^{\mathrm{c}} = (11101000011100)00\cdots$, which can be compared with the output of the encoder in Figure 5.2.

5.2.3 State representation

A *state representation* is given by

$$\begin{cases} s(t+1) &= A\, s(t) + B\, m(t), \\ m^{\mathrm{c}}(t) &= C\, s(t) + D\, m(t). \end{cases} \tag{5.1}$$

Here, m and m^{c} are the message and the encoded message respectively, and s is the state. The matrices A, B, C and D are of appropriate size. The first equation in (5.1) represents the shifting of the registers from one state to another. The second equation represents the form of the encoder. Once again, an example will make things clear.

Example 5.2. *For the convolutional encoder given in Figure 5.2 the state representation is given by*

$$\begin{cases} s(t+1) &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} s(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} m(t), \\ m^{\mathrm{c}}(t) &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} s(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} m(t). \\ s(0) &= 0 \end{cases} \tag{5.2}$$

The coding of a message $m = (1101)00\cdots$ *is done as follows*

$$\begin{aligned} t = 0 \quad &\Longrightarrow \quad s(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad m^{\mathrm{c}}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ t = 1 \quad &\Longrightarrow \quad s(2) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad m^{\mathrm{c}}(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ t = 2 \quad &\Longrightarrow \quad s(3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad m^{\mathrm{c}}(2) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &\vdots \end{aligned}$$

This gives as encoded message $m^{\mathrm{c}} = (11101000011100)00\cdots$. *Compare this with Figure 5.2 and Example 5.1.*

5.3 Viterbi decoder

Convolutional codes can be decoded with a number of decoding algorithms, one of which is Viterbi decoding. When a decoder receives a sequence, it has to estimate the sequence that was really sent. The decoder will be optimal if it chooses the estimate that maximizes some log-likelihood function. Viterbi decoding is convenient in the sense that it is maximum likelihood [23], but the complexity is an exponential function of the memory depth M.

The trellis representation of the encoding process, see Figure 5.4, can be used to decode a CC. But it is needed to label the edges of the trellis with some distance measure. This distance measure is for hard decision decoding often the Hamming distance, and for soft decision decoding the Euclidean distance. In Figure 5.5 the trellis is shown, with edges labelled with the Hamming distance between the received encoded message bits and the possible encoder outputs. The total Hamming distance between the received encoded message r and a codeword c is the sum of all labels on the path corresponding to that codeword. This sum is just the length of the trellis path corresponding to that codeword.

Decoding the received message r is a matter of finding the codeword closest to r. This is equivalent to finding the shortest path from the start to the end of the trellis. Finding the shortest path can be done using the Viterbi algorithm. The *Viterbi algorithm* is a form of dynamic programming. It is based upon the simple observation that once the shortest paths to all states at time t is found, the shortest paths at time $t + 1$ can be constructed from the former paths. The Viterbi algorithm is illustrated by decoding a received encoded message for the example CC, as shown in Figure 5.6.

More formally, the Viterbi decoding algorithm is given by the following scheme. The time is denoted by t. The states are in the set S, and the all-zero state is denoted by $\mathbf{0}$. The labels of the

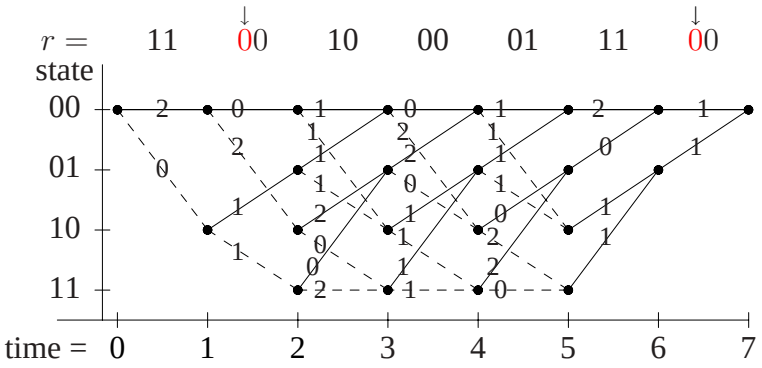


Figure 5.5: Reproduced from [25]. Another truncated trellis for the encoder of Figure 5.2. This time the transitions are labelled with the Hamming distance between the encoder output and the received message r.

trellis of Figure 5.5 are denoted by $l_{t-1,t}(u, v)$, where $u, v \in S$ are states. For example the label from time 2 to 3 and state 10 to 01 in Figure 5.5 is equal to 2, so in this case $l_{2,3}(10, 01) = 2$. The cost of state $u \in S$ up to a time t is the length of a path from the zero state at time 0 up to state u at the current time, and is denoted by $C_t(u)$. It is necessary to keep track of the path followed from the zero state at time 0 to the state u at time t. This is denoted by $P_t(u)$. Finally, $B(u, v)$ is equal to 0 or 1, depending whether a transition is labelled with 0 or 1 (a solid or a dashed line, respectively). Remember l is the message length and M the memory depth.

Step 1: (Initialization) For all states, set the costs to zero, set the followed path to an empty set and set the time to 1, i.e.,

$$\begin{aligned} P_0(\mathbf{0}) &= \emptyset \\ \forall u \in S \Rightarrow C_0(u) &= 0 \\ t &= 1 \end{aligned}$$

Step 2: (Recursion) $\forall u \in S$, do

$$v_0 = \arg\min_{v \in S} \{C_{t-1}(v) + l_{t-1,t}(v, u)\}$$

and $\forall u \in S$ set

$$\begin{aligned} C_t(u) &= C_{t-1}(v_0) + l_{t-1,t}(v_0, u) \\ P_t(u) &= P_{t-1}(v_0) * B(v_0, u) \end{aligned}$$

Step 3: If $t = l + M$ stop, else set $t = t + 1$ and go to Step 2.

5.4 Experimental results for ECC in the SCS

5.4.1 Experiment description

We have implemented repetition coding and convolutional coding into the watermarking embedding scheme. For our implementation it is also possible to use the *concatenation* of both codes,

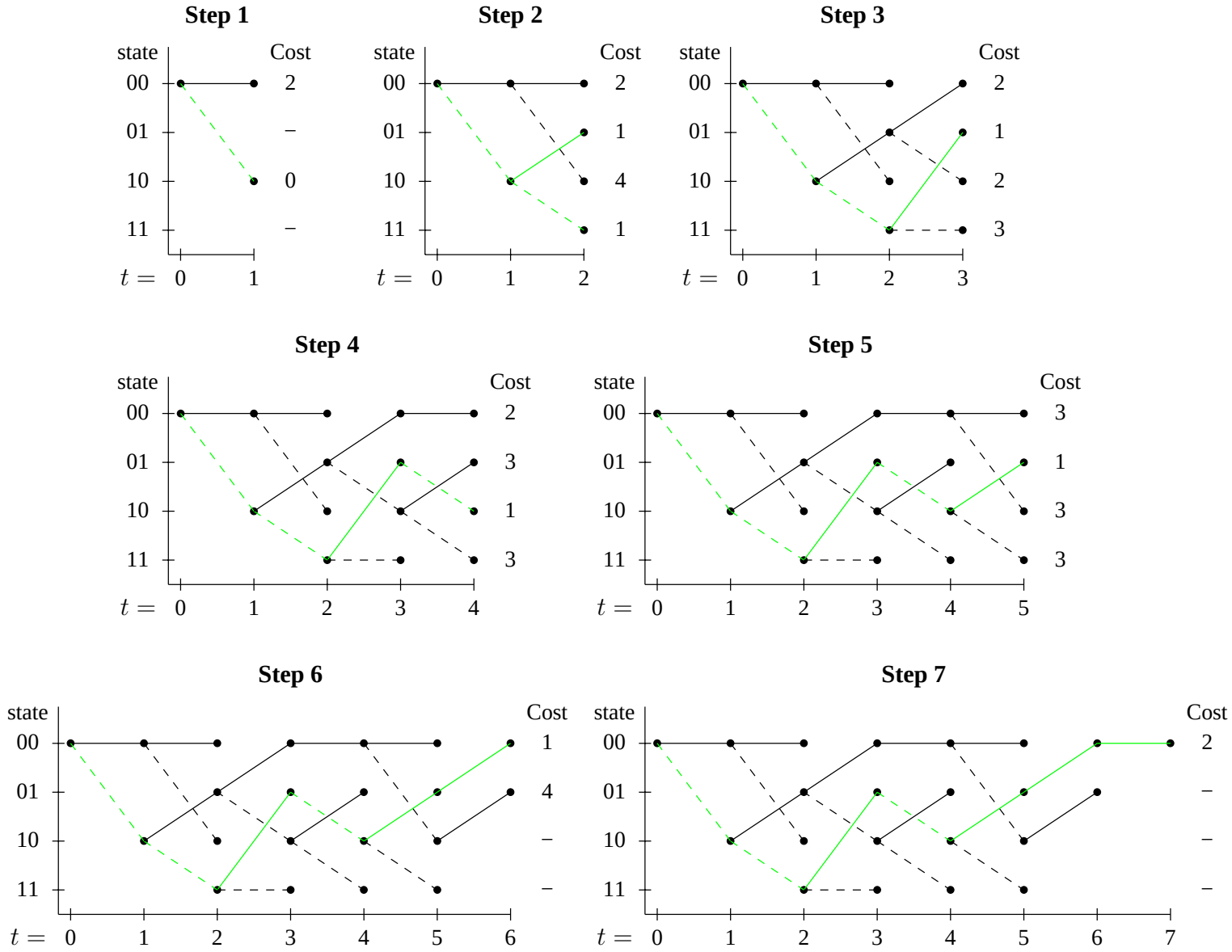


Figure 5.6: The Viterbi algorithm. For each state and each time, the shortest path and the corresponding costs are calculated using the Viterbi algorithm. For each step the shortest path is depicted in green. The decoded message can easily be read off, by following the shortest path (in green), and is found to be $\hat{m} = (1101)$.

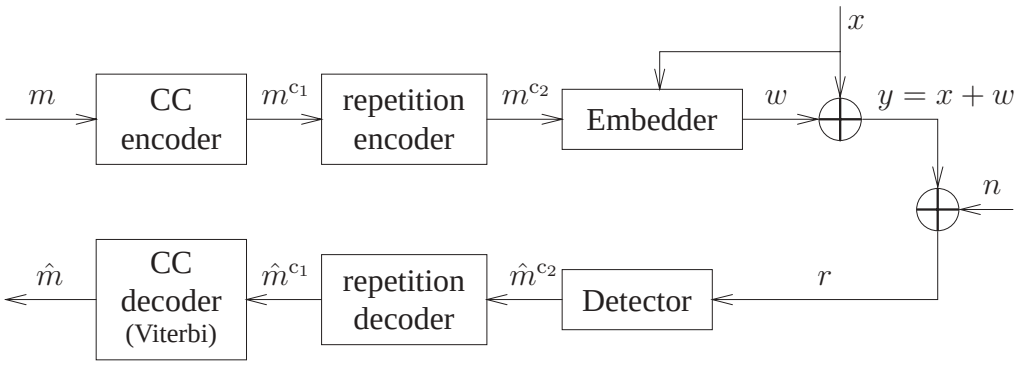


Figure 5.7: A watermarking model using concatenated codes.

with the convolutional code as an outer code. This means that the massage m is first encoded using convolutional coding, resulting in m^{c_1} with length l^{c_1}, and than m^{c_1} is encoded using a repetition code, resulting in m^{c_2}, with length l^{c_2}, where usually $l^{c_2} = N$. See Figure 5.7 for illustration.

It is possible for our implementation to use an arbitrary number of inputs k and outputs n, so any rate $\frac{k}{n}$ can be achieved. In the detector a Viterbi decoder is used. The Viterbi algorithm uses more computer memory if a larger CC memory size M is used; This relation is exponential. There is a linear relationship between the message length l and the decoding time. For our implementation it is possible on a PIV 1.7 GHz with 256 MB internal memory, to use a CC memory size M up till 9. We observed that the complexity of a watermarking algorithm with convolutional coding is much larger than one without, because decoding time increases dramatically.

In order to be able to determine the performance of the watermarking algorithm, we have carried out some experiments. Testing was done using repetition coding or a concatenated code (CC and repetition coding together), for different rates, different watermark strengths and different attacks with different strengths. We performed two kinds of attacks: namely AWGN and JPEG attacks. We varied the watermark variance σ_w^2, the noise variance σ_n^2 of the AWGN and the quality factor of the JPEG compression. In order to be able to know the relative strength of the watermark compared to the host signal, the *Document-to-Watermark Ratio* (DWR) is used:

$$\mathrm{DWR} = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_w^2} \right), \tag{5.3}$$

where the document is in fact the host signal. In our experiments we have used the NASA CC, depicted in Figure 5.8, because it is known that this code has a large minimum distance.

We have done experiments on four images, namely `lena.bmp`, `baboon.bmp`, `peppers.bmp` and `tulips.bmp`, see Appendix B. Because the experiments are limited only qualitative conclusions can be drawn, not quantitative.

5.4.2 Results and discussion

The results of these experiments are displayed in Figure 5.9 - 5.13. The Bit Error Rates (BER's) in the graphs are the average BER's for the four images used for the experiments.

Some qualitative conclusions can be drawn from these figures. We see that outcome of the experiments is in line with what we would expect. In Figure 5.9 it is seen that if the AWGN variance σ_n^2 increases, also the BER increases. In Figure 5.10 we see that the BER decreases if the JPEG

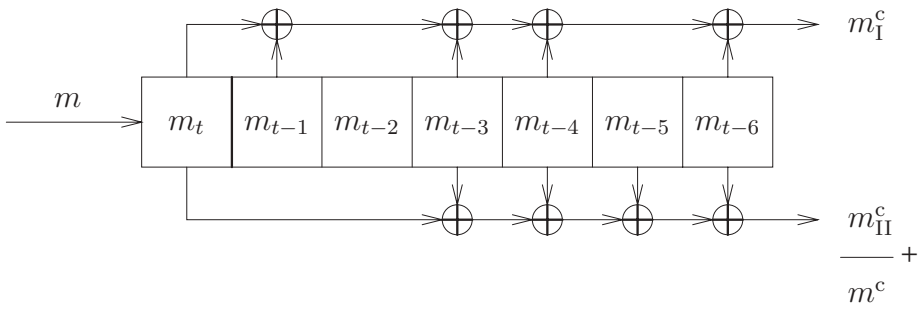
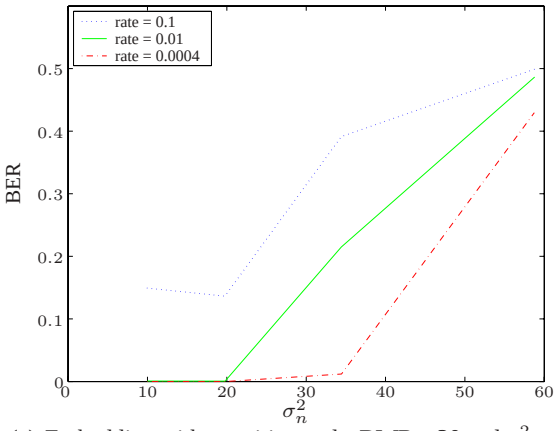
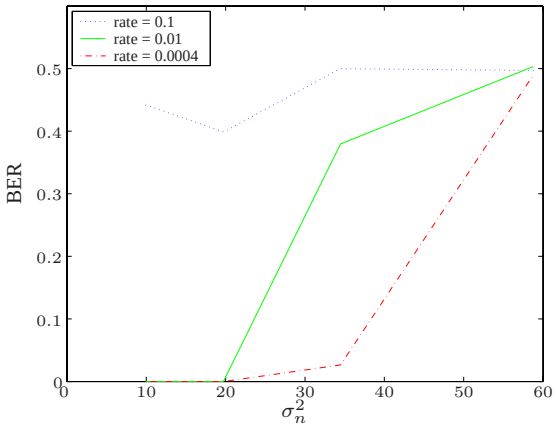


Figure 5.8: The convolutional encoder used by NASA for deep space communication. This CC has rate $\frac{1}{2}$ and memory depth $M = 6$.

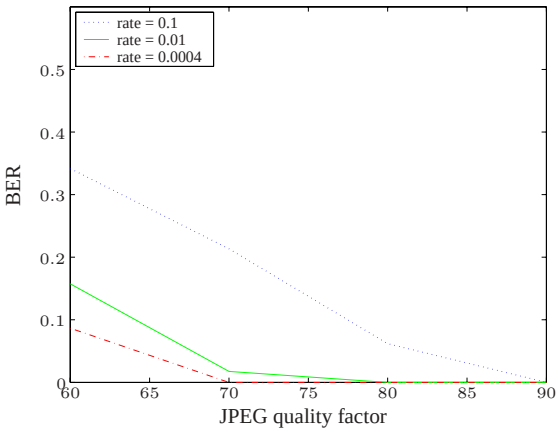


(a) Embedding with repetition code, DWR = 30 and $\sigma_w^2 = 3.2$.

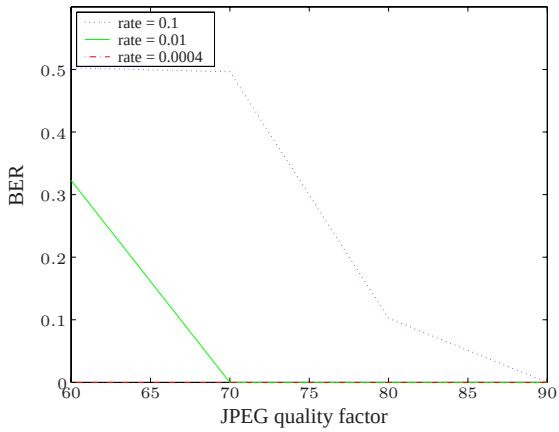


(b) Embedding with concatenated code, DWR = 30 and $\sigma_w^2 = 3.2$.

Figure 5.9: The BER as a function of the AWGN variance σ_n^2, for different rates.

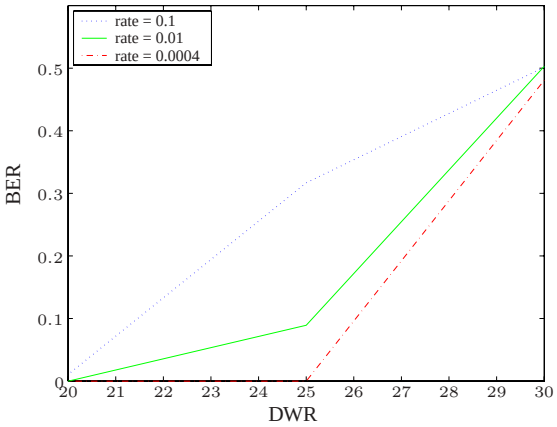


(a) Embedding with repetition code, DWR = 20 and $\sigma_w^2 = 38.8$.

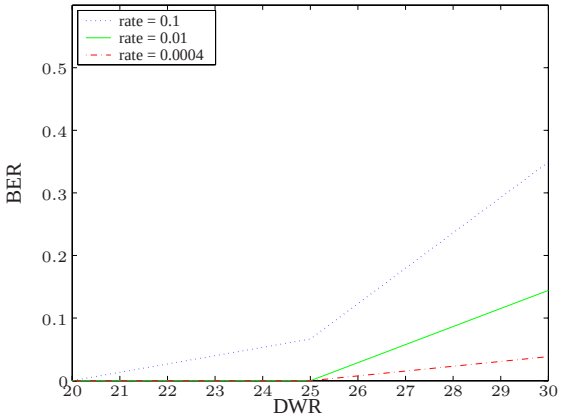


(b) Embedding with concatenated code, DWR = 20 and $\sigma_w^2 = 38.8$.

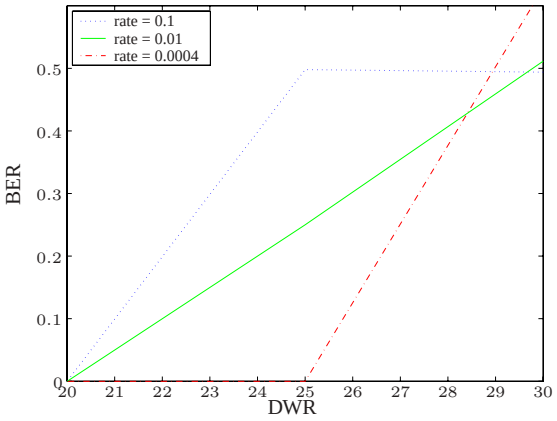
Figure 5.10: The BER as a function of the JPEG quality factor, for different rates.



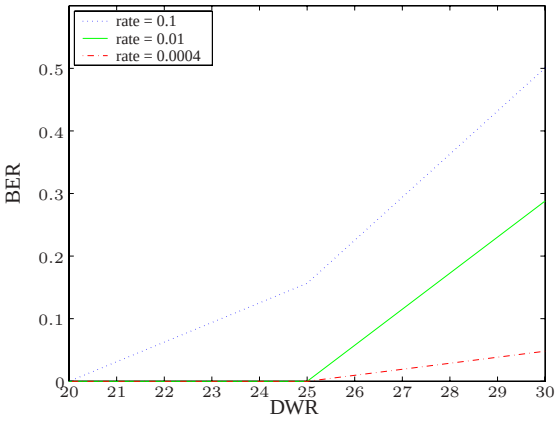
(a) Embedding with repetition code, an AWGN attack with $\sigma_n^2 = 58.8$.



(b) Embedding with repetition code, a jpeg attack with quality factor 90 and $\sigma_n^2 = 42.0$.

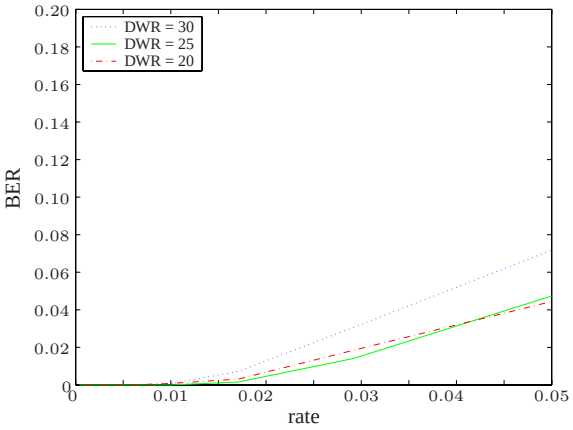


(c) Embedding with concatenated code, an AWGN attack with $\sigma_n^2 = 58.8$.

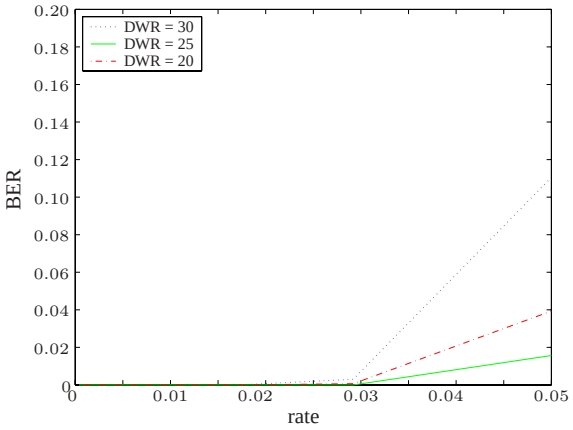


(d) Embedding with concatenated code, a jpeg attack with quality factor 90 and $\sigma_n^2 = 42.0$.

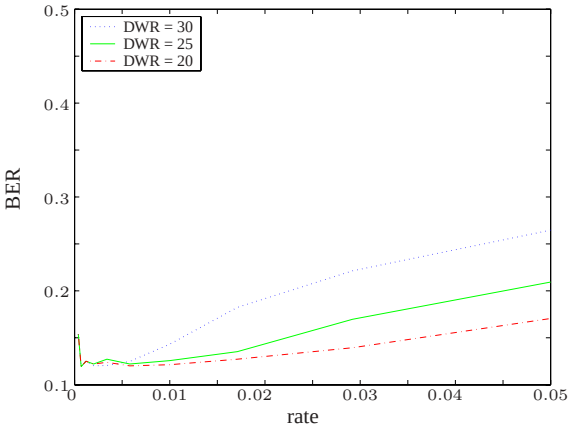
Figure 5.11: The BER as a function of the DWR, for different rates.



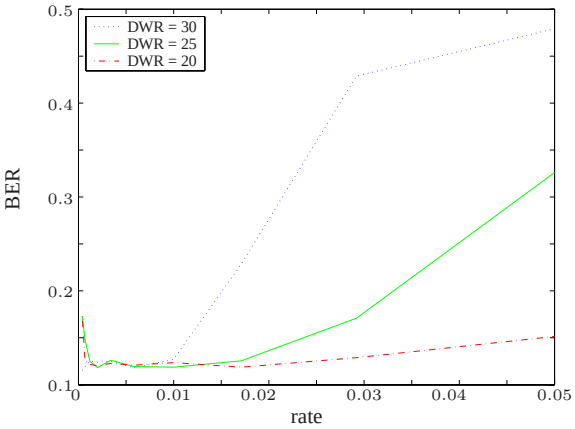
(a) Embedding with repetition code, a noise attack with $\sigma_n^2 = 10.0$.



(b) Embedding with concatenated code, a noise attack with $\sigma_n^2 = 10.0$.

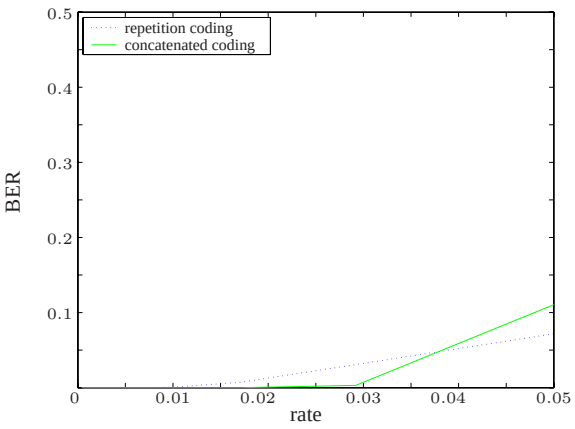


(c) Embedding with repetition code, a jpeg attack with quality factor 90 and $\sigma_n^2 = 41.0$.

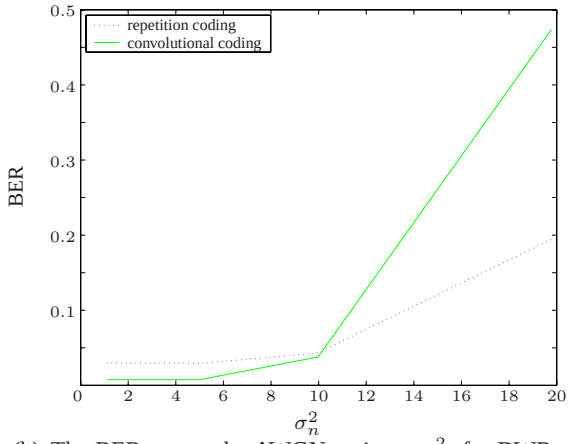


(d) Embedding with concatenated code, a jpeg attack with quality factor 90 and $\sigma_n^2 = 41.0$.

Figure 5.12: The BER as a function of the rate, for different DWR's.



(a) The BER versus the rate, for DWR = 30, $\sigma_w^2 =$ and $\sigma_n^2 = 10.0$.



(b) The BER versus the AWGN variance σ_n^2, for DWR = 25, $\sigma_w^2 =$ and rate = $\frac{1}{2}$.

Figure 5.13: Repetition coding compared with concatenated / convolutional coding.

quality factor increases. In Figure 5.11 we see that if the DWR increases, also the BER increases. In all these three figures and in Figure 5.12 we see that if the rate decreases, the BER decreases. All these observations are what one would expect in a watermark environment. Summarizing, we have

- $\sigma_n^2 \uparrow \implies$ BER $\uparrow$
- JPEG quality factor $\uparrow \implies$ BER $\downarrow$
- DWR $\uparrow \implies$ BER $\uparrow$
- rate $\uparrow \implies$ BER $\uparrow$

In Figure 5.13(a) repetition coding is compared with concatenated coding and in Figure 5.13(b) with convolutional coding. In Figure 5.13(a) we see that it is better to use concatenated coding than repetition coding alone up till a certain rate. In Figure 5.13(b) we observe that convolutional coding is better than repetition coding up till a certain noise attack strength. For higher AWGN variances the convolutional code stops bringing in advantage, because to many bits are erroneous. The repetition code continues bringing his modest protection in this noise variance range. This also explains why the concatenated code is doing worse than repetition coding alone if the rate is too high. The inner repetition code does not bring the bit error low enough if rates are too high, and therefore the convolutional code does not work anymore. See also [2] and Section 5.1.

Chapter 6

Enhancing robustness using an adaptive quantization step size

The Scalar Costa Scheme uses a fixed quantization step size Δ, i.e., for each pixel of an image the same Δ is used. The quantization step size is a parameter to control the tradeoff between robustness and visibility. This is true because a large Δ separates two quantization levels by a large distance, so a stronger noise signal is needed in order to get an error. On the other hand, for a larger Δ the perceptual distortion is also larger. From Section 2.4 it is known that larger modifications can be made in bright areas of a signal (brightness sensitivity). Therefore robustness can be gained in those areas by embedding with a larger step size, without losing on the perceptibility aspect.

The most important reason of using an *adaptive quantization* step size is that is gives robustness against brightness changes. This is explained in Section 6.1. Another effect of adaptive quantization is that choosing a larger quantization step size in bright areas and a smaller one in darker regions, is less visible for the human eye.

Some of the advantages and disadvantages are described in Section 6.1. In Section 6.2 options for determining the adaptive quantization step size are given and reflected on.

6.1 Advantages and disadvantages

6.1.1 Advantages

There are two advantages when using an adaptive quantization step size. The first one is robustness against *brightness scaling*. The second one is that the embedding strength is more or less proportional to the perceptual sensitivity to distortions.

Because for bright image areas a larger quantization step size is chosen, a larger robustness is achieved in those areas; Information embedded in bright areas can be retrieved at the detector with larger reliability. The overall robustness will gain from this adaptive quantization, while the watermark is as perceptible as in the case of a fixed quantization step size.

Suppose a simple model for determining the adaptive quantization step size is used: there is simply a linear relation between Δ and the group of pixel values x_i. Such a linear relationship can be defended by referring to Weber's law, that gives a linear relationship between the sensitivity of the

human eye and the luminance value. So, the quantization step size is determined by

$$\Delta(x) = \frac{\gamma}{L} \sum_{i=1}^{L} x_i, \tag{6.1}$$

where L is the number of pixels in the aforementioned group and where γ is some embedding strength parameter. It can easily be seen that this model is brightness scale invariant. If the brightness of an image is scaled by a factor β, then in fact the scaled image is $x_{\text{scaled}} = \beta x$. In the unscaled case, detection is done with the quantization stepsize (6.1) and by calculating

$$m = \left\lfloor \frac{x}{\Delta(x)} \right\rceil \mod 2 = \left\lfloor \frac{x}{\frac{\gamma}{L}\sum_{i=1}^{L} x_i} \right\rceil \mod 2. \tag{6.2}$$

In the scaled case detection is done with quantization step size

$$\Delta_{\text{scaled}}(x) = \frac{\gamma}{L} \sum_{i=1}^{L} \beta x_i, \tag{6.3}$$

and the detection is

$$m = \left\lfloor \frac{x_{\text{scaled}}}{\Delta_{\text{scaled}}(x)} \right\rceil \mod 2 = \left\lfloor \frac{\beta x}{\frac{\gamma}{L}\sum_{i=1}^{L} \beta x_i} \right\rceil \mod 2, \tag{6.4}$$

which is equal to the detection in the unscaled case.

6.1.2 Disadvantages

For a fixed quantization step size, both at the embedder and the detector, Δ is known. For an adaptive quantization step size Δ is a function of x. At the embedder the quantization step size is calculated by $\Delta(x)$, but at the detector Δ has to be calculated from the received signal r, so the quantization step size is $\Delta(r) = \Delta(y+n)$. This $\Delta(r)$ is only an estimation of the quantization step size used at the embedder and therefore may not be completely accurate. Because the detection depends on the adaptive quantization step size, see Equation (3.18), the estimation error causes bit errors in estimating the received message.

6.2 The adaptation rule

Consider the case without dither modulation and distortion compensation. Conceptually, we would like the quantization step size to be a linear function of the corresponding pixel value, so $\Delta(x) = \gamma x$. Then, at the embedder QIM is done by (see Equation (3.14)):

$$y_{\text{QIM}} = \left\lfloor \frac{x - \frac{1}{2}m\gamma x}{\gamma x} \right\rceil \gamma x + \frac{m\gamma x}{2} = \left(\left\lfloor \frac{1}{\gamma} - \frac{m}{2} \right\rceil \gamma + \frac{m\gamma}{2} \right) x. \tag{6.5}$$

In this way y_{QIM} does not contain information about the embedded message bit. This can be seen by applying the detector. In order to estimate the message bit at the detector (see Equation (3.18)), the corresponding quantization step size Δ has to be estimated with $\hat{\Delta} = \gamma r$. The message bit is

then estimated with $\hat{m} = \left\lfloor \frac{r}{\frac{1}{2}\Delta} \right\rceil \mod 2 = \left\lfloor \frac{r}{\frac{1}{2}\gamma r} \right\rceil \mod 2 = \left\lfloor \frac{1}{\frac{1}{2}\gamma} \right\rceil \mod 2$, which does not depend on the encoded bit at all. Therefore, this method of determining an adaptive quantization step size is not usable.

Therefore another adaptation rule has to be developed, which does allow actual embedding and detection of information.

A way of making the estimation of Δ more robust, is to not only use the pixel value itself, but rather the average of that pixel value and the surrounding ones. The surrounding pixels of a pixel x_{ij} are the pixels that fit in a square with center x_{ij} of size $L = F \times F$. The idea is that using this new average value, the detection result still depends on the encoded bit. Also, average values are less sensitive to changes in the host signal x than the pixel value alone. The latter argument is true, since if the adaptive quantization step size is chosen to be (6.1) and if due to some processing zero mean noise is added to the watermarked image, then detection is done with a quantization step size

$$\hat{\Delta}(r) = \frac{\gamma}{L} \sum_{i=1}^{L} r_i = \frac{\gamma}{L} \sum_{i=1}^{L} (x_i + w_i + n_i) \tag{6.6}$$

$$\approx \frac{\gamma}{L} \sum_{i=1}^{L} x_i, \tag{6.7}$$

if L is large enough, because $\mathrm{E}[w] = 0$ and $\mathrm{E}[n] = 0$. By applying the detector, it can be seen that with this procedure, consistent decoding is possible.

Another possible adaptation rule is to let the step size depend in a nonlinear way on the corresponding pixel, e.g. using a staircase function; Every averaged pixel value in a certain range corresponds to the same quantization step size. Now, when at the detector a received value falls into a certain range, no error will be made as long as this received value has not crossed the edges of that range. For example choose

$$\Delta(x) = \begin{cases} 4\,\gamma & \text{if } 0 \leq x \leq 50 \\ 7\,\gamma & \text{if } 50 < x \leq 150 \\ 12\,\gamma & \text{if } 150 < x \leq 255 \end{cases}, \tag{6.8}$$

with γ the embedding strength parameter. See Figure 6.1 for this staircase function. For this staircase function errors can be made if the averaged pixel value is around the jumping points 50 and 150.

Other means of choosing a quantization step size are also possible. For example Δ could be chosen to be the median from a sequence of pixel values. From the median it is known that it is robust against outliers. The visual model of high values of Δ with high values of the host signal, still applies in this case. Other possibilities involve the use of the variance or the second moment or some clever combination of all methods of choosing Δ. In this case another visual model is applied, because a high variance is not directly related to high or low pixel values.

Instead of choosing between methods of estimating the quantization step size, we prefer to combine the strength of some methods. First for each pixel the average value of the surrounding pixels is determined, then this average value is used as input to either a linear function, or a step function.

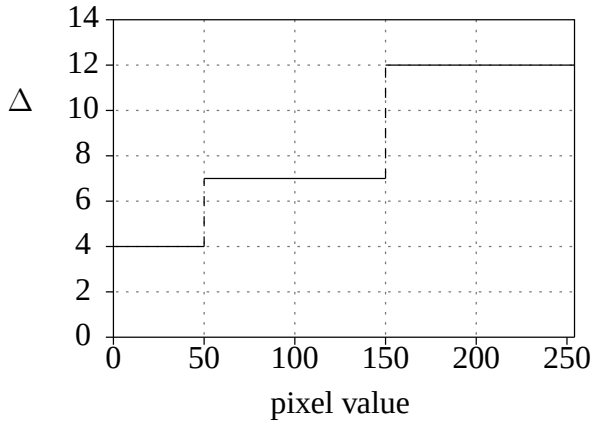


Figure 6.1: An example of a staircase function used to determine the adaptive quantization step size for a pixel.

So, we use two possibilities for determining the quantization step size:

$$\Delta(x) = \frac{\gamma}{L} \sum_{i=1}^{L} x_i, \tag{6.9}$$

and

$$\Delta(x) = f\left(\frac{1}{L} \sum_{i=1}^{L} x_i \right), \tag{6.10}$$

with $f(\cdot)$ a staircase function, for example the staircase function of Figure 6.1. This staircase function is arbitrarily chosen. According to some experiments, it outperforms some other staircase functions we have looked at, but it is not claimed that this is the best staircase function. Methods that include the usage of a median, the variance or other higher moments are not considered in this report.

In order to illustrate the use of an adaptive quantization step size we give an example.

Example 6.1. *Let $\Delta(x)$ be given by (6.9), with $L = 9$ and $\gamma = 0.05$. Let part of the host signal x be given by*

$$x = \begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 185 & 188 & 187 & 190 & 191 & \cdots \\ \cdots & 183 & 170 & 170 & 172 & 180 & \cdots \\ \cdots & 180 & 165 & 181 & 200 & 205 & \cdots \\ \cdots & 175 & 179 & 185 & 198 & 206 & \cdots \\ \cdots & 150 & 190 & 187 & 194 & 201 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

Suppose we want to determine Δ for the pixel in the middle, with value 181. The pixels we have to take into account are the 9 pixels that are in the 3×3 square with center $x = 181$. This gives

$$\Delta = \frac{0.05}{9} (170 + 170 + 172 + 165 + 181 + 200 + 179 + 185 + 198) = 9.$$

If Δ was to be determined with (6.10) and staircase function given by (6.8), then we would have

$$\Delta = f\left(\frac{1}{9}\left(170+170+172+165+181+200+179+185+198\right)\right) = f\left(180\right) = 12.$$

6.3 Experimental results for an adaptive quantization step size

The robustness properties for both methods are determined by experiment and compared in this section. In Section 7.3 we will take a more theoretical look at this robustness.

6.3.1 Experiment description

We perform experiments in order to compare three adaptive quantization methods. We also compare the best of the three methods with fixed quantization. We want to evaluate the robustness of fixed and adaptive quantization against brightness scaling and AWGN and JPEG attacks.

Adaptive quantization is done by taking an average value over a square of size $F \times F = L$, and use that as an input to either one of the two staircase functions or a linear function. So,we consider the following three adaptive quantization methods:

Method 1 $\Delta(x) = f\left(\frac{1}{L}\sum_{i=1}^{L} x_i\right)$, with

$$\Delta(x) = \begin{cases} 3\,\gamma & \text{if } 0 \le x \le 50 \\ 5\,\gamma & \text{if } 50 < x \le 100 \\ 8\,\gamma & \text{if } 100 < x \le 150 \\ 10\,\gamma & \text{if } 150 < x \le 255 \end{cases}. \tag{6.11}$$

Method 2 $\Delta(x) = f\left(\frac{1}{L}\sum_{i=1}^{L} x_i\right)$, with $f(\cdot)$ given by (6.8).

Method 3 $\Delta(x) = \frac{\gamma}{L}\sum_{i=1}^{L} x_i$.

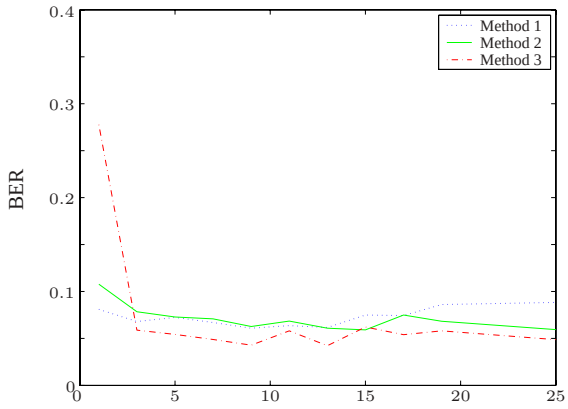
We also want to compare the robustness against brightness scaling of embedding with fixed quantization step size and embedding with adaptive quantization step size. Therefore the four images of Appendix B are embedded after which scaling of the luminance component is performed on the image. Then the altered image is used as an input to the detector. For this experiment, we use the best method for adaptive quantization, which will appear to be method 3, see the next subsection.

Also compared are the robustness of adaptive and fixed quantization against AWGN and JPEG attacks.

6.3.2 Results and discussion

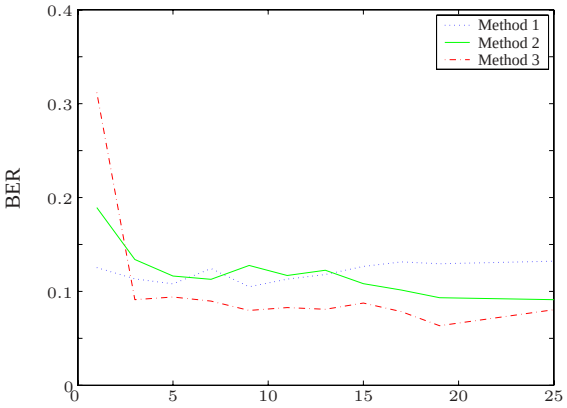
The results of these experiments are displayed in Figure 6.2, 6.3 and 6.4.

In Figure 6.2 the BER is plotted against the square root of the number of pixels in a square around a single pixel. We see that the BER is always lower for Method 3. For higher AWGN variances



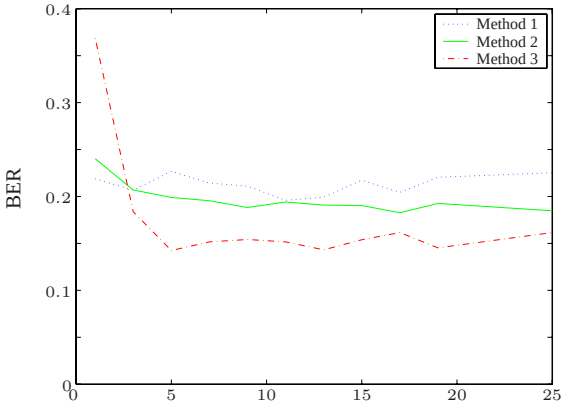
$F = \sqrt{L}$

(a) $\sigma_n^2 = 400$



$F = \sqrt{L}$

(b) $\sigma_n^2 = 500$



$F = \sqrt{L}$

(c) $\sigma_n^2 = 700$

Figure 6.2: The BER as a function of $F = \sqrt{L}$, for different noise variances.

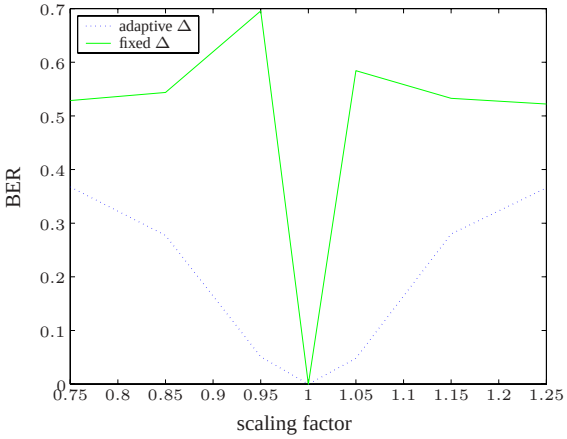
the difference with the two other methods increases. The size of the square $L = F^2$ has some influence on the BER, but only for low values of L. Beyond a certain point an increasing L does not decrease the BER anymore. We conclude that $L = 11$ is a good value to set at the embedder. We conclude that Method 3 is the better method of the three. This is somewhat remarkable, because the staircase functions only make errors in the neighborhood of the jumps and the linear function in the whole range. Apparently, the averaging in a square range of a pixel, averages also the errors to 0. Because of this averaging, only small errors are made by determining Δ at the detector. These small errors are cancelled out, because of the repetition code.

The robustness against brightness scaling for a fixed and an adaptive quantization step size can be seen from Figure 6.3. It can be seen that adaptive quantization is much more robust against brightness scaling than fixed quantization, as predicted in Subsection 6.1.1. It can also be seen that adaptive quantization is much more influenced by an increase in watermark strength compared to fixed quantization, see the Figures 6.3(a), 6.3(c), 6.3(e) and the Figures 6.3(b), 6.3(d), 6.3(f).

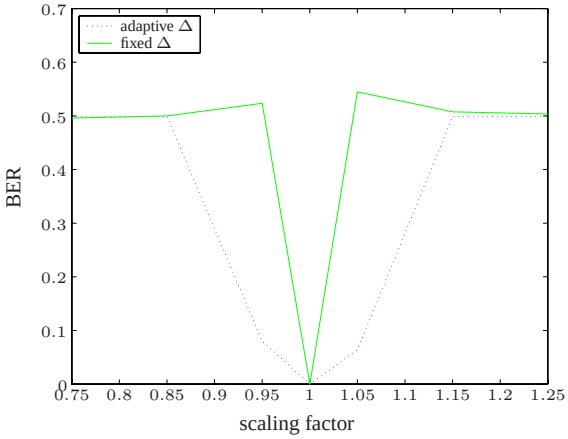
From Figure 6.4 we see that the robustness against AWGN is usually better for adaptive quantization, see Figure 6.4(a) and 6.4(a). For JPEG the robustness is slightly better for fixed quantization.

Summarizing, we have

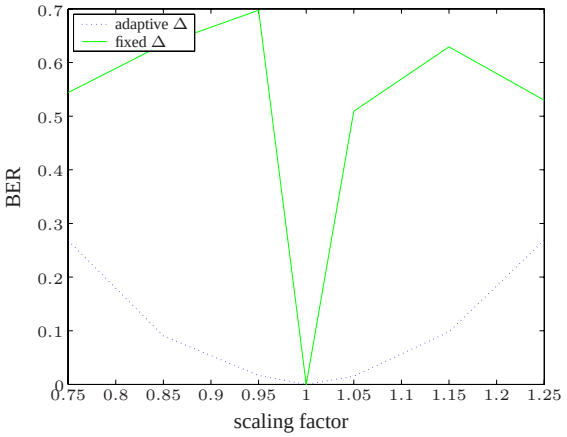
- For adaptive quantization a linear relationship between the quantization step size and the average of surrounding pixels is better than using staircase functions.
- Robustness against brightness scaling is much better for adaptive quantization compared with fixed quantization.
- Robustness against AWGN is slightly better for adaptive quantization compared with fixed quantization.
- Robustness against JPEG is slightly worse for adaptive quantization compared with fixed quantization.



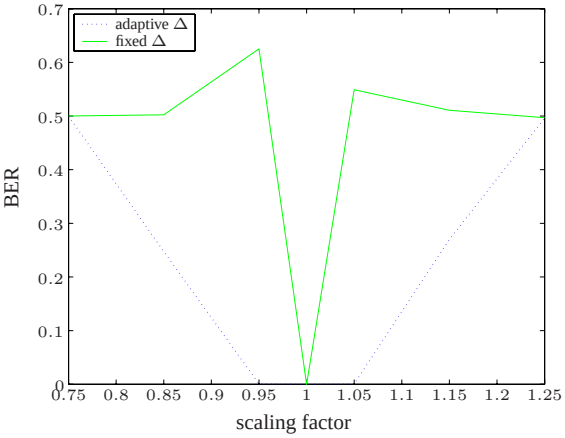
(a) Repetition coding with DWR = 30 and rate = 0.01.



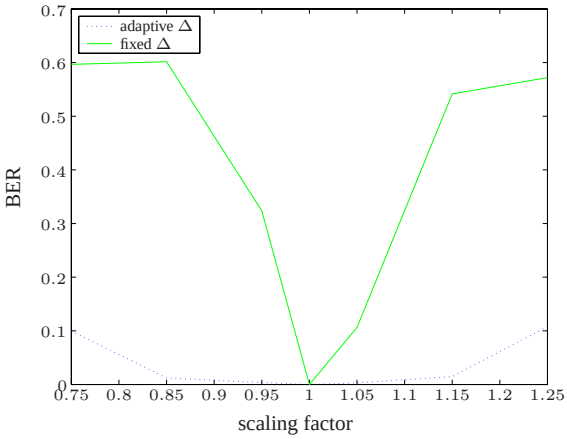
(b) Concatenated coding with DWR = 30 and rate = 0.01.



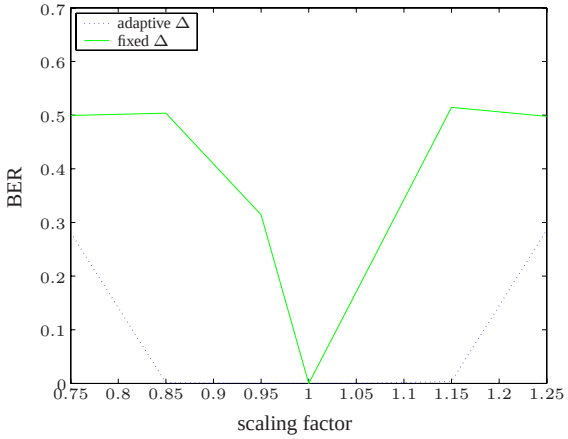
(c) Repetition coding with DWR = 25 and rate = 0.01.



(d) Concatenated coding with DWR = 25 and rate = 0.01.

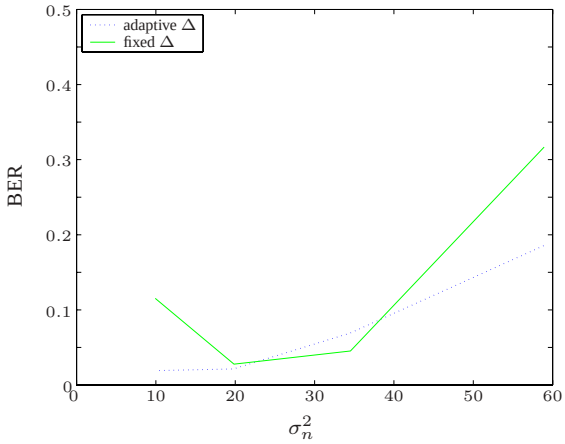


(e) Repetition coding with DWR = 20 and rate = 0.01.

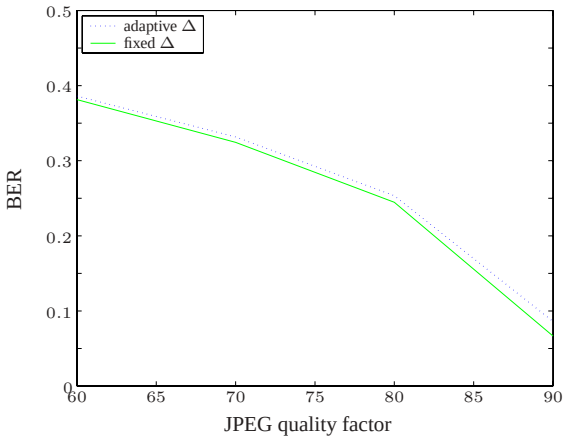


(f) Concatenated coding with DWR = 20 and rate = 0.01.

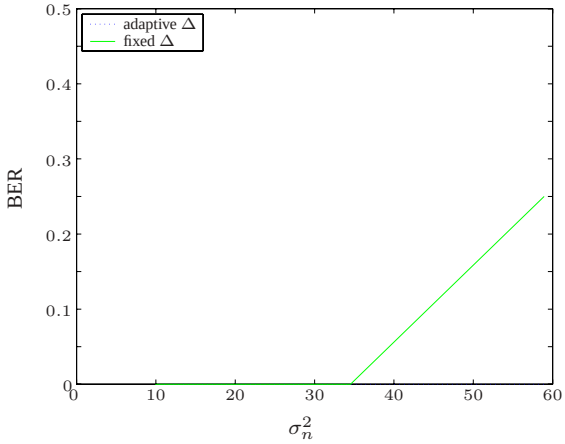
Figure 6.3: Adaptive quantization versus fixed quantization. The BER as a function of the brightness scaling factor.



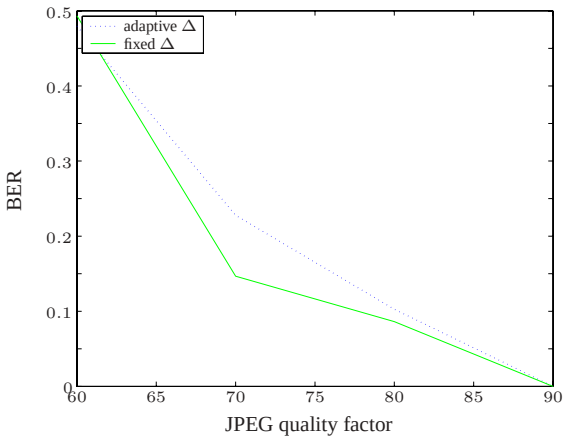
(a) Repetition coding with rate = 0.1 and AWGN attack.



(b) Repetition coding with rate = 0.1 and JPEG attack.



(c) Concatenated coding with rate = 0.01 and AWGN attack.



(d) Concatenated coding with rate = 0.01 and JPEG attack.

Figure 6.4: Adaptive quantization versus fixed quantization. The BER as a function of the AWGN variance or the JPEG quality factor.

Chapter 7

Performance analysis

As seen in Chapter 4, this report seeks an increased robustness and a performance evaluation of the developed model. Robustness is searched especially for lossy compression and noise attacks. In this chapter these attacks are modelled by Additive White Gaussian Noise (AWGN) and uniformly distributed noise attacks. This model is not very good, because JPEG noise is highly correlated with the host signal. Of course, analysis of the effect of noise attacks is made easier if the relevant signals are assumed to be Gaussian distributed.

The effect of the two improvements over Scalar Costa Scheme (SCS), the use of Error Correcting Codes (ECC) and an adaptive Δ, are to be measured. For ECC this is only done by experiment, see Section 5.4. The improvements of an adaptive quantization step size is measured experimentally in Section 6.3. From Subsection 6.1.2 it is known that using an adaptive Δ introduces an additional source of errors, due to the necessity to estimate Δ from the received signal. In this chapter also a theoretical model is developed in order to estimate this error probability.

In Section 7.1 a performance measure is given. The error probability for Gaussian and uniform noise attacks is determined in Section 7.2. In Section 7.3 the error probability due to the use of an adaptive quantization step size is modelled. Finally, in Section 7.4 the results are discussed and summarized.

7.1 A measure for the performance

The performance of a watermarking scheme is measured by the bit error probability. This is the probability that if a bit (0 or 1) is sent, a wrong bit is detected (1 or 0), i.e., the bit error probability is $P(m_i \neq \hat{m}_i)$. Another performance measure is the message error probability $P(m \neq \hat{m})$. These two performance measures are related by $P(m \neq \hat{m}) = 1 - \Pi_{i=1}^{l} (1 - P(m_i \neq \hat{m}_i))$, if the m_i are mutual independent.

A watermarking scheme can also be evaluated by looking at the capacity or the rate-distortion function. The capacity of a watermarking channel and the error probability of that channel are related. This can be understood intuitively as follows; If the capacity of a channel increases, then it is possible to embed more bits, or to use this extra space to add redundancy. This additional redundancy decreases the bit error probability.

The bit error probability is used in order to determine the performance of a watermarking scheme.

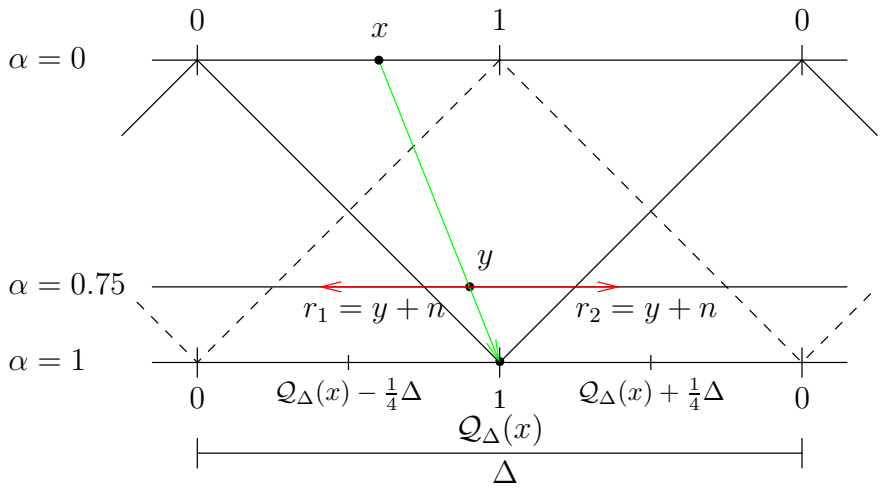


Figure 7.1: The embedding process for embedding 1. Everything in the range $[\mathcal{Q}_\Delta(x) - \frac{1}{2}\Delta,\ \mathcal{Q}_\Delta(x) + \frac{1}{2}\Delta]$ *is projected on* $\mathcal{Q}_\Delta(x)$ *for* $\alpha = 1$ *and on* $[\mathcal{Q}_\Delta(x) - \frac{1}{2}(1-\alpha)\Delta,\ \mathcal{Q}_\Delta(x) + \frac{1}{2}(1-\alpha)\Delta]$ *for* $\alpha \in]0,1[$*. If, after a noise attack,* $r = y + n$ *is still in the correct quantization bin, i.e.,* $r \in [\mathcal{Q}_\Delta(x) - \frac{1}{4}\Delta,\ \mathcal{Q}_\Delta(x) + \frac{1}{4}\Delta]$*, the received value* r *is still in the detection area, and no error is made. For the noise illustrated by the left red arrow (*r_1*) an error will be made, for the right red arrow (*r_2*) a correct detection is made.*

7.2 The bit error probability due to a noise attack

In this section the bit error probability for the SCS is determined. A fixed quantization step size is used and the noise is assumed to be Gaussian or uniformly distributed. Error correction codes can bring the bit error probability down, but for the sake of simple analysis they are not considered.

If the SCS is used, the embedding formula is Equation (3.17). Consider a pixel x. Without distortion compensation ($\alpha = 1$) embedding is done by quantizing x to the nearest odd or even level (1 or 0, respectively). With distortion compensation ($\alpha \in]0,1[$), the pixel value x is altered to a value y somewhere between the original value x and the quantized value $\mathcal{Q}_\Delta(x)$. See Figure 7.1 for illustration. A correct detection is made whenever r is in the correct quantization bin, corresponding to 0 or 1.

Assume that the image data x is uniformly distributed over the range of a quantization bin. According to [17] this is a reasonable assumption, because in most watermarking applications, the host-data power is much stronger than the watermark power ($\sigma_X^2 \gg \sigma_W^2$). If the dither is chosen appropriately, x can be made to be uniformly distributed over a quantization bin [20, 24, 31]. The added noise is assumed to be either normally distributed with zero mean and variance σ_N^2 or uniformly distributed on $[-c, c]$, so

$$N^U \sim U[-c,c], \quad p_{N^U}(n) = \begin{cases} \frac{1}{2c} & \text{if } n \in [-c,c] \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

$$N^G \sim N(0,\sigma_n^2), \quad p_{N^G}(n) = \frac{1}{\sqrt{2\pi}\sigma_n}\, e^{\frac{n^2}{2\sigma_n^2}} \quad ,\forall\, n \in \mathbb{R}. \tag{7.2}$$

The used embedding formula is

$$y_{\mathrm{SCS}} = x + \alpha \left(\left\lfloor \frac{x + d - \frac{1}{2}b\Delta}{\Delta} \right\rceil \Delta - x - d + \frac{b\Delta}{2} \right). \tag{7.3}$$

The bit error is independent of the value of the embedded bit. Without loss of generality, we assume $b = 0$. The dither is considered as something that gives the host signal x some desirable properties, so without loss of generality assume that x has the desired statistics and $d = 0$. Now, the embedding formula

$$y = x + \alpha \left(\mathcal{Q}_\Delta(x) - x \right) \tag{7.4}$$

is considered, where $\mathcal{Q}_\Delta(x) = \left\lfloor \frac{x}{\Delta} \right\rceil \Delta$ is the quantization operator. For the Gaussian case we will need the law of total probability, which reads, for two continuous stochastic variables V and N,

$$p_V(v) = \int_{-\infty}^{\infty} p_{V|N}(v|n) p_N(n) \, \mathrm{d}n. \tag{7.5}$$

So

$$P_V(V \in A) = \int_{-\infty}^{\infty} P_{V|N}(v \in A | N = n) p_N(n) \, \mathrm{d}n. \tag{7.6}$$

7.2.1 The derivation of the bit error probability

The detector makes an error if the received value $r = y + n$ is outside of one of the correct quantization bins. For the bit error probability is does not matter whetter 0 or 1 is embedded. Assuming 0 was embedded, then these correct quantization bins are the sets

$$\left[\mathcal{Q}_\Delta(x) + (4k-1)\frac{1}{4}\Delta, \mathcal{Q}_\Delta(x) + (4k+1)\frac{1}{4}\Delta \right], k \in \mathbb{Z}, \tag{7.7}$$

see Figure 7.2. So, no error is made if $\mathcal{Q}_\Delta(x) + (4k-1)\frac{1}{4}\Delta \leq y + n \leq \mathcal{Q}_\Delta(x) + (4k+1)\frac{1}{4}\Delta$, for some $k \in \mathbb{Z}$.

Therefore the bit error probability is given by:

$$\begin{aligned} P_{\mathrm{BE}} &= 1 - P(R \in \text{the correct quantization bin}) \\ &= 1 - \sum_{k=-\infty}^{\infty} P\left(\mathcal{Q}_\Delta(X) + (4k-1)\frac{1}{4}\Delta \leq Y + N \leq \mathcal{Q}_\Delta(X) + (4k+1)\frac{1}{4}\Delta \right). \end{aligned} \tag{7.8}$$

Theorem 7.1. *The bit error probability P_{BE} (7.8) can be rewritten as*

$$P_{\mathrm{BE}} = 1 - \sum_{k=-\infty}^{\infty} P\left((4k-1)\frac{1}{4}\Delta \leq V + N \leq (4k+1)\frac{1}{4}\Delta \right) \tag{7.9}$$

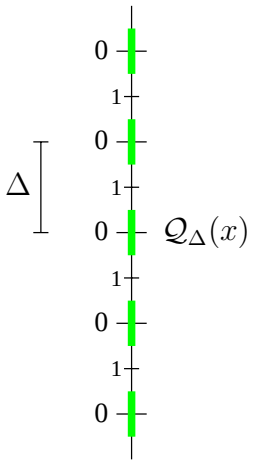


Figure 7.2: The quantization bins for which no error is made, are depicted with the green bars.

Proof: The probability in the sum of Equation (7.8) can be rewritten as

$$P\left(\mathcal{Q}_\Delta(X) + (4k-1)\frac{1}{4}\Delta \leq Y + N \leq \mathcal{Q}_\Delta(X) + (4k+1)\frac{1}{4}\Delta\right) \tag{7.10}$$

$$= P\left((4k-1)\frac{1}{4}\Delta \leq Y - \mathcal{Q}_\Delta(X) + N \leq (4k+1)\frac{1}{4}\Delta\right) \tag{7.11}$$

$$= P\left((4k-1)\frac{1}{4}\Delta \leq (1-\alpha)\left(X - \mathcal{Q}_\Delta(X)\right) + N \leq (4k+1)\frac{1}{4}\Delta\right) \tag{7.12}$$

$$= P\left((4k-1)\frac{1}{4}\Delta \leq V + N \leq (4k+1)\frac{1}{4}\Delta\right), \tag{7.13}$$

where in the second line $\mathcal{Q}_\Delta(X)$ is subtracted, in the third line the embedding formula (7.4) is substituted for Y, and where we defined $V \triangleq (1-\alpha)(X - \mathcal{Q}_\Delta(X))$ and substituted V in the fourth line. ■

V is interpreted as the distribution of the host signal over a quantization bin, except for a scaling with α. By assumption, $X - \mathcal{Q}_\Delta(X)$ is uniformly distributed over the range of a quantization bin. This means that $V \sim U[-\frac{1}{2}(1-\alpha)\Delta, \frac{1}{2}(1-\alpha)\Delta]$, so

$$p_V(v) = \begin{cases} \frac{1}{(1-\alpha)\Delta} & \text{if } v \in [-\frac{1}{2}(1-\alpha)\Delta, \frac{1}{2}(1-\alpha)\Delta] \\ 0 & \text{otherwise} \end{cases}. \tag{7.14}$$

This infinite sum in Equation (7.9) is hard to compute and therefore we define the truncated sum

$$P_{\mathrm{BE}}^K = 1 - \sum_{k=-K}^{K} P\left((4k-1)\frac{1}{4}\Delta \leq V + N \leq (4k+1)\frac{1}{4}\Delta\right) \tag{7.15}$$

to be the K^{th} order approximation to the bit error probability. Note that for all K, $P_{\mathrm{BE}}^K \geq P_{\mathrm{BE}}^{K+1} \geq P_{\mathrm{BE}}$.

The bit error probability of Equation (7.9) can be approximated by P_{BE}^K. Depending on the values of the noise variance σ_n^2 and the quantization step size Δ this approximation is already good for

$K = 1$, because for moderate noise the probability that r will come in another quantization bin, is very small. So, the 0^{th} order approximation for the bit error probability P_{BE}, as given by

$$P^0_{\text{BE}} = 1 - P\left(-\frac{1}{4}\Delta \leq V + N \leq \frac{1}{4}\Delta\right) \tag{7.16}$$

$$= P\left(|V + N| > \frac{1}{4}\Delta\right), \tag{7.17}$$

may already give relevant information. A better approximation is given by

$$\begin{aligned} P^1_{\text{BE}} = 1 &- P\left(-\frac{1}{4}\Delta \leq V + N \leq \frac{1}{4}\Delta\right) - P\left(-\frac{5}{4}\Delta \leq V + N \leq -\frac{3}{4}\Delta\right) \\ &- P\left(\frac{3}{4}\Delta \leq V + N \leq \frac{5}{4}\Delta\right) \end{aligned} \tag{7.18}$$

$$= P^0_{\text{BE}} - P\left(-\frac{5}{4}\Delta \leq V + N \leq -\frac{3}{4}\Delta\right) - P\left(\frac{3}{4}\Delta \leq V + N \leq \frac{5}{4}\Delta\right). \tag{7.19}$$

The two cases of AWGN and uniformly distributed noise, i.e., $N^G \sim N(0, \sigma_N^2)$ and $N^U \sim U[-c, c]$, are considered in the following two sections. There we will derive formulas for the bit error probability.

7.2.2 The bit error probability for Gaussian noise

In this subsection the bit error probability for the Gaussian case is approximated with P^0_{BE} and P^1_{BE}. In the following theorems the formulas for P^0_{BE} and P^1_{BE} are derived.

The following facts about the error function erf and the complementary error function erfc are used

$$\text{erf}(z) = 1 - \text{erfc}(z), \tag{7.20}$$

$$\text{erf}(-z) = -\text{erf}(z), \tag{7.21}$$

$$\int \text{erf}(z)\ \mathrm{d}z = z\,\text{erf}(z) + \frac{1}{\sqrt{\pi}}\, e^{-z^2}. \tag{7.22}$$

Theorem 7.2. *For* $\alpha = 1$,

$$P^0_{\text{BE}} = 1 - \text{erf}\left(\frac{\frac{1}{4}\Delta}{\sqrt{2}\,\sigma_n}\right). \tag{7.23}$$

Proof: For the case of no distortion compensation, $\alpha = 1$, the host signal is quantized to one point, so $v = 0$. Using Equation (7.16) P^0_{BE} is calculated as

$$P^0_{\text{BE}} = 1 - \int_{-\frac{\Delta}{4}}^{\frac{\Delta}{4}} \frac{1}{\sqrt{2\pi}\sigma_n}\, e^{\frac{n^2}{2\sigma_n^2}}\, \mathrm{d}n,$$

This is equal to (7.23), where (7.20) and (7.21) are used. ■

Theorem 7.3. *For* $0 < \alpha < 1$ *we have*

$$P_{\mathrm{BE}}^{0} = \frac{\alpha - \frac{1}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{1}{2})\Delta}{\sqrt{2}\sigma_n}\right) - \frac{\alpha - \frac{3}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{3}{2})\Delta}{\sqrt{2}\sigma_n}\right) + \frac{2\sigma_n}{\sqrt{2\pi}(1-\alpha)\Delta}\left(e^{-\frac{\frac{1}{4}(\alpha-\frac{1}{2})^2\Delta^2}{2\sigma_n^2}} - e^{-\frac{\frac{1}{4}(\alpha-\frac{3}{2})^2\Delta^2}{2\sigma_n^2}}\right) + 1. \tag{7.24}$$

Proof: Using distortion compensation, $\alpha \in]0,1[$, v is distributed with a probability density function as in Equation (7.14). The approximation of the conditional bit error probability given v is

$$P(\mathrm{BE} \mid V = v) = P(|V + N^G| > \frac{\Delta}{4} \mid V = v) \tag{7.25}$$

$$= P(N^G > \frac{\Delta}{4} - V \mid V = v) + P(N^G < -\frac{\Delta}{4} - V \mid V = v) \tag{7.26}$$

$$= \int_{\frac{\Delta}{4}-v}^{\infty} p_{N^G}(n)\,\mathrm{d}n + \int_{-\infty}^{-\frac{\Delta}{4}-v} p_{N^G}(n)\,\mathrm{d}n. \tag{7.27}$$

Then we apply the law of total probability (see (7.6)), to obtain

$$\begin{aligned} P_{\mathrm{BE}}^{0} &= \int_{-\frac{1}{2}(1-\alpha)\Delta}^{\frac{1}{2}(1-\alpha)\Delta} P(\mathrm{BE} \mid V = v) p_V(v)\,\mathrm{d}v \\ &= \int_{-\frac{1}{2}(1-\alpha)\Delta}^{\frac{1}{2}(1-\alpha)\Delta} \left[\int_{\frac{\Delta}{4}-v}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{\frac{-n^2}{2\sigma_n^2}}\,\mathrm{d}n + \int_{-\infty}^{-\frac{\Delta}{4}-v} \frac{1}{\sqrt{2\pi}\sigma_n} e^{\frac{-n^2}{2\sigma_n^2}}\,\mathrm{d}n \right] \frac{1}{(1-\alpha)\Delta}\,\mathrm{d}v \\ &= \int_{-\frac{1}{2}(1-\alpha)\Delta}^{\frac{1}{2}(1-\alpha)\Delta} \left[\frac{1}{2}\operatorname{erfc}\left(\frac{\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) + \left(1 - \frac{1}{2}\operatorname{erfc}\left(\frac{-\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right)\right) \right] \frac{1}{(1-\alpha)\Delta}\,\mathrm{d}v, \end{aligned} \tag{7.28}$$

where we used (7.2) and (7.14). In order to rewrite Equation (7.28), the Equations (7.20), (7.21) and (7.22) are used. Now P_{BE}^{0} can be rewritten as

$$P_{\mathrm{BE}}^{0} = c_0 \int_a^b 1\;\mathrm{d}v - \frac{1}{2}c_0 \int_a^b \operatorname{erf}\left(\frac{\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right)\mathrm{d}v + \frac{1}{2}c_0 \int_a^b \operatorname{erf}\left(\frac{-\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right)\mathrm{d}v, \tag{7.29}$$

with $c_0 = \frac{1}{(1-\alpha)\Delta}$, $a = -\frac{1}{2}(1-\alpha)\Delta$ and $b = \frac{1}{2}(1-\alpha)\Delta$ and where we used (7.20). The first

integral is equal to $\frac{1}{c_0}$. The second is equal to

$$\int_a^b \operatorname{erf}\left(\frac{\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v = -\sqrt{2}\sigma_n \int_{\tilde{a}}^{\tilde{b}} \operatorname{erf}(u)\,\mathrm{d}u = -\sqrt{2}\sigma_n \left[u\,\operatorname{erf}(u) + \frac{1}{\sqrt{\pi}}\, e^{-u^2}\right]_{\tilde{a}}^{u=\tilde{b}} \tag{7.30}$$

$$\begin{aligned} &= -\frac{1}{2}(\alpha - \tfrac{1}{2})\Delta\, \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{1}{2})\Delta}{\sqrt{2}\sigma_n}\right) - \frac{\sqrt{2}\sigma_n}{\sqrt{\pi}}\, e^{-\frac{\frac{1}{4}(\alpha-\frac{1}{2})^2\Delta^2}{2\sigma_n^2}} \\ &\quad + \frac{1}{2}(\alpha - \tfrac{3}{2})\Delta\, \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{3}{2})\Delta}{\sqrt{2}\sigma_n}\right) + \frac{\sqrt{2}\sigma_n}{\sqrt{\pi}}\, e^{-\frac{\frac{1}{4}(\alpha-\frac{3}{2})^2\Delta^2}{2\sigma_n^2}}, \end{aligned} \tag{7.31}$$

where in the first line the integration variable v is changed into u, with $u = \frac{\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}$, $\tilde{a} = \frac{\frac{1}{4}\Delta - a}{\sqrt{2}\sigma_n}$, $\tilde{b} = \frac{\frac{1}{4}\Delta - b}{\sqrt{2}\sigma_n}$, and where also in the first line Equation (7.22) is applied. In the second line the expression is evaluated for $\tilde{a}$ and $\tilde{b}$.

The same procedure is applied on the third integral. Taking the three integrals together again and using Equation (7.21) gives the desired result. ■

A better approximation can be calculated with P_{BE}^1, see Equation (7.19).

Theorem 7.4. *For* $\alpha = 1$

$$P_{\mathrm{BE}}^1 = 1 - \operatorname{erf}\left(\frac{\frac{1}{4}\Delta}{\sqrt{2}\,\sigma_n}\right) + \operatorname{erf}\left(\frac{\frac{3}{4}\Delta}{\sqrt{2}\,\sigma_n}\right) - \operatorname{erf}\left(\frac{\frac{5}{4}\Delta}{\sqrt{2}\,\sigma_n}\right). \tag{7.32}$$

Proof: Use (7.19), $v = 0$ and apply the same reasoning as in Theorem 7.2. ■

Theorem 7.5. *For* $0 < \alpha < 1$ *we have*

$$\begin{aligned} P_{\mathrm{BE}}^1 = {} & \frac{\alpha - \frac{1}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{1}{2})\Delta}{\sqrt{2}\sigma_n}\right) - \frac{\alpha - \frac{3}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{3}{2})\Delta}{\sqrt{2}\sigma_n}\right) \\ & + \frac{\alpha - \frac{5}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{5}{2})\Delta}{\sqrt{2}\sigma_n}\right) - \frac{\alpha - \frac{7}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha - \frac{7}{2})\Delta}{\sqrt{2}\sigma_n}\right) \\ & - \frac{\alpha + \frac{1}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha + \frac{1}{2})\Delta}{\sqrt{2}\sigma_n}\right) + \frac{\alpha + \frac{3}{2}}{2(1-\alpha)} \operatorname{erf}\left(\frac{\frac{1}{2}(\alpha + \frac{3}{2})\Delta}{\sqrt{2}\sigma_n}\right) \\ & + \frac{2\sigma_n}{\sqrt{2\pi}(1-\alpha)\Delta}\left(- e^{-\frac{\frac{1}{4}(\alpha-\frac{7}{2})^2\Delta^2}{2\sigma_n^2}} + e^{-\frac{\frac{1}{4}(\alpha-\frac{5}{2})^2\Delta^2}{2\sigma_n^2}} - e^{-\frac{\frac{1}{4}(\alpha-\frac{3}{2})^2\Delta^2}{2\sigma_n^2}} \right. \\ & \left. + e^{-\frac{\frac{1}{4}(\alpha-\frac{1}{2})^2\Delta^2}{2\sigma_n^2}} - e^{-\frac{\frac{1}{4}(\alpha+\frac{1}{2})^2\Delta^2}{2\sigma_n^2}} + e^{-\frac{\frac{1}{4}(\alpha+\frac{3}{2})^2\Delta^2}{2\sigma_n^2}} \right) + 1. \end{aligned} \tag{7.33}$$

Proof: With (7.18) the conditional bit error probability given $V = v$ can be made. Applying the

law of total probability and after some rewriting, as in the case P^0_{BE}, we get

$$\begin{aligned} P^1_{\mathrm{BE}} = c_0 \int_a^b 1 \ \mathrm{d}v + \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{-\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v - \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{\frac{1}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v \\ + \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{-\frac{5}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v - \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{-\frac{3}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v \\ + \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{\frac{3}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v - \frac{1}{2}c_0 \int_a^b \mathrm{erf}\left(\frac{\frac{5}{4}\Delta - v}{\sqrt{2}\sigma_n}\right) \mathrm{d}v, \end{aligned} \tag{7.34}$$

with c_0, a and b as before.

Following the same procedure as with the calculation of P^0_{BE} for $0 < \alpha < 1$, this gives the desired result. ■

The 0^{th} order approximation for the case of Gaussian noise (7.24) is drawn as a function of α for different WNR's in Figure 8.1.

7.2.3 The bit error probability for uniform noise

Next, we will compute approximations to the bit error probability (7.9) for the case of uniformly distributed noise N and "host" V. In order to establish the bit error probability, first the distribution of the sum $V + N$ of the two uniform distributed variables V and N is calculated. We have that $N \sim U[-c, c]$ (7.1) and $V \sim U[-\frac{1}{2}(1-\alpha)\Delta, \frac{1}{2}(1-\alpha)\Delta]$ (7.14). A distinction is made between two cases:

Weak noise case: The noise variance is smaller than the variance of V: $\sigma_N^2 \leq \sigma_V^2$ or $c \leq \frac{1}{2}(1-\alpha)\Delta$;

Strong noise case: The variance of V is smaller than the noise variance: $\sigma_N^2 > \sigma_V^2$ or $c > \frac{1}{2}(1-\alpha)\Delta$.

In this subsection we will treat the weak noise case. The strong noise case can be treated completely analogously, so only the results are stated. See Figure 7.3 for the probability density functions of V, N and the sum $Z = V + N$ for the weak noise case. In the following the probability density function of Z is derived.

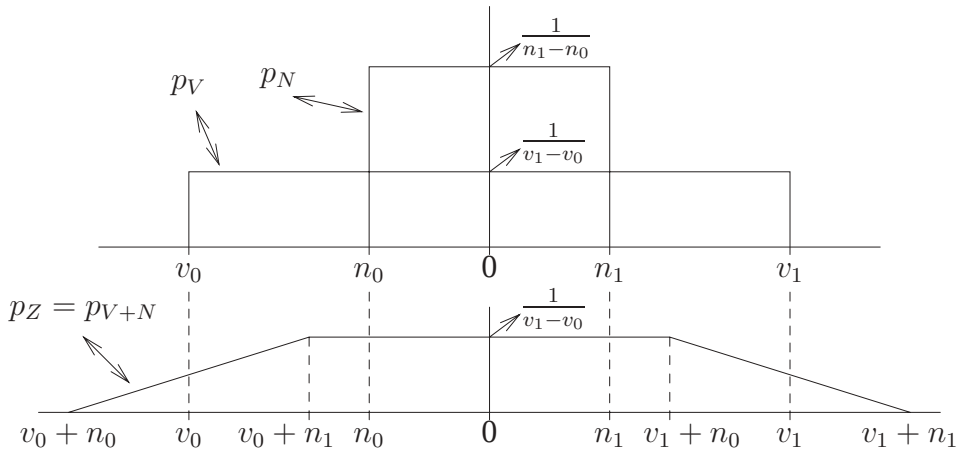


Figure 7.3: The probability density functions of N, V and the sum Z for the weak noise case: $\sigma_N^2 \leq \sigma_V^2$.

Theorem 7.6. *The probability density function of Z for the weak noise case is given by*

$$p_Z(z) = \begin{cases} \frac{z-(v_0+n_0)}{(v_1-v_0)(n_1-n_0)} & \text{if } z > v_0+n_0 \text{ and } z \leq v_0+n_1 \\ \frac{1}{(v_1-v_0)} & \text{if } z > v_0+n_1 \text{ and } z \leq v_1+n_0 \\ \frac{(v_1+n_1)-z}{(v_1-v_0)(n_1-n_0)} & \text{if } z > v_1+n_0 \text{ and } z \leq v_1+n_1 \\ 0 & \text{otherwise} \end{cases} \tag{7.35}$$

$$= \begin{cases} \frac{z+\frac{1}{2}(1-\alpha)\Delta+c}{2(1-\alpha)\Delta c} & \text{if } -\frac{1}{2}(1-\alpha)\Delta - c \leq z \leq -\frac{1}{2}(1-\alpha)\Delta + c \\ \frac{1}{(1-\alpha)\Delta} & \text{if } -\frac{1}{2}(1-\alpha)\Delta + c \leq z \leq \frac{1}{2}(1-\alpha)\Delta - c \\ \frac{\frac{1}{2}(1-\alpha)\Delta+c-z}{2(1-\alpha)\Delta c} & \text{if } \frac{1}{2}(1-\alpha)\Delta - c \leq z \leq \frac{1}{2}(1-\alpha)\Delta + c \\ 0 & \text{otherwise} \end{cases} \tag{7.36}$$

Proof: Consider the general case that $V \sim U[y_0, y_1]$ and $N \sim U[n_0, n_1]$. The weak noise case is divided again in three subcases: subcase A_1 is the case $z \leq v_0 + n_1$ (this is area 1 in Figure 7.4), subcase A_2 is the case $z > v_0 + n_1$ and $z \leq v_1 + n_0$ (this is area 2), and subcase A_3 is the case $z > v_1 + n_0$ (area 3).

In order to calculate the probability of $Z \leq z$ integration is done over area i for subcase A_i, $i = 1, 2, 3$. Subcase A_1 gives

$$P(Z \leq z) = \int_{v_0}^{z-n_0} \int_{n_0}^{z-v} \frac{1}{(v_1-v_0)} \frac{1}{(n_1-n_0)} \, \mathrm{d}n \, \mathrm{d}v = \int_{v_0}^{z-n_0} \frac{z-v-n_0}{(v_1-v_0)(n_1-n_0)} \, \mathrm{d}v \tag{7.37}$$

$$= \frac{1}{2} \frac{(z-n_0)^2}{(v_1-v_0)(n_1-n_0)} + \frac{\frac{1}{2}v_0^2 - v_0(z-n_0)}{(v_1-v_0)(n_1-n_0)}, \tag{7.38}$$

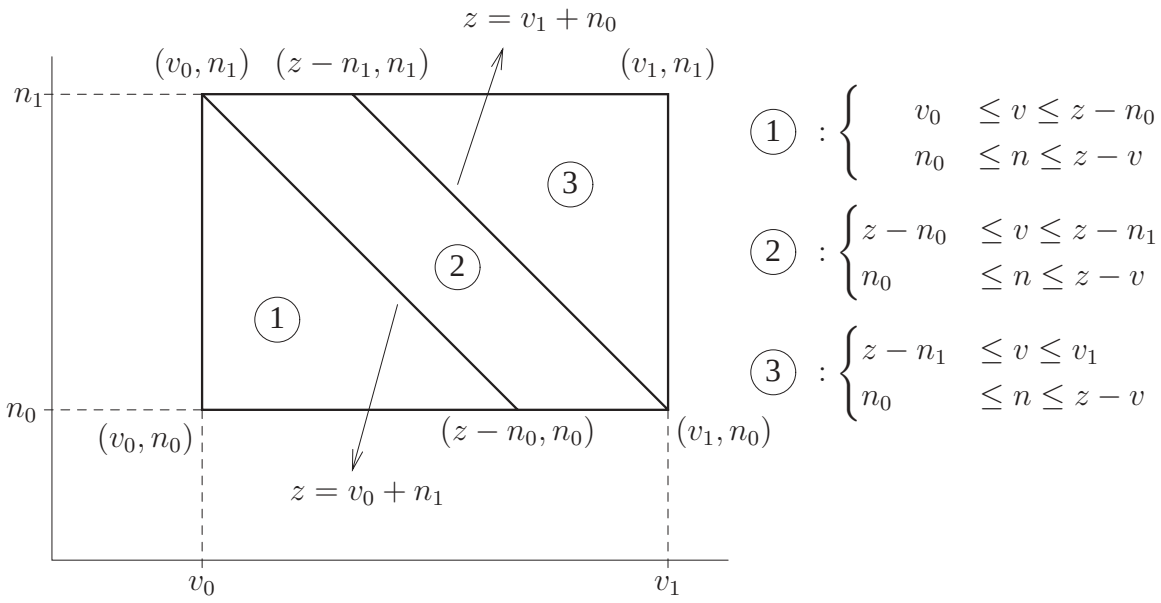


Figure 7.4: The integration areas of the three subcases of the weak noise case.

for subcase A_2

$$P(Z \le z) = \int_{v_0}^{z-n_1} \int_{n_0}^{n_1} \frac{1}{(v_1 - v_0)} \frac{1}{(n_1 - n_0)} \, \mathrm{d}n \, \mathrm{d}v + \int_{z-n_1}^{z-n_0} \int_{n_0}^{z-v} \frac{1}{(v_1 - v_0)} \frac{1}{(n_1 - n_0)} \, \mathrm{d}n \, \mathrm{d}v \tag{7.39}$$

$$= \frac{z - n_1 - v_0}{(v_1 - v_0)} + \int_{z-n_1}^{z-n_0} \frac{z - v - n_0}{(v_1 - v_0)(n_1 - n_0)} \, \mathrm{d}v \tag{7.40}$$

$$= \frac{\frac{1}{2}(n_1 - n_0)^2}{(v_1 - v_0)(n_1 - n_0)} + \frac{z - (n_1 + v_0)}{(v_1 - v_0)}, \tag{7.41}$$

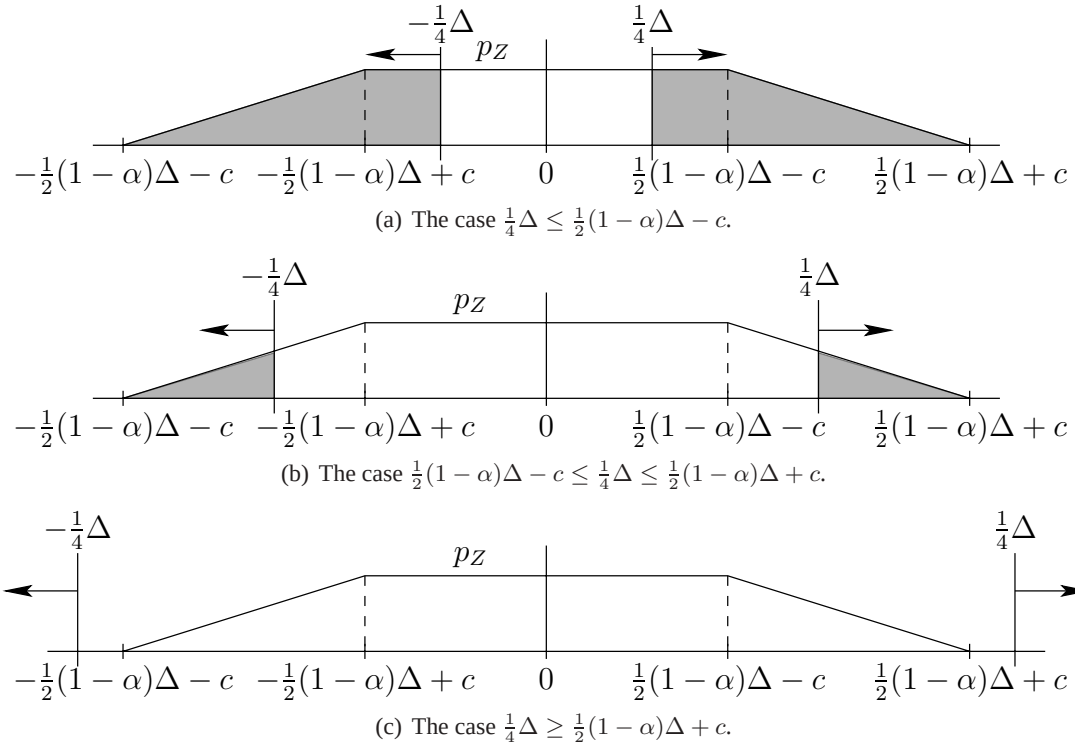
and for subcase A_3

$$P(Z \le z) = \int_{v_0}^{z-n_1} \int_{n_0}^{n_1} \frac{1}{(v_1 - v_0)} \frac{1}{(n_1 - n_0)} \, \mathrm{d}n \, \mathrm{d}v + \int_{z-n_1}^{v_1} \int_{n_0}^{z-v} \frac{1}{(v_1 - v_0)} \frac{1}{(n_1 - n_0)} \, \mathrm{d}n \, \mathrm{d}v \tag{7.42}$$

$$= \frac{z - n_1 - v_0}{(v_1 - v_0)} + \int_{z-n_1}^{v_1} \frac{z - v - n_0}{(v_1 - v_0)(n_1 - n_0)} \, \mathrm{d}v \tag{7.43}$$

$$= \frac{z - (n_1 + v_0)}{(v_1 - v_0)} + \frac{(z - n_0)(v_1 + n_1 - z) + \frac{1}{2}(z - n_1)^2 - \frac{1}{2}v_1^2}{(v_1 - v_0)(n_1 - n_0)}. \tag{7.44}$$

The probability density function of Z for the weak noise case is given by $p_Z(z) = \frac{\mathrm{d}}{\mathrm{d}z} P(Z \le z)$. For our case we have $v_0 = -\frac{1}{2}(1-\alpha)\Delta$, $v_1 = \frac{1}{2}(1-\alpha)\Delta$, $n_0 = -c$ and $n_1 = c$, which completes the proof. ■



(a) The case $\frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta - c$.

(b) The case $\frac{1}{2}(1-\alpha)\Delta - c \leq \frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta + c$.

(c) The case $\frac{1}{4}\Delta \geq \frac{1}{2}(1-\alpha)\Delta + c$.

Figure 7.5: The bit error probability for the weak noise case is the shaded area under $p_Z(z)$. There are three possible cases, depicted in Subfigure 7.5(a), 7.5(b) and 7.5(c).

For the strong noise case the pdf is given by

$$p_Z(z) = \begin{cases} \frac{z+\frac{1}{2}(1-\alpha)\Delta+c}{2(1-\alpha)\Delta c} & \text{if } -\frac{1}{2}(1-\alpha)\Delta - c \leq z \leq \frac{1}{2}(1-\alpha)\Delta - c \\ \frac{1}{2c} & \text{if } \frac{1}{2}(1-\alpha)\Delta - c \leq z \leq -\frac{1}{2}(1-\alpha)\Delta + c \\ \frac{\frac{1}{2}(1-\alpha)\Delta+c-z}{2(1-\alpha)\Delta c} & \text{if } -\frac{1}{2}(1-\alpha)\Delta + c \leq z \leq \frac{1}{2}(1-\alpha)\Delta + c \\ 0 & \text{otherwise} \end{cases} \tag{7.45}$$

The next step is to derive the 0^{th} order approximation of the bit error probability P_{BE}^0 (see Equation (7.16)) for uniform noise. For the weak noise case the probability density function is given by Equation (7.36).

Theorem 7.7. *For the weak noise case the 0^{th} order approximation of the bit error probability is given by*

$$P_{\text{BE}}^0 = \begin{cases} 1 - \frac{1}{2(1-\alpha)} & \text{if } 0 < \frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta - c \\ \frac{1}{2} - \frac{\alpha\Delta}{8c} + \frac{(c-\frac{1}{4}\Delta)^2}{2(1-\alpha)\Delta c} & \text{if } \frac{1}{2}(1-\alpha)\Delta - c \leq \frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta + c \\ 0 & \text{if } \frac{1}{4}\Delta \geq \frac{1}{2}(1-\alpha)\Delta + c \end{cases} \tag{7.46}$$

Proof: In order to calculate the error probability, the area under the graph of $p_Z(z)$ is calculated, with z ranging from $-\frac{1}{4}\Delta$ till $\frac{1}{4}\Delta$. There are three possibilities, see Figure 7.5: $\frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta - c$, $\frac{1}{2}(1-\alpha)\Delta - c \leq \frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta + c$ and $\frac{1}{4}\Delta \geq \frac{1}{2}(1-\alpha)\Delta + c$. The areas

under the pdf of Figure 7.5 are easy to calculate. For the last case $P^0_{\mathrm{BE}} = 0$. The first case gives $P^0_{\mathrm{BE}} = 1 - 2 \times \left(\frac{1}{4}\Delta \cdot \frac{1}{(1-\alpha)\Delta}\right)$ and the second case

$$P^0_{\mathrm{BE}} = 1 - 2 \times \left(\frac{1}{2}(1-\alpha)\Delta - c\right) \cdot \frac{1}{(1-\alpha)\Delta} - 2 \times \int_{\frac{1}{2}(1-\alpha)\Delta - c}^{\frac{1}{4}\Delta} \frac{\frac{1}{2}(1-\alpha)\Delta + c - z}{2(1-\alpha)\Delta c}\,\mathrm{d}z \tag{7.47}$$

$$= 1 - \left(1 - \frac{c}{\frac{1}{2}(1-\alpha)\Delta}\right) - \left[\frac{(1-\alpha)\Delta + 2c}{2(1-\alpha)\Delta c}z - \frac{1}{2(1-\alpha)\Delta c}z^2\right]_{\frac{1}{2}(1-\alpha)\Delta - c}^{z=\frac{1}{4}\Delta} \tag{7.48}$$

$$= \frac{1}{2} - \frac{\alpha\Delta}{8c} + \frac{(c - \frac{1}{4}\Delta)^2}{2(1-\alpha)\Delta c}. \tag{7.49}$$

The three integrals together give the desired result. ■

Without proof we state the result for the strong noise case ($c \geq \frac{1}{2}(1-\alpha)\Delta$).

Theorem 7.8. *For the strong noise case the 0^{th} order approximation of the bit error probability is given by*

$$P^0_{\mathrm{BE}} = \begin{cases} 1 - \frac{\Delta}{4c} & \text{if } 0 < \frac{1}{4}\Delta \leq -\frac{1}{2}(1-\alpha)\Delta + c \\ \frac{1}{2} - \frac{\alpha\Delta}{8c} + \frac{(c-\frac{1}{4}\Delta)^2}{2(1-\alpha)\Delta c} & \text{if } -\frac{1}{2}(1-\alpha)\Delta + c \leq \frac{1}{4}\Delta \leq \frac{1}{2}(1-\alpha)\Delta + c \\ 0 & \text{if } \frac{1}{4}\Delta \geq \frac{1}{2}(1-\alpha)\Delta + c \end{cases} \tag{7.50}$$

The 0^{th} order approximation for the case of uniform noise given by (7.46) and (7.50) is drawn as a function of α for different WNR's in Figure 8.1.

7.3 The bit error probability due to the adaptive quantization step size

At the embedder for each pixel a quantization step size Δ is determined. In order to make a detection it is necessary to estimate this quantization step size at the detector (this estimate is called $\hat{\Delta}$). Due to estimation errors, bit errors may be introduced in the detection. In this section we analyze the dependence of the bit error probability on the estimation performance.

In Subsection 7.3.1 a model to determine this bit error probability is derived. An approximation using Fourier series is given in Subsection 7.3.2 and the convergence of the Fourier series is examined in Subsection 7.3.3. The expected error probability due to the estimation of Δ at the detector is calculated in Subsection 7.3.4.

7.3.1 Modelling the bit error probability P_Δ

The bit error probability due to quantization step size estimation errors is to be modelled. Because only this error source is considered, other sources of errors are assumed to be absent. Therefore it is assumed that no noise is added and that embedding is done without distortion compensation.

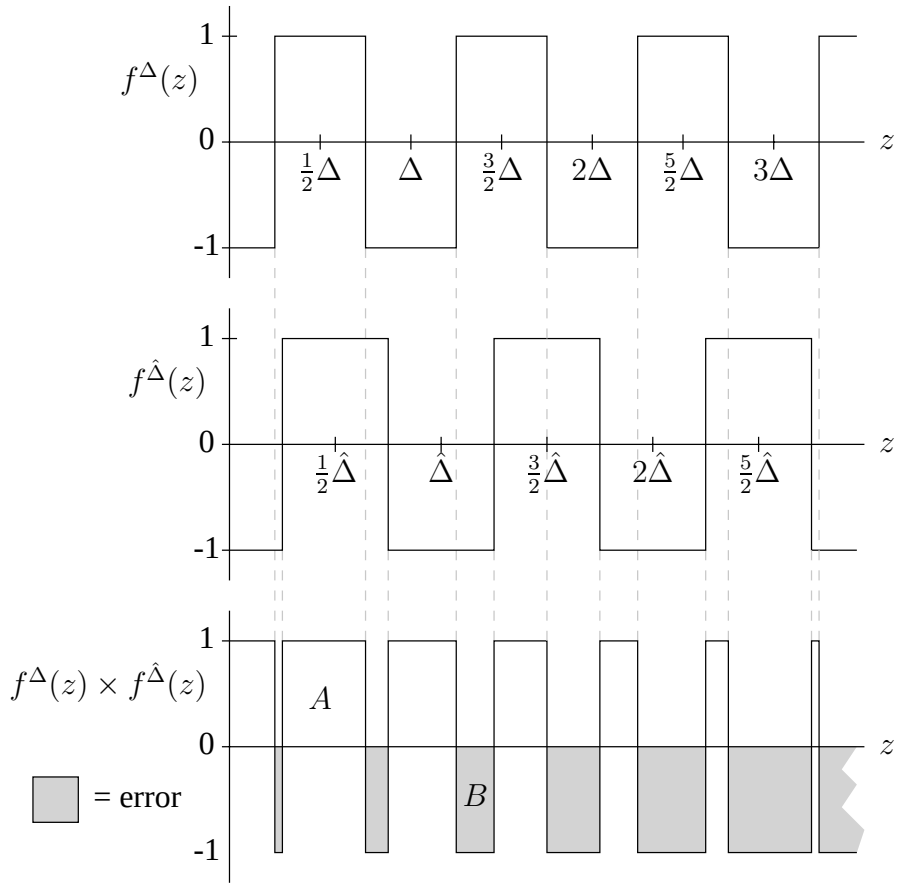


Figure 7.6: The detection error made due to the errors in the estimation of the quantization step size. In the upper and middle plots detection with Δ and $\hat{\Delta}$, respectively, is shown. The values of z for which an error is made is shown in the lower plot, which is a plot of the product of $f^{\Delta}(z)$ and $f^{\hat{\Delta}}(z)$.

For simplicity, we also assume that the dither equals zero. The bit error probability due to the estimation of the adaptive quantization step size at the detector is denoted by P_{Δ}.

Consider Figure 7.6. This shows for some quantization step size which bit is detected, depending on the pixel value z. The upper plot of Figure 7.6 shows which bit is detected if the correct quantization step size Δ is used, that is if at the detector the quantization step size is estimated correctly ($\hat{\Delta} = \Delta$). We call this function $f^{\Delta}(z)$, with

$$f^{\Delta}(z) = \text{sign}\left(-\cos\left(\frac{\pi}{\frac{1}{2}\Delta} z\right)\right) \tag{7.51}$$

where $\text{sign}(z) \triangleq \frac{z}{|z|}$ if $z \neq 0$ and $\text{sign}(z) = 0$ if $z = 0$. The middle plot of Figure 7.6 shows detection with an estimated $\hat{\Delta}$ which is slightly bigger than the quantization step size Δ used at the embedding side. This function is called $f^{\hat{\Delta}}(z)$. An error is made for all pixel values z for which $f^{\Delta}(z) \neq f^{\hat{\Delta}}(z)$, i.e., for which the product $f^{\Delta}(z) \times f^{\hat{\Delta}}(z)$ equals -1, see the lower plot of Figure 7.6.

For ease of computation, we scale the variable z, and all related variables (Δ and $\hat{\Delta}$), such that

the range of z becomes the interval $[0, 1]$. The probability that an error will be made is equal to the area between the graph of $f^{\Delta}(z) \times f^{\hat{\Delta}}(z)$ and the horizontal line $y = 0$. If we call the area corresponding to correct detection A (the white area in Figure 7.6) and the area corresponding to erroneous detection B (the shaded area in Figure 7.6), then we have:

$$A + B = 1, \tag{7.52}$$

$$\int_0^1 \left(f^{\Delta}(z) \times f^{\hat{\Delta}}(z) \right) p_z(z)\, \mathrm{d}z = A - B, \tag{7.53}$$

where $p_z(z)$ is the pdf of z. Then the bit error probability equals:

$$P_{\Delta} = B = \frac{1}{2} - \frac{1}{2} \int_0^1 \left(f^{\Delta}(z) \times f^{\hat{\Delta}}(z) \right) p_z(z)\, \mathrm{d}z \tag{7.54}$$

$$= \frac{1}{2} - \frac{1}{2} \int_0^1 \mathrm{sign}\left(\cos\left(\frac{\pi}{\Delta} z \right) \cos\left(\frac{\pi}{\hat{\Delta}} z \right) \right) p_z(z)\, \mathrm{d}z. \tag{7.55}$$

Plots for this bit error probability P_{Δ} are given in Appendix C. From these plot it is clear that the quantization step size should be estimated with large precision, because else P_{Δ} increases rapidly.

This integral is very hard to evaluate due to the sign-function. Therefore, we have to approximate the integral. We do this in the next subsection by approximating the functions $f^{\Delta}(z)$ and $f^{\hat{\Delta}}(z)$ by their Fourier series.

7.3.2 Fourier approximation

Note that $f^{\Delta}(z)$ is a periodic function with period Δ. This function can be approximated by the Fourier series on $[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$. According to Fourier theory the Fourier series of $f^{\Delta}(z)$ is given by

$$\sum_{i=0}^{\infty} \alpha_i g_i(z), \tag{7.56}$$

where $S = \{g_0(z), g_1(z), \cdots\}$ is an orthonormal system in the Hilbert space H of squared integrable functions on $[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$ and where $\alpha_i = \langle f, g_i(z) \rangle$, with $\langle \cdot \rangle$ an inner product. The L_2-norm on $[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$ is used, which is

$$\|f\|_2^{[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]} = \left(\frac{1}{\frac{1}{2}\Delta} \int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} |f(z)|^2\, \mathrm{d}z \right)^{\frac{1}{2}} \tag{7.57}$$

and is derived from the inner product

$$\langle f, g \rangle^{[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]} = \frac{1}{\frac{1}{2}\Delta} \int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} |f(z)g(z)|\, \mathrm{d}z. \tag{7.58}$$

If for S the orthonormal system of sinuses and cosinuses is used, then $f^{\Delta}(z)$ can be written as

$$f^{\Delta}(z) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right), \tag{7.59}$$

where the a_i and b_i are the Fourier coefficients $\alpha_i = \langle f, g_i(z)\rangle$, so

$$a_0 = \frac{1}{\frac{1}{2}\Delta}\int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} f^{\Delta}(z)\,\mathrm{d}z, \tag{7.60}$$

$$a_n = \frac{1}{\frac{1}{2}\Delta}\int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} f^{\Delta}(z)\cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z \quad \text{and} \tag{7.61}$$

$$b_n = \frac{1}{\frac{1}{2}\Delta}\int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} f^{\Delta}(z)\sin\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z. \tag{7.62}$$

Because $f^{\Delta}(z)$ is an even function, $b_n = 0\ \forall n \in \mathbb{N}$. Also

$$a_0 = \frac{1}{\frac{1}{2}\Delta}\int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} f^{\Delta}(z)\,\mathrm{d}z = 0 \quad \text{and} \tag{7.63}$$

$$a_n = \frac{1}{\frac{1}{2}\Delta}\int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} \operatorname{sign}\left(-\cos\left(\frac{\pi}{\frac{1}{2}\Delta}z\right)\right)\cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z \tag{7.64}$$

$$= \frac{1}{\frac{1}{2}\Delta}\left(\int_{-\frac{1}{2}\Delta}^{-\frac{1}{4}\Delta}\cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z - \int_{-\frac{1}{4}\Delta}^{\frac{1}{4}\Delta}\cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z + \int_{\frac{1}{4}\Delta}^{\frac{1}{2}\Delta}\cos\left(\frac{n\pi}{\frac{1}{2}\Delta}z\right)\,\mathrm{d}z\right) \tag{7.65}$$

$$= \cdots = \frac{-4}{n\pi}\sin\left(\frac{1}{2}n\pi\right) = \begin{cases} 0 & \text{if } n \text{ even,} \\ (-1)^{(n+1)/2}\frac{4}{n\pi} & \text{if } n \text{ odd.} \end{cases} \tag{7.66}$$

This gives

$$f^{\Delta}(z) = \sum_{k=0}^{\infty}(-1)^{k+1}\frac{4}{(2k+1)\pi}\cos\left(\frac{(2k+1)\pi}{\frac{1}{2}\Delta}z\right), \tag{7.67}$$

where $n = 2k+1$ is used. For $f^{\hat{\Delta}}(z)$ the derivation is done similarly:

$$f^{\hat{\Delta}}(z) = \sum_{k=0}^{\infty}(-1)^{k+1}\frac{4}{(2k+1)\pi}\cos\left(\frac{(2k+1)\pi}{\frac{1}{2}\hat{\Delta}}z\right). \tag{7.68}$$

So we have

$$P_\Delta = \frac{1}{2} - \frac{1}{2}\int_0^1 \left\{ \sum_{k=0}^{\infty} (-1)^{k+1} \frac{4}{(2k+1)\pi} \cos\left(\frac{(2k+1)\pi}{\frac{1}{2}\Delta} z \right) \right. \\ \left. \times \sum_{k=0}^{\infty} (-1)^{k+1} \frac{4}{(2k+1)\pi} \cos\left(\frac{(2k+1)\pi}{\frac{1}{2}\hat{\Delta}} z \right) \right\} p_z(z) \mathrm{d}z. \tag{7.69}$$

7.3.3 Convergence analysis

In this subsection the convergence of (7.69) is established. According to Theorem 11.4 of [1] the Fourier series of a function f converges pointwise to f if and only if

$$\left(\|f\|_2^{[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]} \right)^2 = \sum_{i=1}^{\infty} |\alpha_i|^2, \tag{7.70}$$

with α_i the Fourier coefficients of f and where the L_2-norm on $[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$ is used. Equation (7.70) is known as Parseval's formula. For f^Δ this means

$$\left(\|f^\Delta\|_2^{[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]} \right)^2 = \frac{1}{\frac{1}{2}\Delta} \int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} f^\Delta(z)^2 \, \mathrm{d}z = 2 \tag{7.71}$$

(7.72)

and

$$\sum_{i=1}^{\infty} |\alpha_i|^2 = \sum_{k=0}^{\infty} \left| (-1)^{k+1} \frac{1}{2k+1} \frac{4}{\pi} \right|^2 = \frac{16}{\pi^2} \sum_{k=0}^{\infty} \frac{1}{(2k+1)^2} = \frac{16}{\pi^2} \frac{\pi^2}{8} = 2, \tag{7.73}$$

from which it can be concluded that the Fourier series of $f^\Delta(z)$ and $f^{\hat{\Delta}}(z)$ converges pointwise to $f^\Delta(z)$ and $f^{\hat{\Delta}}(z)$ respectively. Because $f^\Delta(z)$ and $f^{\hat{\Delta}}(z)$ are not continuous functions, their Fourier series does not converge uniformly, see [1]. It is known that $f^\Delta(z)$ and $f^{\hat{\Delta}}(z)$ are continuous on the interval $[0,1]$, except for some points of discontinuity $z = (2k+1)\frac{1}{4}\Delta$ or $z = (2k+1)\frac{1}{4}\hat{\Delta}$, $k \in \mathbb{N}$. Therefore, we have that their Fourier series are uniform convergent on any closed interval $I \subset [0,1]$ not containing a point of discontinuity.

According to Theorem 11.4 of [1], because Equation (7.70) holds, it also holds that

$$\lim_{n\to\infty} \|f - s_n\|_2^{[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]} = 0, \tag{7.74}$$

where $\{s_n\}$ is the sequence of partial sums defined by $s_n(z) = \sum_{k=0}^{n} \alpha_k \, g_k(z)$. Because $f^\Delta(z)$ is a periodical function, not only $\lim_{n\to\infty} \frac{1}{\frac{1}{2}\Delta} \int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} |f^\Delta(z) - s_n|^2 \, \mathrm{d}z = 0$, but also $\lim_{n\to\infty} \frac{1}{\frac{1}{2}\Delta} \int_{\frac{1}{2}\Delta}^{\frac{3}{2}\Delta} |f^\Delta(z) - s_n|^2 \, \mathrm{d}z = 0$, and so on. Also, $\lim_{n\to\infty} \frac{1}{\frac{1}{2}\Delta} \int_{-\frac{1}{2}\Delta}^{\frac{1}{2}\Delta} |f^\Delta(z) - s_n|^2 \, \mathrm{d}z = 0$ implies $\lim_{n\to\infty} \frac{1}{\frac{1}{2}\Delta} \int_a^b |f^\Delta(z) - s_n|^2 \, \mathrm{d}z = 0$, for $a, b \in [-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$, because the integration term is positive. Therefore

$$\lim_{n\to\infty} \|f - s_n\|_2^{[0,1]} = \lim_{n\to\infty} \|f - s_n\|_2^{[0, \frac{1}{2}\Delta]} + \lim_{n\to\infty} \|f - s_n\|_2^{[\frac{1}{2}\Delta, \frac{3}{2}\Delta]} \\ + \cdots + \lim_{n\to\infty} \|f - s_n\|_2^{[h,1]} = 0, \tag{7.75}$$

with $h = (2k+1)\frac{1}{2}\Delta$, where $(2k+1)$ the largest integer such that it is smaller than 1.

Denote the Fourier series of $f^{\Delta}(z)$ by s_n and the Fourier series of $f^{\hat{\Delta}}(z)$ by t_n. Then it can be shown that $s_n\, t_n$ converges to $f^{\Delta}(z)f^{\hat{\Delta}}(z)$ in the L_1-norm on $[0,1]$:

$$\begin{aligned}
\|s_n t_n - f^{\Delta} f^{\hat{\Delta}}\|_1^{[0,1]} &= \|(s_n - f^{\Delta})t_n + f^{\Delta}(t_n - f^{\hat{\Delta}})\|_1^{[0,1]} && (7.76)\\
&\leq \|(s_n - f^{\Delta})t_n\|_1^{[0,1]} + \|f^{\Delta}(t_n - f^{\hat{\Delta}})\|_1^{[0,1]} && (7.77)\\
&= \int_0^1 |(s_n - f^{\Delta})t_n|\, \mathrm{d}z + \int_0^1 |f^{\Delta}(t_n - f^{\hat{\Delta}})|\, \mathrm{d}z && (7.78)\\
&= \left\langle |s_n - f^{\Delta}|, |t_n| \right\rangle_2^{[0,1]} + \left\langle |f^{\Delta}|, |t_n - f^{\hat{\Delta}}| \right\rangle_2^{[0,1]} && (7.79)\\
&\leq \|(s_n - f^{\Delta})\|_2^{[0,1]}\, \|t_n\|_2^{[0,1]} + \|f^{\Delta}\|_2^{[0,1]}\, \|(t_n - f^{\hat{\Delta}})\|_2^{[0,1]}, && (7.80)
\end{aligned}$$

where the second line follows from the triangle inequality, the third from the definition of the L_1-norm on $[0,1]$, the fourth from the definition of the L_2-inner product on $[0,1]$ and the fifth from the Cauchy-Schwarz inequality. Because $\lim_{n\to\infty} \|f^{\Delta} - s_n\|_2^{[0,1]} = 0$, $\lim_{n\to\infty} \|f^{\hat{\Delta}} - t_n\|_2^{[0,1]} = 0$ (see Equation (7.75)), and because $\|s_n\|_2^{[0,1]}$ and $\|t_n\|_2^{[0,1]}$ are finite, the limit of the right-hand-side of (7.80) goes to zero and therefore

$$\lim_{n\to\infty} \|s_n t_n - f^{\Delta} f^{\hat{\Delta}}\|_1^{[0,1]} = 0. \tag{7.81}$$

So $s_n t_n$ goes to $f^{\Delta} f^{\hat{\Delta}}$ for $n \to \infty$ in L_1 sense. It is also known that

$$\left| \int_0^1 s_n t_n\, \mathrm{d}z - \int_0^1 f^{\Delta} f^{\hat{\Delta}}\, \mathrm{d}z \right| \leq \int_0^1 \left| s_n t_n - f^{\Delta} f^{\hat{\Delta}} \right|\, \mathrm{d}z \xrightarrow{n\to\infty} 0, \tag{7.82}$$

so the integral (7.69) of the product of the two Fourier-sums is convergent.

7.3.4 Statistics

In this subsection the expected value of P_{Δ} as given in (7.69) is calculated. Until now we have considered P_{Δ} as a deterministic function. It is also possible to see it as a stochastic variable depending on the stochastic variables Δ and $\hat{\Delta}$, or alternatively on Δ and the estimation error $\epsilon_1 = \hat{\Delta} - \Delta$. The mean of this stochastic variable can be calculated as

$$\mathrm{E}P_{\Delta} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P_{\Delta}(\epsilon_1, \Delta) p_{\epsilon_1,\Delta}(\epsilon_1, \Delta)\, \mathrm{d}\epsilon_1\, \mathrm{d}\Delta. \tag{7.83}$$

We need to know the probability density function $p_{\epsilon_1,\Delta}(\epsilon_1, \Delta)$. Therefore we derive the distributions of the stochastic variables ϵ_1 and Δ from the distribution of the host signal samples x.

Assume that the host signal samples $x = (x_1, x_2, \cdots, x_N)$ are identically and independently distributed with mean $\mathrm{E}x = \mu_x$ and variance $\mathrm{var}(x) = \sigma_x^2$. Let the quantization step size be determined according to $\Delta = \frac{\gamma}{L} \sum_{i=1}^{L} x_i$. Because of the Central Limit Theorem (see [19]), we have

for sufficiently large L that $S_L = \sum_{i=1}^{L} x_i$ can be approximated by a normal distribution with mean

$$\mu_S = \mathrm{E}\left[\sum x_i\right] = \sum \mathrm{E}\left[x_i\right] = L\mu_x \tag{7.84}$$

and variance

$$\sigma_S^2 = \mathrm{var}\left[\sum x_i\right] = \sum \mathrm{var}\left[x_i\right] = L\sigma_x^2. \tag{7.85}$$

Hence, $\Delta_L = \frac{\gamma}{L} S_L$ has a normal distribution, as well, with mean

$$\mu_\Delta = \mathrm{E}\left[\frac{\gamma}{L} S_L\right] = \frac{\gamma}{L} L\mu_x = \gamma\mu_x \tag{7.86}$$

and variance

$$\sigma_\Delta^2 = \mathrm{var}\left[\frac{\gamma}{L} S_L\right] = \frac{\gamma^2}{L^2}\mathrm{var}\left[S_L\right] = \frac{\gamma^2}{L}\sigma_x^2. \tag{7.87}$$

So the probability density function of Δ_L is given by

$$p_\Delta(\Delta) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{L}}{\gamma\sigma_x}\, e^{-L\frac{(\Delta-\gamma\mu_x)^2}{2\gamma^2\sigma_x^2}}. \tag{7.88}$$

By assumption the watermark $w = y - x$ is uniformly distributed over $[-\Delta, \Delta]$, if $\alpha = 1$, so

$$p_{w|\Delta}(w|\Delta) = \begin{cases} \frac{1}{2\Delta} & \text{if } w \in [-\Delta, \Delta] \\ 0 & \text{otherwise} \end{cases} \tag{7.89}$$

We defined $\epsilon_1 = \hat{\Delta} - \Delta$, therefore

$$\epsilon_1 = \frac{\gamma}{L}\sum_{i=1}^{L}(y_i - x_i) = \frac{\gamma}{L}\sum_{i=1}^{L}(x_i + w_i - x_i) = \frac{\gamma}{L}\sum_{i=1}^{L} w_i. \tag{7.90}$$

We calculate the distribution of ϵ_1 given Δ the same way as we did with calculating the distribution of Δ and we get that ϵ_1 given Δ is normally distributed with zero mean and variance $\frac{\gamma^2\Delta^2}{6L}$, so

$$p_{\epsilon_1|\Delta}(\epsilon_1) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{6L}}{\gamma\Delta}\, e^{-6L\frac{\epsilon_1^2}{2\gamma^2\Delta^2}} \tag{7.91}$$

Using the definition of conditional probabilities we have

$$p_{\epsilon_1,\Delta}(\epsilon_1, \Delta) = p_{\epsilon_1|\Delta}(\epsilon_1) \cdot p_\Delta(\Delta) \tag{7.92}$$

$$= \frac{1}{2\pi} \frac{\sqrt{6}L}{\gamma^2\sigma_x\Delta}\, e^{-6L\frac{\epsilon_1^2}{2\gamma^2\Delta^2}}\; e^{-L\frac{(\Delta-\gamma\mu_x)^2}{2\gamma^2\sigma_x^2}}. \tag{7.93}$$

From this the expected value for the error probability due to the estimation of Δ at the detector can be derived with (7.83). With $P_\Delta(\Delta, \epsilon_1)$ given by (7.69), where we have substituted $\hat{\Delta} = \epsilon_1 + \Delta$, we get the desired formula for the expected bit error probability

$$\mathrm{E}P_\Delta(\mu_x, \sigma_x, \gamma, L) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P_\Delta(\Delta, \epsilon_1) \times \frac{1}{2\pi} \frac{\sqrt{6}L}{\gamma^2\sigma_x\Delta}\, e^{-6L\frac{\epsilon_1^2}{2\gamma^2\Delta^2}}\; e^{-L\frac{(\Delta-\gamma\mu_x)^2}{2\gamma^2\sigma_x^2}}\, \mathrm{d}\epsilon_1\, \mathrm{d}\Delta. \tag{7.94}$$

This integral can be evaluated numerically.

7.4 Summary

In this chapter we have developed two models in order to quantify the performance of the watermarking model. Bit error probabilities are used as performance measure.

- The first model uses the SCS with a fixed quantization step size and an AWGN or uniform noise channel. For Gaussian noise two approximations of the bit error probability P_{BE} are calculated. These two bit error probabilities are given by Equation (7.24) and (7.33). For uniform noise one approximation is calculated, which is given by (7.46) and (7.50).
- The second model uses SCS with an adaptive quantization step size and an error free channel. The error probability P_{Δ} due to the estimation of Δ at the detector is given by (7.55).

The bit error probability P_{Δ} is approximated with the integral over two Fourier series. This is proved to be convergent in Subsection 7.3.3. The expected value of P_{Δ} is calculated in Subsection 7.3.4 and is given by (7.94).

The two models can be used in order to give a measure of the total performance of the watermarking system.

Chapter 8

Parameter optimization

In this chapter we determine the optimal value of the distortion compensation parameter α. This optimization is based on the bit error probability P_{BE}, derived in Chapter 7. Our work is related to the work of Joachim Eggers who also optimized α. His approach is a information-theoretic one, based on channel capacity maximization. We have an analytical expression, which can be optimized numerically, while Eggers has a numerical expression, which he numerically optimizes. The resulting optimal values cannot be compared exactly, but they turn out to be very similar.

In Section 8.1 the optimization problem is given. A solution is given in Section 8.2 and compared to the result of Eggers.

8.1 The optimization problem for α

It is assumed that embedding is done by SCS, with a fixed quantization step size Δ and an AWGN channel. So, the same conditions are used as in Subsection 7.2.2. In Section 7.2 an expression is derived for the error probability under these conditions, see Equation (7.9). For Gaussian noise this equation can be approximated with Equation (7.24) or (7.33). Both of these expressions for the bit error probability P_{BE} are dependent on the distortion compensation parameter α, the quantization step size Δ and on the noise power σ_n^2.

Recall that the watermark is defined as $w \triangleq y - x$. Using Equation (7.4) this gives $w = \alpha\left(\mathcal{Q}_\Delta(x) - x\right)$; Recall that $\mathcal{Q}_\Delta(X) - X \sim U[-\frac{1}{2}\Delta, \frac{1}{2}\Delta]$, see Subsection 7.2.1. Therefore, the watermark is uniformly distributed as $W \sim U[-\frac{1}{2}\alpha\Delta, \frac{1}{2}\alpha\Delta]$. The watermark variance is given by

$$\sigma_w^2 = \frac{(\frac{1}{2}\alpha\Delta - -\frac{1}{2}\alpha\Delta)^2}{12} = \frac{\alpha^2\Delta^2}{12}, \tag{8.1}$$

so Δ, α and σ_w^2 are related by

$$\Delta = \frac{2\sqrt{3}\sigma_w}{\alpha}. \tag{8.2}$$

In [17] Eggers reports the same relationship.

Substituting this relation for Δ in Equation (7.24) leads to the following expression of P_{BE}^0 as a function of α, σ_w and σ_n:

$$P^0_{\mathrm{BE}}\left(\alpha;\frac{\sigma_w^2}{\sigma_n^2}\right)=\frac{\alpha-\frac{1}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{1}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)-\frac{\alpha-\frac{3}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{3}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)+\frac{\alpha}{\sqrt{6\pi}(1-\alpha)}\frac{\sigma_n}{\sigma_w}\left(e^{-3/2\left(\frac{1}{4\alpha^2}-\frac{1}{\alpha}+1\right)\frac{\sigma_w^2}{\sigma_n^2}}-e^{-3/2\left(\frac{9}{4\alpha^2}-\frac{3}{\alpha}+1\right)\frac{\sigma_w^2}{\sigma_n^2}}\right)+1. \tag{8.3}$$

Note that P^0_{BE} only depends on α and the ratio of σ_w and σ_n. The Watermark-to-Noise Ratio is defined to be $\mathrm{WNR} \triangleq 10\ \log_{10}\left(\frac{\sigma_w^2}{\sigma_n^2}\right)$ dB. For constant WNR the bit error probability of Equation (8.3) is only a function of the distortion compensation parameter α. The value of the WNR depends on the application. For some applications the power of the noise attack can be quite large compared to the watermark power, for other applications the ratio between σ_n^2 and σ_w^2 is around 1.

The parameter α is set at the embedder. The question is now: How to set this parameter such that the bit error probability is minimal? Or, in mathematical terms, we need to find

$$\alpha^* = \arg\min_{\alpha\in]0,1]} P_{\mathrm{BE}}\left(\alpha;\frac{\sigma_w^2}{\sigma_n^2}\right). \tag{8.4}$$

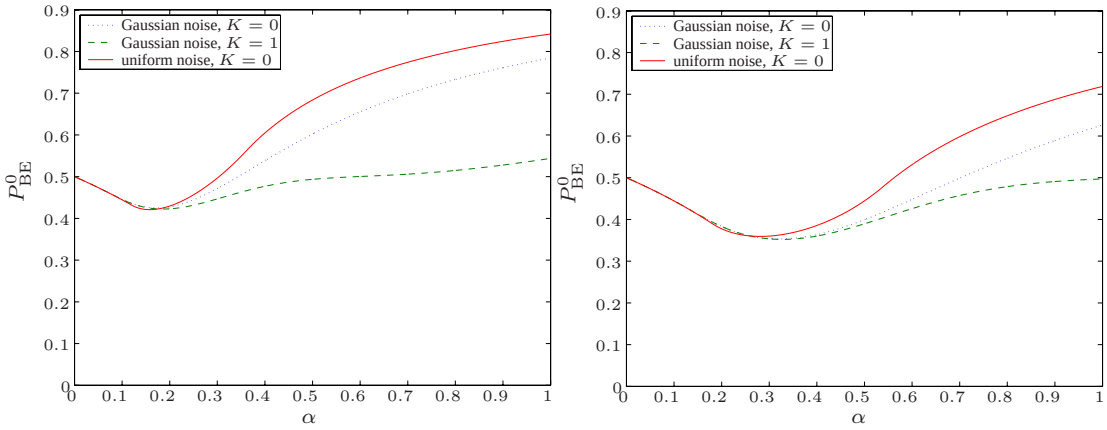
This problem can be approximated by the problems

$$\alpha^*_K = \arg\min_{\alpha\in]0,1]} P^K_{\mathrm{BE}}\left(\alpha;\frac{\sigma_w^2}{\sigma_n^2}\right). \tag{8.5}$$

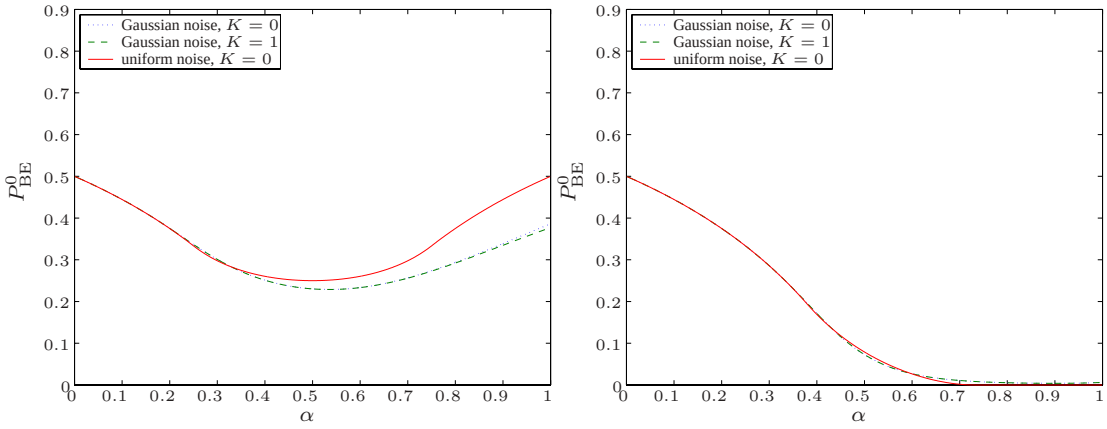
An explicit expression for P^K_{BE} is given by Equation (8.3) for $K=0$ and for $K=1$ by

$$\begin{aligned}P^1_{\mathrm{BE}}\left(\alpha;\frac{\sigma_w^2}{\sigma_n^2}\right)=&\frac{\alpha-\frac{1}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{1}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)-\frac{\alpha-\frac{3}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{3}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)\\&+\frac{\alpha-\frac{5}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{5}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)-\frac{\alpha-\frac{7}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{7}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)\\&-\frac{\alpha+\frac{1}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{1}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)+\frac{\alpha+\frac{3}{2}}{2(1-\alpha)}\,\mathrm{erf}\left(\frac{\sqrt{3/2}(\alpha-\frac{3}{2})}{\alpha}\frac{\sigma_w}{\sigma_n}\right)\\&+\frac{\alpha}{\sqrt{6\pi}(1-\alpha)}\frac{\sigma_n}{\sigma_w}\left(-e^{-\frac{3}{2}\frac{\left(\alpha-\frac{7}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}+e^{-\frac{3}{2}\frac{\left(\alpha-\frac{5}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}-e^{-\frac{3}{2}\frac{\left(\alpha-\frac{3}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}\right.\\&\left.+e^{-\frac{3}{2}\frac{\left(\alpha-\frac{1}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}-e^{-\frac{3}{2}\frac{\left(\alpha+\frac{1}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}+e^{-\frac{3}{2}\frac{\left(\alpha+\frac{3}{2}\right)^2}{\alpha^2}\frac{\sigma_w^2}{\sigma_n^2}}\right)+1.\end{aligned} \tag{8.6}$$

In Chapter 7 we also derived an expression for P^K_{BE} for the case of uniform noise. In this case, $c=\sqrt{3}\sigma_n$, see (7.1). Substituting this and (8.2) in (7.46) and (7.50) we get expressions for P^0_{BE} as a function of α and $\frac{\sigma_w}{\sigma_n}$ for the assumption of uniform noise,



(a) P^K_{BE}'s for WNR $= -10$ (b) P^K_{BE}'s for WNR $= -5$



(c) P^K_{BE}'s for WNR $= 0$ (d) P^K_{BE}'s for WNR $= 5$

Figure 8.1: The error probabilities for different Watermark-to-Noise Ratio's: P^0_{BE} for the case of uniform and Gaussian noise and P^1_{BE} for the Gaussian case.

$$
\begin{aligned}
&\text{If} \quad \frac{\sigma_n}{\sigma_w} \le \frac{1-\alpha}{\alpha} \quad \text{(weak noise case) then} \\
&P^0_{\mathrm{BE}} = \begin{cases} 1 - \frac{1}{2(1-\alpha)} & \text{if } \frac{\sigma_n}{\sigma_w} \le \frac{1-2\alpha}{2\alpha} \\ \frac{1}{2} - \frac{1}{4(1-\alpha)} + \left(\frac{1}{16\alpha(1-\alpha)} - \frac{1}{4}\right)\frac{\sigma_w}{\sigma_n} + \frac{\alpha}{4(1-\alpha)}\frac{\sigma_n}{\sigma_w} & \text{if } \frac{\sigma_n}{\sigma_w} \ge \frac{1-2\alpha}{2\alpha} \wedge \frac{\sigma_n}{\sigma_w} \ge \frac{2\alpha-1}{2\alpha} \\ 0 & \text{if } \frac{\sigma_n}{\sigma_w} \le \frac{2\alpha-1}{2\alpha} \end{cases} \\
&\text{If} \quad \frac{\sigma_n}{\sigma_w} > \frac{1-\alpha}{\alpha} \quad \text{(strong noise case) then} \\
&P^0_{\mathrm{BE}} = \begin{cases} 1 - \frac{1}{2\alpha}\frac{\sigma_w}{\sigma_n} & \text{if } \frac{\sigma_n}{\sigma_w} \le \frac{1-2\alpha}{2\alpha} \\ \frac{1}{2} - \frac{1}{4(1-\alpha)} + \left(\frac{1}{16\alpha(1-\alpha)} - \frac{1}{4}\right)\frac{\sigma_w}{\sigma_n} + \frac{\alpha}{4(1-\alpha)}\frac{\sigma_n}{\sigma_w} & \text{if } \frac{\sigma_n}{\sigma_w} \ge \frac{1-2\alpha}{2\alpha} \wedge \frac{\sigma_n}{\sigma_w} \ge \frac{2\alpha-1}{2\alpha} \\ 0 & \text{if } \frac{\sigma_n}{\sigma_w} \le \frac{2\alpha-1}{2\alpha} \end{cases}
\end{aligned}
\tag{8.7}
$$

The bit error probabilities (8.3), (8.6) and (8.7) are shown in Figure 8.1 for different WNR's.

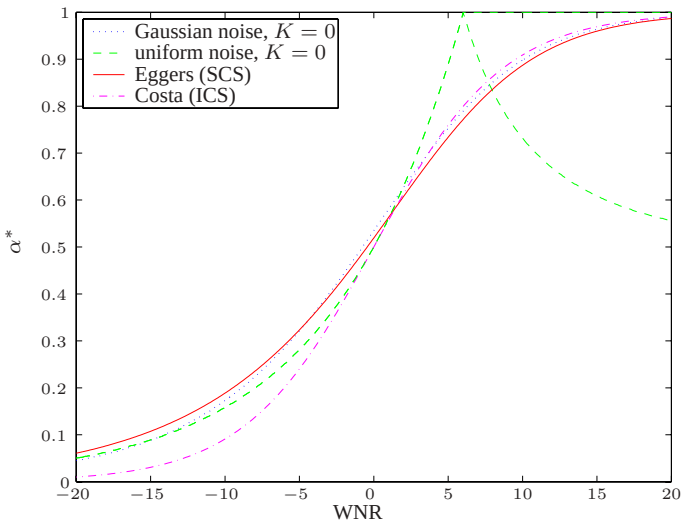


Figure 8.2: The optimal distortion compensation parameter α for Gaussian and uniform noise attacks and $K = 0$. For uniform noise there is a clipping point in the figure. For WNR's larger than this clipping point the optimal distortion compensation parameter α^* is the set of values between the two green dashed lines.

8.2 Bit error minimization and Eggers

Finding the optimal distortion compensation parameter α^*, concerns finding the function $\alpha^*\left(\frac{\sigma_w}{\sigma_n}\right)$ that minimizes (8.4). This is hard to solve analytically, but for a given value of $\frac{\sigma_w}{\sigma_n}$ the corresponding value α^* can easily be calculated numerically.

Eggers also optimized the distortion compensation parameter. However, where we minimized the channel bit error probability, he maximized the channel capacity C

$$\alpha^* = \arg\max_{\alpha \in]0,1[} C\left(\alpha; \frac{\sigma_w}{\sigma_n}\right). \tag{8.8}$$

Eggers found a numerical approximation of the capacity function. He optimized this function numerically and found a sequence of α^*'s for a given WNR. He approximated this sequence with the formula

$$\alpha^*_{\text{SCS}} = \sqrt{\frac{\sigma_w^2}{\sigma_w^2 + 2.71\sigma_n^2}}, \tag{8.9}$$

for the Scalar Costa Scheme (SCS).

A similar optimization problem can be defined for the case of uniform noise. In Figure 8.2 the optimal distortion compensation parameter α^* is plotted against the WNR for the different problems and approaches: problem (8.5) with $K = 1$, the optimal α^* for uniform noise, Eggers' formula (Equation (8.9)) and for the Ideal Costa Scheme (ICS) found by Costa given by

$$\alpha^*_{\text{ICS}} = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_n^2}. \tag{8.10}$$

We see that our model for $K = 0$ is almost equal to the result of Eggers. For $K = 1$ we also derived the optimal α^*. This improved bit error probability is plotted in Figure 8.3.

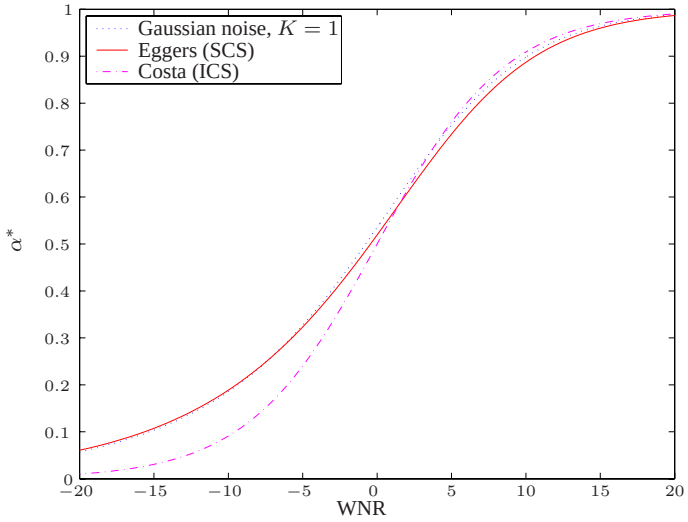


Figure 8.3: Optimal distortion compensation parameter for the improved model ($K = 1$).

The improvement for $K = 1$ is very small, but still it is even closer to Eggers' function than our earlier result with $K = 0$. Because of the small difference between the models for $K = 0$ and $K = 1$, see Figure 8.4, an increase in the value of K is not expected to improve the estimate for the optimal distortion compensation parameter α^*.

In order to determine the bit error probability in the presence of relatively strong noise, i.e., low WNR's, it is necessary to use higher values of K. This can be seen from Figure 8.1, where for low WNR's there is difference between P_{BE}^0 and P_{BE}^1. This is explained from the fact that for strong noise, more quantization bins need to be taken into account. This is also why higher bit error probabilities are seen than 0.5. It is also seen from Figure 8.1 that for higher WNR's, from $\mathrm{WNR} = 0$, P_{BE}^0 and P_{BE}^1 are practically indifferent.

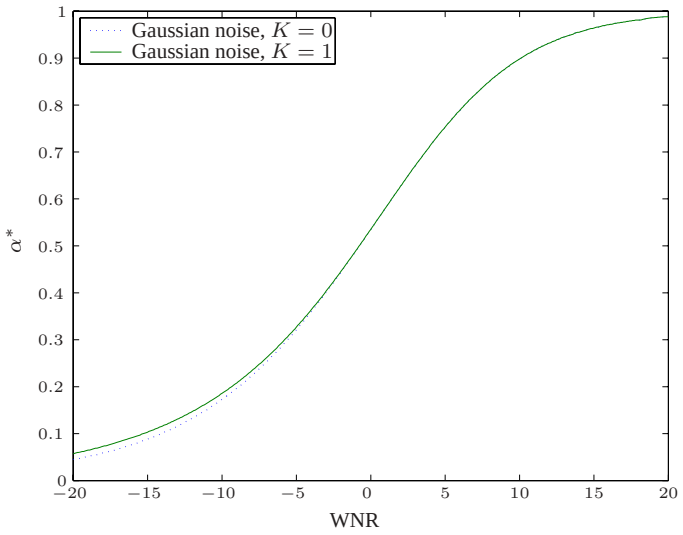


Figure 8.4: Optimal distortion compensation parameter for the two models ($K = 0$ and $K = 1$) of Gaussian noise.

Chapter 9

Contributions, Conclusions and Recommendations

9.1 Contributions

During the internship we have done the following:

- A literature study of the state of the art of quantization watermarking was made. An overview of watermarking in general and quantization based watermarking is given in Chapter 2 and 3, respectively;
- The SCS is implemented in Matlab and C++;
- Two improvements of this scheme are developed: the use of error correcting codes and the use of an adaptive quantization step size. The aim of the improvements is to get a better robustness of the watermarking algorithm at an equal perceptibility level of the watermark;
- Several possibilities of the adaptive quantization method are implemented in C++;
- A convolutional encoder and a corresponding Viterbi decoder are implemented in C++. This implementation uses an arbitrary number of inputs and outputs, so any rate is achievable. Decoding has a high complexity, therefore it is only possible to use a memory size up to $M = 9$;
- A complete watermarking embedder and detector of SCS with the two improvements is implemented in C++;
- Two models for estimating the performance of the watermarking algorithm are developed. This results in two bit error probabilities P_{BE} and P_{Δ};
- A detection threshold T is calculated;
- The distortion compensation parameter α is optimized for a Gaussian host signal and an AWGN communication channel. This optimal α^* is compared with a similar result of Eggers and is found to be equal;
- Experiments are done in order to estimate the result of the improvements.

9.2 Conclusions

As stated in Chapter 4 we are seeking an increased robustness against AWGN and JPEG attacks and against brightness scaling. The methods we have used were Error Correcting Codes and adaptive quantization.

From Chapter 5 we conclude that ECC are a valuable tool in order to decrease the bit error probability and in this way increase robustness against all sorts of attacks. Concatenation of convolutional codes and repetition codes is valuable as long as the noise introduced by the channel is low enough. Beyond a certain noise strength CC do not add significant robustness improvements.

Adaptive quantization strongly adds to robustness against brightness scaling; It is much better than fixed quantization. Against AWGN the improvement from adaptive quantization is limited and for JPEG it is slightly better to use fixed quantization.

Another goal of this report is to give an analytical model for the performance of the watermark model. This was done in Chapter 7. The performance is measured by the bit error probability. The bit error probability for the case of fixed quantization, distortion compensation and a uniform or Gaussian noise channel is determined. This bit error probability P_{BE} is a function of the distortion compensation parameter α and the WNR. Also the bit error probability for adaptive quantization, without distortion compensation and without a noise attack is determined. The expectation of this bit error probability P_{Δ} is a function of the mean and variance of the host signal x, and the two parameters in order to be able to determine the adaptive quantization step size, namely γ and L. The developed models can be used to measure the effects of attacks and the use of adaptive quantization on robustness.

For the case of a fixed quantization step size, distortion compensation and an AWGN channel the optimal distortion compensation parameter α^* is determined. Using this parameter at the embedder, robustness in an AWGN situation is maximized. The α^* we found was compared with values found by Eggers [17], and was found to be identical.

If the maximum allowed distortion from the watermark w is known i.e., if σ_w^2 and γ are known, and if it is known in which range the attack strength is, i.e., σ_n^2 is known, then the WNR can be calculated. For this WNR the optimal distortion compensation parameter α^* can be derived from Chapter 8. From Chapter 6 we have that $L = 11$ is a reasonable value to set L to. For a given host, the two derived bit error probabilities can be calculated, giving a measure of the performance in this situation.

9.3 Recommendations

The following topics deserve to be subject of further study:

- $P_{\mathrm{BE}}(\alpha)$ can be compared with values found by experiment, using the software that was developed during the internship.
- The bit error probability for the case of adaptive quantization, with distortion compensation, and with a noise attack has to be determined in order to get a reliable measure for the overall performance of the watermarking model. Is there a relationship between the two derived bit error probabilities and the overall one? Maybe it is possible to prove that $P_{\mathrm{total}} \leq P_{\mathrm{BE}} + P_{\Delta}$.
- Are there other improvements, besides the use of ECC and adaptive quantization?

- The present watermarking algorithm is not robust against synchronization and geometrical attacks. Countermeasures have to be sought.

Appendix A

Notation

Variables and Functions

x	host signal
y	watermarked signal
y_{M}	watermarked signal, watermarked using a watermarking scheme M
w	watermark $w \triangleq y - x$
n	processing on a channel, usually considered as noise
r	signal received at the detection point
m	message
l	message length
m^{c}	encoded message
l^{c}	encoded message length
$\hat{m}$	message estimated at the detector
k	key sequence
d	dither sequence
α	distortion compensation parameter
γ	embedding strength parameter
Δ	quantization step size
L	number of samples in an environment
N	length of signals x, y, w, n, r, d
T	threshold setting
P_{BE}	bit error probability due to a noise attack
P_{BE}^{K}	K^{th} order approximation of P_{BE}
P_{Δ}	bit error probability due to the estimation of Δ at the detector
$\mathrm{erf}(z)$	error function
$\mathrm{erfc}(z)$	complementary error function
$\mathcal{Q}_{\Delta}(z)$	quantizer with step size Δ
$\lfloor \cdot \rceil$	rounding to the nearest integer

Abbreviations and Acronyms

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BMP	BitMaP
CC	Convolutional Code
DC-QIM	Distortion Compensated Quantization Index Modulation
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DM	Dither Modulation
DS-CDMA	Direct Sequence Code Division Multiple Access
DVD	Digital Versatile Disc
DWT	Discrete Wavelet Transform
ECC	Error Correcting Code
GIF	Graphics Interchange Format
HAS	Human Auditory System
HVS	Human Visual System
ICS	Ideal Costa Scheme
IH	Information Hiding
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MPEG	Motion Pictures Experts Group
MSE	Mean Square Error
NASA	National Aeronautics and Space Administration
pdf	probability density function
SCS	Scalar Costa Scheme
SNR	Signal-to-Noise Ratio
SS	Spread Spectrum
QIM	Quantization Index Modulation
WNR	Watermark-to-Noise Ratio

Appendix B

Images

(a) `lena.bmp`

(b) `baboon.bmp`

(c) `peppers.bmp`

(d) `tulips.bmp`

Figure B.1: The images used for experiments.

Appendix C

Graphs for $P_{\triangle}$

We have plotted the bit error probability P_{Δ} due to the estimation of Δ at the detector, as a function of Δ and the absolute error $\epsilon_1 = \hat{\Delta} - \Delta$, see Figure C.1 and C.2. In Figure C.1 Δ is plotted for $0 \leq \Delta \leq 1$ after scaling of the parameters, see Section 7.3. For large values of $Delta$, the distortion becomes too high. Therefore, those large values are disregarded. Therefore, the relevant range for Δ is more like $0 \leq \Delta \leq 0.1$ after scaling of the parameters, which is 0 to 25 before scaling. See Figure C.2 for a plot of P_{Δ} in this range.

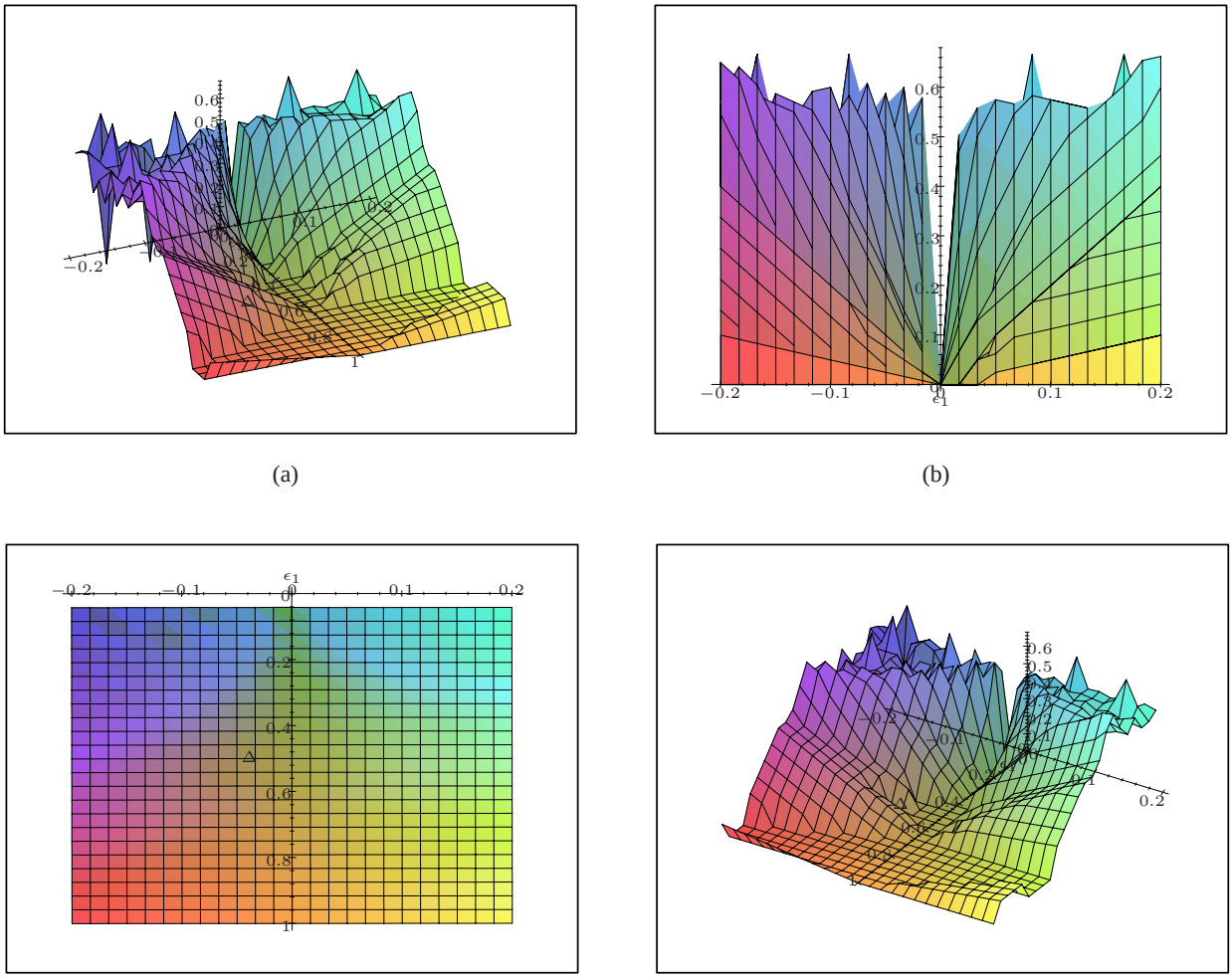
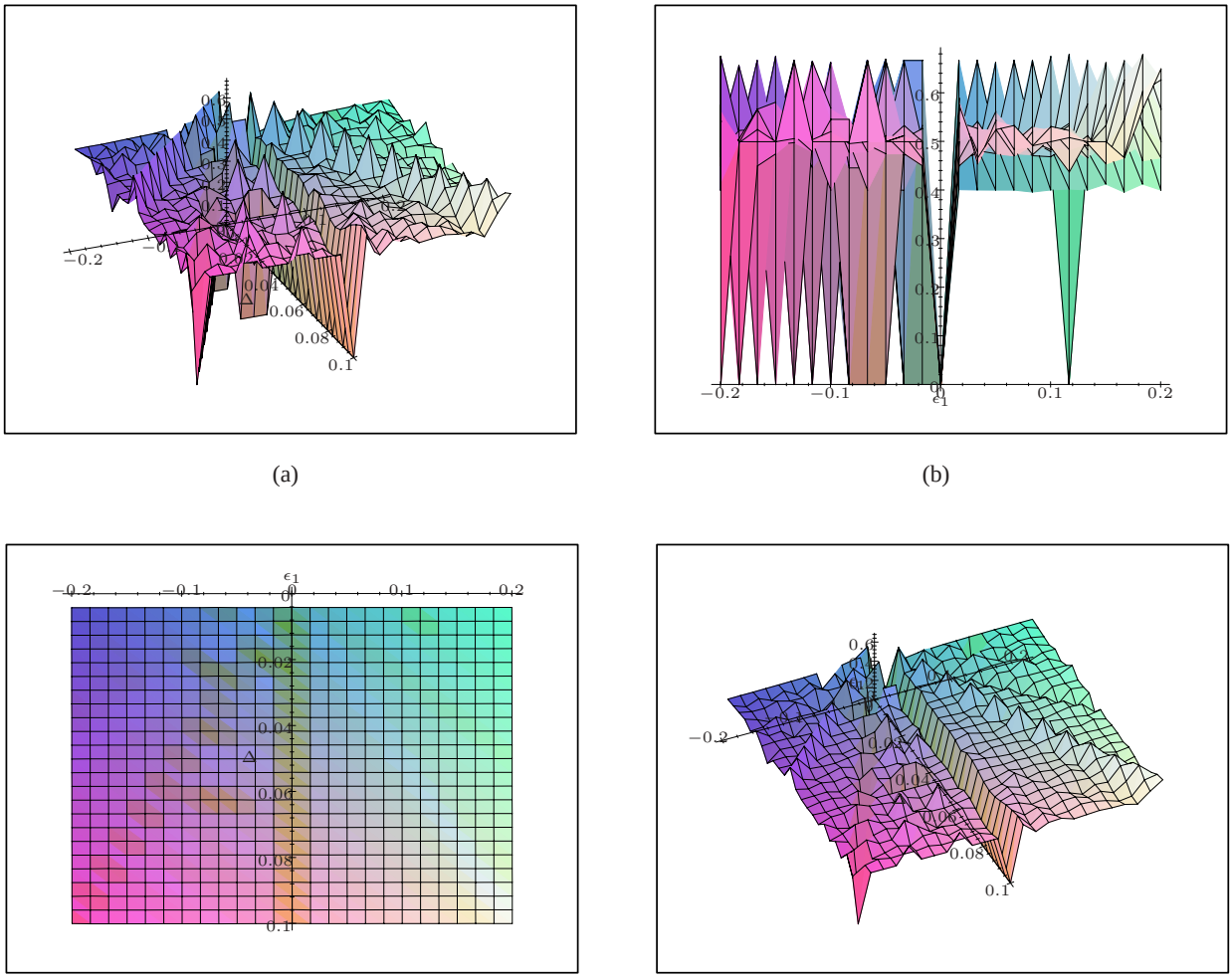


Figure C.1: The bit error rate P_Δ as a function of Δ and ϵ_1. P_Δ is shown from different angles. Δ is in the range $[0, 1]$.



(a) (b)

(c) (d)

Figure C.2: The bit error rate P_Δ as a function of Δ and ϵ_1. P_Δ is shown from different angles. Δ is in the range $[0, 0.1]$.

Bibliography

[1] T.M. Apostol, *Mathematical Analysis*, Addison-Wesley series in Mathematics, Addison-Wesley, Reading, Massachusetts, USA, 1974.

[2] S. Baudry, Jean-François Delaigle, B. Sankur et al., "Analysis of error correction strategies for typical communication channels in watermarking," *Signal Processing*, vol. 81, no. 6, pp. 1239 - 1250, June 2001.

[3] B. Chen and G.W. Wornell, "Digital Watermarking and Information Embedding using Dither Modulation," *Proc. of IEEE Workshop on Multimedia Signal Processing*, pp. 273-278, Redondo Beach, CA, USA, December 1998.

[4] B. Chen and G.W. Wornell, "Dither modulation: a new approach to digital watermarking and information embedding," *Proc. of SPIE Vol. 3657: Security and Watermarking of Multimedia Contents*, pp. 342 - 353, San Jose, Ca, USA, January 1999.

[5] B. Chen and G.W. Wornell, "An information-theoretic approach to the design of robust digital watermarking systems", *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1999 (ICASSP '99)*, vol. 4, pp. 2061 - 2064, Phoenix, USA, March 1999.

[6] B. Chen and G. W. Wornell, " Achievable performance of digital watermarking systems," *Proc. of IEEE International Conference on Multimedia Computing and Systems 1999 (ICMCS-99)*, Special Session on Security and Watermarking, vol. 1, pp. 13 - 18, Florence, Italy, June 1999.

[7] B. Chen and G.W. Wornell, "Preprocessed and postprocessed quantization index modulation methods for digital watermarking," *Proc. of SPIE Vol. 3971: Security and Watermarking of Multimedia Contents II*, pp. 48 - 59, San Jose, Ca, USA, January 2000.

[8] B. Chen, "*Design and Analysis of Digital Watermarking, Information Embedding, and Data Hiding Systems*," PhD. thesis, Department of Electrical Engineering and Computer Science, MIT, Boston, MA, USA, June 2000.

[9] B. Chen and G. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, May 2001.

[10] L. Chiang, *Human Visual System: Visually Lossless Image Compression*, [Online] Available: `http://www.image.ee.cityu.edu.hk/~loben/thesis/node30.html`, Accessed: 16 September 2002.

[11] M.H.M. Costa, ”Writing on Dirty Paper,” *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439 - 441, May 1983.

[12] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, John Wiley & Sons, New York, USA, June 1991.

[13] I.J. Cox, M. Miller, J.P.M.G. Linnartz and A.C.C. Kalker, ”A review of watermarking principles and practices”, Chapter 17 of *Digital Signal Processing for Multimedia Systems*, pp. 461 - 486, K.K. Parhi and T. Nishitani (eds.), Marcel Dekker, Inc., New York, March 1999.

[14] I.J. Cox, M.L. Miller and J.A. Bloom, *”Digital Watermarking*,” Morgan Kaufmann Publishers, Inc., San Francisco, The Morgan Kaufmann Series in Multimedia and Information Science, Edward Fox (Ed.), 2001.

[15] J.J. Eggers, J.K. Su and B. Girod, ”A Blind Watermarking Scheme Based on Structured Codebooks,” *Secure Images and Image Authentication, Proc. IEE Colloquium*, pp. 4/1 - 4/6, London, UK, April 2000.

[16] J.J. Eggers, J.K. Su and B. Girod, ”Robustness of a blind image watermarking scheme,” *Proc. of the IEEE Intl. Conference on Image Processing 2000*, Vancouver, Canada, September 2000.

[17] J.J. Eggers, R. Bäuml, R. Tzschoppe and B. Girod, ”Scalar Costa Scheme for Information Embedding,” submitted to *IEEE Transactions on Signal Processing*, 2002.

[18] J.J. Eggers and B. Girod, *Informed Watermarking*, The Kluwer international series in engineering and computer science 685, Kluwer Academic Publishers, Boston, April 2002.

[19] W. Feller, *An Introduction to Probability Theory and Its Applications*, Volume II, Wiley Series in Probability and Mathematical Statistics, R.A. Bradley et al. (eds), John Wiley & Sons, New York, USA, 1971.

[20] R.M. Gray and T.G. Stockham, ”Dithered quantizers”, *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 805 - 812, May 1993.

[21] A. Kerckhoffs, ”La cryptographie militaire,” *Journal des Sciences Militaires*, vol. 9, pp. 5 - 38, January 1883.

[22] G.C. Langelaar, I. Setyawan and R.L. Lagendijk, ”Watermarking Digital Image and Video Data: A State-of-the-Art Overview,” *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20 - 46, September 2000.

[23] S. Lin and D.J. Costello, *”Error Control Coding: Fundamentals and Applications*”, Prentice Hall, Englewood Cliffs, NJ, February 1983.

[24] S.P. Lipschitz, R.A. Wannamaker and J. Vanderkooy, ”Quantization and dither: A theoretical survey,” Journal of the Audio Engineering Society, vol. 40, no. 5, pp. 355 - 375, May 1992.

[25] R.J. McEliece, *”The Theory of Information and Coding: a mathematical framework for communication*”, vol. 3, Addison-Wesley Publishing Company, Reading, MA, Encyclo-

pedia of Mathematics and its Applications, Gian-Carlo Rota (Ed.), 1977.

[26] A.M. Mayache, T. Eude and H. Cherifi, ”A comparison of image quality models and metrics based on human visual sensitivity,” *IEEE International Conference on Image Processing*, vol. 3, pp. 409 - 413, October 1998.

[27] S.R.M. Oliveira, M.A. Nascimento and O.R. Zaïane, ”Digital Watermarking: Its Status, Limitations and Prospects,” Technical Report, Department of Computing Science, University of Alberta, Canada, Januari 2002.

[28] V.S. Pless, W.C. Huffman and R.A. Brualdi (eds), *Handbook of Coding Theory*, Elsevier Publishing Company, Amsterdam, The Netherlands, October 1998.

[29] S. Schimmel, ”Motion Sensitive Video Watermarking, preliminary study”, Nat.Lab. unclassified report, 2001/824, Philips Research, Eindhoven, April 2001.

[30] S. Schimmel, ”Motion Sensitive Video Watermarking, final report”, Nat.Lab. unclassified report, 2001/825, Philips Research, Eindhoven, August 2001.

[31] L. Schuchman, ”Dither signals and their effect on quantization noise,” *IEEE Transactions on Communications Technology (COM)*, vol. 12, pp. 162 - 165, December 1964.

[32] I. Setyawan, *Attacks on Watermarking Systems*, Technical Report, Information and Communication Theory Group, TU Delft, 2000.

[33] C.E. Shannon, ”A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379 - 423 and 623 - 656, July and October 1948.

[34] C.E. Shannon, ”Channels with side information at the transmitter,” *IBM J. Res. Develop.*, vol. 2, pp. 289 - 293, 1958.

Index