Real-time Watermarking Techniques for Compressed Video Data

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op dinsdag 1 februari 2000 te 16:00 uur

door

Gerrit Cornelis LANGELAAR

elektrotechnisch ingenieur,
geboren te Leersum.

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr.ir. J. Biemond
Prof.dr.ir. R.L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. J. Biemond,	Technische Universiteit Delft, promotor
Prof.dr.ir. R.L. Lagendijk,	Technische Universiteit Delft, promotor
Dr.ir. J.C.A. van der Lubbe,	Technische Universiteit Delft
Prof.dr. S. Vassiliadis,	Technische Universiteit Delft
Prof.dr.ir. L.J. van Vliet,	Technische Universiteit Delft
Prof.dr. T. Kalker,	Technische Universiteit Eindhoven
Prof.dr. E.J. Delp,	Purdue University, USA

Real-time Watermarking Techniques for Compressed Video Data
Langelaar, Gerrit Cornelis
Thesis Delft University of Technology - With ref. - With Summary in Dutch
Printed by Universal Press, Veenendaal
ISBN 90-9013190-6

Aan mijn ouders

Table of Contents

Real-time watermarking techniques for compressed video data

Summary

In the past few years there has been an explosion in the use and distribution of digital multimedia data. Personal computers with internet connections have taken the homes by storm, and have made the distribution of multimedia data and applications much easier and faster. Furthermore the analog audio and video equipment in the home are in the process of being replaced by their digital successors.

Although digital data have many advantages over analog data, service providers are reluctant to offer services in digital form because they fear unrestricted duplication and dissemination of copyrighted material. The lack of adequate protection systems for copyrighted content was for instance the reason for the delayed introduction of the Digital Versatile Disk (DVD). Several media companies initially refused to provide DVD material until the copy protection problem had been addressed.

To provide copy protection and copyright protection for digital audio and video data, two complementary techniques are being developed: encryption and watermarking. Encryption techniques can be used to protect digital data during the transmission from the sender to the receiver. However, after the receiver has received and decrypted the data, the data is in the clear and no longer protected. Watermarking techniques can complement encryption by embedding a secret imperceptible signal, a watermark, directly into the clear data. This watermark signal is embedded in such a way that it cannot be removed without affecting the quality of the audio or video data. The watermark signal can for instance be used for copyright protection as it can hide information about the author in the data. The watermark can now be used to prove ownership in court. Another interesting application for which the watermark signal can be used is to trace the source of illegal copies by means of *fingerprinting* techniques.

In this case, the media provider embeds watermarks in the copies of the data with a serial number that is related to the customer's identity. If illegal copies are found, for instance on the Internet, the intellectual property owner can easily identify customers who have broken their license agreement by supplying the data to third parties. The watermark signal can also be used to control digital recording devices as it can indicate whether certain data may be recorded or not. In such case the recording devices must be equipped with watermark detectors, of course. Other applications of the watermark signal include: automated monitoring systems for radio and TV broadcasting, data authentication and transmission of secret messages.

Each watermarking application has its own specific requirements. Nevertheless, the most important requirements that are to be met by most watermarking techniques are that the watermark is imperceptible in the data in which it is hidden, that the watermark signal can contain a reasonable amount of information and that the watermark signal cannot be removed easily without affecting the data in which the watermark is hidden.

In this thesis an extensive overview is given of different existing watermarking methods. However, the emphasis is on the particular class of watermarking techniques that is suitable for real-time embedding watermarks in and extracting them from compressed video data. This class of techniques is for instance suitable for fingerprinting and copy protection systems in home-recording devices.

To qualify as a real-time watermarking technique for compressed video data, a watermark technique should meet the following requirements besides the already mentioned ones. The techniques for watermark embedding and extracting may not be too complex, for which there are two reasons: they are to be processed in real time, and as they are to be used in consumer products, they must be inexpensive. This means that fully decompressing the compressed data, adding a watermark and subsequently compressing the data again is not an option. It should be possible to add a watermark directly to the compressed data. Furthermore, it is important that the addition of a watermark does not influence the size of the compressed data. For instance, if the size of a compressed MPEG video stream increases, transmission over a fixed bit-rate channel can cause problems: the buffers in hardware decoders can run out of space, or the synchronization of audio and video can be disturbed.

The most efficient way to reduce the complexity of real-time watermarking algorithms is to avoid computationally demanding operations by exploiting the compression format of the video data. In this thesis two new watermarking concepts are introduced that directly operate on the compressed data stream, namely the least significant bit (LSB) modification concept and the Differential Energy Watermark (DEW) concept.
If the LSB concept is used to add a watermark, only fixed or variable length codes in the compressed data stream are replaced by other codes. Advantages of this concept are the high computational efficiency and the enormous amount of information that can be stored in the watermark signal. A drawback of this concept is that the watermark embedding and extraction procedures are completely dependent on the data structure of the compressed video stream. Once a compressed video stream is decompressed, the watermark is lost. Since fully decompressing and re-compressing a video stream is a task that is computationally quite demanding, this is not really a problem for consumer applications requiring moderate robustness.

For real-time applications that require a higher level of robustness, we have developed the DEW watermarking concept. The DEW algorithm adds a watermark by enforcing energy differences between video regions. The energy differences are enforced by selectively discarding high frequency coefficients. To embed a watermark in or extract a watermark from a compressed video stream, the DEW algorithm only requires partial decoding steps; it does not require partial video encoding steps. The complexity of the DEW watermarking algorithm is therefore only slightly higher than the LSB-based methods. Since the watermarks embedded with the DEW concept are not dependent on the data

structure of the compressed video stream, the watermarks remain present after the video stream has been decompressed.

The last part of this thesis is dedicated to the evaluation of the DEW concept. Several approaches to evaluate watermarking methods from literature are discussed and applied. Furthermore, watermark removal attacks discussed in literature are explained and a new watermark removal attack is proposed.

Chapter 1

Introduction

1.1 The need for watermarking

In the past few years there has been an explosion in the use and distribution of digital multimedia data. Personal computers with internet connections have taken the homes by storm, and have made the distribution of multimedia data and applications much easier and faster. Electronic commerce applications and on-line services are rapidly being developed. Even the analog audio and video equipment in the home are in the process of being replaced by their digital successors. As a result, we can see the digital mass recording devices for multimedia data enter the consumer market of today.

Although digital data have many advantages over analog data, service providers are reluctant to offer services in digital form because they fear unrestricted duplication and dissemination of copyrighted material. Because of possible copyright issues, the intellectual property of digitally recorded material must be protected [Sam91]. The lack of such adequate protection systems for copyrighted content was the reason for the delayed introduction of the Digital Versatile Disk (DVD) [Tay97]. Several media companies initially refused to provide DVD material until the copy protection problem had been addressed [Rup96], [Ren96]. Representatives of the consumer electronics industry and the motion picture industry have agreed to seek legislation concerning digital video recording devices. Recommendations describing ways that would protect both intellectual property and consumers' rights have been submitted to the US Congress [Ren96] and resulted in the Digital Millennium Copyright Act [DCM98], which was signed by President Clinton October 28, 1998.

To provide copy protection and copyright protection for digital audio and video data, two complementary techniques are being developed: encryption and watermarking [Cox97]. Encryption techniques can be used to protect digital data during the transmission from the sender to the receiver [Lan99a]. However, after the receiver has received and decrypted the data, the data is in the clear and no longer protected. Watermarking techniques can complement encryption by embedding a secret imperceptible signal, a watermark, directly into the clear data in such a way that it always remains present. Such a watermark can for instance be used for the following purposes:

- **Copyright protection:** For the protection of intellectual property, the data owner can embed a watermark representing copyright information in his data. This watermark can prove his ownership in court when someone has infringed on his copyrights.

- **Fingerprinting:** To trace the source of illegal copies, the owner can use a fingerprinting technique. In this case, the owner can embed different watermarks in the copies of the data that are supplied to different customers. Fingerprinting can be compared to embedding a serial number that is related to the customer's identity in the data. It enables the intellectual property owner to identify customers who have broken their license agreement by supplying the data to third parties. In Section 1.4 a fingerprinting application is explained in more detail.

- **Copy protection:** The information stored in a watermark can directly control digital recording devices for copy protection purposes [Lan98a]. In this case, the watermark represents a copy-prohibit bit and watermark detectors in the recorder determine whether the data offered to the recorder may be stored or not. A complete copy-protection system is discussed in Section 1.4.

- **Broadcast monitoring:** By embedding watermarks in commercial advertisements an automated monitoring system can verify whether advertisements are broadcasted as contracted [And98]. Not only commercials but also valuable TV products can be protected by broadcast monitoring [Kal99]. News items can have a value of over 100.000 USD per hour, which make them very vulnerable to intellectual property rights violation. A broadcast surveillance system can check all broadcast channels and charge the TV stations according to their findings.

- **Data authentication:** *Fragile* watermarks [Wol99a] can be used to check the authenticity of the data. A fragile watermark indicates whether the data has been altered and supplies localization information as to where the data was altered.

Watermarking techniques are not only used for protection purposes. Other applications include:

- **Indexing:** Indexing of video mail, where comments can be embedded in the video content; indexing of movies and news items, where markers and comments can be inserted that can be used by search engines.
- **Medical safety:** Embedding the date and the patient's name in medical images could be a useful safety measure [And98].

- **Data hiding:** Watermarking techniques can be used for the transmission of secret private messages. Since various governments restrict the use of encryption services, people may hide their messages in other data.

1.2 Watermarking requirements

Each watermarking application has its own specific requirements. Therefore, there is no set of requirements to be met by all watermarking techniques. Nevertheless, some general directions can be given for most of the applications mentioned above:

- **Perceptual transparency:** In most applications the watermarking algorithm must embed the watermark such that this does not affect the quality of the underlying host

data. A watermark-embedding procedure is truly imperceptible if humans cannot distinguish the original data from the data with the inserted watermark [Swa98]. However, even the smallest modification in the host data may become apparent when the original data is compared directly with the watermarked data. Since users of watermarked data normally do not have access to the original data, they cannot perform this comparison. Therefore, it may be sufficient that the modifications in the watermarked data go unnoticed as long as the data are not compared with the original data [Voy98].

- **Payload of the watermark:** The amount of information that can be stored in a watermark depends on the application. For copy protection purposes, a payload of one bit is usually sufficient.
  According to a recent proposal for audio watermarking technology from the International Federation for the Phonographic Industry, (IFPI), the minimum payload for an audio watermark should be 20 bits per second, independently of the signal level and music type [Int97]. However, according to [Pet98a] this minimum is very ambitious and should be lowered to only a few bits per second.
  For the protection of intellectual property rights, it seems reasonable to assume that one wants to embed an amount of information similar to that used for ISBN, International Standard Book Numbering, (roughly 10 digits) or better ISRC, International Standard Recording Code, (roughly 12 alphanumeric letters). On top of this, one should also add the year of copyright, the permissions granted on the work and rating for it [Kut99]. This means that about 60 bits [Fri99a] or 70 bits [Kut99] of information should be embedded in the host data, the image, video-frame or audio fragment.

- **Robustness:** A fragile watermark that has to prove the authenticity of the host data does not have to be robust against processing techniques or intentional alterations of the host data, since failure to detect the watermark proves that the host data has been modified and is no longer authentic. However, if a watermark is used for another application, it is desirable that the watermark always remains in the host data, even if the quality of the host data is degraded, intentionally or unintentionally. Examples of unintentional degradations are applications involving storage or transmission of data, where lossy compression techniques are applied to the data to reduce bit-rates and increase efficiency. Other unintentional quality-degrading processing techniques include filtering, re-sampling, digital-analog (D/A) and analog-digital (A/D) conversion. On the other hand, a watermark can also be subjected to processing solely intended to remove the watermark [Cox97]. In addition, when many copies of the same content exist with different watermarks, as would be the case for fingerprinting, watermark removal is possible because of collusion between several owners of copies. In general, there should be no way in which the watermark can be removed or altered without sufficient degradation of the perceptual quality of the host data so as to render it unusable.

- **Security:** The security of watermarking techniques can be interpreted in the same way as the security of encryption techniques. According to Kerckhoffs [And98], one should assume that the method used to encrypt the data is known to an unauthorized party, and that the security must lie in the choice of a key. Hence a watermarking technique is truly secure if knowing the exact algorithms for embedding and extracting the

watermark does not help an unauthorized party to detect the presence of the watermark [Swa98].

- **Oblivious vs. non-oblivious watermarking:** In some applications, like copyright protection and data monitoring, watermark extraction algorithms can use the original unwatermarked data to find the watermark. This is called *non-oblivious* watermarking [Kut99]. In most other applications, e.g. copy protection and indexing, the watermark-extraction algorithms do not have access to the original unwatermarked data. This renders the watermark extraction more difficult. Watermarking algorithms of this kind are referred to as *public*, *blind* or *oblivious* watermarking algorithms.

The requirements listed above are all related to each other. For instance, a very robust watermark can be obtained by making many large modifications to the host data for each bit of the watermark. However, large modifications in de host data will be noticeable and many modifications per watermark bit will limit the maximum amount of watermark bits that can be stored in a data object. Hence, a trade-off should be found between the different requirements so that an optimal watermark for each application can be developed. The mutual dependencies between the basic requirements are shown in Figure 1.1.

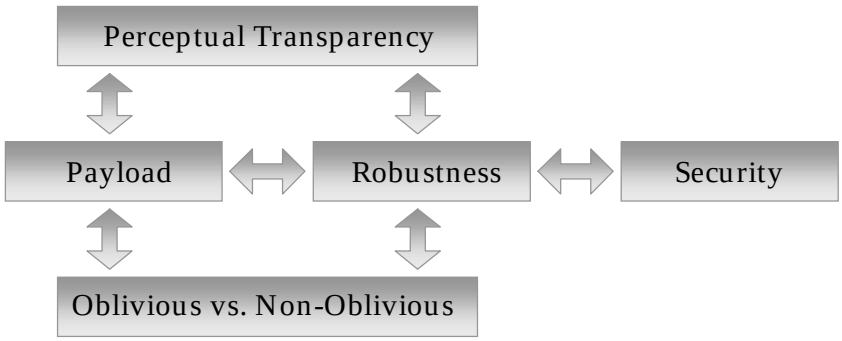


Figure 1.1. Mutual dependencies between the basic requirements.

The relation between the basic requirements for a well-designed secure watermark is represented in Figure 1.2. The security of a watermark influences the robustness enormously. If a watermark is not secure, it cannot be very robust.

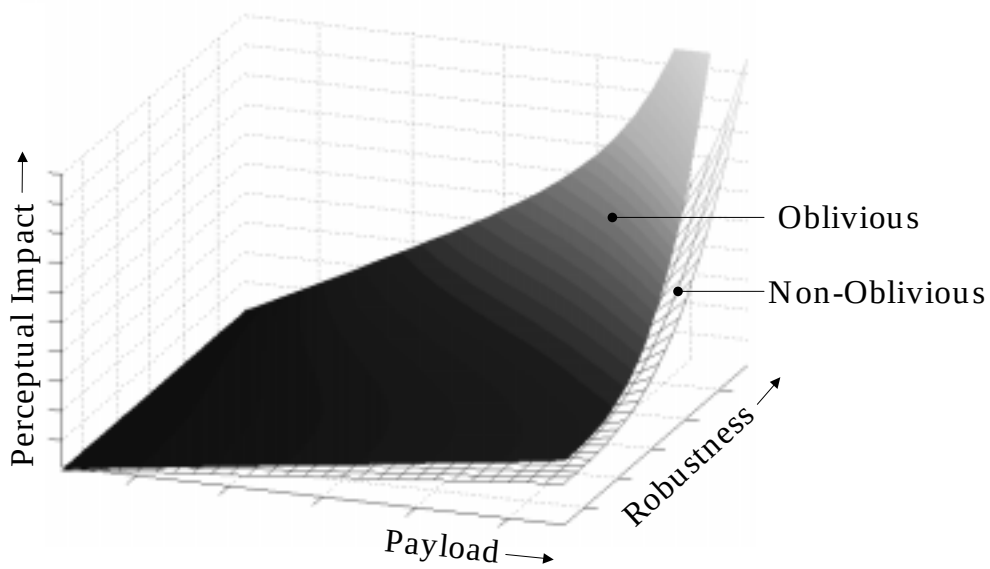


Figure 1.2. Relation between the basic requirements for a secure watermark.

1.3 Brief history of watermarking

Watermarking techniques are not new. Watermarking forms a particular group in the steganography field. *Steganography* stems from the Greek words στεγανος for "covered" and γραφω for "to write", and means covered or secret writing. While classical cryptography is about rendering messages unintelligible to unauthorized persons, steganography is about concealing the existence of the messages. Kahn has traced the roots of steganography to Egypt 4000 years back, where hieroglyphic symbol substitutions were used to inscribe information in the tomb of a nobleman, Khnumhoteb II [Kah67, Swa98].

Herodotus wrote about how the Greeks received a warning of Xerxes' hostile intentions through a message underneath the wax of a writing tablet [Her72]. Another secret writing method he described was to shave the head of a messenger and tattoo a message or image on the messenger's head. After the hair had grown back, the message would be undetectable until the head was shaved again [Joh98, Kob97].

A method suggested by Aenas the Tactician was to mark successive letters in a cover text with secret ink, barely visible pin pricks or small dots and dashes [Kah67]. The marked letters formed the secret message.

Johannes Trithemius (1462-1526), a German monk, was the first who used the term steganography. He encoded letters as religious words in such a way as to turn covert messages into apparently meaningful prayers. As a reward for this artifice the first printing of his manuscript Steganographia in 1606 was placed on the Vatican's prohibited Index and was characterized as "full of peril and superstition" [Kah67, Lea96].


DE
OCCVLTIS
LITERARVM NO-
TIS
SEV
ARTIS ANIMI SENSA
occulte alijs significandi, aut ab alijs si-
gnificata expiscandi enodan-
dique
LIBRI IIII.
IO. BAPTISTA PORTA
NEAPOLITANO AVCTORE.
MONTISBELIGARDI
M. D. XCIII.

Figure 1.3. Title page of Porta's book: *De occultis notis*.

In 1593, Giovanni Baptista Porta published a book about cryptography under the title: *De occultis literarum notis seu artis animemi occulte alijs significadi, aut ab alijs significata expiscandi enodandique. Libri III* (Figure 1.3). In his book, he describes amongst others a method for concealing a secret text message in a cover message by means of a mask. In the following example the secret message can be extracted by ignoring the masked (gray) text [Por93]:

> Honor Militiae tuus suit Carolus pater, nam cum infini to victus est, cum minima exercitu inuitus parte hostis fugit, ac prope ultimum diem iniurius peribit, necabunt Bere illum; atque extemplo puer Arato peribit, res omnes deprehensae bonae si sunt, ante Sillam, & optimo capite non poenitentias amplius decidere sperabit. Vale.

In the 17^{th} century it was not unusual to publish manuscripts anonymously, especially if it concerned the writing of histories. The risk of offending powerful political parties, which could have severe consequences to the author, was far too great. Therefore, Bishop Francis Godwin coded his name as the initial capital letters of each chapter of his manuscript [Lea96]. This is an early example of copyright protection.

An example of embedding copyright or authorship information in musical scores was practiced by Bach, who embedded his name in many of his pieces. For instance, in his organ chorale "Vor deinem Thron", he used null cipher coding by spelling out *B-A-C-H* in notes, where B-flat represents *B*, and B represents *H* or by counting the number of occurrences of a note, one occurrence for *A*, two for *B*, three for *C* and eight for *H* [Swa98].

In World War II steganogaphic techniques were widely used [Kah67, Joh98]. In the USA the post banned a large class of objects that could conceal messages, like chess games,

crosswords and newspaper clippings. Other objects were changed before these were delivered, lovers' Xs were deleted, watch hands were shifted, loose stamps and blank paper were replaced. Censors even rephrased telegrams to prevent that people hid secret messages in normal text messages. In one case, a censor changed "father is dead" to "father is deceased", which resulted in the reply "is father dead or deceased?". Thousands of people were involved in reading mail, looking for language which appeared to be forced. For example, the following message was actually sent by a German spy [Kah67]:

> Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

Extracting the second letter in each word reveals the following message:

> Pershing sails from NY June 1.

During the 1980s steganographic techniques were used for fingerprinting. Prime Minister Margaret Thatcher became so irritated at press leaks of cabinet documents that she had the word processors reprogrammed to encode the user's identity in the word spacing, so that disloyal ministers could be traced [And98].

From this brief history overview we can conclude that most applications mentioned in Section 1.1 are nothing else than variations on the historical ones.

1.4 Scope of this thesis

There are many types of watermarking techniques. The scope of this thesis is the techniques for *real-time embedding of watermarks in and the extraction of watermarks from compressed image and video data*. These watermarking techniques can for instance be used in fingerprinting and copy protection systems for home-recording devices.

- **Fingerprinting:** A consumer can receive digital services, like pay TV or video on demand, by cable or satellite dish using a set-top box and a smart card, which he has to buy and can therefore be related to his identity. To prevent other non-paying consumers to make use of the same services, the service provider encrypts the data, for which he uses one or more keys. This protects the services during transmission. The set-top box in the home of the consumer decrypts the data if a valid smart card is used, and adds a watermark, representing the identity of the user, to the compressed clear data. The fingerprinted data can now be fed to the internal video decoder to view the data or the data can be stored in compressed form.

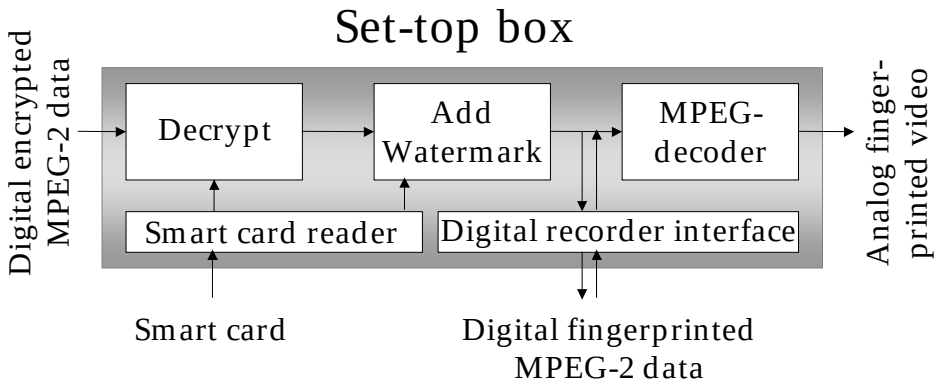


Figure 1.4. Set-top box with fingerprinting capabilities.

The service provider can now identify consumers who supply data to third parties breaking their license agreement. The complete scheme of a set-top box with fingerprinting facilities is depicted in Figure 1.4.

- **Copy protection:** Service providers are reluctant to accept digital recording devices, because of they fear unrestricted copying of services like Pay TV, Pay-Per-View and Video-On-Demand. However, digital video recorders enable consumers to use services on another time than the time the services are actually broadcasted (*time-shifting*), or to insert longer breaks in a movie. A compromise between the conflicting desires of the service providers and the consumers would be the embedding of an SCMS-like [IEC958] copy protection system in each digital recorder [Han96].

  Using the Serial Copy Management System, consumers can make copies of any digital source, but they cannot make copies of copies. An example of an SCMS-like copy protection scheme using watermarking techniques is shown in Figure 1.5.

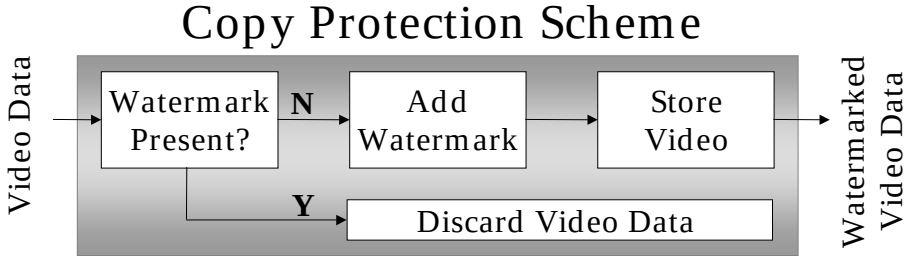


Figure 1.5. A copy protection scheme for digital recorders.

This copy protection system checks all incoming video streams for a predefined copy-prohibit watermark. If such a watermark is found, the incoming video must already have been copied before and is therefore refused by the recorder. If the copy-prohibit watermark is not found, the watermark is embedded and the watermarked video is stored. This means that video data stored on this recorder always contains a watermark and cannot be duplicated if a recorder is used equipped with such a copy protection system.

Besides the basic requirements mentioned in Section 1.2, a watermarking technique should meet the following extra requirements to qualify as a real-time technique for compressed image and video data applicable to recording devices:

- **Oblivious:** It should be possible to extract watermark information without using the original unwatermarked data, since a recorder and a set-top box do not have the original data at their disposal.

- **Low complexity:** There are two reasons why the watermarking techniques cannot be too complex: they are to be processed in real time, and as they are to be used in consumer products, they must be inexpensive. This means that fully decompressing the data, adding a watermark and finally compressing the data is not an option for embedding a watermark.

- **Preserve host data size:** The watermark should not increase the size of the compressed host data. For instance, if the size of a compressed MPEG-video stream increases, transmission over a fixed bit-rate channel can cause problems, the buffers in hardware decoders can run out of space, or the synchronization of audio and video can be disturbed.

Protection systems that make use of watermarking techniques consist in general of a chain of cryptographic techniques. The watermark information can be encrypted first. Subsequently, the processed watermark information is added to the host data by means of embedding techniques. The encryption and embedding techniques use keys; these keys may vary in time. Cryptography protocols have to take care of the key-management problem. In Figure 1.6 the involved fields of cryptography are represented graphically. The subjects of encryption and protocol development are outside the scope of this thesis. The focus is on developing, analyzing and testing the embedding techniques for watermarks.

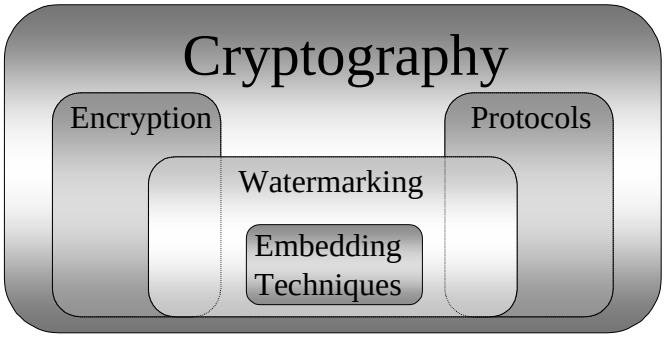


Figure 1.6. Fields of cryptography involved in watermarking applications.

1.5 Outline

This thesis is structured as follows. In *Chapter 2* the state of the art in watermarking techniques for digital image and video data is presented. Since the most commonly used watermarking techniques use additive noise for watermark embedding and correlation techniques for watermark detection, the correlation-based techniques are discussed in full detail here. Various correlation-based techniques are explained for embedding video content dependent or independent watermarks representing one bit, multiple bits or logos in the spatial, Fourier, Discrete Cosine or Discrete Wavelet Transform domain which do or do not use Human Visual System models to maximize the watermark energy. In addition extra measures are discussed that make these watermarks resistant to lossy compression techniques and geometrical transformations. Other non-correlation-based techniques, like least significant bit modification, DCT-coefficient ordering, salient point modification and fractal-based techniques are briefly explained at the end of this chapter. This chapter is partly based on the publication [Lan96a].

In *Chapter 3* the state of the art in real-time watermarking algorithms for compressed video data is discussed. Furthermore, two new algorithms are proposed and evaluated that are computationally highly efficient and very suitable for consumer applications requiring moderate robustness. These real-time watermarking algorithms are based on the basic Least Significant Bit (LSB) modification principle, which is here directly applied to MPEG compressed video streams. Since the watermarking methods discussed in this chapter rely heavily on the MPEG video compression standard, this chapter starts with a brief description of the relevant parts of the MPEG standard. This chapter is partly based on the publications [Lan96b], [Lan97b] and [Lan98a].

In *Chapter 4* the slightly more complex Differential Energy Watermarking (DEW) concept is proposed which is applicable for real-time consumer applications requiring more robustness. The DEW concept is suitable for directly embedding watermarks in and extracting watermarks from MPEG/JPEG or embedded zero tree wavelet encoded video and image data. The DEW algorithm embeds the label bits of the watermark by selectively discarding high frequency coefficients in certain video frame regions. The label bits of the watermark are encoded in the pattern of energy differences between DCT blocks or hierarchical wavelet trees. This chapter is based on the publications [Lan97a], [Lan97b], [Lan98a] and [Lan99b].

Chapter 5 describes how a statistical model is derived and experimentally validated to find optimal parameter settings for the DEW algorithm. The performance of the DEW algorithm has been defined as its robustness against re-encoding attacks, its label size, and its visual impact. We show analytically how the performance is controlled by three embedding parameters. The derived statistical model gives us an expression for the label bit error probability as a function of these three parameters. Using this expression, we show how we can optimize a watermark for robustness, label size or visibility and how we can add adequate error correcting codes to the label bits. This chapter is based on the publications [Lan99b] and [Lan99c].

In *Chapter 6* the DEW algorithm is evaluated. For this purpose, benchmarking approaches for watermarking algorithms and watermark removal attacks described in literature are discussed. Next, the performance of the DEW algorithm for MPEG compressed video data

is compared to a real-time spread spectrum technique for MPEG compressed video data. Finally, the DEW algorithm for JPEG compressed and uncompressed still images is compared to a basic spread spectrum method, which is not specially designed for real-time operation on compressed data. The real-time aspect is neglected in this comparison and for the evaluation the guidelines of the benchmarking methods from literature are followed and the removal attacks are taken into account. This chapter is partly based on the publications [Lan98b] and [Lan98c].

In *Chapter 7* the main results of the LSB and DEW concepts are discussed and directions are given into which further research can take place.

Chapter 2

State of the Art in Watermarking Digital Image and Video Data

2.1 Introduction

In order to embed watermark information in host data, watermark embedding techniques apply minor modifications to the host data in a perceptually invisible manner, where the modifications are related to the watermark information. The watermark information can be retrieved afterwards from the watermarked data by detecting the presence of these modifications.

A wide range of modifications in any domain can be used for watermarking techniques. Prior to embedding or extracting a watermark, the host data can be converted to, for instance, the spatial, the Fourier, the Wavelet, the Discrete Cosine Transform or even the Fractal domain, where the properties of the specific transform domains can be exploited. In these domains modifications can be made like: Least Significant Bit modification, noise addition, coefficient re-ordering, coefficient removal, warping or morphing data parts and block similarities enforcing. Further, the impact of the modifications can be minimized with the aid of Human Visual Models, whereas modifications can be adapted to the anticipated post-processing techniques or to the compression format of the host data.

Since the most commonly used techniques use additive noise for watermark embedding and correlation techniques for watermark detection, we discuss the oblivious correlation-based techniques extensively in this chapter, together with all its possible variations. Other oblivious techniques are briefly explained at the end of this chapter. The cryptographic security of the methods described here lies in the key that is used to generate a pseudorandom watermark pattern or to pseudorandomly select image regions or coefficients to embed the watermark. In general, the robustness of the watermark against processing techniques depends on the embedding depth and the amount of information bits of the watermark.

2.2 Correlation-based watermark techniques

2.2.1 Basic technique in the spatial domain

The most straightforward way to add a watermark to an image in the spatial domain is to add a pseudorandom noise pattern to the luminance values of its pixels. Many methods are

based on this principle [Sch94], [Ben95], [Pit95], [Car95], [Har96], [Lan96a], [Pit96a], [Smi96], [Wol96], [Lan97a], [Wol97], [Zen97], [Fri99b], [Wol98], [Wol99a], [Kal99]. In general, the pseudorandom noise pattern consists of the integers {-1,0,1}, however also floating-point numbers can be used. The pattern is generated based on a key using, for instance, seeds, linear shift registers or randomly shuffled binary images. The only constraints are that the energy in the pattern is more or less uniformly distributed and that the pattern is not correlated with the host image content. To create the watermarked image $I_w(x,y)$ the pseudorandom pattern $W(x,y)$ is multiplied by a small gain factor k and added to the host image $I(x,y)$, as illustrated in Figure 2.2.1.

$$I_w(x,y) = I(x,y) + k \cdot W(x,y) \tag{2.2.1}$$

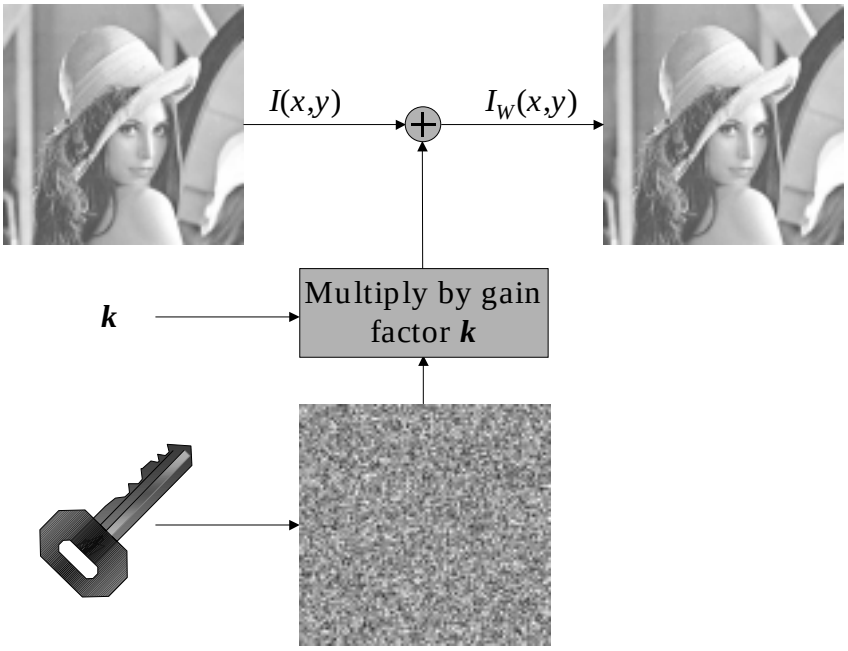


Figure 2.2.1. Watermark embedding procedure.

To detect a watermark in a possibly watermarked image $I'_w(x,y)$ we calculate the correlation between the image $I'_w(x,y)$ and the pseudorandom noise pattern $W(x,y)$. In general, $W(x,y)$ is normalized to a zero mean before correlation. If the correlation R_{XY} exceeds a certain threshold T the watermark detector determines that image $I'_w(x,y)$ contains watermark $W(x,y)$:

$$\begin{aligned} R_{I'_W(x,y)W(x,y)} > T \quad &\rightarrow \quad W(x,y) \text{ detected} \\ < T \quad &\rightarrow \quad \text{No } W(x,y) \text{ detected} \end{aligned} \tag{2.2.2}$$

If $W(x,y)$ only consists of the integers {-1,1} and if the number of –1s equals the number of 1s, we can estimate the correlation as:

$$R_{I'_W(x,y)W(x,y)} = \frac{1}{Z}\sum_{i=1}^{Z} I'_{W_i}(x,y)W_i(x,y) = \frac{1}{Z}\sum_{i=1}^{Z/2} I'_{W_i} W_i^{+} + \frac{1}{Z}\sum_{i=1}^{Z/2} I'_{W_i} W_i^{-}$$
$$= \frac{1}{2}\left\{\mu\left[I'^{+}_{W}(x,y)\right] - \mu\left[I'^{-}_{W}(x,y)\right]\right\} \qquad (2.2.3)$$

Where Z is the number of pixels in the image I'_W, and $^{+,-}$ indicates the set of pixels where the corresponding noise pattern is positive or negative, and $\mu[I'^{+}_{W}(x,y)]$ represents the average value of set pixels in $I'^{+}_{W}(x,y)$. From Equation 2.2.3 it follows that the watermark detection problem corresponds to testing the hypothesis whether two randomly selected sets of pixels in a watermarked image have the same mean.

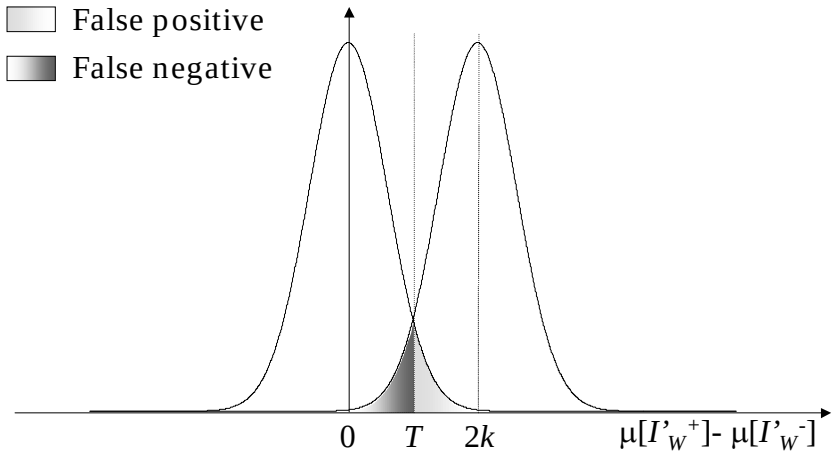


Figure 2.2.2. Watermark detection procedure.

Figure 2.2.2 shows that the watermark detector can make two types of errors. In the first place, it can detect the existence of a watermark, although there is none. This is called a false positive. In the second place, the detector can reject the existence of the watermark, even though there is one. This is called a false negative. In [Kal98a] the probabilities of these two types of errors are derived based on a first-order autoregressive image model:

$$P_{fp} = \frac{1}{2}erfc\left(\frac{T\sqrt{Z}}{\sigma_W\sigma_I\sqrt{2}}\right) \quad \text{and} \quad P_{fn} = \frac{1}{2}erfc\left(\frac{(\sigma_W^2 - T)\sqrt{Z}}{\sigma_W\sigma_I\sqrt{2}}\right) \qquad (2.2.4)$$
$$\text{where} \quad erfc(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-t^2/2}dt$$

Here, σ_W^2 represents the variance of the watermark pixels and σ_I^2 denotes the variance of the image pixels. If the watermark pattern $W(x,y)$ only consists of the integers {-1,1} and the number of –1s equals the number of 1s, the variance of the watermark σ_W^2 equals k^2. The errors P_{fp} and P_{fn} can be minimized by increasing the gain factor k. However, using larger values for the gain factor decreases the visual quality of the watermarked image.

Since the image content can interfere with the watermark, especially in the low-frequency components, the reliability of the detector can be improved by applying matched filtering

before correlation [Dep98], [Sch94], [Lan96a]. This decreases the contribution of the original image to the correlation. For instance, a simple edge-enhance FIR filter F_{edge} can be used, where F_{edge} is given by the following convolution kernel:

$$F_{edge} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix} / 2 \qquad (2.2.5)$$

The experimental results presented in the next section show that applying this filter before correlation reduces the error probability significantly, even when the visual quality of the watermarked image was affected seriously before correlation [Lan96a], [Lan97a].

2.2.2 Extensions to embed multiple bits or logos in one image

From the watermark detector's point of view, an image I can be regarded as Gaussian noise, which distorts the watermark information W. Further, the watermarked image I_W can be seen as the output of a communication channel subject to Gaussian noise over which the watermark information is transmitted. In this case, reliable transmission of the watermark is theoretically possible if its information rate does not exceed the channel capacity, which is given by [Sha49]:

$$C_{ch} = W_b \log_2\left(1 + \frac{\sigma_W^2}{\sigma_I^2}\right) \qquad \text{bit/pixel} \qquad (2.2.6)$$

Here, C_{ch} is given in units of watermark information bits per image pixel and the available bandwidth W_b is equal to 1 cycle per pixel. However, for practical systems a tighter empirically lower bound can be determined [Smi96]:

$$C_{ch} = W_b \log_2\left(1 + \frac{\sigma_W^2}{\alpha \cdot \sigma_I^2}\right) \text{ bit/pixel} \qquad (2.2.7)$$

Here, α is a small headroom factor, which is larger than 1 and typically around 3. Since the signal-to-noise ratio σ_W^2/σ_I^2 is significantly smaller than 1, Equation 2.2.7 can be approximated by:

$$C_{ch} \approx \frac{1}{\ln 2}\left(\frac{\sigma_W^2}{\alpha \cdot \sigma_I^2}\right) \text{ bit/pixel} \qquad (2.2.8)$$

According to this equation, it should be possible to store much more information in an image than just 1 bit using the basic technique described in the previous section. For instance, a watermark consisting of the integers $\{-k,k\}$ added to the 512x512 Lena image (Figure 2.2.1) can carry approximately 50, 200 or 500 bits of information for k=1,2 or 3 respectively and for α=3.

There are several ways to increase the payload of the basic watermarking technique. The simplest way to embed a string of l watermark bits $b_0 b_1 \ldots b_{l-1}$ in an image is to divide the image I into l sub-images $I_0 I_1 \ldots I_{l-1}$ and to add a watermark to each sub-image, where each

watermark represents one bit of the string [Smi96], [Lan96a] and [Lan97a]. This procedure is depicted in Figure 2.2.3.

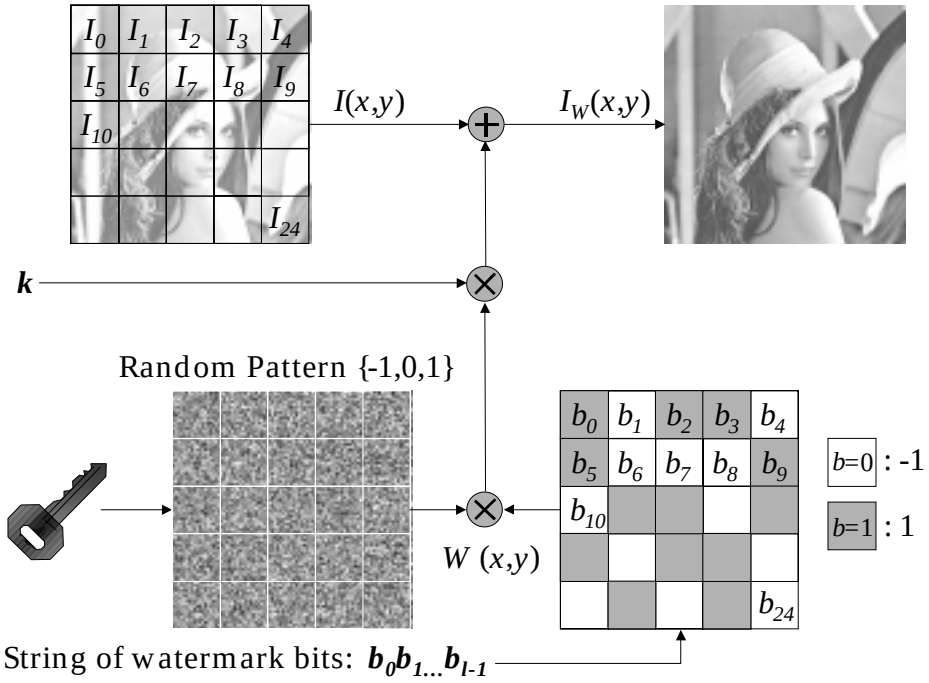


Figure 2.2.3. Watermark bit string embedding procedure.

Using Equation 2.2.8 we can calculate the number of pixels P required per sub-image for reliable detection of a single bit in a sub-image:

$$P \approx \frac{\alpha \sigma_I^2 \ln 2}{\sigma_W^2} \quad \text{pixels} \tag{2.2.9}$$

The watermark bits can be represented in several ways. A pseudorandom pattern can be added if the watermark bit equals one, and the sub-image can be left unaffected if the watermark bit equals zero. In this case, the detector calculates the correlation between the sub-image and the pseudorandom pattern and assigns the value 1 to the watermark bit if the correlation exceeds a certain threshold T; otherwise the watermark bit is assumed to be 0.

The use of a threshold can be circumvented by adding two different pseudorandom patterns RP_0 and RP_1 for watermark bit 0 and 1. The detector now calculates the correlation between the sub-image and the two patterns. The bit value corresponding with the pattern that gives the highest correlation is assigned to the watermark bit. In [Smi96] the two patterns are chosen in such a way that they only differ in sign, $RP_0 = -RP_1$. In this case, the detector only has to calculate the correlation between the sub-image and one of the patterns; the sign of the correlation determines the watermark bit value.

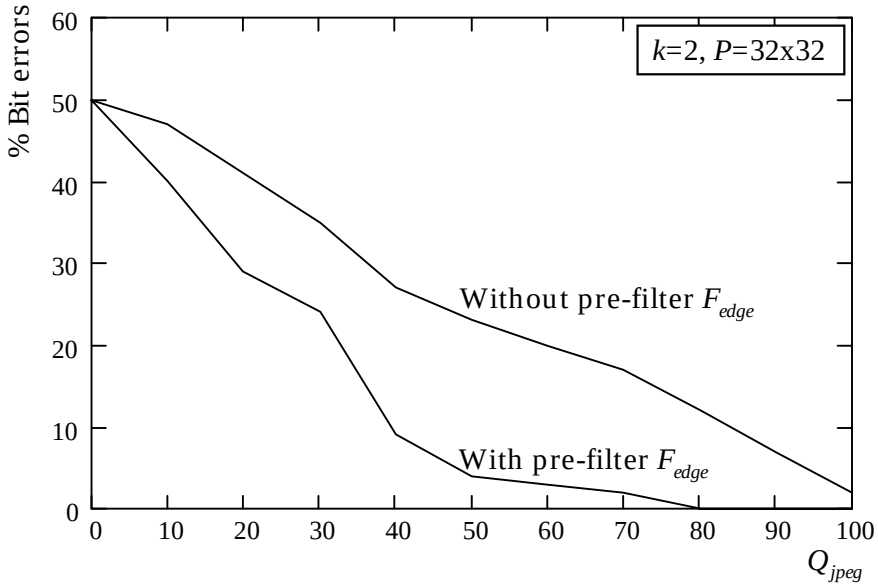


Figure 2.2.4. Watermark detection with and without pre-filtering.

To investigate the effect on the robustness of the watermark of the pre-filter in the detector, the gain factor k, and the number of pixels P per watermark bit, we perform the following experiments. We first add a watermark to an image with the method of [Smi96]. Next, we compress the watermarked image with the JPEG algorithm [Pen93], where the quality factor Q_{jpeg} of the compression algorithm is made variable. Finally, the watermark is extracted from the decompressed image and compared bit by bit with the originally embedded watermark bits. From this experiment, we find the percentages of watermark bit errors due to JPEG compression as a function of the JPEG quality factor.

The first experiment shows the effect of applying the pre-filter given by Equation 2.2.5 before detection of a watermark embedded with a gain factor k=2, and P=32x32 pixels per watermark bit. In Figure 2.2.4 the percentages bit errors caused by JPEG compression are plotted for a detector that uses this pre-filter and for a plain detector. It can clearly be seen that pre-filtering significantly increases the robustness of the watermark.

The second experiment shows the effect of increasing the gain factor k for a watermark embedded with P=32x32 pixels per watermark bit and detected using a pre-filter. From Figure 2.2.5 it follows that the robustness of a watermark can be improved significantly by increasing the gain factor.

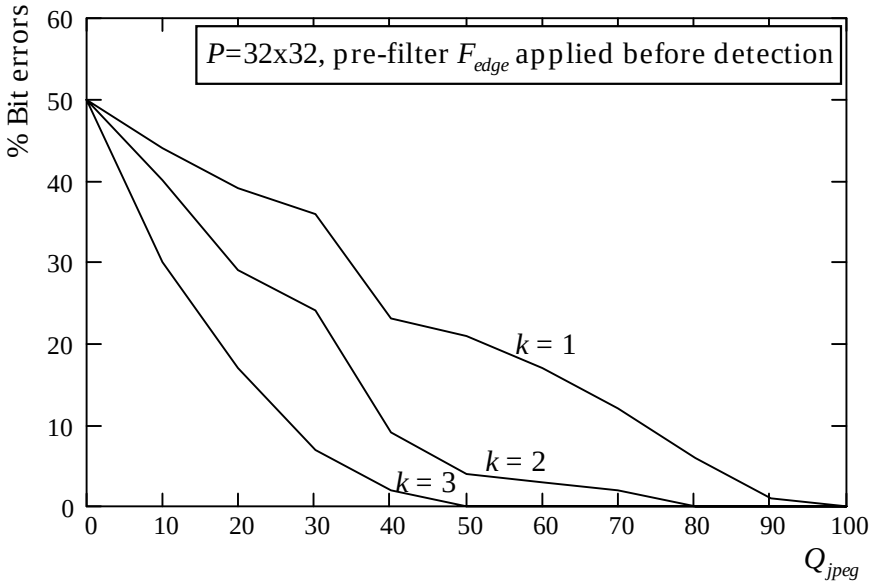


Figure 2.2.5. Influence of the gain factor *k* on the robustness of a watermark.

The third experiment shows the influence of the number of pixels *P* per watermark bit on the robustness of a watermark embedded with a gain factor *k*=2 and detected using a pre-filter. From Figure 2.2.6 it follows that decreasing the payload of the watermark by increasing *P* improves the robustness significantly.

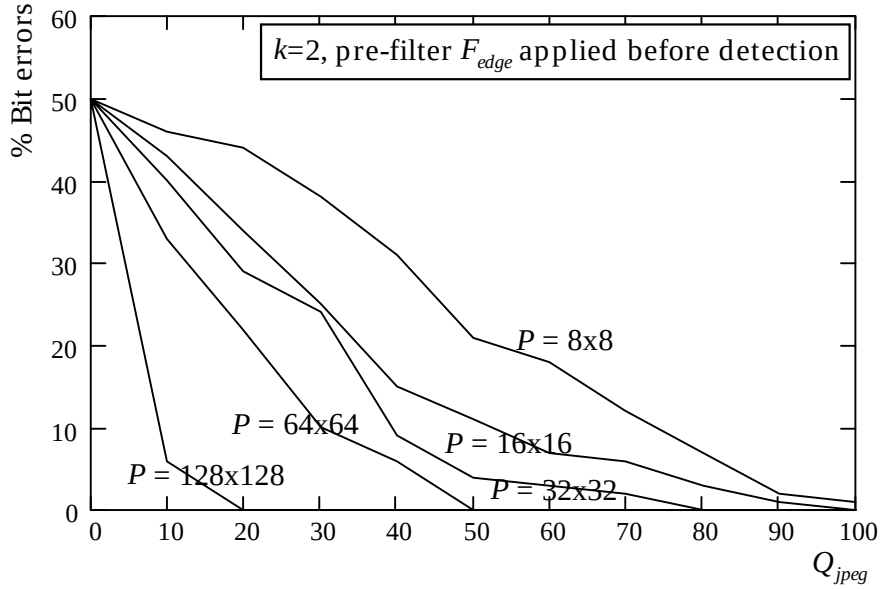


Figure 2.2.6. Influence of the number of pixels per watermark bit *P* on the robustness of a watermark.

Another way to increase the payload of the basic watermarking technique is the use of Direct Sequence Code Division Multiple Access (DS-CDMA) spread spectrum communications [Rua98a] [Rua98b]. Here, for each bit b_j out of the watermark bit string $b_0 b_1 \ldots b_{l-1}$ a different stochastically independent pseudorandom pattern RP_j is generated that

has the same size as the image. This pattern is dependent on the bit value b_j. Here we use the pattern $+RP_j$ if b_j represents a 0 and $-RP_j$ if b_j represents a 1. The summation of all l random patterns $\pm RP_j$ forms the watermark. Prior to adding the watermark to an image, we can scale the watermark by a gain factor or limit it to a certain small range. An example of the 1-dimensional watermark generation is presented in Figure 2.2.7. This example uses 7 different pseudorandom patterns to embed the 7 watermark bits 0011010.

```
RP0:-1 1 1-1-1 1-1-1 1 1-1  b0:0  →  +RP0:-1 1 1-1-1 1-1-1 1 1-1
RP1: 1 1-1-1 1-1-1 1 1-1 1  b1:0  →  +RP1: 1 1-1-1 1-1-1 1 1-1 1
RP2: 1-1-1 1-1-1 1 1-1 1-1  b2:1  →  -RP2:-1 1 1-1 1 1-1-1 1-1 1
RP3:-1-1 1-1-1 1 1-1 1-1-1  b3:1  →  -RP3: 1 1-1 1 1-1-1 1-1 1 1
RP4:-1 1-1-1 1 1-1 1-1-1 1  b4:0  →  +RP4:-1 1-1-1 1 1-1 1-1-1 1
RP5: 1-1-1 1 1-1 1-1-1 1 1  b5:1  →  -RP5:-1 1 1-1-1 1-1 1 1-1-1
RP6:-1-1 1 1-1 1-1-1 1 1 1  b6:0  →  +RP6:-1-1 1 1-1 1-1-1 1 1 1 +
                                      W  :-3 5 1-3 1 3-7 1 3-1 3
```

Figure 2.2.7. Example of a CDMA watermark generation for 7 bits $b_0b_1\ldots b_7$.

Each bit b_j out of the watermark bit string $b_0b_1\ldots b_{l-1}$ can be extracted by calculating the correlation between the normalized image I'_W and the corresponding pseudorandom pattern RP_j. If the correlation is positive, the value 0 is assigned to the watermark bit, otherwise the watermark bit is assumed to be 1. Figure 2.2.8 shows as an example the extraction of the embedded watermark bits in Figure 2.2.7.

```
W    : -3   5  1 -3  1  3 -7  1  3 -1  3
I    : 98  98 97 98 97 96 97 96 95 94 94 +
Iw   : 95 103 98 95 98 99 90 97 98 93 97

                    E[(RP0-E[RP0])·(Iw-E[Iw])] = +15.6 → b0=0
                    E[(RP1-E[RP1])·(Iw-E[Iw])] = +16.4 → b1=0
                    E[(RP2-E[RP2])·(Iw-E[Iw])] = -26.4 → b2=1
                    E[(RP3-E[RP3])·(Iw-E[Iw])] = - 3.1 → b3=1
                    E[(RP4-E[RP4])·(Iw-E[Iw])] = +21.6 → b4=0
                    E[(RP5-E[RP5])·(Iw-E[Iw])] = -23.6 → b5=1
                    E[(RP6-E[RP6])·(Iw-E[Iw])] = + 0.4 → b6=0
```

Figure 2.2.8. Example of CDMA watermark extraction, compare to Figure 2.2.7.

Both ways of extending the watermark payload have their advantages and disadvantages. If each watermark bit has its own image tile, there is no interference between the bits and only a small number of multiplications are required to calculate the correlations. However, if the image is cropped, the watermark bits located at the border are lost. If CDMA techniques are used, the probability that all bits can be recovered after cropping the image is high. However, the watermark bits may interfere with each other and many multiplications are required to calculate the correlations, since each bit is completely spread over the image.

The watermark bits embedded using the methods mentioned above can represent anything: copyright messages, serial numbers, plain text, control signals etc. The content represented by these bits can be compressed, encrypted and protected by error correcting codes. In

some cases it may be more useful to embed a small logo instead of a bit string as a watermark. If the watermarked image is distorted, the watermark logo will also be affected. But now the sophisticated pattern-recognition capabilities of the human visual system can be exploited to detect the logo [Bra97], [Hsu96], [Voy96]. For instance, we can embed a binary watermark logo with 128x32 pixels in an image with 512x512 pixels using the techniques described in this section. Each logo pixel is embedded in an image tile of 8x8 pixels by adding the pseudorandom pattern $+RP$ or $-RP$ to the image tile for a black or white logo pixel respectively. As an example in Figure 2.2.9 the results are shown of the logos extracted after the watermarked image has been degraded with the lossy JPEG [Pen93] compression algorithm using several quality factors. From Figure 2.2.9 it can be seen that, although it is heavily corrupted, the logo can still be recognized.

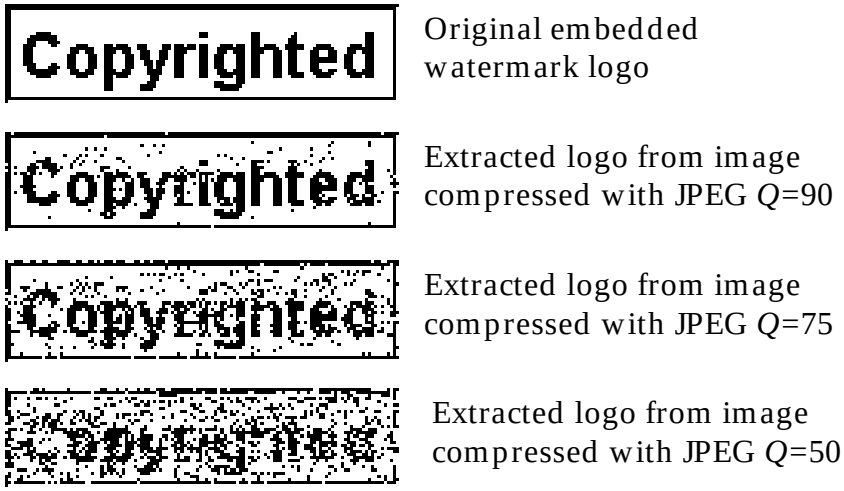


Figure 2.2.9. Extracted watermark logos from a JPEG distorted image.

2.2.3 Techniques for other than spatial domains

The techniques described in the previous section can also be applied in other non-spatial domains. Each transform domain has it own advantages and disadvantages. In [Rua96c] the phase of the Discrete Fourier Transform (DFT) is used to embed a watermark, because the phase is more important than the amplitude of the DFT values for the intelligibility of an image. Putting a watermark in the most important components of an image improves the robustness of the watermark, since tampering with these important image components to remove the watermark will severely degrade the quality of the image. The second reason to use the phase of the DFT values is that it is well known from communications theory that often phase modulation possesses superior noise immunity in comparison with amplitude modulation [Rua96c].

Many watermarking techniques use DFT amplitude modulation because of its translation or shift invariant property [Her98a], [Her98b], [Per99], [Rua96a], [Rua97], [Rua98a], [Rua98b]. Because cyclic translations of the image in the spatial domain do not affect the DFT amplitude, the watermark embedded in this domain will be translation invariant and, in case a CDMA watermark is used, it is even slightly resistant to cropping. Furthermore, the watermark can directly be embedded in the most important middle band frequencies, since modulation of the lowest frequency coefficients results in visible artifacts while the highest frequency coefficients are very vulnerable to noise, filtering and lossy

compression algorithms. Finally the watermark can easily be made image content dependent by modulating the DFT amplitude coefficients $|I(u,v)|$ in the following way [Cox95]:

$$|I_W(u,v)| = |I(u,v)| \cdot (1 + k \cdot W(u,v)) \tag{2.2.10}$$

Here, $W(u,v)$ represents a CDMA watermark, a 2-dimensional pseudorandom pattern, and k denotes the gain factor. Now, the modification of a DFT coefficient is not fixed but proportional to the amplitude of the DFT coefficient. Small DFT coefficients are hardly affected, whereas larger DFT coefficients are affected more severely. This complies with Weber's law [Jai81]. The human visual system does not perceive equal changes in images equally, but visual sensitivity is nearly constant with respect to relative changes in an image. If ΔI is a just noticeable difference, then $\Delta I/I$ = constant. Rewriting Equation 2.2.10 gives:

$$\frac{|I_W(u,v)| - |I(u,v)|}{|I(u,v)|} = \frac{\Delta I(u,v)}{|I(u,v)|} = k \cdot W(u,v) \cong \text{constant} \tag{2.2.11}$$

Since the watermark is here mainly embedded in the larger DFT coefficients, the perceptually most significant components of the image, the robustness of the watermark improves.

Note that the symmetry of the Fourier coefficients must be preserved to ensure that the image data is still real valued after the inverse transform to the spatial domain. If the coefficient $|I(u,v)|$ in an image with NxM pixels is modified according to Equation 2.2.10, its counterpart $|I(N-u,M-v)|$must be modified in the same way. In Figure 2.2.10b an example is given of an image in which a watermark is embedded using all DFT amplitude coefficients according to Equation 2.2.10 and using a relatively small gain factor k. Figure 2.2.10c presents the strongly amplified difference between the original image and the watermarked image. Figure 2.2.10d shows an image watermarked using a large value for the gain factor k.

(a) Original image

(b) Watermarked image

Figure 2.2.10. Fourier Amplitude Watermark.

(c) Difference $W(x,y)=I-I_W$ scaled for visibility

(d) Heavily watermarked image

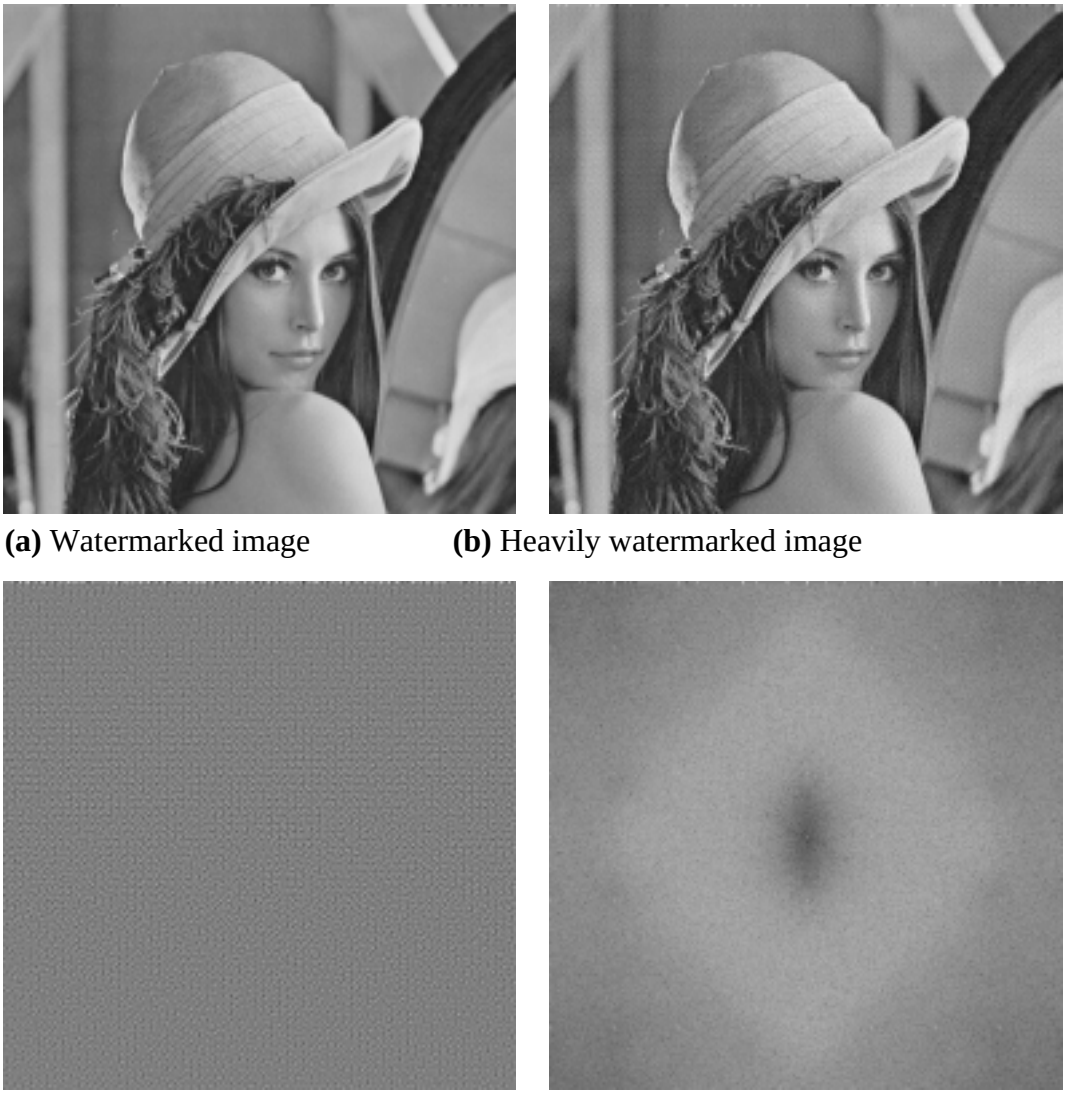
Figure 2.2.10. Fourier Amplitude Watermark.

Another commonly used domain for embedding a watermark is the Discrete Cosine Transform (DCT) domain [Bol95], [Cox95], [Cox96a], [Cox96b], [Hsu96], [Piv97], [Pod97], [Tao97], [Rua96b], [Wol99c]. Using the DCT an image can easily be split up in pseudo frequency bands, so that the watermark can conveniently be embedded in the most important middle band frequencies. Furthermore, the sensitivity of the human visual system (HVS) to the DCT basis images has been extensively studied, which resulted in a default JPEG quantization table [Pen93]. These results can be used for predicting and minimizing the visual impact of the distortions caused by the watermark. Finally, the block-based DCT is widely used for image and video compression. By embedding a watermark in the same domain we can anticipate lossy compression and exploit the DCT decomposition to make real-time watermark applications.

In Figure 2.2.11a an example is given of an image in which a 2-dimensional CDMA watermark W is embedded in the 8x8 block DCT middle band frequencies. The 8x8 DCT coefficients $F(u,v)$ are modulated according to the following Equation:

$$I_{W_{x,y}}(u,v) = \begin{cases} I_{x,y}(u,v) + k \cdot W_{x,y}(u,v) & u,v \in F_M \\ I_{x,y}(u,v) & u,v \notin F_M \end{cases} \qquad x,y=1,8,16\ldots \qquad (2.2.12)$$

Here F_M denotes the middle band frequencies, k the gain factor, (x,y) the spatial location of an 8x8 pixel block in image I and (u,v) the DCT coefficient in the corresponding 8x8 DCT block (Figure 2.2.12).



(a) Watermarked image **(b)** Heavily watermarked image

(c) Difference $W(x,y)=I(x,y)-I_w(x,y)$ **(d)** Fourier Spectrum $W(u,v)$

Figure 2.2.11. 8x8 DCT middle band image content independent watermark.

In Figure 2.2.11c the strongly amplified difference between the original image and the watermarked image is presented. Figure 2.2.11d shows the Fourier Spectrum of the watermark. Here, it can clearly be seen that watermark only affects the middle band frequencies.

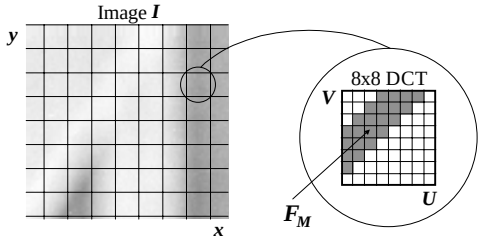


Figure 2.2.12. Definition of the middle band frequencies in a DCT block.

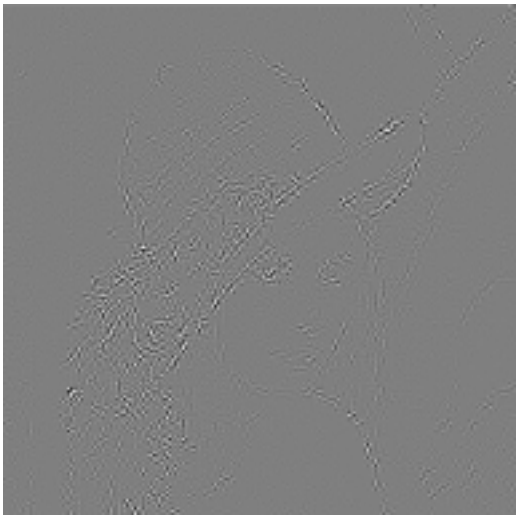
The watermark can be made image dependent by changing the modulation function to:

$$I_{W_{x,y}}(u,v) = \begin{cases} I_{x,y}(u,v) \cdot (1 + k \cdot W_{x,y}(u,v)) & u,v \in F_M \\ I_{x,y}(u,v) & u,v \notin F_M \end{cases} \quad x,y=1,8,16,\ldots \qquad (2.2.13)$$

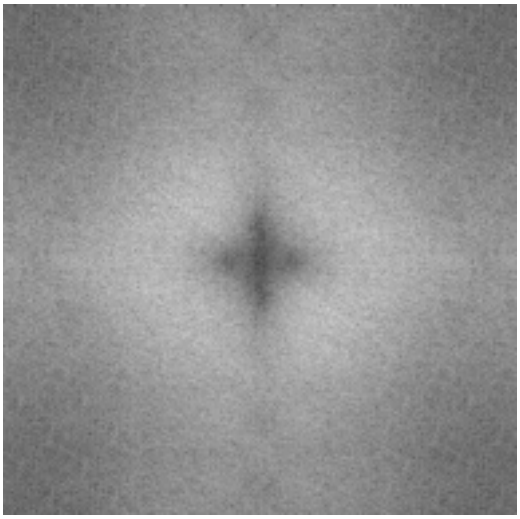
If this modulation function is applied, the results from Figure 2.2.11 change into the results shown in Figure 2.2.13. From Figure 2.2.13b and c it appears that most distortions introduced by the watermark are located around the edges and in the textured areas.

(a) Watermarked image

(b) Heavily watermarked image



(c) Difference $W(x,y)=I(x,y)-I_w(x,y)$



(d) Fourier Spectrum $W(u,v)$

Figure 2.2.13. 8x8 block DCT middle band image content dependent watermark.

If watermarking techniques can exploit the characteristics of the Human Visual System (HVS), it is possible to hide watermarks with more energy in an image, which makes watermarks more robust. From this point of view the Digital Wavelet Transform (DWT) is a very attractive tool, because it can be used as a computationally efficient version of the frequency models for the HVS [Bar99]. For instance, it appears that the human eye is less sensitive to noise in high resolution DWT bands and in the DWT bands having an

orientation of 45° (i.e. *HH* bands). Furthermore, DWT image and video coding, such as embedded zero-tree wavelet (EZW) coding, will be included in the up-coming image and video compression standards, such as JPEG2000 [Xia97]. By embedding a watermark in the same domain we can anticipate lossy EZW compression and exploit the DWT decomposition to make real-time watermark applications. Many approaches apply the basic techniques described at the beginning of this section to the high resolution DWT bands, LH_1, HH_1 and HL_1 (Figure 2.2.14) [Bar99], [Bol95], [Kun97], [Rua96b], [Xia97].

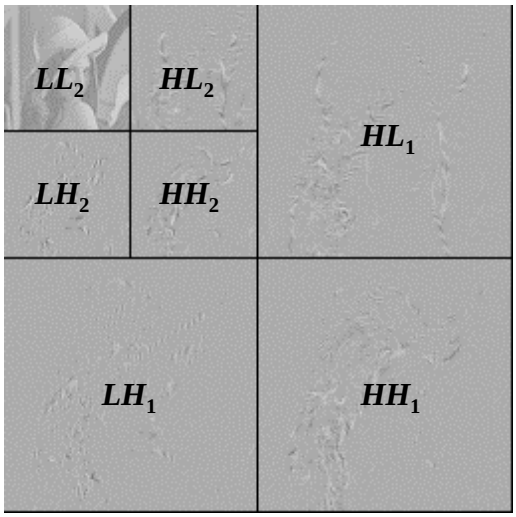


Figure 2.2.14. DWT 2-level decomposition of an image.

In Figure 2.2.15a an example is given of an image in which a 2-dimensional CDMA watermark W is embedded in the LH_1, HH_1 and HL_1 DWT bands using a large gain factor k. The DWT coefficients in each of the three DWT bands are modulated as follows:

$$I_W(u,v) = I(u,v) + k \cdot W(u,v) \tag{2.2.14}$$

Figure 2.2.15b shows the strongly amplified difference between the original image and the watermarked image.

(a) Heavily watermarked image **(b)** Difference $W(x,y)=I(x,y)-I_w(x,y)$

Figure 2.2.15. DWT image content independent watermark.

The DWT watermark can be made image dependent by modulating the DWT coefficients in each of the three DWT bands as follows:

$$I_W(u,v) = I(u,v) \cdot (1 + k \cdot W(u,v)) \tag{2.2.15}$$

In Figure 2.2.16a an example is given of an image in which the same CDMA watermark W is embedded in the LH_1, HH_1 and HL_1 DWT bands using Equation 2.2.15 with a large gain factor k. Figure 2.2.16b shows the strongly amplified difference between the original image and the watermarked image.

(a) Heavily watermarked image **(b)** Difference $W(x,y)=I(x,y)-I_w(x,y)$

Figure 2.2.16. DWT image content dependent watermark.

2.2.4 Watermark energy adaptation based on HVS

The robustness of a watermark can be improved by increasing the energy of the watermark. However, increasing the energy degrades the image quality. By exploiting the properties of the Human Visual System (HVS), the energy can be increased locally in places where the human eye will not notice it. As a result, by exploiting the HVS, one can embed perceptually invisible watermarks that have higher energy than if this energy were to be distributed evenly over the image.

If a visual signal is to be perceived, it must have a minimum amount of contrast, which depends on its mean luminance and frequency. Furthermore, a signal of a given frequency can mask a disturbing signal of a similar frequency [Wan95] and [Bar98]. This masking effect is already used in the image-dependent DCT watermarking method described in the previous section, where the DCT-coefficients are modulated by means of Equation 2.2.13. Here, to each sinusoid present in the image (masking signal), another sinusoid

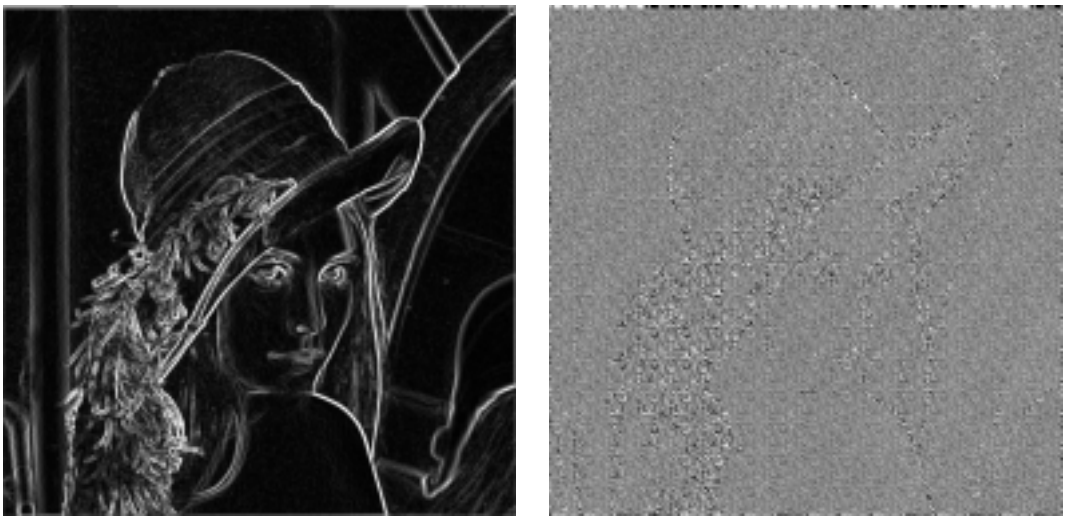
(watermark) is added, having an amplitude proportional to the masking signal. If the gain factor k is properly set, frequency masking occurs.

The HVS is less sensitive to changes in regions of high luminance. This fact can be exploited by making the watermark gain factor luminance dependent [Kut97]. Furthermore, since the human eye is least sensitive to the blue channel, a perceptually invisible watermark embedded in the blue channel can contain more energy than a perceptually invisible watermark embedded in the luminance channel of a color image [Kut97].

Around edges and in textured areas of an image, the HVS is less sensitive to distortions than in smooth areas. This effect is called spatial masking and can also be exploited for watermarking by increasing the watermark energy locally in these masked image areas [Mac95]. The basic spatial watermarking techniques described in Sections 2.2.1 and 2.2.2 can be extended with spatial masking compensation by, for instance, using the following modulation function.

$$I_W(x,y) = I(x,y) + Msk(x,y) \cdot k \cdot W(x,y) \tag{2.2.16}$$

Here $W(x,y)$ represents the 2-dimensional pseudorandom pattern of the watermark, k denotes the fixed gain factor and $Msk(x,y)$ represents a masking image. The values of the masking image range from 0 to k'_{max} and give a measure of insensitivity to distortions for each corresponding point in the original image $I(x,y)$. In [Kal99] the masking image Msk is generated by filtering the original image with a Laplacian high-pass filter and by taking the absolute values of the resulting filtered image.



(a) Masking image **(b)** Difference $W(x,y)=I(x,y)-I_W(x,y)$

Figure 2.2.17. Watermarking using masking image based on Prewitt operator.

In Figure 2.2.17a a mask is shown for the Lena image (Figure 2.2.10a) which is generated by a simple Prewitt edge detector. Figure 2.2.17b shows the strongly amplified watermark modulated with this mask.

Experiments have shown that a perceptually invisible watermark modulated with a gain factor locally adapted to such a mask can contain twice as much energy as a perceptually invisible watermark modulated with a fixed gain factor.

To investigate the effect of this energy doubling on the robustness of the watermark we perform the following experiment. We add a watermark $W_{\text{fixed}}(x,y)$ to the Lena image with the method of [Smi96] using a fixed gain factor k=2. Increasing this fixed gain factor causes visible artefacts in the resulting watermarked image. Next, we add a watermark $W_{\text{var}}(x,y)$ to another Lena image with the same method, but now we use a variable gain factor locally adapted to the masking image presented in Figure 2.2.17a. Although the watermark $W_{\text{var}}(x,y)$ contains about twice as much energy as $W_{\text{fixed}}(x,y)$ the watermark is not noticeable in the resulting watermarked image. Then we compress both watermarked images with the JPEG algorithm [Pen93], where the quality factor Q_{jpeg} of the compression algorithm is made variable. Finally, the watermarks are extracted from the decompressed image and compared bit by bit with the originally embedded watermark bits. From this experiment, we find the percentages of watermark bit errors due to JPEG compression as a function of the JPEG quality factor. In Figure 2.2.18 the error curves are plotted for both watermarks $W_{\text{fixed}}(x,y)$ and $W_{\text{var}}(x,y)$. It can be seen that the robustness can be slightly improved by applying a variable gain factor adapted to the HVS.

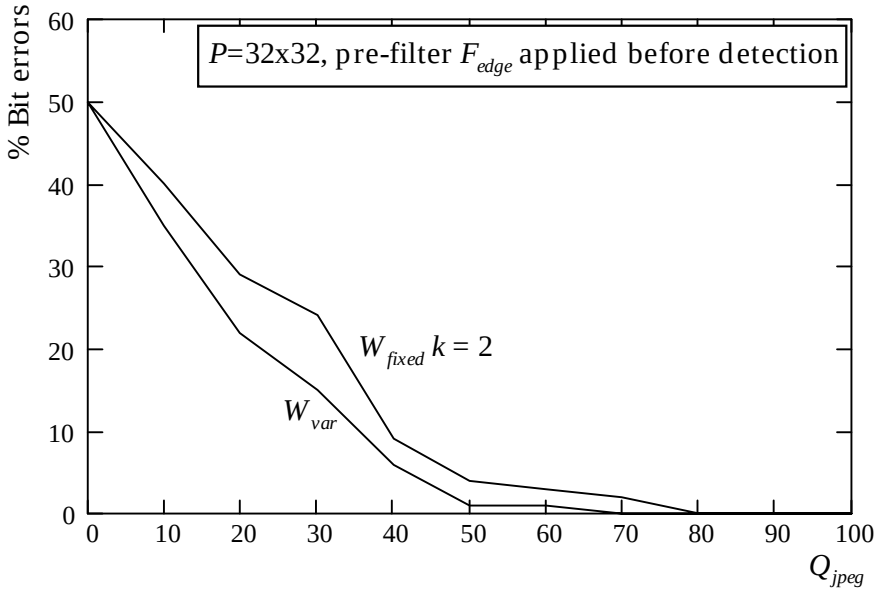
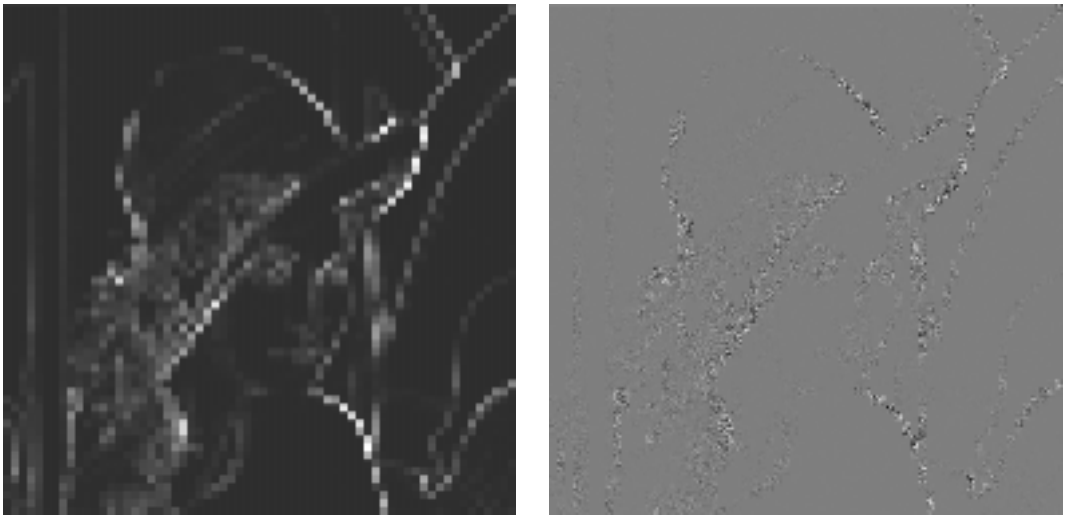


Figure 2.2.18. Influence of a variable gain factor adapted to the HVS on the robustness of a watermark.

In [Ng99] the squared sum of the 8x8 DCT AC-coefficients is used to generate a masking image. Figure 2.2.19a shows a mask generated using this DCT-AC energy for the Lena image. Figure 2.2.19b presents the strongly amplified watermark modulated with this mask.



(a) Masking image **(b)** Difference $W(x,y)=I(x,y)-I_w(x,y)$

Figure 2.2.19. Watermarking where a masking image is used based on DCT-AC energy.

Spatial masking can also be applied if the watermark is embedded in another domain e.g. DFT, DCT or DWT. In this case, the non-spatial watermark is first embedded in an image I, resulting in the temporary image I_{wt}. The watermarked image I_w is now constructed by mixing the original image I and this temporary image I_{wt} by means of a masking image Msk as described above [Bar98] and [Piv97]:

$$I_W(x,y) = \big(1 - Msk(x,y)\big)I(x,y) + Msk(x,y) \cdot I_{Wt}(x,y) \tag{2.2.17}$$

Here the masking image must be scaled to values in the range from 0 to 1. Watermarking methods based on more sophisticated models for the HVS can be found in [Bar98], [Bar99], [Fle97], [Gof97], [Kun97], [Piv97], [Pod97], [Swa96a], [Swa96b], [Wol99b] and [Wol99c].

2.3 Extended correlation based watermark techniques

2.3.1 Anticipating lossy compression and filtering

Watermarks that have been embedded in an image by means of the spatial watermarking techniques described in Sections 2.2.1 and 2.2.2 cannot be detected reliably after the watermarked image has been highly compressed with the lossy JPEG compression algorithm. This is due to the fact that such watermarks consist essentially of low-power, high-frequency noise. Since JPEG allocates fewer bits to the higher frequency components, such watermarks can easily be distorted. Furthermore, these watermarks can also be affected severely by low-pass operations like linear or median filters.

The robustness to JPEG compression can be improved in several ways. In [Smi96] the pseudorandom pattern W is first compressed and then decompressed using the JPEG algorithm. The energy of the resulting pattern W is increased to compensate for the energy lost through the compression. Finally, this pattern is added to the image to generate the

watermarked image. The idea here is to use the compression algorithm to filter out in advance all the energy that would otherwise be lost later in the course of the compression. It is assumed that a watermark formed in this way is invariant to further JPEG compression that uses the same quality factor, except for small numerical artifacts. Analogous pre-distortion of the watermark pattern, such as filtering, can be applied to prevent other anticipated degradations of the watermarked image.

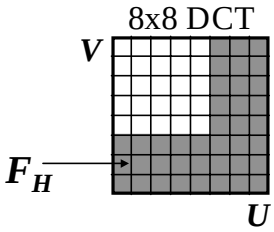


Figure 2.3.1. DCT bands F_H in which the watermark energy Φ is minimized.

In [Nik96] the energy of the watermark pattern is shifted to the lower frequencies by calculating an individual gain factor $k_{x,y}$ for each pixel of the watermark pattern instead of using the same gain factor k for all pixels. First a pseudorandom pattern $W(x,y)$ is generated consisting of the integers 0 and k. Next, the pattern is divided into 8x8 blocks and the DCT transform $W(u,v)$ is calculated for each 8x8 block. The non-zero elements in the 8x8 blocks are now regarded as gain factors $k_{x,y}$ and are adapted in such a way that the energy Φ in the vulnerable high frequency DCT bands F_H is minimized (Figure 2.3.1):

$$\Phi = \sum_{u,v \in F_H} W(u,v)^2 \qquad F_H=\{u,v|\ 5 < u \le 8,\ 5 < v \le 8,\} \tag{2.3.1}$$

The energy Φ is minimized under the following constraints:

$$\sum_{x=1}^{8}\sum_{y=1}^{8} W(x,y)\cdot k = \sum_{x=1}^{8}\sum_{y=1}^{8} W(x,y)\cdot k_{x,y} \tag{2.3.2}$$

$$k_{min} \le k_{x,y} \le k_{max}$$

The effect of this high-energy minimization on the watermark pattern is illustrated in Figure 2.3.2. Figure 2.3.2a shows the watermark pattern within an 8x8 block, where a constant gain factor of k=3 is used. After the high-energy minimization with k_{min}=0 and k_{max}=6 the watermark pattern fades smoothly to zero (Figure 2.3.2.b) although the sum of the non-zero pixels still equals to the sum of the non-zero pixels in the original pattern.

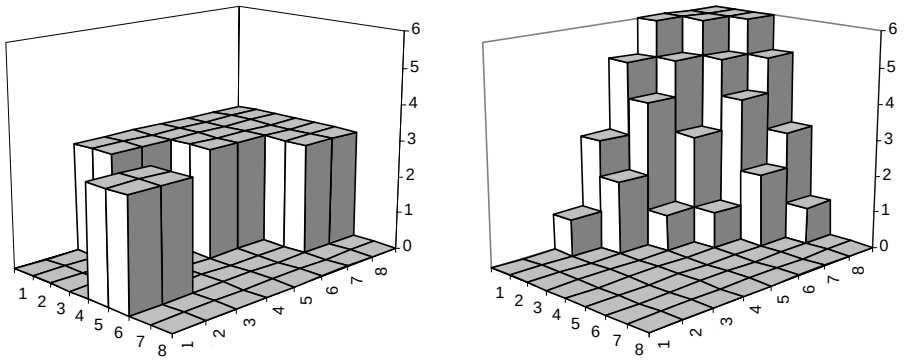


Figure 2.3.2. (a) Original watermark block **(b)** Low frequency watermark block.

In [Lan96a] and [Lan97a] JPEG compression immunity is obtained by deriving a different gain factor *k* for each 32x32 pixel block based on a lower quality JPEG compressed image. A 32x32 pseudorandom pattern representing a watermark bit is added to an 32x32 image tile. A copy of this watermarked image tile is degraded according to the JPEG standard for which end a relatively low quality factor is used. If the watermark bit cannot be extracted correctly from this degraded copy, the watermark pattern is added to the image by means of a higher gain factor and a new degraded copy is formed to check the bit. This procedure is repeated iteratively for each bit until all bits can be extracted reliably from the degraded copies. A watermark formed in this way is resistant to JPEG compression using a quality factor equal to or greater than the quality factor used to degrade the copies. In Figure 2.3.3 an example of such a watermark is shown, amplified for visibility purposes.

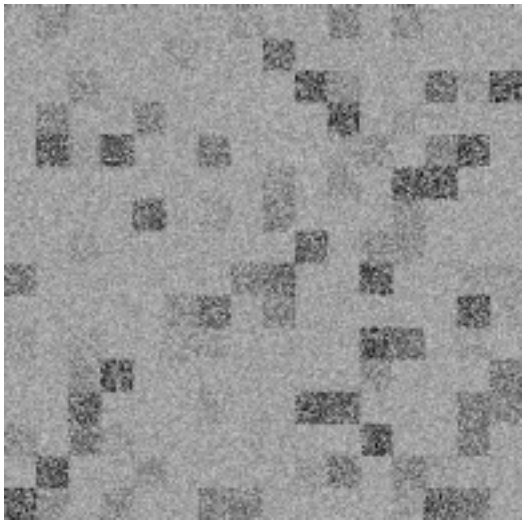


Figure 2.3.3. Watermark where the local gain factor per block is based on a lower quality image.

2.3.2 Anticipating geometrical transforms

A watermark should not only be robust to lossy compression techniques, but also to geometrical transformations such as shifting, scaling, cropping, rotation etc. Geometrical transforms hardly affect the image quality, but they do make most of the watermarks that have been embedded by means of the techniques described in the previous sections

undetectable for the watermark detectors. Since geometrical transforms affect the synchronization between the pseudorandom pattern of the watermark and the watermarked image, the synchronization must be retrieved before the detector performs the correlation calculations.

The most obvious way to achieve shift invariance is using the DFT amplitude modulation technique described in Section 2.2.3. However if for some reason another watermarking embedding domain is preferred and shift invariance is required, a marker can be added in the spatial domain to determine the translation. This marker can be a pseudorandom pattern like the watermark itself. The detector first determines the spatial position of this marker by shifting the marker over all possible locations in the image and calculating the correlation between the marker and the corresponding image part. The translation with the highest correlation defines the spatial position of the marker. Finally, the image is shifted back to its original position and the normal watermarking detection procedure is applied.
An exhaustive search for a marker is computationally quite demanding. Therefore, in [Kal99] a different approach is proposed: adding a pseudorandom pattern twice, but at different locations in the image. The content of the watermark, i.e. the watermark bits, is here embedded in the relative positions of the two watermark patterns. To detect the watermark, the detector computes the phase correlation between the image and the watermark pattern using the fast Fourier transform and it detects the two correlation peaks of the two patterns. The content of the watermark is derived from relative position of the peaks. If the whole image is shifted before detection, the absolute positions of the correlation peaks will change, but the relative positions will remain unchanged, leaving the watermark bits readable for the detector.

In [Fle97] a method is proposed to add a grid to an image that can be used to scale, rotate and shift an image back to its original size and orientation. The grid is represented by a sum of sinusoidal signals, which appear as peaks in the FFT frequency domain. These peaks are used to determine the geometrical distortions.

In [Kut98] a method is proposed which embeds a pseudorandom pattern multiple times at different locations in the spatial domain of an image. The detector estimates the watermark *W'* by applying a high pass filter F_{HP} to the watermarked image:

$$W' = I_W \otimes F_{HP} \qquad F_{HP} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 12 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} / 12 \qquad (2.3.3)$$

Next, the autocorrelation function of the estimated watermark W' is calculated. This function will have peak values at the center and the positions of the multiple embedded watermarks. If the image has undergone a geometrical transformation, the peaks in the

autocorrelation function will reflect the same transformation, and hence provide a grid that can be used to transform the image back to its original size and orientation.

In [Her98a], [Her98b], [Rua97], [Per99], [Rua98a] and [Rua98b] a method is proposed that embeds the watermark in a rotation, scale and translation invariant domain using a combination of Fourier Transforms (DFT) and a Log Polar Map (LPM). Figure 2.3.4 presents a scheme of this watermarking method.

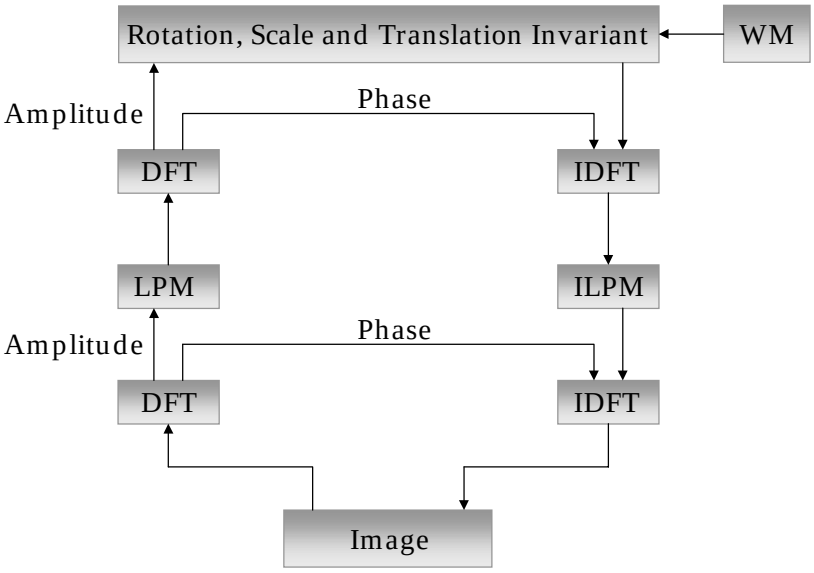


Figure 2.3.4. Rotation, scale and translation invariant watermarking scheme.

First the amplitude of the DFT is calculated to get a translation invariant domain. Next, for every point (u,v) of the DFT amplitude a corresponding point in the Log Polar Map (μ,θ) is determined:

$$u = e^{\mu}\cos(\theta) \qquad v = e^{\mu}\sin(\theta) \tag{2.3.4}$$

This coordinate system of the Log Polar Map converts rotation and scaling into translations along the horizontal and vertical axis. By taking the amplitude of the DFT of this Log Polar map, we obtain a rotation, scale and translation invariant domain. In this domain a CDMA watermark can be added, for instance by modulating the coefficients using Equation 2.2.10.

(a) Original image **(b)** LPM of (a) **(c)** Scaled, rotated **(d)** LPM of (c)

Figure 2.3.5. Example of the properties of the Log Polar Map.

Figure 2.3.5 demonstrates an example of the properties of the Log Polar Map. Figure (b) shows the Log Polar Map of the Lena image (a). Figure (c) depicts a rotated and scaled version of the Lena image and Figure (d) shows its corresponding Log Polar Map. It can clearly be seen that the rotation and scaling are converted into translations.

In practice it has proven to be difficult to implement a watermarking scheme as illustrated in Figure 2.3.4. The authors therefore propose a different approach, where a CDMA watermark is embedded in the translation invariant amplitude DFT domain as described in Section 2.2.3. To make the watermark scale and rotation invariant, they embed a second watermark, a template, in this domain. To extract the watermark, they first determine the scale and orientation of the watermarked image by using the template in the following way:

- The DFT of the watermarked image is calculated.
- The Log Polar Map of the DFT amplitudes and the template pattern is calculated.
- The horizontal and vertical offsets between the two log polar maps are calculated using exhaustive search and cross-correlation techniques, resulting in a scale and rotation factor.

Next, the image is transformed back to its original size and orientation, and the information-carrying watermark is extracted.

2.3.3 Correlation-based techniques in the compressed domain

Not only robustness, but also computational demands play an important role in real-time watermarking applications. In general image data is transmitted in compressed form. To embed a watermark in real time the compressed format must be taken into account, because first decompressing the data, adding a watermark and then re-compressing the data is computationally too demanding. In [Har96], [Har97a], [Har97b], [Har97c] and [Wu97] a method is proposed that adds a DCT transformed pseudorandom pattern directly to selected DCT coefficients of an MPEG compressed video signal. To extract the watermark they decompress the video data and apply the correlation techniques described in Section 2.2. Since the scope of this thesis is real-time watermarking algorithms, the above-mentioned method and novel alternatives are described in full in Chapters 3 and 4.

2.4 Non-correlation-based watermarking techniques

2.4.1 Least significant bit modification

The simplest example of a spatial domain watermarking technique that is not based on correlation is the least significant bit modification method. If each pixel in a gray level image is represented by an 8-bit value, the image can be sliced up in eight bit planes. In Figure 2.4.1 these eight bit planes are represented for the Lena image, where the upper left image represents the most significant bit plane and the lower right image represents the least significant bit plane.

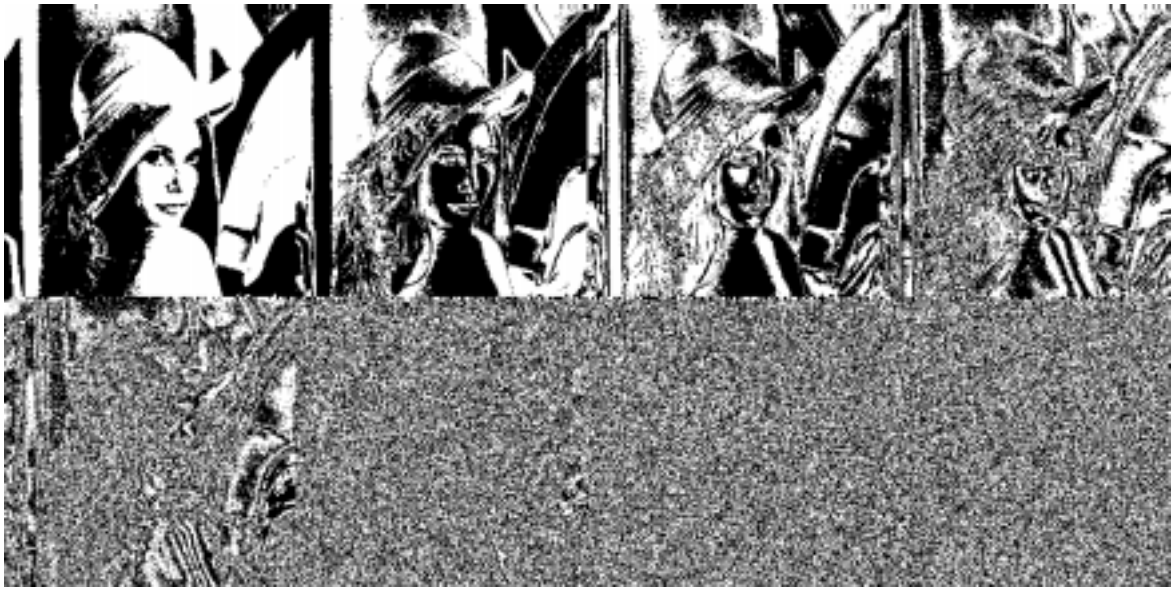


Figure 2.4.1. Bit planes for the Lena image.

Since the least significant bit plane does not contain visually significant information, it can easily be replaced by an enormous amount of watermark bits. More sophisticated watermarking algorithms that make use of LSB modifications can be found in [Sch94], [Aur95], [Aur96], [Hir96] and [Fri99c]. These watermarking techniques are not very secure and not very robust to processing techniques because the least significant bit plane can easily be replaced by random bits, effectively removing the watermark bits.

2.4.2 DCT coefficient ordering

In [Koc95], [Zha95], [Koc94] and [Bur98] a watermarking method is proposed that adds a watermark bit string in the 8x8 block DCT domain. To watermark an image, the image is divided into 8x8 blocks. From these 8x8 blocks the DCT transform is calculated and two or three DCT coefficients are selected in each block in the middle band frequencies F_M (Figure 2.4.2). The selected coefficients are quantized using the default JPEG quantization table [Pen93] and a relatively low JPEG quality factor. The selected coefficients are then adapted in such a way that their magnitudes form a certain relationship. The relationships among the selected coefficients compose 8 patterns (combinations), which are divided into 3 groups. Two groups are used to represent the watermark bits '1' or '0', and the third group represents invalid patterns. If the modifications which are needed to hold a desired pattern become too large, the block is marked as invalid. For example, if a watermark bit with value '1' must be embedded in a block, the third coefficient should have a lower value than the two other coefficients. The embedding process and the list of patterns are represented in Figure 2.4.2.

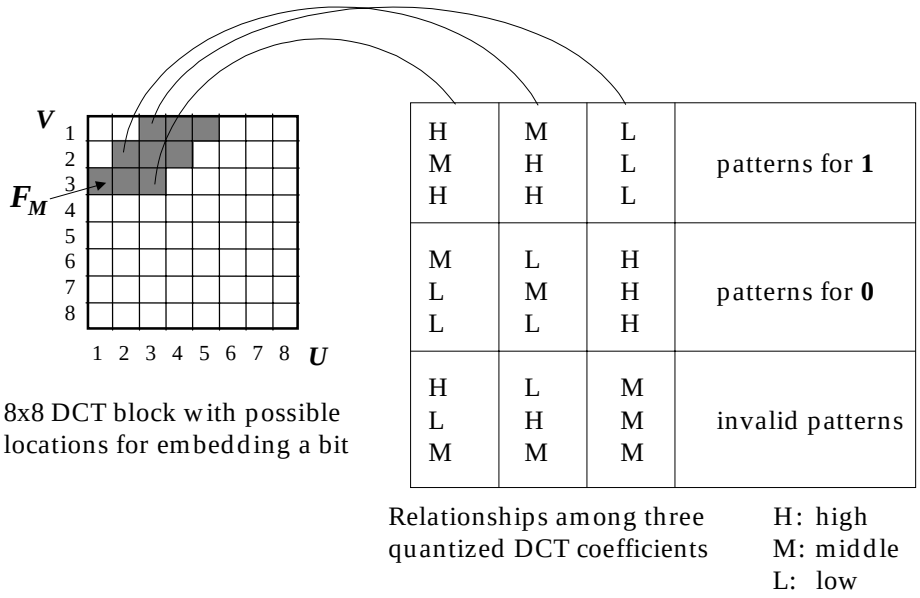


Figure 2.4.2. Watermarking based on adapting relationship between 3 coefficients.

In Figure 2.4.3 the heavily amplified difference between the original Lena image and the watermarked version is shown. In [Bor96a] and [Bor96b] a similar watermarking method is proposed, but here the DCT coefficients are modified in such a way that they fulfill a linear or circular constraint imposed by the watermark code.

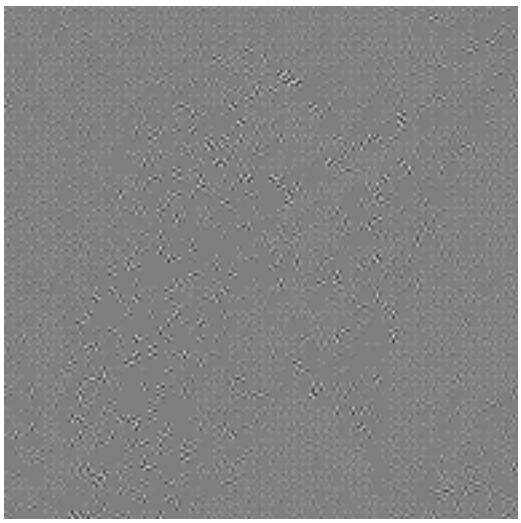


Figure 2.4.3. Watermark $W(x,y)=I(x,y)-I_w(x,y)$ created by adapting relationships between DCT coefficients.

In the methods described here, the relationships between a few middle band coefficients within an 8x8 DCT block define the watermark bits. In [Lan97a], [Lan97b], [Lan98a] and [Lan99b] a method is proposed that uses the relationship between a large amount of high frequency band DCT coefficients in different DCT blocks to define the watermark bits. This new algorithm, its performance and its statistical modeling are described in full in Chapters 4 and 5.

2.4.3 Salient-point modification

In [Ron99] a watermarking method is proposed that is based on modification of salient points in an image. Salient points are defined as isolated points in an image for which a given saliency function is maximal. These points could be corners in an image or locations of high energy for example.

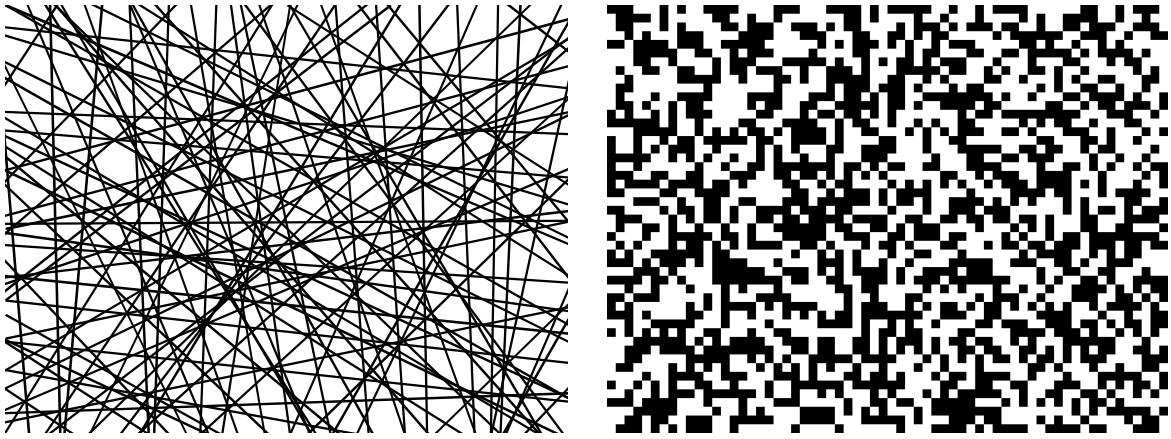


Figure 2.4.4. Examples of watermark patterns for salient-point modification.

To embed a watermark we extract the set of pixels with highest saliency S from the image. Next, a binary pseudorandom pattern $W(x,y)$ with the same dimensions as the image is generated. This can be a line or block pattern as represented in Figure 2.4.4. If this pattern is sufficiently random and covers 50% of all the image pixels, 50% of all salient points in set S will be located on the pattern and 50% off the pattern $W(x,y)$. Finally, the salient points in set S are adapted in such a way that a statistically significant high percentage of them lies on the watermark pattern (i.e. the black pixels in the pattern). There are two ways to adapt the salient points:

- The location of the salient points can be changed by warping the points towards the watermark pattern. In this case small, local geometrical changes are introduced in the image.
- The saliency of the points can be decreased or increased by adding well-chosen pixel patterns to the neighborhood of a salient point.

To detect the watermark we extract the set of pixels with highest saliency S from the image and compare the percentages of the salient points on the watermark pattern and off the pattern. If both percentages are about 50%, no watermark is detected. If there is a statistically significant high percentage of salient points on the pattern, the watermark is detected. The payload of this watermark is 1 bit.

2.4.4 Fractal-based watermarking

Some watermark embedding algorithms are proposed that are based on Fractal compression techniques [Dav96], [Pua96], [Bas98] and [Bas99]. They mainly use block-based local iterated function system coding [Jac92]. We first briefly describe the basic principles of this fractal compression algorithm here. An image is partitioned at two different resolution levels. On the first level, the image is partitioned in range blocks of size nxn. On the second level the image is partitioned in domain blocks of size 2nx2n. For

each range block, a transformed domain block is searched for which the mean square error between the two blocks is minimal. Before the range blocks are matched on the domain blocks, the following transformations are performed on the domain blocks. First, the domain blocks are sub-sampled by a factor two to get the same dimensions as the range blocks. Subsequently, the eight isometries of the domain blocks are determined (the original block and its mirrored version rotated over 0, 90, 180 and 270 degrees). Finally, the scale factor and the offset for the luminance values is adapted. The image is now completely described by a set of relations for each range block, by the index number of the best fitting domain block, its orientation, the luminance scaling and the luminance offset. Using this set of relations, an image decoder can reconstruct the image by taking any initial random image and calculating the content of each range block from its associated domain block using the appropriate geometric and luminance transformations. Taking the resulting image as initial image one repeats this process iteratively until the original image content is approximated closely enough.

In [Pua96] a watermarking technique is proposed which embeds a watermark of 32 bits $b_0 b_1 \ldots b_{31}$ in an image. The embedding procedure consists of the full fractal encoding and decoding process as described above, where the watermark embedding takes place in the fractal encoding process. First, the image $I(x,y)$ is split in two regions $A(x,y)$ and $B(x,y)$. For each watermark bit b_j U range blocks are pseudorandomly chosen from $I(x,y)$. If b_j equals one, the domain blocks to code the U range blocks are searched in region $A(x,y)$. If b_j equals zero, the domain blocks to code the U range blocks are searched in region $B(x,y)$. For range blocks which are not involved in the embedding process, domain blocks are searched in regions $A(x,y)$ and $B(x,y)$. To extract the watermark information, we must select and re-encode the U range blocks for each bit b_j. If most of the best fitting domain blocks are found in region $A(x,y)$, the value 1 is assigned to bit b_j, otherwise the bit is assumed to be zero.

In [Bas98] and [Bas99] a watermark is embedded by forcing range blocks to map exactly on specific domain blocks. The watermark pattern here consists of this specific mapping. This mapping is enforced by adding artificial local similarities to the image. The size of the range blocks may be chosen equal to the size of the domain blocks. In Figure 2.4.5 an example is given of this process.

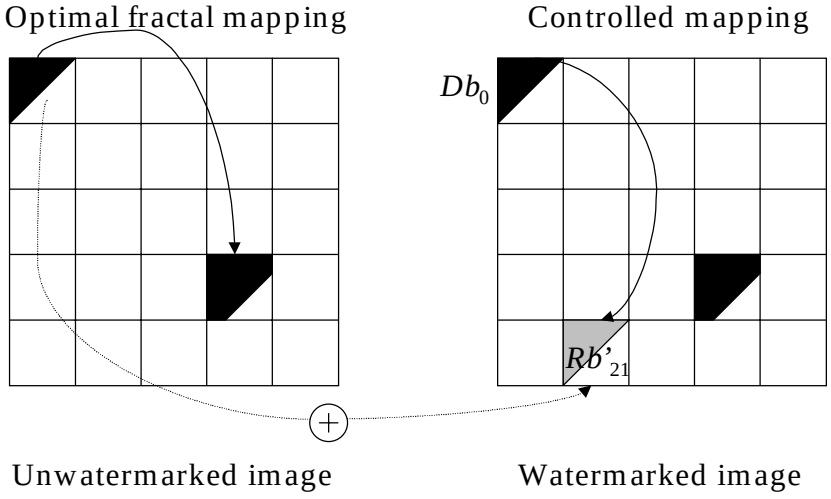


Figure 2.4.5. Modifying the mapping between range and domain blocks.

The left image illustrates how a fractal encoder would map the range block Rb_{18} on domain block Db_0 in an unwatermarked image. To embed the watermark, this mapping $Db_0 \rightarrow Rb_{18}$ must for instance be changed to $Db_0 \rightarrow Rb_{21}$. To force the mapping to this form, a block Rb'_{21} is generated from block Db_0 by changing its luminance values. By adding block Rb' to the image, we change the optimal fractal mapping to its desired form $Db_0 \rightarrow Rb_{21}$, because the quadratic error between Db_0, corrected for luminance scale and offset and Rb_{21} is now smaller than the error between Db_0 and Rb_{18}.

To detect the watermark we calculate the optimal fractal mapping between the range blocks and the domain blocks. If a statistically significant high percentage of the mappings between range blocks and domain blocks match the predefined mappings of the watermark pattern, the watermark is detected.

2.5 Discussion

Not all existing watermarking techniques are discussed in this chapter, because some techniques are specifically designed for e.g. printing purposes, and others are not so extensively represented in literature as the methods described in this chapter. We will therefore only enumerate the most important principles of some of these other methods here:

- For printed images dithering patterns can be adapted to hide watermark information [Tan90] and [Che99].

- Instead of the pixel values, the histogram of an image can be modified to embed a watermark [Col99].

- Quantization can be exploited to hide a watermark. In [Rua96c] a method is proposed in which the pixel values of an image are first coarsely quantized, before some small adaptations are made to the image. To detect these adaptations the watermarked image is subtracted from its coarsely quantized version. In [Kun98] selected wavelet coefficients are quantized using different quantizers for watermark bits 0 and 1.

In this chapter we discussed the two most important classes of watermarking techniques. The first class comprises the correlation-based methods. Here a watermark is embedded by adding pseudorandom noise to image components and detected by correlating the pseudorandom noise with these image components. The second class comprises the non-correlation based techniques. This class of watermarking methods can roughly be divided into two groups: the group based on least significant bit (LSB) modification and group based on geometrical relations.

Chapter 3

Low Complexity Watermarks for MPEG Compressed Video

3.1 Introduction

The scope of Chapters 3, 4 and 5 is on real-time watermarking algorithms for MPEG compressed video. In this chapter the state of the art in real-time watermarking algorithms is discussed and two new computationally highly efficient algorithms are proposed, which are very suitable for consumer applications requiring moderate robustness. In Chapter 4 the slightly more complex DEW watermarking algorithm is proposed which is applicable for applications requiring more robustness. In Chapter 5 a statistical model is derived to find optimal parameter settings for the DEW method.

A real-time watermarking algorithm should meet several requirements. In the first place it should be an oblivious low complexity algorithm. This means that fully decompressing the video data, adding a watermark to the raw video data and finally compressing the data again is not an option for real-time watermark embedding. The watermark should be embedded and detected directly in the compressed stream to avoid computational demanding operations as shown in Figure 3.1.1.

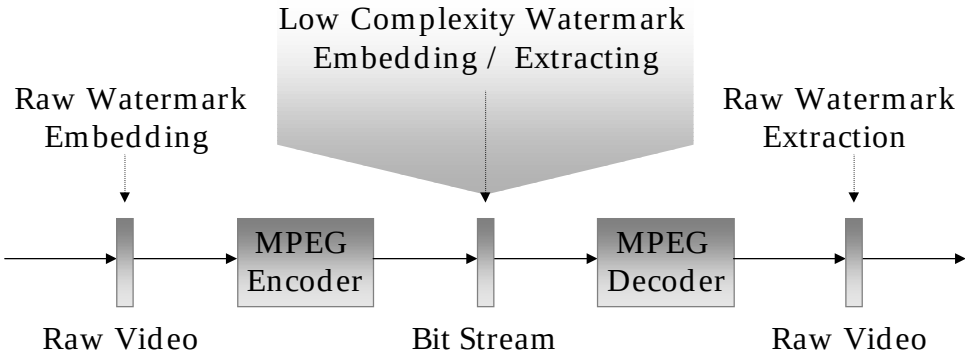


Figure 3.1.1. Watermark embedding / extraction in raw vs. compressed video.

Furthermore, the watermark embedding operation should not increase the size of the compressed video stream. If the size of the stream increases, transmission over a fixed bit-rate channel can cause problems, the buffers in hardware decoders can run out of space, or the synchronization of audio and video can be disturbed.

Since the watermarking methods discussed in the following chapters heavily rely on the MPEG video compression standard [ISO96] the relevant parts of the MPEG-standard and the different domains in which a low complexity watermark can be added are described in Section 3.2. In Section 3.3 an overview is given of two real-time correlation-based watermarking algorithms from literature. In Sections 3.4 and 3.5 two new computationally highly efficient algorithms are proposed, which are very suitable for consumer applications requiring moderate robustness [Lan96b], [Lan97b] and [Lan98a].

3.2 Watermarking MPEG video bit streams

Before we discus the low complexity watermarking techniques, we first briefly describe the MPEG video compression standard [ISO96] itself. The MPEG video bit stream has a layered syntax. Each layer contains one or more subordinate layers as illustrated in Figure 3.2.1. A video *Sequence* is divided into multiple *Group of Pictures* (GOP), representing sets of video frames which are contiguous in display order. Next, the frames are split in slices and macro blocks. The lowest layer, the block layer, is formed by the luminance and chrominance blocks of a macro block.

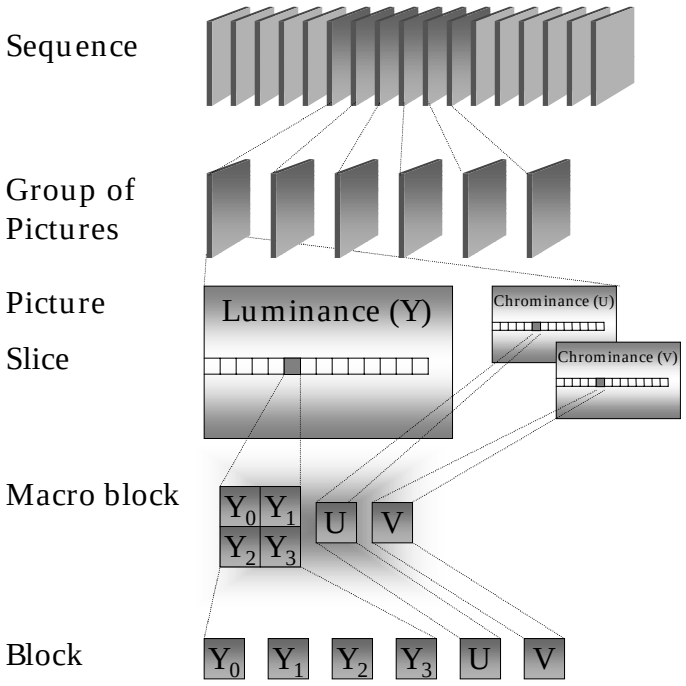


Figure 3.2.1. The layered MPEG syntax.

The MPEG video compression algorithm is based on the basic hybrid coding scheme [Gir87]. As can be seen in Figure 3.2.2 this scheme combines inter (DPCM) and intra frame coding to compress the video data.

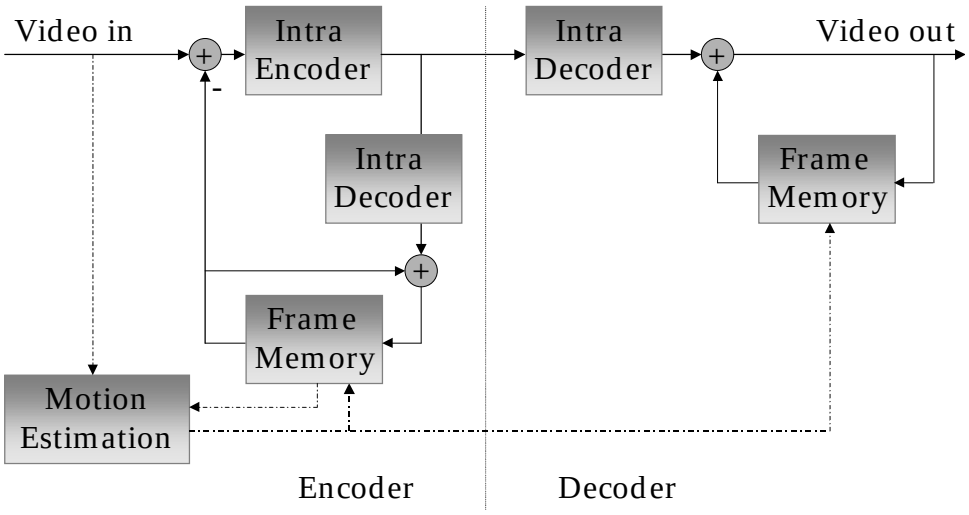


Figure 3.2.2. Motion compensated hybrid coding scheme.

Within a GOP the temporal redundancy among the video frames is reduced by applying temporal DPCM. This means that the frames are temporally predicted by other motion compensated frames. Subsequently, the resulting prediction error, which is called the displaced frame difference, is encoded. Three types of frames are used in the MPEG standard: (I) Intra-frames, which are coded without any reference to other frames, (P) Predicted-frames, which are coded with reference to past I- or P- frames, and (B) Bi-directionally interpolated frames, which are coded with references to both past and future frames. An encoded GOP always starts with an I-frame, to provide access points for random access of the video stream. In Figure 3.2.3 an example of a GOP with 3 frame types and their references is shown.

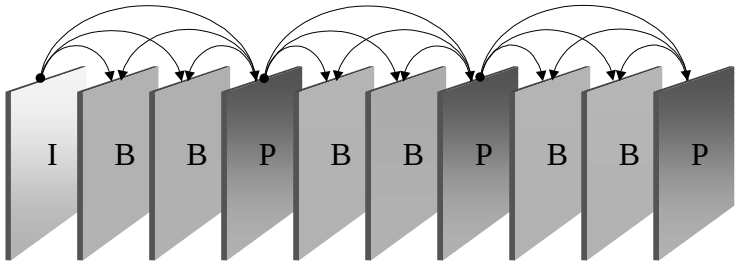


Figure 3.2.3. GOP with 3 frame types and the references between the frames.

The spatial redundancy in the prediction error of the predicted frames and the I-frames, represented by the luminance component *Y* and the chrominance components *U* and *V*, is reduced using the following operations: First the chrominance components *U* and *V* are subsampled. Next, the DCT transform is performed on the 8x8 pixel blocks of the *Y*, *U* and *V* components, and the resulting DCT coefficients are quantized. Since the de-correlating DCT transform concentrates the energy in the lower frequencies, and the human eye is less sensitive to the higher frequencies, the high frequency components can be quantized more coarsely. The DCT coefficient with index (0,0) is called the DC-coefficient, since it represents the average value of the 8x8 pixel block. The other DCT coefficients are called AC-coefficients.

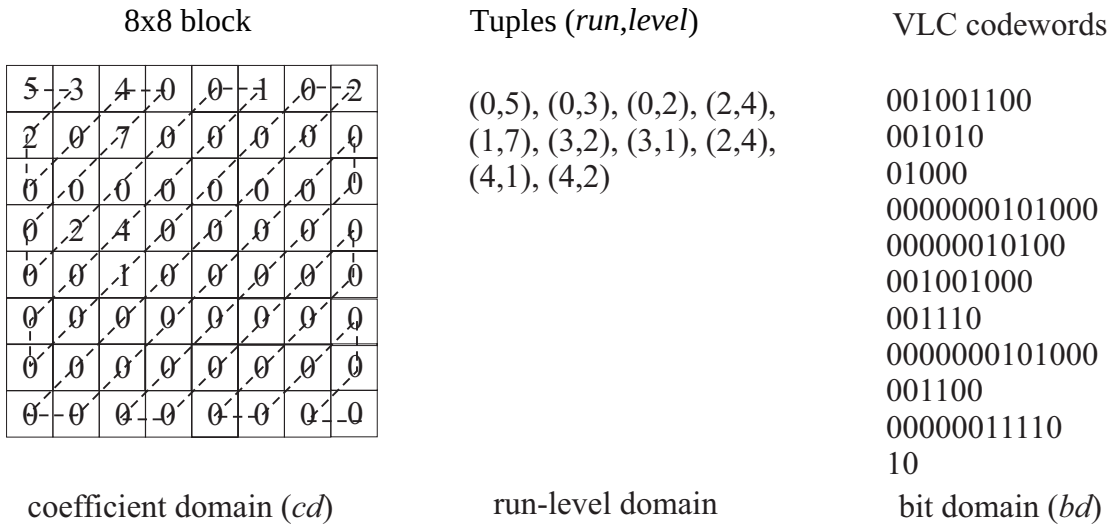


Figure 3.2.4. DCT-block representation domains.

In the lowest MPEG layer, the block-layer, the spatial 8x8 pixel blocks are represented by 64 quantized DCT coefficients. Figure 3.2.4 shows the three domains in which the block layer can be divided. The first domain is the *coefficient domain (cd)*, where a block contains 8x8 integer entries that correspond with the quantized DCT coefficients. Many of the entries are usually zero, especially those entries that correspond with the spatial high frequencies. In the *run-level domain*, the non-zero AC coefficients are re-ordered in a zig-zag scan fashion and are subsequently represented by a (*run,level*) tuple, where the run is equal to the number of zeros preceding a certain coefficient and the level is equal to the value of the coefficient. In lowest level domain, the *bit domain (bd)*, the (*run,level*) tuples are entropy coded and represented by variable length coded (VLC) codewords. The codewords for a single DCT-block are terminated by an end of block (EOB) marker.

A real-time watermarking algorithm for MPEG compressed video should closely follow the MPEG compression standard to avoid computationally demanding operations, like DCT and inverse DCT transforms or motion vector calculation. Therefore, the algorithm should work on the block-layer, the lowest layer of the MPEG stream. A watermarking algorithm that operates on the *coefficient domain* level only needs to perform VLC coding, tuple coding and quantization steps. This process is illustrated in Figure 3.2.5.

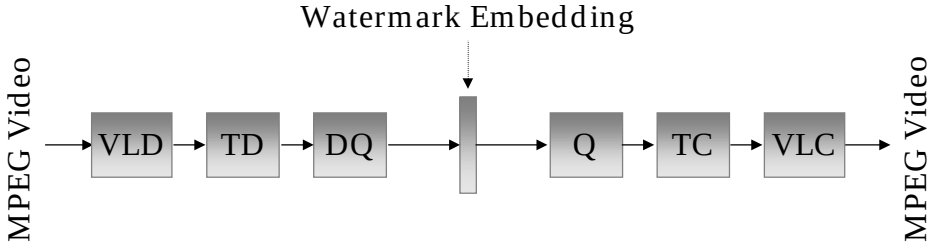


Figure 3.2.5. Coefficient domain watermarking concept.

A watermarking algorithm that operates on the *bit domain* level only needs the VLC coding processing step. Here, a complete watermark embedding procedure can consists of VLC-decoding, VLC-modification and VLC-encoding. This process is illustrated in Figure 3.2.6.

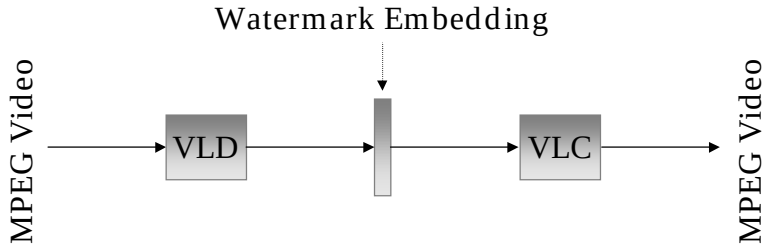


Figure 3.2.6. Bit domain watermarking concept.

In Section 3.3 an overview is given of two real-time correlation-based watermarking algorithms from literature. The first method described in this section is applied in the *coefficient domain*. The second method is more advanced and operates on a slightly higher level than the *coefficient domain*, since it needs a full MPEG decoding operation for drift compensation and watermark detection, and an additional DCT operation. The new watermarking methods proposed in Sections 3.4 and 3.5 operate on the lowest level domain, the *bit domain*, and are therefore the most computationally efficient methods. The DEW algorithm proposed in Chapters 4 and 5 is completely applied in the *coefficient domain*.

3.3 Correlation-based techniques in the coefficient domain

3.3.1 DC-coefficient modification

In [Wu97] a method is proposed that adds a DCT transformed pseudorandom pattern directly to the DC-DCT coefficients of an MPEG compressed video stream. The watermarking process only takes the luminance values of the I-frames into account. To embed a watermark the following procedure is performed: First a pseudorandom pattern consisting of the integers {–1,1} is generated based on a secret a key. This pattern has the same dimensions as the I-frames. Next, the pattern is modulated by a watermark bit string and multiplied by a gain factor as described in Section 2.2.2. Finally, the 8x8 block DCT transform is applied on the modulated pattern and the resulting DC-coefficients are added to the corresponding DC-values of each I-frame. The watermark can be detected using correlating techniques in the DCT domain or in the spatial domain as described in Section 2.2.2.

The authors report that the algorithm decreases the visual quality of the video stream drastically. Therefore, the gain factor of the watermark has to be chosen very low (<1) and the number of pixels per watermark bit has to be chosen extremely high (>> 100,000) to maintain reasonable visual quality for the resulting video stream. This is mainly due to the fact that the watermark pattern is embedded in just one of the 64 DCT coefficients, the DC-component. Furthermore, the pattern consists only of low frequency components to

which the human eye is quite sensitive. For comparison, the algorithm described in Section 2.2.2 uses a gain factor of 2 and about 1000 pixels per watermark bit.

3.3.2 DC- and AC-coefficient modification with drift compensation

3.3.2.1 Basic watermarking concept

In [Har96], [Har97a], [Har97b], [Har97c] and [Har98] a more sophisticated watermarking algorithm is proposed, that embeds a watermark not only in the DC-coefficients, but also in the AC-coefficients of each I-, P- and B-frame. The watermark is here also a pseudorandom pattern consisting of the integers {–1,1} generated based on a secret a key. This pattern has the same dimensions as the video frames. The pattern is modulated by a watermark bit string and multiplied by a gain factor k as described in Section 2.2.2.

To embed the watermark, the watermark pattern $W(x,y)$ is divided into 8x8 blocks. These blocks are transformed to the DCT domain and denoted by $W_{x,y}(u,v)$, where x,y=0,8,16... and u,v=0…7. Next, the 2-dimensional blocks $W_{x,y}(u,v)$ are re-ordered in a zig-zag scan fashion and become arrays $W_{x,y}(i)$, where i=0…63. $W_{x,y}(0)$ represents the DC-coefficient and $W_{x,y}(63)$ denotes the highest frequency AC-coefficient of a 8x8 watermark block. Since the corresponding MPEG encoded 8x8 video content blocks are encoded in the same way as $I_{x,y}(i)$, these arrays can directly be used to add the watermark. For each video block $I_{x,y}(i)$ out of an I-, P-, or B-frame the following steps are performed:

1. The DC-coefficient is modulated as follows:

$$I_{W_{x,y}}(0) = I_{x,y}(0) + W_{x,y}(0) \tag{3.3.1}$$

   Which means that the average value of the watermark block is added to the average value of the video block.

2. To modulate the AC-coefficients the bit stream of the encoded video block is searched VLC-by-VLC for the next VLC code word, representing the next non-zero DCT coefficient. The run and level of this code word are decoded to determine its position i along the zig-zag scan and its amplitude $I_{x,y}(i)$.

   A candidate DCT coefficient for the watermarked video block is generated, which is defined as:

$$I_{W_{x,y}}(i) = I_{x,y}(i) + W_{x,y}(i) \qquad i \neq 0 \tag{3.3.2}$$

   Now the constraint that the video bit-rate may not increase comes into play. The size Sz_I of the VLC needed to encode $I_{x,y}(i)$ and the size Sz_{I_W} of the VLC needed to encode $I_{W_{x,y}}(i)$ are determined using the VLC-Tables B.14 and B.15 of the MPEG-2 standard [ISO96]. If the size of VLC encoding the candidate DCT coefficient is equal or smaller than the size of the existing VLC, the existing VLC is replaced. Otherwise the VLC is left unaffected. This means that the DCT coefficient $I_{x,y}(i)$ is modulated in the following way:

$$\text{If} \quad Sz_{I_W} \le Sz_I \quad \text{then} \quad I_{W_{x,y}}(i) = I_{x,y}(i) + W_{x,y}(i) \qquad (3.3.3)$$
$$\text{else} \quad I_{W_{x,y}}(i) = I_{x,y}(i)$$

This procedure is repeated until all AC-coefficients of the encoded video block are processed.

To extract the watermark information, the MPEG encoded video stream is first fully decoded and the watermark bits are retrieved by correlating the decoded frames with the watermark pattern $W(x,y)$ in the spatial domain using the standard techniques as described in Section 2.2.2.

3.3.2.2 Drift compensation

A major problem of directly modifying DCT-coefficients in an MPEG encoded video stream is drift or error accumulation. In an MPEG encoded video stream predictions from previous frames are used to reconstruct the actual frame, which itself may serve as a reference for future predictions. The degradations caused by the watermarking process may propagate in time, and may even spatially spread. Since all video frames are watermarked, watermarks from previous frames and from the current frame may accumulate and result in visual artefacts. Therefore, a drift compensation signal *Dr* must be added. This signal must be equal to the difference of the (motion compensated) predictions from the unwatermarked bit stream and the watermarked bit stream. Equation 3.3.2 changes for a drift compensated watermarking scheme into:

$$I_{W_{x,y}}(i) = I_{x,y}(i) + W_{x,y}(i) + Dr_{x,y}(i) \qquad (3.3.4)$$

A disadvantage of this drift signal is that the complexity of the watermark embedding algorithm increases substantially, since an additional DCT operation and a complete MPEG decoding step are required to calculate the drift compensation signal. The increase in complexity compared to the coefficient domain methods is illustrated in Figure 3.3.1.

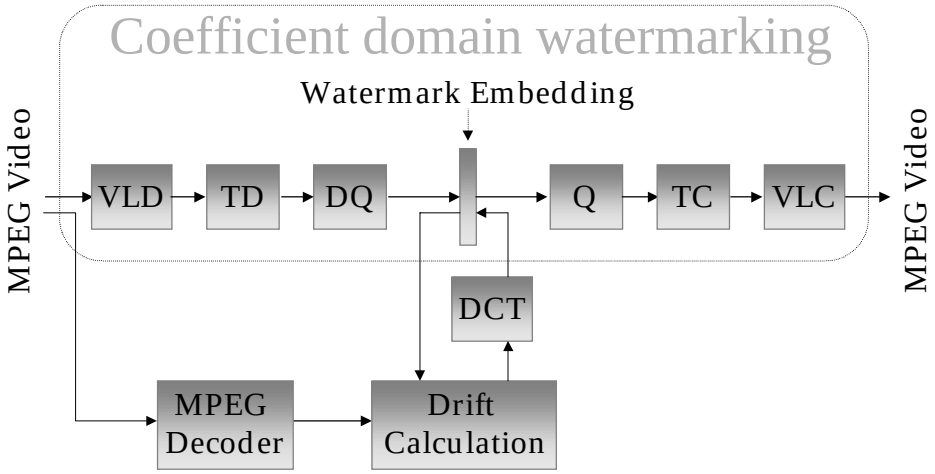


Figure 3.3.1. Increase of complexity due to drift compensation.

3.3.2.3 Evaluation of the correlation-based technique

Due to the bit-rate constraint, only around 10-20% of the DCT coefficients are altered by the watermark embedding process, depending on the video content and the coarseness of the MPEG quantizer. In some cases, especially for very low bit-rate video, only the DC-coefficients are modified. This means that only a fraction of the watermark pattern $W(x,y)$ can be embedded, typically around 0.5…3% [Har98]. Since only existing (non-zero) DCT coefficients of the video stream are watermarked, the embedded watermark is video content dependent. In areas with only low-frequency content, the watermark automatically consists of only low frequency components. This complies with the Human Visual System. The watermark energy is mainly embedded in areas containing a lot of video content energy.

The authors [Har98] report that the complexity of the watermark embedding process is much lower than the complexity of a decoding process followed by watermarking in the spatial domain and re-encoding. The complexity is somewhat higher than the complexity of a full MPEG decoding operation. Typical parameter settings for the embedding are k=1…5 for the gain factor of the watermark and P=500,000…1,000,000 for the number of pixels per watermark bit, yielding watermark label bit-rates of only a few bytes per second. The authors claim that the watermark is not visible, except in direct comparison to the unwatermarked video, and that the watermark is robust against linear and non-linear operations like filtering, noise addition and quantization in the spatial or frequency domain.

3.4 Parity bit modification in the bit domain

3.4.1 Bit domain watermarking concept

In Section 2.4.1 we saw that watermarking algorithms based on LSB (least significant bit) modification have an enormous payload and are not computationally demanding. In this section, this LSB modification principle is directly applied in the bit domain of MPEG compressed video, resulting in a computationally highly efficient watermarking algorithm with an extremely high payload [Lan96b], [Lan97b] and [Lan98a].

A watermark consisting of l label bits b_j ($j = 0, 1, 2,\ldots, l$-1) is embedded in the MPEG-stream by selecting suitable VLCs and forcing the least significant bit of their *quantized level* to the value of b_j. To ensure that the change in the VLC is perceptually invisible after decoding and that the MPEG-bit stream keeps its original size, we select only those VLCs for which another VLC exists with:

- the same run length
- a level difference of 1
- the same code word length

A VLC that meets this requirement is called a label-bit-carrying-VLC (*lc-VLC*). According to Table B.14 and B.15 of the MPEG-2 standard [ISO96], an abundance of such *lc-VLCs* exists. Furthermore, all fixed-length-coded DCT-coefficients following an Escape-code meet the requirement. Some examples of *lc-VLCs* are listed in Table 3.4.1, where the symbol *s* represents the sign-bit. This sign-bit represents the sign of the DCT coefficient level.

Table 3.4.1. Example of *lc-VLCs* in Table B.14 of the MPEG-2 Standard.

Variable length code	VLC size	Run	Level	LSB of Level
0010 0110 s	8 + 1	0	5	1
0010 0001 s	8 + 1	0	6	0
0000 0001 1101 s	12 + 1	0	8	0
0000 0001 1000 s	12 + 1	0	9	1
0000 0000 1101 0 s	13 + 1	0	12	0
0000 0000 1100 1 s	13 + 1	0	13	1
0000 0000 0111 11 s	14 + 1	0	16	0
0000 0000 0111 10 s	14 + 1	0	17	1
0000 0000 0011 101 s	15 + 1	1	10	0
0000 0000 0011 100 s	15 + 1	1	11	1
0000 0000 0001 0011 s	16 + 1	1	15	1
0000 0000 0001 0010 s	16 + 1	1	16	0

The VLCs in the intra and inter coded macro blocks can be used in the watermarking process. The DC coefficients are not used, because they are predicted from other DC coefficients and coded with a different set of VLCs and Escape-codes. Furthermore, replacing each DC coefficient in intra and inter coded frames can result in visible artefacts due to drift. By only taking the AC coefficients into account the watermark will adapt itself more to the video content and the drift will be limited.

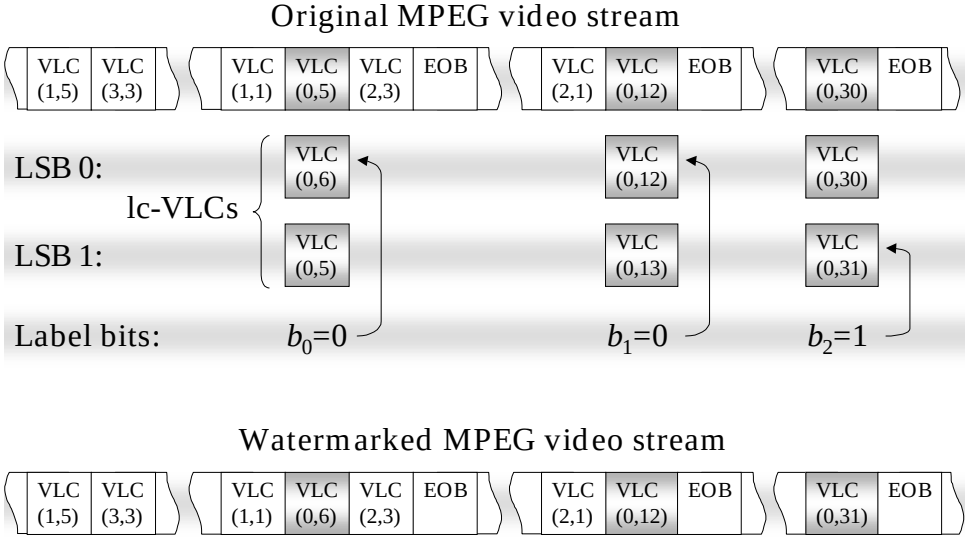


Figure 3.4.1. Example of the LSB watermarking process.

To add the label bit stream L to an MPEG-video bit stream, the VLCs in each macro block are tested. If an *lc-VLC* is found and the least significant bit of its level is unequal to the label bit b_j ($j=0,1,2,\ldots,l$-1), this VLC is replaced by another, whose LSB-level represents the label bit. If the LSB of its level equals the label bit b_j the VLC is not changed. The procedure is repeated until all label bits are embedded. In Figure 3.4.1 an example is given of the watermarking process, where 3 label bits are embedded in the MPEG video stream.

To extract the label bit stream *L* the VLCs in each macro blocks are tested. If an *lc-VLC* is found, the value represented by its LSB is assigned to the label bit b_j. The procedure is repeated for *j=0,1,2,…,l*-1 until no *lc-VLC*s can be found anymore.

3.4.2 Evaluation of the bit domain watermarking algorithm

3.4.2.1 Test sequence

The maximum label bit-rate is the maximum number of label bits that can be added to the video stream per second. This label bit-rate is determined by the number of *lc-VLCs* in the video stream and is not known in advance. Therefore, we first experimentally evaluate the maximum label bit-rate by applying the watermarking technique to an MPEG-2 video-sequence. The sequence lasts 10 seconds, has a size of 720 by 576 pixels, is coded with 25 frames per second, has a GOP-length of 12 and contains P-, B- and I-frames. The sequence contains smooth areas, textured areas and sharp edges. During the 10 seconds of the video there is a gradual frame-to-frame transition and the camera turns fast to another view at the end. A few frames of the sequence are shown in Figure 3.4.2. This sequence coded at different bit-rates (1.4, 2, 4, 6 and 8 Mbit/s) is used for all experiments in this thesis and will be referred to as the "sheep-sequence".

Figure 3.4.2. A few frames of the "sheep-sequence".

3.4.2.2 Payload of the watermark

In Table 3.4.2 the results of the watermark embedding procedure are listed. Only the *lc-VLC*s in the intra coded macro blocks, excluding the DC coefficients, are used to embed watermark label bits. In this table the "number of VLCs" equals the number of all coded DCT-coefficients in the intra coded macro blocks, including the fixed length coded coefficients and the DC-values. It appears that it is possible to store up to 7 kbit of watermark information per second in the MPEG streams if only intra coded macro blocks are used.

Table 3.4.2. Total number of VLCs and number of *lc-VLCs* in the intra-coded macro blocks of **10 seconds** MPEG-2 video coded using different bit-rates and the maximum label bit-rate.

Video bit-rate	Number of VLCs	Number of *lc-VLCs*	Max. label bit-rate
1.4 Mbit/s	334,433	1,152 (0.3%)	0.1 kbit/s
2.0 Mbit/s	670,381	11,809 (1.8%)	1.2 kbit/s
4.0 Mbit/s	1,401,768	34,650 (2.5%)	3.5 kbit/s
6.0 Mbit/s	1,932,917	52,337 (2.7%)	5.2 kbit/s
8.0 Mbit/s	2,389,675	69,925 (2.9%)	7.0 kbit/s

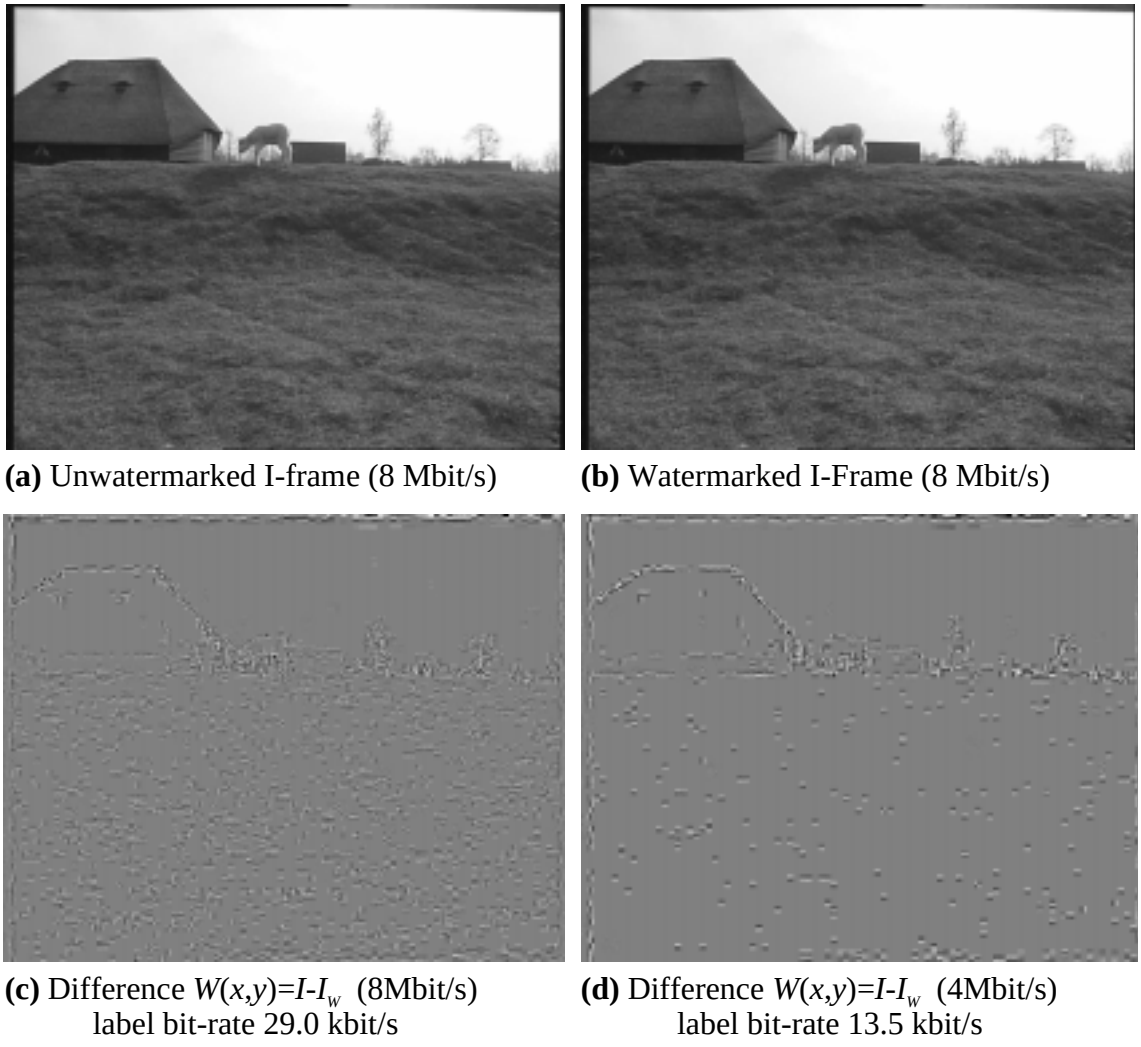
If also the *lc-VLC*s in the inter coded blocks are used, the maximum label bit-rate increases to 29 kbit/s. The results of this experiment are listed in Table 3.4.3. In this case the "number of VLCs" equals the number of all coded DCT-coefficients in the intra and inter coded macro blocks, including the fixed length coded coefficients and the DC-values.

Table 3.4.3. Total number of VLCs and number of *lc-VLCs* in the intra and inter coded macro blocks of **10 seconds** MPEG-2 video coded using different bit-rates and the maximum label bit-rate.

Video bit-rate	Number of VLCs	Number of *lc-VLCs*	Max. label bit-rate
1.4 Mbit/s	350,656	1,685 (0.5%)	0.2 kbit/s
2.0 Mbit/s	1,185,866	30,610 (2.6%)	3.1 kbit/s
4.0 Mbit/s	4,057,786	135,005 (3.3%)	13.5 kbit/s
6.0 Mbit/s	7,131,539	222,647 (3.1%)	22.3 kbit/s
8.0 Mbit/s	10,471,557	289,891 (2.8%)	29.0 kbit/s

3.4.2.3 Visual impact of the watermark

Informal subjective tests show that the watermarking process does not result in any visible artefacts in the streams coded at 4, 6 and 8 Mbit/s. It was not possible to reliably evaluate the quality degradation due to watermark embedding at less than 2 Mbit/s, because the unwatermarked MPEG-streams are already of poor quality, as it contains many compression artefacts. Although the visual degradation of the video due to the watermarking is not noticeable, the degradations are numerically measurable. In particular the maximum local degradations and the drift due to accumulation are of relevance. In Figure 3.4.3a an original I-frame of the "sheep-sequence" is represented. The sequence is MPEG-2 encoded at 8 Mbit/s. Figure 3.4.3b shows the corresponding watermarked frame. In Figure 3.4.3c the strongly amplified difference between the original I-frame and the watermarked frame is presented. Figure 3.4.3d shows the difference between the original I-frame coded at 4 Mbit/s and the corresponding watermarked frame. Since more bits are stored in an I-frame of a video stream coded at 8 Mbit/s more degradations are introduced (Figure 3.4.3c) than in an I-frame of a video stream coded at 4 Mbit/s (Figure 3.4.3d).



(a) Unwatermarked I-frame (8 Mbit/s)

(b) Watermarked I-Frame (8 Mbit/s)

(c) Difference $W(x,y)=I-I_w$ (8Mbit/s) label bit-rate 29.0 kbit/s

(d) Difference $W(x,y)=I-I_w$ (4Mbit/s) label bit-rate 13.5 kbit/s

Figure 3.4.3. Watermarking by VLC parity bit modification.

According to Figure 3.4.3 most differences are located around the edges and in the textured areas. The smooth areas are left unaffected. In order to explain this effect the location of the *lc-VLC*s is investigated. In Figure 3.4.4 a histogram is shown of the sheep-sequence coded at 8 Mbit/s. The number of all VLCs (including the fixed length codes) that code non-zero DCT coefficients and the number of *lc-VLCs* are plotted along the logarithmic vertical axis, represented by respectively white and gray bars. The DCT-coefficient index scanned in the zig-zag order ranging from 0 to 63 is shown on the horizontal axis.

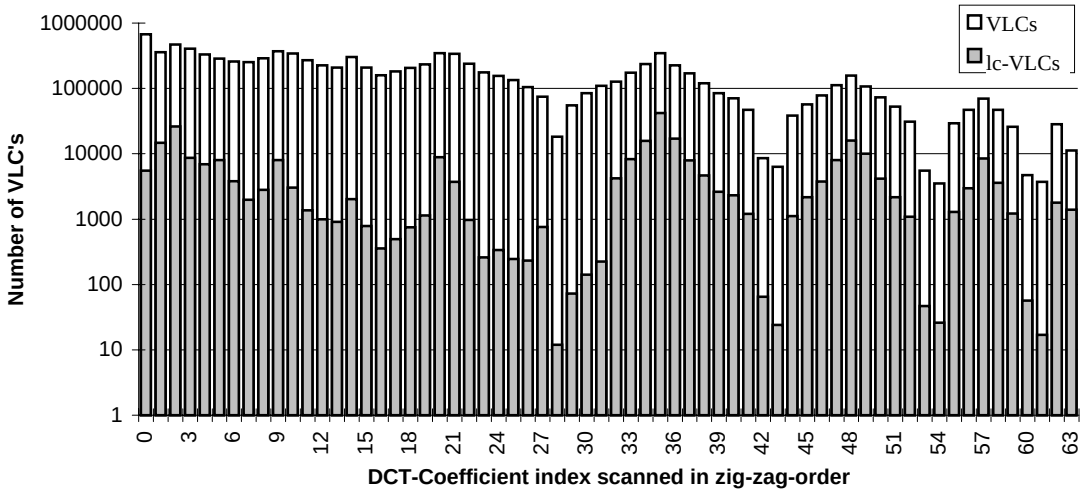


Figure 3.4.4. Number of VLCs and lc-VLCs in 10s MPEG-2 video coded at 8Mb/s.

From Figure 3.4.4 it appears that the *lc-VLC*s are fairly uniformly distributed over the DCT-spectrum. Therefore, we can expect each non-zero DCT-coefficient represented by a VLC to have an equal probability of being modified. If we take into account that according to Table 3.4.3 at most 3.3% of all VLCs are *lc-VLCs*, the probability of a VLC being modified can roughly be estimated as follows:

$$P[\textit{VLC modified}] = P[\textit{VLC = lc-VLC}] \cdot P[\textit{label bit} \neq \textit{LSB level VLC}] \qquad (3.4.1)$$
$$P[\textit{VLC modified}] < 0.033 \cdot \tfrac{1}{2} = 0.016$$

Smooth blocks are coded with only one or a few DCT-coefficients. Because only 1.6% of them is replaced, most of the smooth areas are left unaffected. The textured blocks and the blocks containing sharp edges are coded with far more VLCs. These blocks will therefore contain the greater part of the *lc-VLCs*.

The maximum *local* degradation or the number of *lc-VLCs* per block must be as low as possible. The visual impact of the watermarking process will be much smaller if the degradations introduced by modifying an *lc-VLC* are distributed more or less uniformly over the frame, instead of being concentrated and accumulated in a relative small area of the frame or even worse being accumulated in a single DCT-block.
In Figure 3.4.5 a histogram is shown of 10 seconds of the watermarked "sheep-sequence" coded at 8 Mbit/s. On the vertical axis the number of *lc-VLCs* per 8x8 block is shown. The number of 8x8 blocks that contain this amount of *lc-VLC*s is plotted along the logarithmic horizontal axis.

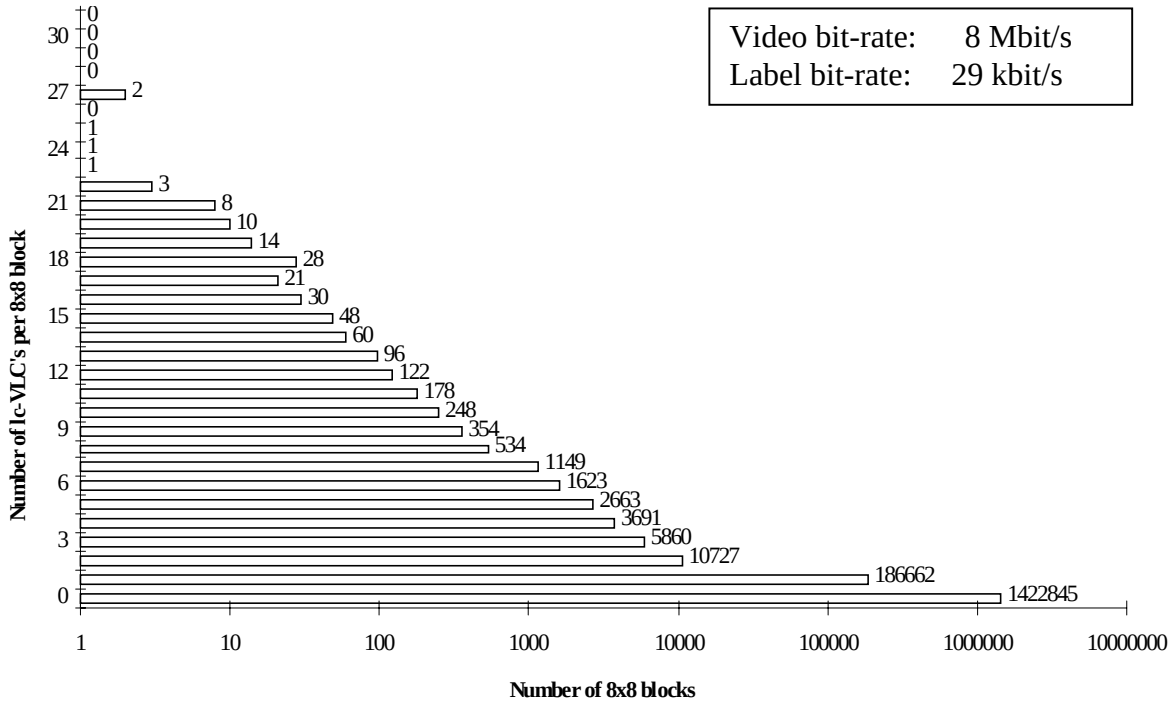


Figure 3.4.5. Log-histogram of the number of *lc-VLCs* per 8x8 block.

This figure shows that 87% of all coded 8x8 blocks do not contain any *lc-VLC*. The rest of the coded 8x8 blocks contain one ore more *lc-VLCs*. Most blocks (186.662) contain only one *lc-VLC*, which is about 64% of all *lc-VLCs* in the sequence. These numbers can be explained by examining Table B.14 and B.15 of the MPEG-2 standard [ISO96]. The most frequently occurring run-level pairs are coded with short VLCs. Almost all short VLCs do not qualify as an *lc-VLC*. This means that the chance of a large number of *lc-VLCs* in one 8x8 block is relatively low.

To limit the maximum number of *lc-VLC* replacements per DCT-block to T_m, a threshold mechanism can be used. If the number of *lc-VLC*s exceeds T_m, only the first T_m *lc-VLC*s are used for the watermark embedding, the other *lc-VLC*s are left unchanged. In Table 3.4.4 the label bit-rates for the "sheep-sequence" coded at 8 Mbit/s are listed for several values of T_m. If at most two *lc-VLC* replacements per block are allowed ($T_m = 2$), the label bit-rate is only decreased to 83% of the maximum label bit-rate for which T_m = *unlimited*. So limiting the number of *lc-VLC* replacements per block can avoid unexpected large local degradations without drastically affecting the maximum label bit-rate.

Table 3.4.4. Label bit-rates using a threshold for at most T_m *lc-VLC* replacements per 8x8 DCT-block (Video bit-rate 8 Mbit/s).

T_m =max. *lc-VLC* replacements per block	Max. label bit-rate
2	24.2 Kbit/s
4	26.9 Kbit/s
6	28.1 Kbit/s
8	28.6 Kbit/s
10	28.8 Kbit/s
Unlimited	29.0 Kbit/s

3.4.2.4 Drift

In an MPEG-video stream P-frames are predicted from the previous I- or P-frame. The B-frames are predicted from the two nearest I- or P-frames. Since intra and inter coded macro blocks are used for the watermark embedding, errors are introduced in all frames. However, error accumulation (drift) from the frames used for the prediction occurs in the predicted P- and B-frames. The drift can clearly be seen in Figure 3.4.6, where the difference $\Delta MSE = MSE_l - MSE_u$ is plotted. The MSE_u is the MSE per frame between the original uncoded "sheep-sequence" and the sequence coded at 8 Mbit/s. The MSE_l is the MSE per frame between the uncompressed sequence and the watermarked sequence coded at 8 Mbit/s.

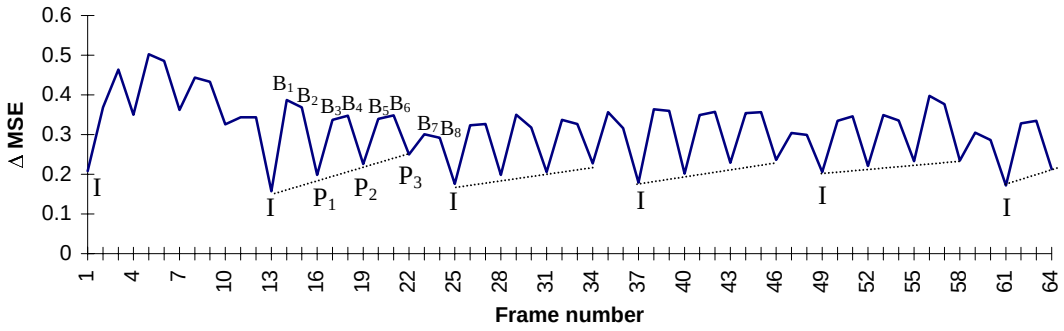


Figure 3.4.6. *ΔMSE* of the watermarked sheep-sequence coded at 8 Mbit/s with a label bit-rate of 29.0 kbit/s.

In Figure 3.4.6 it can be seen that the I-frames (numbered 1,13,25,37…) have the smallest *ΔMSE*, the *ΔMSE* of a predicted B-frame is 2 to 3 times larger than the error in the I-frames in the worst case. The average Peak-Signal-to-Noise-Ratio (*PSNR*) between the MPEG-compressed original and the uncompressed original is 37dB. If the watermarked compressed video stream at 8 Mbit/s is compared with the original compressed stream, the *ΔMSE* causes an average *ΔPSNR* of 0.1dB and a maximum *ΔPSNR* of 0.2dB. From these *ΔPSNR* values we conclude that the drift can be neglected and no drift compensation signal is required.

3.4.3 Robustness

A large label bit stream can be added and extracted in a very fast and simple way, but it can also be removed without significantly affecting the quality of the video. However, it still takes a lot of effort to completely remove a label from a large MPEG video stream. For example decoding the watermarked MPEG-stream and encoding it again using another bit-rate will destroy the label bit string. But re-encoding is a computationally and memory (disk) demanding operation.

The easiest way to remove the label is by watermarking the stream again using another random label bit stream. In this case the quality is slightly affected. During the re-labeling

phase the adapted *lc-VLCs* in the watermarked video stream can either return to their original values or change to VLCs that represent DCTs that differ two quantization levels from the original ones in the unwatermarked video stream. Non-adapted *lc-VLCs* in the watermarked video stream can change to a value that differs one quantization level from the one in the original video stream. This means that there is some extra distortion, although the quality is only slightly affected. Since re-labeling of a large MPEG video stream still requires special hardware or a very powerful computer, the bit domain watermarking method is suitable for consumer applications requiring moderate robustness.

3.5 Re-labeling resistant bit domain watermarking method

By reducing the payload of the watermark drastically we can easily change the bit domain watermarking algorithm described in Section 3.4.1 to a re-labeling resistant algorithm. The watermark label bits b_j are now not directly stored in the least significant bits of the VLCs, but a 1-dimensional pseudorandom watermark pattern $W(x)$ is generated consisting of the integers {-1,1} based on a secret key, which is modulated with the label bits b_j as described in Section 2.2.2. The procedure to add this modulated pattern to the video stream is similar to the procedure described in Section 3.4.1.

However, we now select only those VLCs for which two other VLCs exist, with the same run length and the same codeword length. One VLC must have a level difference of $+\delta$ and the other VLC must have a level difference of $-\delta$. Most lc-VLCs meet these requirements for a relative small δ (e.g. $\delta = 1,2,3$). For notational simplicity we call these VLCs, pattern-carrying-VLCs (*pc-VLC*s).

To embed a watermark in a video stream, we simply add the modulated watermark pattern to the levels of the *pc-VLC*s. To extract the watermark, we collect the *pc-VLC*s in an array. The watermark label bits can now be retrieved by calculating the correlation between this array of *pc-VLC*s and the secret watermark pattern $W(x)$. In Figure 3.5.1 an example is given of the watermark embedding process. About 1,000…10,000 *pc-VLC*s are now required to encode one watermark label bit b_j, but several watermark label bit strings can be added without interfering with each other, if independent pseudorandom patterns are used to form the basic pattern $W(x)$.

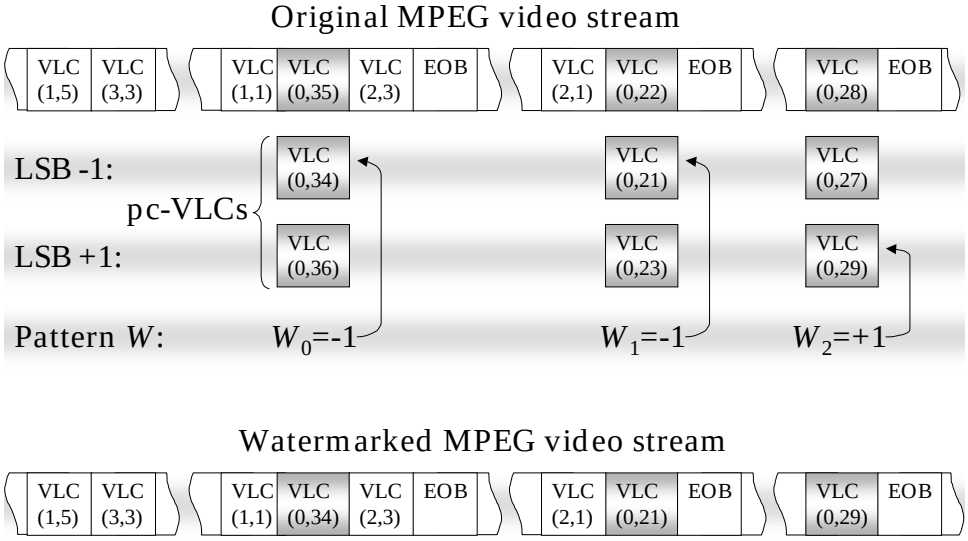


Figure 3.5.1. Example of the re-labeling resistant watermarking method.

3.6 Discussion

The most efficient way to reduce the complexity of real-time watermarking algorithms is to avoid computationally demanding operations by exploiting the compression format of the host video data. An advantage of this approach is that the watermark automatically becomes video content dependent. Since lossy compression algorithms discard video information to which the human visual system is less sensitive and only encode visual important information, the watermark is only embedded in visual important areas. A disadvantage of closely following a compression standard and applying the constraint that the compressed video stream may not increase in size, is that the number of locations to embed watermark information is limited significantly. The distortions caused by the watermark applied on a compressed video stream differ also from the distortions caused by a watermark applied on an uncompressed video stream. Due to block-based transformations and motion compensated frame prediction, distortions may spread over blocks and accumulate over the consecutive frames.

In this chapter we discussed four low complexity watermarking algorithms. The first correlation-based algorithm only uses the DC-coefficients. Although the algorithm can completely be performed in the coefficient domain, the low frequency watermark causes too many visible artefacts. The second correlation-based method takes besides the DC-coefficients also the AC-coefficients into account and applies drift compensation to prevent that the watermark becomes visible. Since it utilizes more locations to embed watermark energy, the watermark is more robust. However, adding a drift compensation signal and extracting the watermark information can not be performed in the coefficient domain, since a full MPEG decoding operation is required. The algorithm is therefore more complex than an algorithm that can fully be applied in the coefficient domain. The third LSB-modification method that we proposed, fully operates in the bit domain, and is therefore the most computational efficient, but least robust method. Other advantages of this method are the enormous payload and the invisibility of the watermark. The fourth

method extends the LSB-modification method and achieves a higher robustness by reducing the payload of the watermark.

There are two important differences between the correlation-based methods and the LSB-modification methods. A watermark embedded by a correlation-based method can still be extracted from the decoded raw video, since the watermarking procedure adds a spatial noise pattern to the pixel values. If the pixel values are available in raw format or another compressed format the watermark can still be detected. Once a video stream watermarked by the LSB-modification methods is decoded, the watermark is lost, because the watermark embedding and extraction procedures are completely dependent on the MPEG structure of the video. This structure disappears or changes when the video is decoded or re-encoded at another bit-rate. Since full MPEG decoding and encoding is a quite computationally demanding task this is not really an issue for consumer applications requiring moderate robustness. Furthermore, correlation-based methods and LSB-modification methods differ considerably in complexity. LSB-modification methods are far more computational efficient since they can operate on the lowest level in the bit-domain.

For real-time applications that require the same level of robustness as the correlation-based methods, but have not enough computational power to perform full MPEG decoding for drift compensation and watermark detection, we have developed a completely new watermarking concept, which is presented in Chapters 4 and 5.

Chapter 4

Differential Energy Watermarks (DEW) for Compressed Video

4.1 Introduction

In Chapter 3 we noticed that correlation-based watermarking techniques have the advantage that watermarks can be extracted from decoded or re-encoded video streams. However, in order to embed or detect an invisible correlation-based watermark, a full MPEG decoding operation is required. This might be too computationally demanding. On the contrary, we have seen that the Least Significant Bit (LSB) based algorithms are computationally highly efficient. But watermarks embedded by these algorithms can not be extracted from decoded or re-encoded video streams. For real-time consumer applications that require the same level of robustness as the correlation based methods and the same computational efficiency as the LSB-based methods, we therefore developed the Differential Energy Watermarking (DEW) concept [Lan97a], [Lan97b], [Lan98a] and [Lan99b]. As can be seen in Figure 4.1.1 the DEW concept can be applied directly on MPEG/JPEG compressed video as well as on raw video.

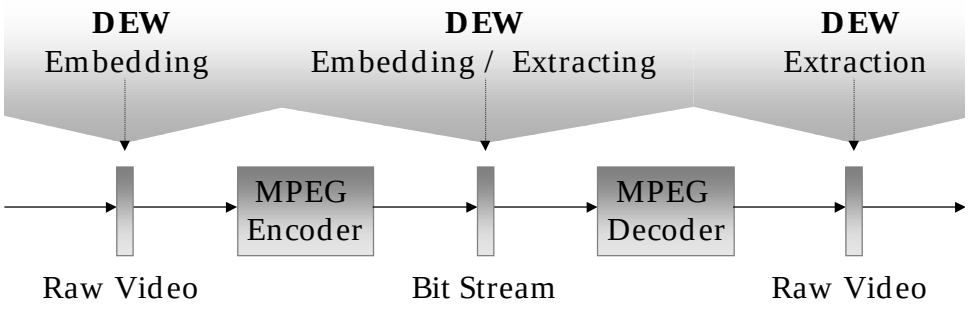


Figure 4.1.1. DEW embedding / extracting in compressed and raw video.

In the case of MPEG/JPEG encoded video data, the DEW embedding and extracting procedures can completely be performed in the coefficient domain (see Section 3.2). The encoding parts of the coefficient-domain watermarking concept can even be omitted. This means that the complexity of the DEW algorithm is only slightly higher than the LSB-based methods discussed in Section 3.4, but its complexity is considerably lower than the correlation-based method with drift compensation discussed in Section 3.3.

The DEW concept is not limited to MPEG/JPEG coded video only, it is also applicable to video data compressed using other coders, for instance embedded zero-tree wavelet coders [Sha93]. The DEW algorithm embeds label bits by selectively discarding high frequency coefficients in certain video frame regions. The label bits of the watermark are encoded in the pattern of energy differences between DCT blocks or hierarchical wavelet trees.

In Section 4.2 the general DEW concept for MPEG/JPEG coders is explained, followed by a more detailed description in Section 4.3. In Section 4.4 the DEW concept is evaluated for MPEG compressed video. Section 4.5 explains the general DEW concept for embedded zero-tree wavelet coded video. Finally the results are discussed in Section 4.6.

4.2 The DEW concept for MPEG/JPEG encoded video

The Differential Energy Watermarking (DEW) method embeds a watermark consisting of l label bits b_j (j = 0, 1, 2,…, l-1) in a JPEG image or in the I-frames of an MPEG video stream. Each bit out of the label bit string has its own label-bit-carrying-region, *lc-region*, consisting of n 8x8 DCT luminance blocks.

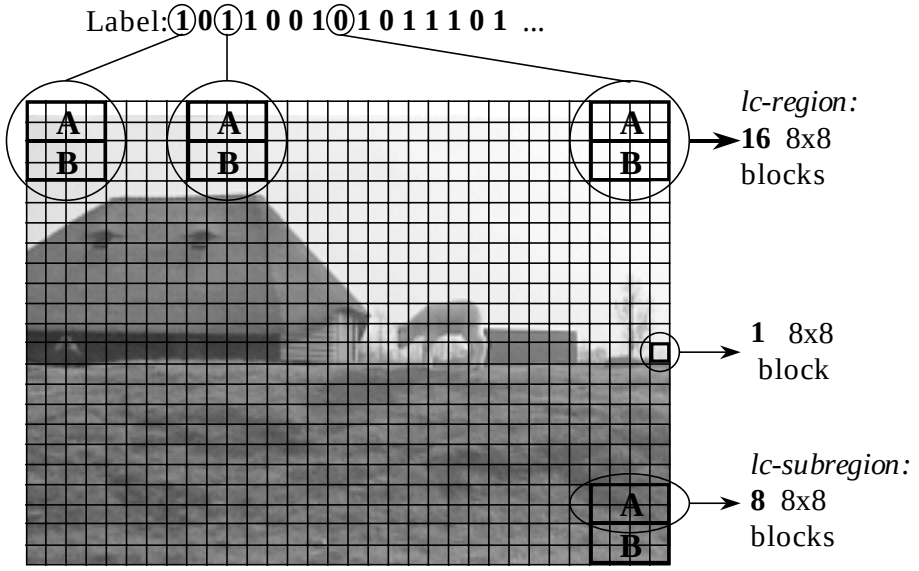


Figure 4.2.1. Label bit positions and region definitions in a frame.

For instance the first label bit is located in the top-left-corner of the image or I-frame in an lc-region of n=16 8x8 DCT blocks as illustrated in Figure 4.2.1. The size of this lc-region determines the label bit-rate. The higher n, the lower the label bit-rate. In case the video data is not DCT compressed, but in raw format, the DEW algorithm requires a block-based DCT transformation as a preprocessing step.

A label bit is embedded in an lc-region by introducing an "energy" difference D between the high frequency DCT-coefficients of the top half of the lc-region (denoted by *lc-subregion A*) and the bottom half (denoted by *B*). The energy in an lc-subregion equals the squared sum of a particular subset of DCT-coefficients in this lc-subregion. This subset is denoted by $S(c)$, and is illustrated in Figure 4.2.2 by the white triangularly shaped areas in the DCT-blocks.

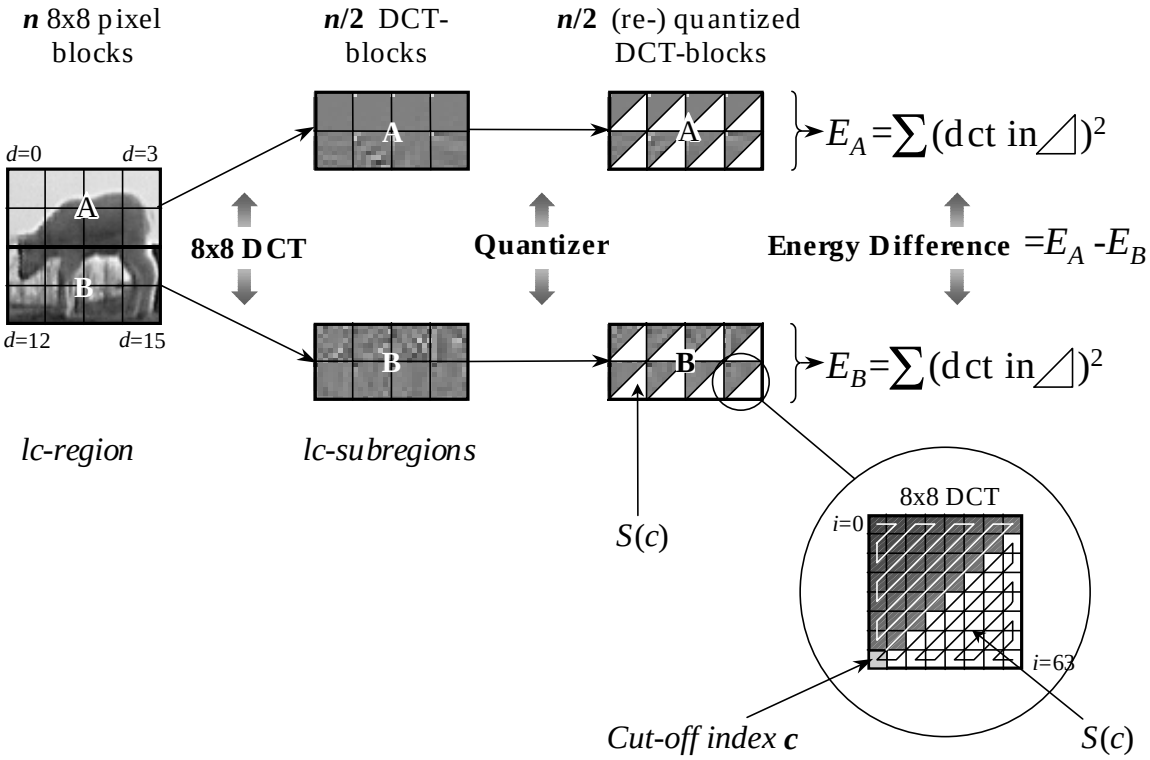


Figure 4.2.2. Energy definitions in an lc-region of n=16 8x8 DCT blocks.

We define the total energy in $S(c)$, computed over the *n/2* blocks in *subregion A*, as:

$$E_A(c,n,Q_{jpeg}) = \sum_{d=0}^{n/2-1} \sum_{i \in S(c)} ([\theta_{i,d}]_{Q_{jpeg}})^2 \qquad (4.2.1)$$

Here $\theta_{i,d}$ denotes the non-weighted zig-zag scanned DCT coefficient with index i in the d-th DCT block of the lc-subregion A under consideration. The notation $[]_{Q_{jpeg}}$ indicates that, prior to the calculation of E_A, the DCT-coefficients of JPEG compressed video are optionally re- or pre-quantized using the standard JPEG quantization procedure [Pen93] with quality factor Q_{jpeg}. For embedding labels bits into MPEG compressed I-frames a similar approach can be followed, but here we confine ourselves to the JPEG notation without loss of generality. The pre-quantization is done only in determining the energies, but is *not* applied to the actual video data upon embedding the label. The energy in lc-subregion B, denoted by E_B, is defined similarly.

$S(c)$ is typically defined according to a *cut-off index c* in the zig-zag scanned DCT-coefficients.

$$S(c) = \{h \in \{1,63\} \mid (h \geq c)\} \qquad (4.2.2)$$

The selection of suitable cut-off indices for lc-regions is very important for the robustness and the visibility of the label bits and will be discussed in the next section. First we focus on how the watermarking procedure works, assuming that we have available suitable cut-off indices *c* for each lc-region. The energy difference *D* between top and bottom half of an lc-region is defined as:

$$D(c,n,Q_{jpeg}) = E_A(c,n,Q_{jpeg}) - E_B(c,n,Q_{jpeg}) \quad (4.2.3)$$

In Figure 4.2.2 the complete procedure to calculate the energy difference *D* of an lc-region *(n=16)* is graphically illustrated.

We now define the label bit value as the sign of the energy difference *D*. Label bit "0" is defined as *D>0* and label bit "1" as *D<0*. The watermark embedding procedure must therefore adapt E_A and E_B to manipulate the energy difference *D*. If label bit "0" must be embedded, all energy after the cut-off index in the DCT-blocks of lc-subregion *B* is eliminated by setting the corresponding DCT-coefficients to zero, so that:

$$D = E_A - E_B = E_A - 0 = +\boldsymbol{E_A} \quad (4.2.4)$$

If label bit "1" must be embedded, all energy after the cut-off index in the DCT-blocks of lc-subregion *A* is eliminated, so that:

$$D = E_A - E_B = 0 - E_B = -\boldsymbol{E_B} \quad (4.2.5)$$

There are several reasons for computing this energy difference over the *triangularly shaped areas*. The most important reason is that calculating the energy difference and changing E_A and E_B can easily be done on the compressed stream. All DCT-coefficients needed for the calculation of E_A or E_B are conveniently located at the end of the compressed 8x8 DCT-block after zig-zag ordering. The coefficients can be forced to zero to adapt the energy without re-encoding the stream by shifting the end of block marker (EOB) towards the DC-coefficient. Figure 4.2.3 graphically illustrates the procedure to calculate *E* in a single compressed DCT-block and to change *E* by removing DCT-coefficients located at the end of the zig-zag scan (i.e. high frequency DCT-coefficients).

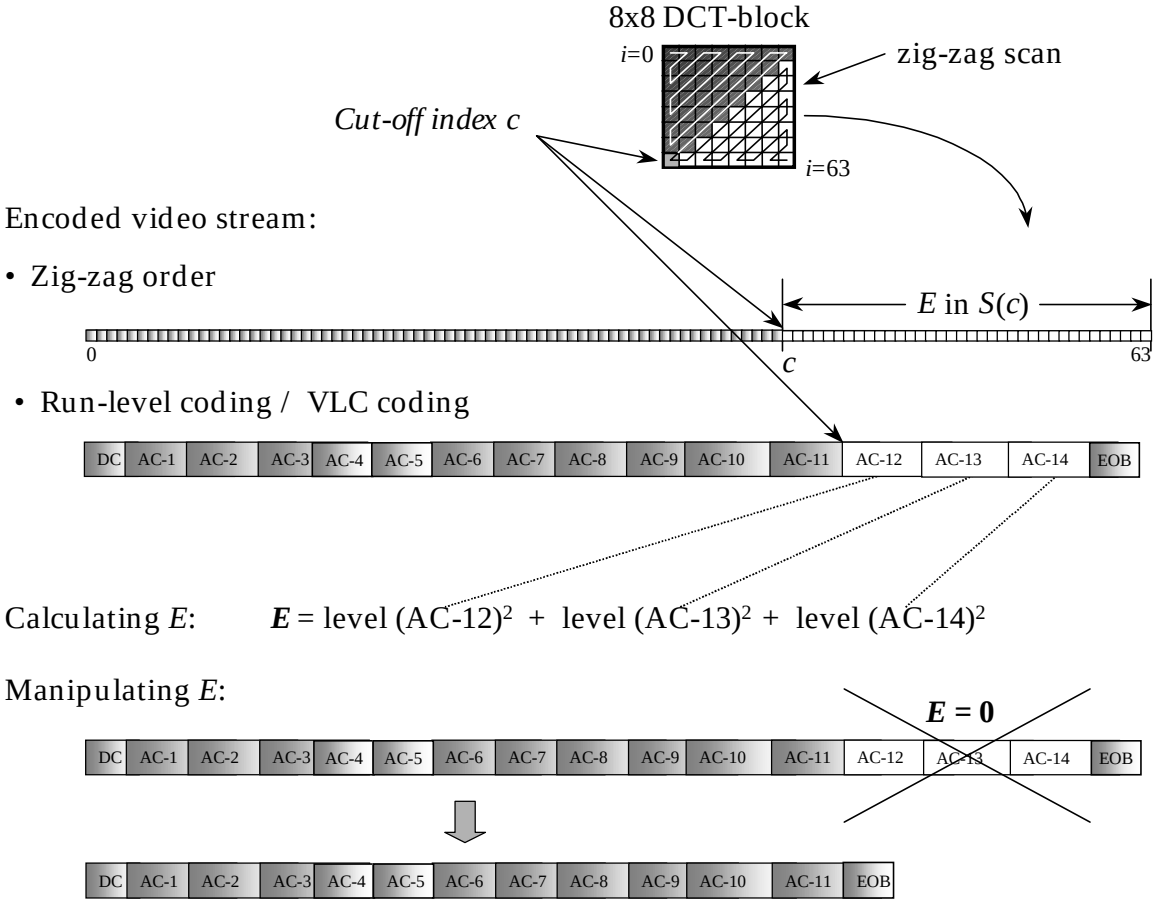


Figure 4.2.3. Calculating and adapting energy in an 8x8 compressed DCT-block.

The fact that a watermark is added only by removing coefficients has two advantages. Since no coefficients are adapted or added to the stream, the encoding parts of the coefficient domain watermarking concept can be omitted as illustrated in Figure 4.2.4. This means that the DEW algorithm has only half the complexity of other coefficient domain watermarking algorithms.

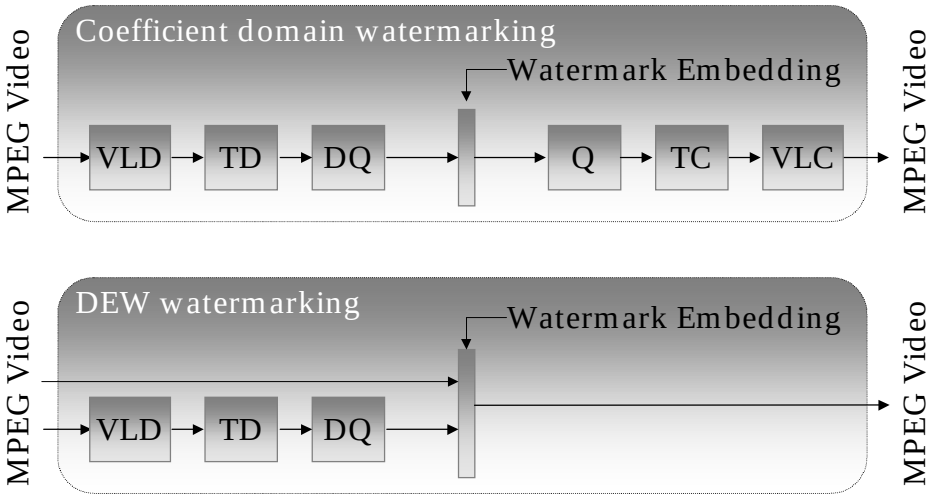


Figure 4.2.4. Complexity difference between the DEW algorithm and other Coefficient domain watermarking algorithms.

Furthermore, removing coefficients will always make the watermarked compressed video stream smaller in size than the unwatermarked video stream. If it is necessary that the watermarked compressed video stream keeps its original size, stuffing bits can be inserted before each macro block.

4.3 Detailed DEW algorithm description

The energies present in lc-subregions *A* and *B* defined by Equations 4.2.1 and 4.2.2 play a central role in the watermark embedding and extraction process. The values of E_A and E_B are determined by 4 factors:

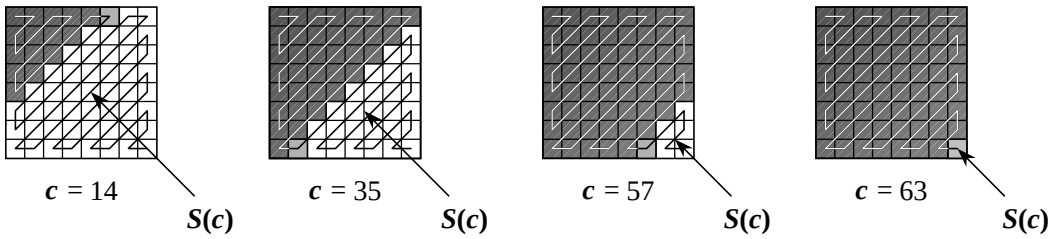
- the spatial content of the lc-subregions *A* and *B*
- the number of blocks *n* per lc-region
- the pre- or re-quantization JPEG quality factor Q_{jpeg}
- the size of subset $S(c)$ (i.e. the *triangular shaped areas*)

If the spatial content of an lc-region is very smooth and only coded by de DC-DCT coefficients, the AC-energy will be zero. The energy will be larger for regions containing a lot of texture or edges. The more DCT-blocks are taken to form the lc-region, the higher the energy will be, since the energy is the sum of the energies in all individual DCT-blocks in the lc-region.

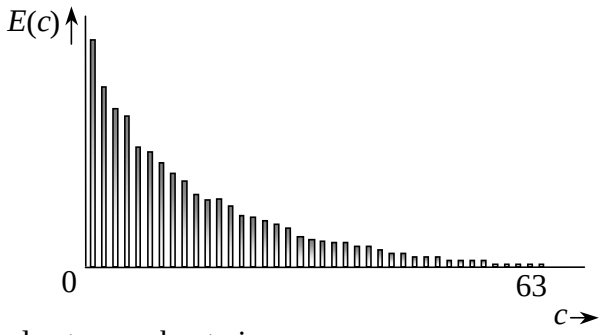
The optional pre- or re-quantization JPEG quality factor Q_{jpeg} controls the robustness of the watermark against re-encoding attacks. In a re-encoding attack the watermarked video data is partially or fully decoded and subsequently re-encoded at a lower bit-rate. Our method anticipates the re-encoding at lower bit-rates up to a certain minimal rate. The smaller Q_{jpeg} is chosen, the more robust the watermark becomes against re-encoding attacks. However, the smaller Q_{jpeg} is chosen, the smaller the energies E_A and E_B will be,

since most high frequency coefficients are quantized to zero, and can not contribute to the energy anymore.

The size of subset $S(c)$ (Equation 4.2.2) is determined by the standard zig-zag scan and a cut-off index c. If the zig-zag scanned DCT coefficients are numbered from 0 to 63, where the coefficient with index 0 represents the DC-component and the coefficient with index 63 the highest frequency component, this subset consists of the DCT coefficients with indices c…63 (c>0). In Figure 4.3.1 some examples are shown of subsets defined by increasing cut-off indices. The corresponding experimentally determined energies are plotted below. This figure shows that increasing the cut-off index decreases the energy.



(a) Subset $S(c)$ of DCT coefficients defined by zig-zag scan and cut-off index



(b) Energy dependent on subset size

Figure 4.3.1. (a) Examples of subsets and **(b)** energies for several cut-off indices.

To enforce an energy difference, the watermark embedding process has to discard all DCT coefficients in the subset $S(c)$ in lc-subregion A or B. Since discarding coefficient introduces visual distortion, the number of discarded DCT coefficients has to be minimized. This means that the watermark embedding algorithm has to find a suitable cut-off index for each lc-region that defines the smallest subset $S(c)$ for which the energy in both lc-subregions A and B exceeds the desired energy difference. To find the cut-off index that defines the desired subset, we first calculate the energies $E_A(c,n,Q_{jpeg})$ and $E_B(c,n,Q_{jpeg})$ for all possible cut-off indices c = 1...63. If D is the energy difference that is needed to represent a label bit in an lc-region, the cut-off index c is found as the *largest* index of the DCT coefficients for which (4.2.1) gives an energy *larger* than the required difference D in *both* subregions A and B.

In controlling the visual quality of the watermarked video data, we wish to avoid the situation that the important low frequency DCT coefficients are discarded. To this end, we

require the selected cut-off index to always be larger than a certain minimum c_{min}. Mathematically, this gives the following expression for determining c:

$$c(n,Q_{jpeg},D,c_{min}) = \max\{c_{min}, \max\{g \in \{1,63\} | (E_A(g,n,Q_{jpeg}) > D) \wedge (E_B(g,n,Q_{jpeg}) > D)\}\} \tag{4.3.1}$$

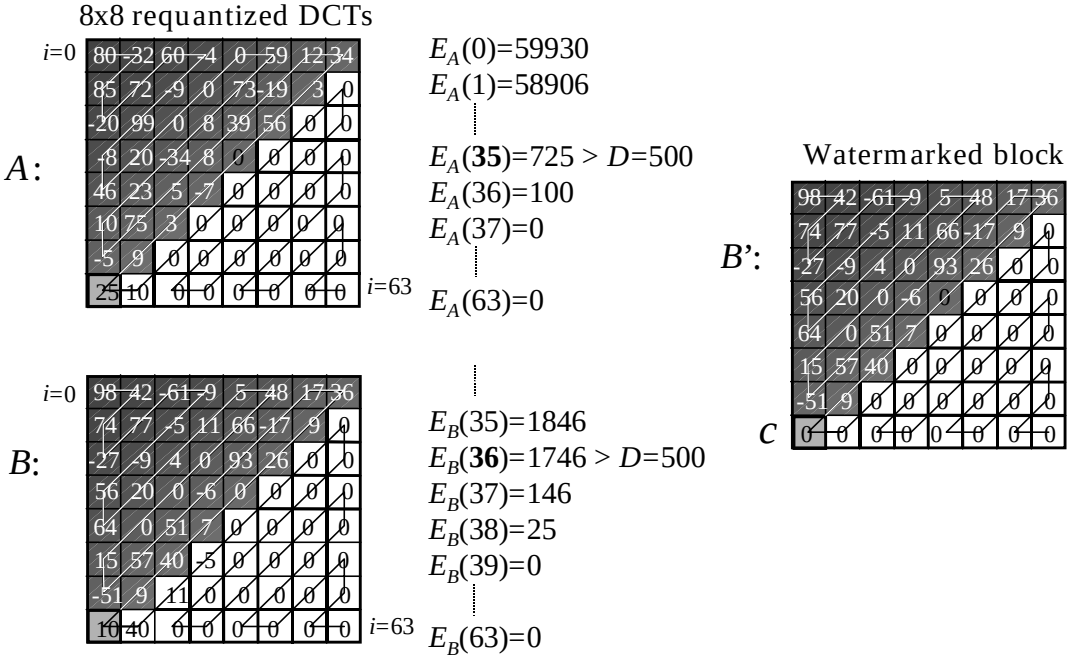


Figure 4.3.2. Embedding label bit b_0=0 in an lc-region of n=2 DCT blocks.

In Figure 4.3.2 an example is given of the embedding of label bit b_0=0 with an energy difference of D=500 in an lc-region consisting of n=2 DCT blocks. The maximum cut-off index for which the energy E_A exceed D=500 is 35, for E_B a cut-off index of 36 is sufficient. This means that the algorithm has to select a cut-off index c of 35 to have enough energy in both lc-subregions *A* and *B*. Since the label bit that has to be embedded is zero, a positive energy difference has to be enforced by setting E_B to zero (Equation 4.2.4). This is done by discarding all non-zero DCT coefficients with indices 35…63 in lc-subregion *B*.

To extract a label bit from an lc-region we have to find back the cut-off index that was used for that lc-region during the embedding process. We therefore first calculate the energies $E_A(c,n,Q_{jpeg})$ and $E_B(c,n,Q_{jpeg})$ for all possible cut-off indices c = 1...63. Since either in lc-subregion *A* or lc-subregion *B* several DCT-coefficients have been eliminated during the watermark embedding, we first find the *largest* index of the DCT coefficients for which Equation 4.2.1 gives an energy *larger* than a threshold $D' \leq D$ in either of the two lc-subregions. The actually used cut-off index is then found as the maximum of these two numbers:

$$c^{(extract)}(n,Q'_{jpeg},D') = \max\{\ \max\{g \in \{1,63\} \mid E_A(g,n,Q'_{jpeg}) > D'\},\ \max\{g \in \{1,63\} \mid E_B(g,n,Q'_{jpeg}) > D'\}\ \} \tag{4.3.2}$$

In the above procedure, the parameters D' and Q'_{jpeg} can be chosen equal to the parameters D and Q_{jpeg}, which are used in the embedding phase. The detection threshold D' influences the determination of the cut-off index. This value must be smaller than the enforced energy difference D, but larger than 0. If $D' = 0$ the label can correctly be extracted only if the video-stream is not affected by processing like adding noise, filtering or re-encoding. However, if a small amount of noise is introduced in the highest DCT-coefficients, cut-off indices will be detected, which are higher than the originally enforced ones. D' determines which amount of energy will be seen as noise. The re-quantization step can also be omitted (Q'_{jpeg}=100) without significantly influencing the reliability of the label bit extraction. Since Q_{jpeg} and D are not fixed parameters but may vary per image, the label extraction procedure must be able to determine suitable values for Q'_{jpeg} and D' itself. The most reliable way for doing this is to start the label bit string with several fixed label bits, so that during the label extraction those values for Q'_{jpeg} and D' can be chosen that result in the fewest errors in the known label bits.

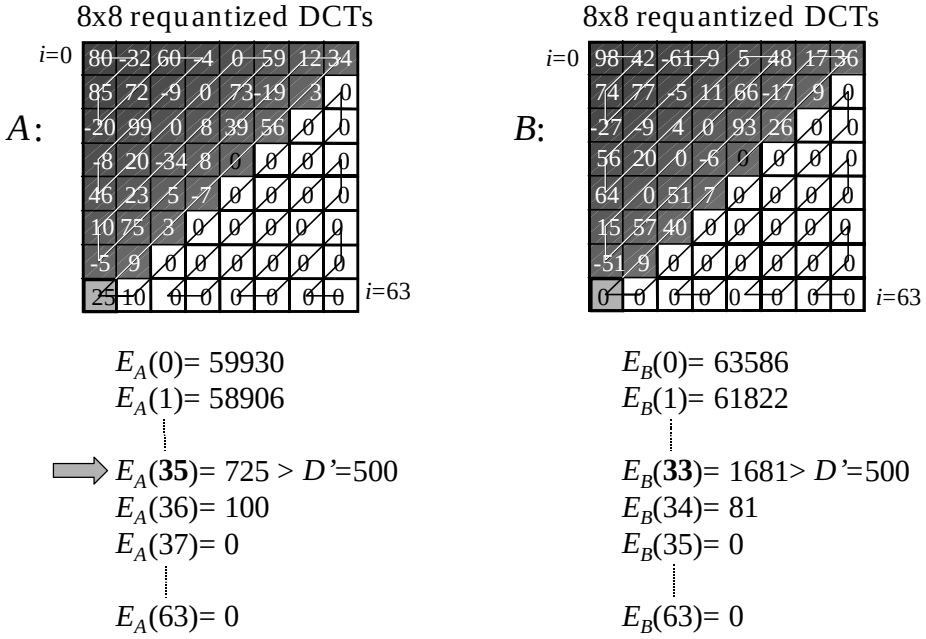


Figure 4.3.3. Extracting label bit b_0 from an lc-region of n=2 DCT blocks.

In Figure 4.3.3 an example is given of the extraction of label bit b_0 from the lc-region consisting of n=2 DCT blocks that was watermarked in Figure 4.3.2. For the extraction D'=D=500 is used. The maximum cut-off index for which the energy E_A exceed D'=500 is 35, for E_B this cut-off index is 33. This means that the watermark embedding algorithm has used a cut-off index of 35. The energy difference $E_A(35)$-$E_B(35)$=+725. Since the energy difference is positive, the value zero is assigned to label bit b_0.

The algorithm applied in this form is heavily dependent on the video content. Figure 4.3.4 shows several examples of this content dependency. In Figure 4.3.4a an lc-region is depicted in which the lc-subregions A and B both contain edges, smooth and textured areas. These are typical examples of regions with average energy in the AC DCT-coefficients. In this case, the watermark embedding procedure will select a subset $S(c)$ with a cut-off index somewhere in the middle of the range 1…63. This means that some

coefficients in the highest and middle frequency bands are discarded. If the amount of energy that is discarded in these frequency bands is limited, the label bit will not be noticeable. Since re-quantization by re-encoding at a lower bit-rate will not affect the energy difference in the middle frequency band seriously, the label bit will survive a re-encoding attack.

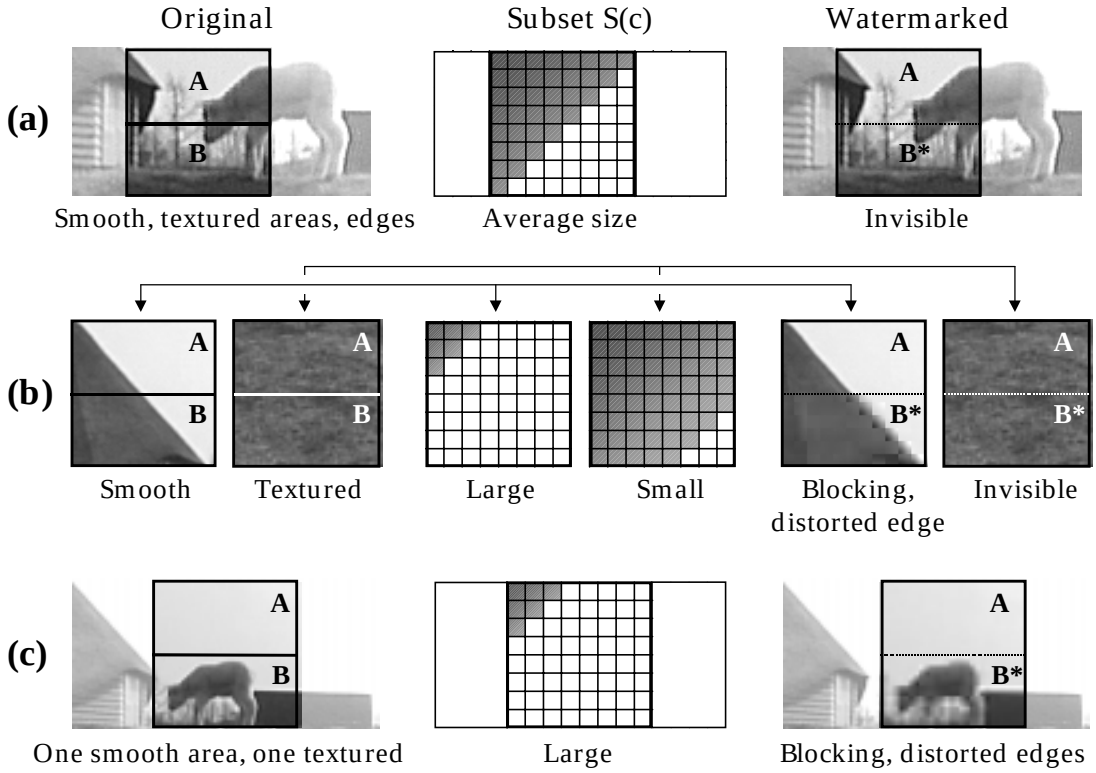


Figure 4.3.4. Examples of subset sizes depending on video content.

In Figure 4.3.4b two lc-regions are presented in which the lc-subregions are both very smooth or both very textured. If there is not much energy in a smooth lc-region, a very large subset $S(c)$ has to be chosen. This means that low-frequency DCT coefficients are discarded to which the human eye is quite sensitive, resulting in block artefacts and distorted edges. If there is much energy in a textured lc-region, a very small subset $S(c)$ is sufficient to find the required energy difference. Since here only the highest frequency components are discarded, the label bit will not be noticeable. However, since re-quantization by re-encoding at a lower bit-rate will affect the energy difference in the highest frequency bands seriously, the label bit will not survive a re-encoding attack.

The worst-case-situation is depicted in Figure 4.3.4c, where one lc-subregion is completely smooth, while the other is textured and contains sharp edges. If a positive energy difference $D = E_A - E_B$ must be generated in this lc-region, all AC DCT-coefficients in lc-subregion B must be eliminated by selecting an extremely large subset $S(c)$ to make $E_A > E_B$. The presence of the label bit obviously becomes clearly visible in lc-subregion B.

From these situations we conclude that it is not desirable to select very small subsets $S(c)$ defined by high cut-off indices, since energy differences embedded in the highest

frequency bands do not survive re-encoding attacks. Furthermore, selecting large subsets defined by low cut-off indices should be avoided, since energy differences enforced in the lowest frequency bands cause visible artefacts like blocking and distortion of sharp edges.

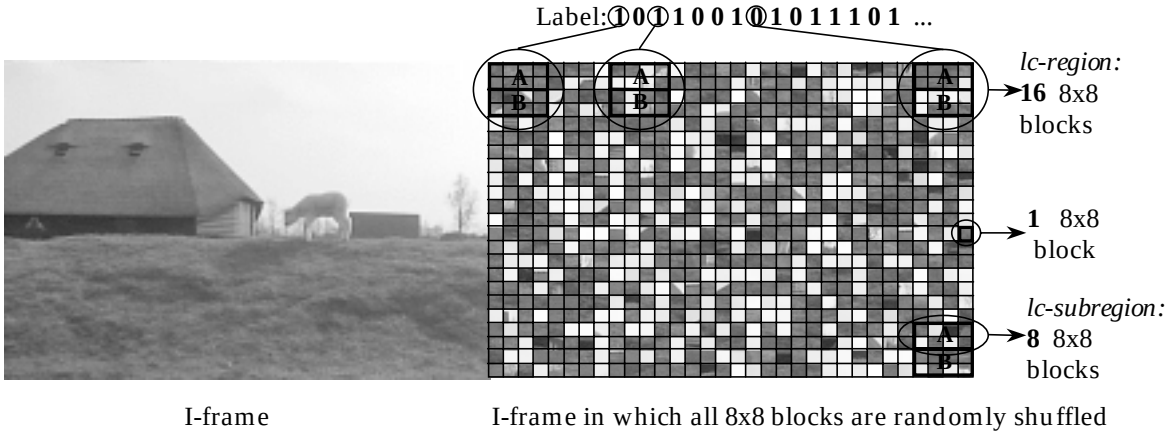


Figure 4.3.5. Label bit positions and region definitions in a shuffled frame.

In order to avoid the use of an extremely high or low cut-off index, we pseudorandomly shuffle all DCT-blocks in the image or I-frame using a secret key prior to embedding the label bits as illustrated in Figure 4.3.5.

Watermark embedding procedure:

- Shuffle all 8x8 DCT luminance blocks of an image or I-frame pseudorandomly
- FOR all label bits b_j in label string L DO
  - Select *lc-subregion A* consisting of *n/2* 8x8 DCT-blocks,
    Select *lc-subregion B* consisting of *n/2* other blocks (Fig. 4.3.5)
  - Calculate cut-off index c:

$$c(n,Q_{jpeg},D,c_{min}) = \max\{c_{min}, \max\{g \in \{1,63\} \mid (E_A(g,n,Q_{jpeg}) > D) \wedge (E_B(g,n,Q_{jpeg}) > D)\}\}$$

where $$E_{A,B}(c,n,Q_{jpeg}) = \sum_{d=0}^{n/2-1} \sum_{i \in S(c)} ([\theta_{i,d}]_{Q_{jpeg}})^2$$

$$S(c) = \{h \in \{1,63\} \mid (h \geq c)\}$$

  - IF ($b_j = 0$) THEN discard coefficients of area B in $S(c)$
    IF ($b_j = 1$) THEN discard coefficients of area A in $S(c)$
- Shuffle all 8x8 DCT luminance blocks back to their original locations

Watermark extraction procedure:

- Shuffle all 8x8 DCT luminance blocks of an image or I-frame pseudorandomly

- FOR all label bits b_j in label string L DO
  - Select *lc-subregion A* consisting of *n/2* 8x8 DCT-blocks,
    Select *lc-subregion B* consisting of *n/2* other blocks (Fig. 4.3.5)
  - Calculate cut-off index *c*:

$$c^{(\text{extract})}(n,Q'_{jpeg},D') = \max\{\ \max\{g\in\{1,63\} \mid E_A(g,n,Q'_{jpeg}) > D'\},\ \max\{g\in\{1,63\} \mid E_B(g,n,Q'_{jpeg}) > D'\}\ \}$$

$$\text{where} \quad E_{A,B}(c,n,Q_{jpeg}) = \sum_{d=0}^{n/2-1} \sum_{i\in S(c)} ([\theta_{i,d}]_{Q_{jpeg}})^2$$

$$S(c) = \{h \in \{1,63\} \mid (h \geq c)\}$$

  - Calculate energy difference:

$$D = E_A(c^{(\text{extract})},n,Q'_{jpeg}) - E_B(c^{(\text{extract})},n,Q'_{jpeg})$$

IF $(D > 0)$ THEN $b_j=0$
ELSE $b_j=1$

Figure 4.3.6. Complete procedure for watermark embedding and extraction.

This does not pose any problems in practice when using MPEG or JPEG streams, because effectively we now select randomly DCT-blocks from the compressed stream to define an *lc-region* instead of spatially neighboring blocks. As a result of the shuffling operation, smooth 8x8 DCT-blocks and textured 8x8 DCT-blocks will alternate in the lc-subregions. The energy is now distributed more equally over all lc-regions, significantly diminishing the chance of a completely smooth or textured lc-subregion. Another major advantage of the shuffle operation is that each label bit is scattered over the image or frame, which makes it impossible for an attacker to localize the lc-subregions. The complete watermark embedding and extracting procedures are shown in Figure 4.3.6.

4.4 Evaluation of the DEW algorithm for MPEG video data

4.4.1 Payload of the watermark

To evaluate the effect of the label bit-rate on the visual quality of the video stream we applied the DEW algorithm to the "sheep-sequence" coded at different bit-rates. The label bit-rate is fixed and determined by *n*, the number of 8x8 DCT-blocks per lc-region. In the experiments we omitted the optional re-quantization stage (Q_{jpeg}=100). Over a wide range of sequences we have found a reasonable setting for the energy difference D = 20 and the detection threshold D' = 15. The cut-off indices *c* for each label bit are allowed to vary in the range from 6 to 63 (c_{min}=6). Informal subjective tests show that the watermark, embedded with n = 32, is not noticeable in video streams coded at 8 and 6 Mbit/s. If MPEG streams coded at a lower bit-rate are labeled with n = 32, blocking artefacts around edges of smooth objects appear. By increasing *n* further to 64 the artefacts disappear in the MPEG stream coded at 4 Mbit/s. At a rate of 1.4 and 2 Mbit/s the compression artefacts always dominate the additional degradations due to watermarking.

Table 4.4.1. Number of 8x8 DCT-blocks per bit, number of bits discarded by the watermarking process, percentage label bit errors and label bit-rate for the "Sheep-sequence" coded at different bit-rates.

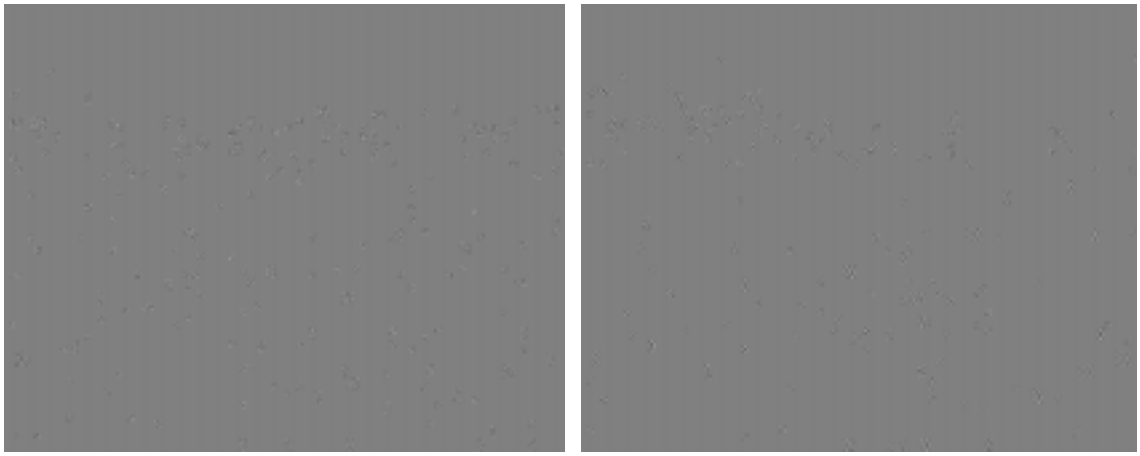
Video bit-rate	*n*	Discarded bits	% Bit errors	Label bit-rate
1.4 Mbit/s	64	1.6 kbit/s	24.6	0.21 kbit/s
2.0 Mbit/s	64	4.6 kbit/s	0.1	0.21 kbit/s
4.0 Mbit/s	64	3.8 kbit/s	0.0	0.21 kbit/s
6.0 Mbit/s	32	7.2 kbit/s	0.0	0.42 kbit/s
8.0 Mbit/s	32	6.6 kbit/s	0.0	0.42 kbit/s

In Table 4.4.1 the results of the experiments are listed. The third column shows the number of bits, which are discarded by the watermark embedding process. The fourth column presents the percentage bit errors found by extracting the label *L'* from the watermarked stream and comparing *L'* with the originally embedded one, *L*. Bit errors occur if the embedding algorithm selects cut-off indices below c_{min}. In this case the energy difference can not be enforced. It appears that not enough high frequency coefficients exist in the compressed stream coded at 1.4Mbit/s to create the energy differences *D* for the label bits, since only 75% of the extracted label bits are correct.

4.4.2 Visual impact of the watermark

In Figure 4.4.1a the original I-frame of the MPEG-2 coded sheep-sequence is represented. The sequence is MPEG-2 encoded at 8 Mbit/s. Figure 4.4.1b shows the corresponding watermarked I-frame. In Figure 4.4.1c the strongly amplified difference between the original I-frame and the watermarked frame is presented. Figure 4.4.1d shows the difference between the original I-frame coded at 4Mbit/s and the corresponding watermarked frame.

(a) Unwatermarked frame *I* (8 Mbit/s) **(b)** Watermarked Frame I_w (8 Mbit/s)



(c) Difference $W(x,y)=I-I_w$ (8Mbit/s)
label bit-rate 0.42 kbit/s

(d) Difference $W(x,y)=I-I_w$ (4Mbit/s)
label bit-rate 0.21 kbit/s

Figure 4.4.1. DEW watermarking by discarding DCT coefficients.

It appears that all degradations are located in DCT-blocks with a relatively large number of high frequency DCT-components, textured blocks and blocks with edges. If we compare Figure 4.4.1 with Figure 3.4.3, we see that the DEW watermarking method causes fewer differences per frame than the LSB-based method described in Section 3.4, although the differences per block are larger. Using the Bit Domain Labeling method a DCT-coefficient is only altered by one quantization level, here DCT-coefficients are completely discarded.

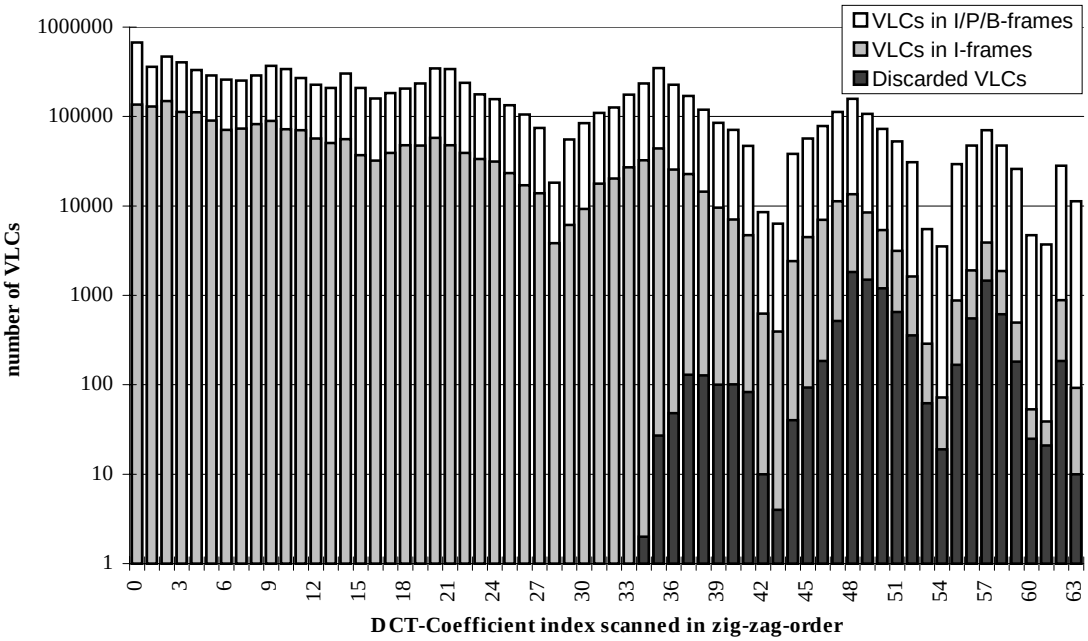


Figure 4.4.2. Number of VLCs coding non-zero DCT coefficients in 10 seconds MPEG-2 video coded at 8Mb/s vs. number of VLCs discarded by the watermark.

In Figure 4.4.2 a histogram is shown of the sheep-sequence coded at 8 Mbit/s. The number of all VLCs (including the fixed length codes) that code non-zero DCT coefficients, the number of all VLCs in the I-frames and the number of discarded VLC*s* are plotted along the logarithmic vertical axis. The DCT-coefficient index scanned in the zig-zag order ranging from 0 to 63 is shown on the horizontal axis. From Figure 4.4.2 it appears that only high frequency DCT-coefficients with an index above 33 are discarded for this particular parameter setting.

The histograms of the cut-off indices in the "sheep-sequence coded at 1.4 and 8Mbit/s" are plotted in Figure 4.4.3. The minimum cut-off index for the "sheep-sequence" coded at 8Mbit/s is 33, for a stream coded at 1.4Mbit/s the minimum is equal to the minimum cut-off index c_{min}=6. The lower the bit-rate is, the lower the cut-off indices have to be because of the lack of high energy components in the compressed video stream.

The visual impact of the labeling will be much smaller if the degradations introduced by discarding DCT-coefficients are distributed more or less uniformly over the frame. Removing all VLCs from a few textured blocks will cause highly visible artefacts.

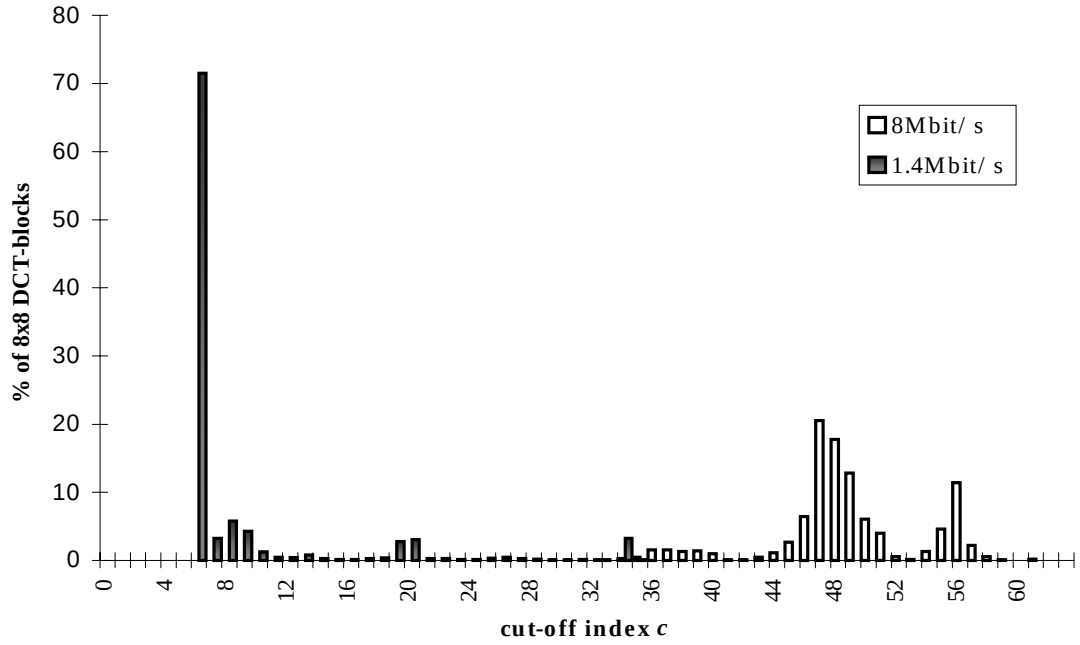


Figure 4.4.3. Histograms of the cut-off indices in an MPEG-2 sequence coded at 1.4 and 8Mb/s, label bit-rates are respectively 0.21kbit/s and 0.42kbit/s.

In Figure 4.4.4 a histogram is shown of 10 seconds of the watermarked "sheep-sequence" coded at 8 Mbit/s. On the vertical axis the number of discarded VLCs per 8x8 DCT-block is shown. The number of 8x8 DCT-blocks that contain this amount of discarded VLCs is plotted along the logarithmic horizontal axis.

It appears that 95% of all coded 8x8 blocks in the I-frames are not affected by the DEW algorithm. From an *lc-region* only the DCT-coefficients above a certain cut-off index in the half, an *lc-subregion*, are eliminated. This means that from an *lc-subregion* only a few (average 10%) 8x8 blocks have energy above the cut-off index.

Video bit-rate: 8 Mb/s
Label bit-rate: 0.42kb/s

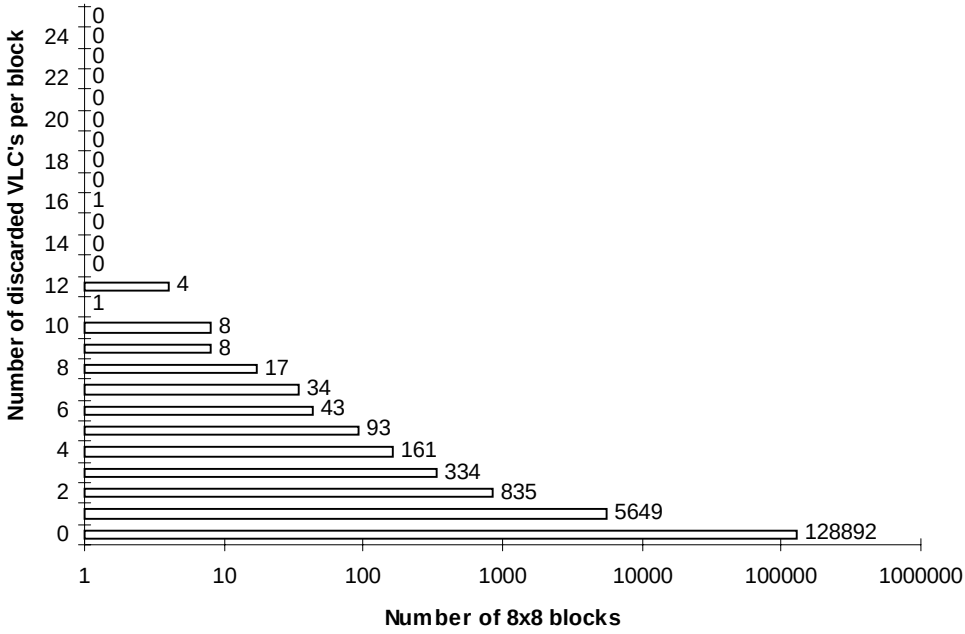


Figure 4.4.4. Number of discarded VLCs per 8x8 DCT-block.

Like in the Bit Domain watermarking algorithm described in Section 3.4 a limit T_m can be set on the number of VLCs per 8x8 block that are discarded during the watermarking process. Whereas in the Bit Domain watermarking algorithm this limit decreases the label bit-rate, the DEW algorithm has a fixed label bit-rate. Instead, setting a limit T_m affects the robustness of the label. If some DCT-coefficients in one 8x8 block of an *lc-subregion* are not eliminated, because the limit T_m prohibits it, in the worst case one label bit error can occur if the label extracted from this stream is compared with the originally embedded one. However, since each label bit is dependent on *n* 8x8 blocks, the likelihood that this error occurs is relatively small.

Table 4.4.2. Worst case % label bit errors introduced by limit T_m, the maximum number of discarded VLCs per 8x8 block (Video bit-rate 8Mbit/s, Label bit-rate 0.42kbit/s).

T_m =Max. number of discarded VLCs per block	Worst case % bit errors
2	17%
3	9%
4	5%
5	3%
6	2%
Unlimited	0%

In Table 4.4.2 the worst case percentages bit errors, which are introduced in the label of the "sheep-sequence" coded at 8 Mbit/s, are listed for several values of T_m. With proper error correcting codes on the label stream, the number of VLCs to be removed can be greatly limited at the advantage of a better visual quality without significantly effecting the label retrieval.

4.4.3 Drift

Since P- and B-frames are predicted from I-and P-frames, the degradations introduced by watermarking in the I-frames appear also in the predicted frames. Because the P- and B-frames are only partially predicted from other frames and partially intra coded, the degradations will fade out. No degradations are introduced in the intra coded parts of the predicted frames by the labeling. The error fade-out can clearly be seen in Figure 4.4.5, where the difference MSE_l-MSE_u is plotted. The MSE_u is the MSE per frame between the uncompressed "sheep-sequence" and the sequence coded at 8Mbit/s. The MSE_l is the MSE per frame between the uncompressed sequence and the labeled sequence coded at 8Mbit/s.

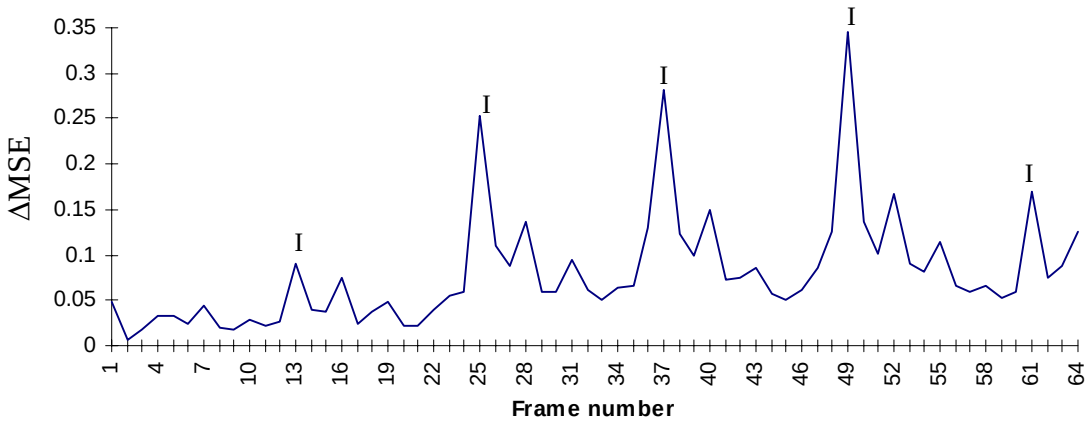


Figure 4.4.5. ΔMSE of the watermarked "sheep-sequence" coded at 8Mbit/s with a label bit-rate of 0.42kbit/s.

The average *PSNR* between the MPEG-compressed original and the uncompressed original is 37dB. If the labeled compressed video stream at 8Mbit/s is compared with the original compressed stream, the *ΔMSE* causes an average *ΔPSNR* of 0.06dB and a maximum *ΔPSNR* of 0.3dB. It appears that this method has less impact on the average *ΔPSNR* and more impact on the maximum *ΔPSNR* than the method described in Section 3.4. From the *ΔPSNR* values we conclude that no drift compensation signal is required.

4.4.4 Robustness

Unlike the LSB-based methods described in Section 3.4, the watermark embedded by the DEW algorithm can not be removed by watermarking the video stream again using another watermark if another pseudorandom block shuffling is used. Other more time-consuming, computationally and memory (disk) demanding methods have to be applied to

the watermarked compressed video stream to attempt to remove the watermark. For simple filtering techniques the compressed stream must be decoded and completely re-encoded. A less complex and disk demanding, but still very computationally demanding operation would be transcoding. To see if the watermark is resistant to transcoding or re-encoding at a lower bit-rate, the following experiment is performed. The "sheep-sequence" is MPEG-2 encoded at 8 Mbit/s and this compressed stream is watermarked (n = 32). Hereafter, the watermarked video sequence is transcoded at different lower bit-rates.

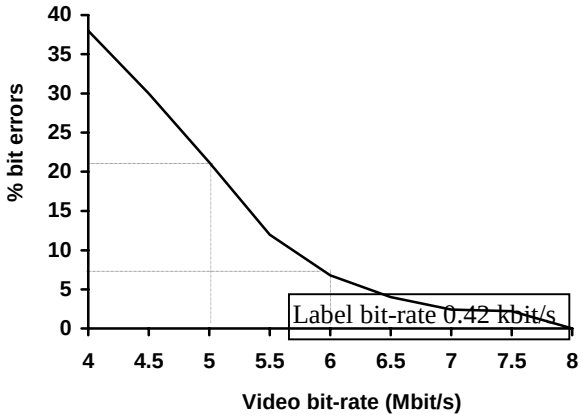


Figure 4.4.6. % Bit errors after transcoding a watermarked 8Mbit/s MPEG-2 sequence at a lower bit-rate.

The label bit strings are extracted from the transcoded video streams and each label bit string is compared with the originally embedded label bit string. If 50% bit errors are made the label is completely removed. The bit errors introduced by decreasing the bit-rate are represented in Figure 4.4.6. It appears that if the video bit-rate is decreased by 25%, only 7% label bit errors are introduced. Even if the video bit-rate is decreased by 38%, 79% of the label bit stream can be extracted correctly. Error correcting codes can further improve this result.

For embedding a label bit in an lc-region the DEW algorithm removes some high frequency DCT-coefficients in one of the lc-subregions. This can be seen as locally applying a low-pass filter to an lc-subregion. To detect the label-bit, the amount of high frequency components in the two lc-subregions is compared. If small geometrical distortions are applied to the video data e.g. shifting, there is a mismatch between the lc-regions chosen during the embedding phase and the lc-regions chosen during the detection phase. Parts of the lc-region chosen during the embedding phase are in the detection phase replaced by adjacent lc-regions. Although, the adjacent lc-regions introduce high frequency components in the low-pass filtered lc-subregions, the difference in high frequency components is still measurable if the geometrical distortions are relative small. The DEW algorithm should therefore exhibit some degree of resistance to geometrical distortions like line-shifting. The experiments performed in the next chapter show that the DEW algorithm is resistant to line-shifts up to 3 pixels.

4.5 Extension of the DEW concept for EZW-coded images

The DEW concept is not only suitable for MPEG/JPEG compressed video data, but can also be applied to video compressed using embedded zero-tree wavelets [Sha93]. For an explanation about wavelet-based compression the reader is referred to [Aka96], [Bar94] and [Vet95]. In MPEG/JPEG compressed video data the natural starting point for computing energies and creating energy differences are the DCT-blocks. In embedded zero-tree wavelet compressed video data the natural starting point is the hierarchical tree structure. Instead of embedding a label bit by enforcing an energy difference between two lc-subregions of DCT-blocks, we now enforce energy differences between two sets of hierarchical trees. Figure 4.5.1 shows a typical tree structure that is used in the wavelet compression of images or video frames.

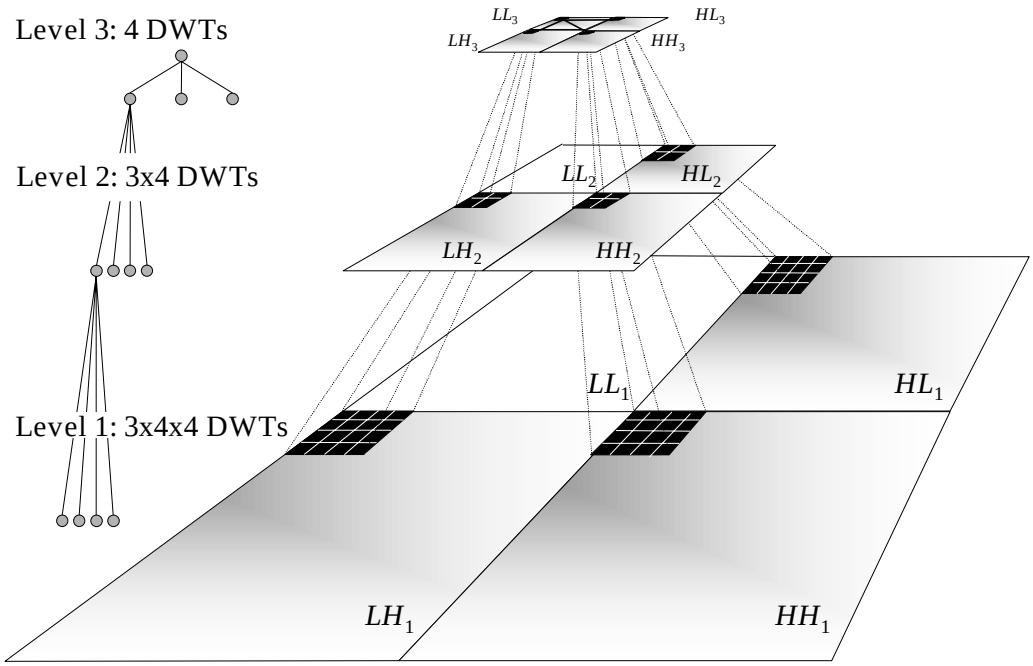


Figure 4.5.1. Hierarchical tree structure of a DWT 3-level decomposition.

As can be seen in Figure 4.5.1, a tree in this 3-level wavelet decomposed image or video frame starts with a root Discrete Wavelet Transform (DWT) coefficient in the LL_3 band and counts 64 DCT coefficients. Unlike the DCT situation where the discarding of high frequency DCT coefficients is implicitly restricted by the zig-zag scan order, in wavelet compressed video data different ways of pruning the hierarchical trees can be envisioned.

The simplest case to remove energy is to truncate the trees below the hierarchical levels. A scheme in which trees are pruned coefficient-by-coefficient allows for finer tuning of the energy difference and for minimization of the visual impact. Therefore we have numbered the DWT coefficients of the hierarchical tree and defined a pseudo zig-zag scan order as illustrated in Figure 4.5.2.

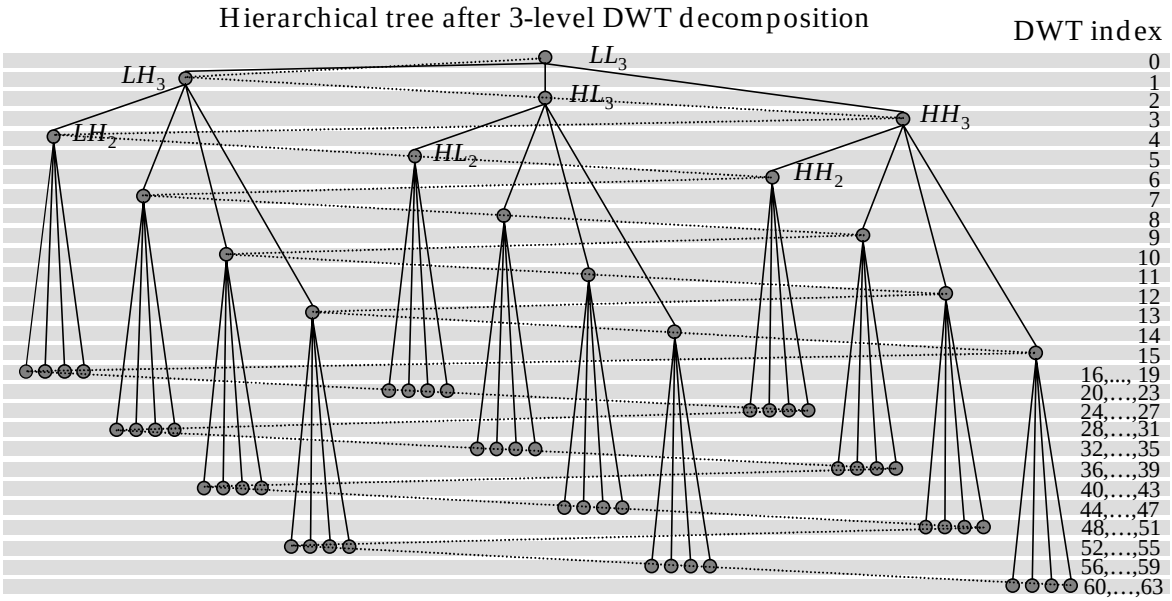


Figure 4.5.2. DWT coefficient numbering and pseudo zig-zag scan order.

This pseudo zig-zag order is not the only possible way to order the DWT coefficients. More sophisticated orderings are possible that take the human visual system into account. The advantage of using the straightforward numbering defined by Figure 4.5.2 is that we now can use the same scheme as we used for the DCT situation. Only two minor changes are required. First, the quantization step in the energy definitions has to be adapted, the DWT coefficients are now optionally re- or pre-quantized using a uniform quantizer instead of the standard JPEG quantization procedure. Second, not the 8x8 blocks are shuffled, but the roots of the hierarchical trees are pseudorandomly shuffled.

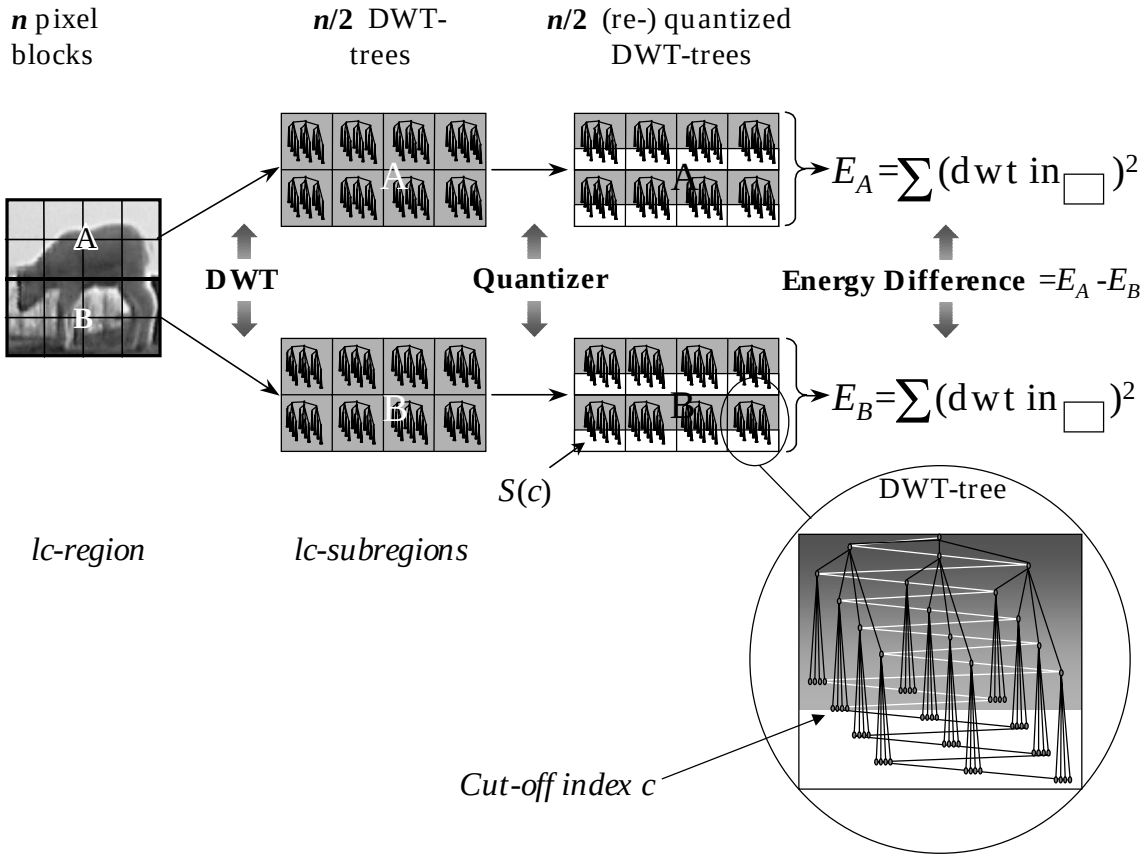


Figure 4.5.3. Energy difference calculation in an lc-region.

The complete procedure to calculate the energy difference in an lc-region is graphically illustrated in Figure 4.5.3.

(a) Watermarked image **(b)** Difference $W(x,y)=I(x,y)-I_w(x,y)$

Figure 4.5.4. Level 3 EZW coded image watermarked using the DEW concept.

In Figure 4.5.4a an example is given of the DEW algorithm applied to the embedded zero-tree wavelet coded Lena-image using a 3-level wavelet decomposition. Here a label bit string of 64 label bits is embedded, using lc-regions of 64 hierarchical trees. It can clearly be seen that the watermark in this variation of the DEW algorithm also adapts to the image content.

4.6 Discussion

In this chapter we introduced the Differential Energy Watermarking (DEW) concept. Unlike the correlation based method with drift compensation described in Section 3.3.2, the DEW embedding and extraction algorithm can completely be performed in the coefficient domain and does not require a drift compensation signal. The encoding parts of the coefficient domain watermarking concept can even be omitted. The complexity of the DEW watermarking algorithm is therefore only slightly higher than the LSB methods described in Section 3.4. Furthermore, the DEW label bit-rate is about 25 times higher than the label bit-rate of the correlation based methods described in Section 3.3. Like these correlation based methods, a watermark embedded with the DEW concept can also be embedded and extracted from raw video data and the label string is resistant to re-labeling. Besides the low complexity and the much higher label bit-rate the advantages of the DEW concept over other methods are that it provides a parameter Q_{jpeg} to anticipate to re-encoding attacks, that it exhibits some degree of resistance to geometrical distortions like line-shifting and that it is directly applicable to video data compressed using other coders, for instance embedded zero-tree wavelet coders.

Since many parameters are involved in the watermark embedding process of the DEW algorithm (n, Q_{jpeg}, D and c_{min},) heuristically determining optimal parameter settings is quite an elaborate task. Therefore in the next chapter a statistical model is derived that can be used to find these optimal parameter settings for DCT based coders.

Chapter 5

Finding Optimal Parameters by Modeling the DEW Algorithm

5.1 Introduction

The performance of the DEW algorithm proposed in the previous chapter heavily depends on the four parameters used in the watermark embedding phase. All parameters involved in the watermarking process are presented in Figure 5.1.

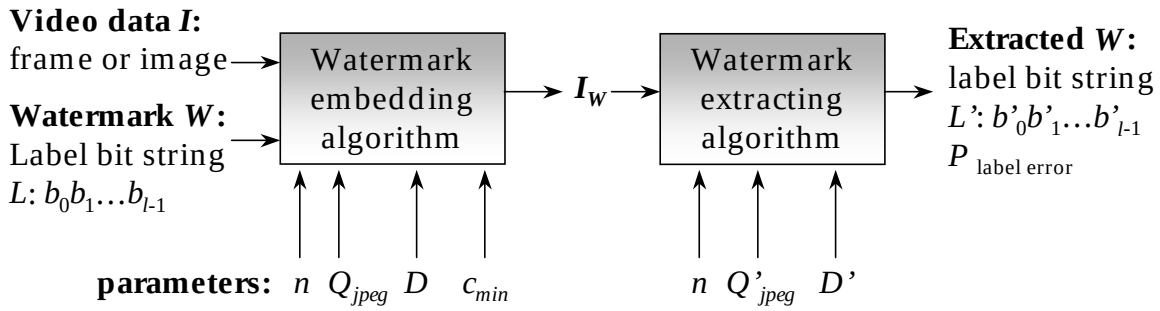


Figure 5.1. Parameters involved in the DEW watermarking process.

The first parameter is the number of 8×8 DCT blocks n that is used to embed a single information bit of the label bit string. The larger n is chosen, the more robust the watermark becomes against watermark-removal attacks, but the fewer information bits can be embedded into an image or a single frame of a video sequence.

The second parameter controls the robustness of the watermark against re-encoding attacks. In a re-encoding attack the watermarked image or video is partially or fully decoded and subsequently re-encoded at a lower bit-rate. Our method anticipates the re-encoding at lower bit-rates up to a certain minimal rate. Without loss of generality we will elaborate on the re-encoding of *JPEG* compressed images, in which case the anticipated re-encoding bit-rate can be expressed by the JPEG quality factor setting Q_{jpeg}. The smaller Q_{jpeg} is the more robust the watermark becomes against re-encoding attacks. However, for decreasing Q_{jpeg} increasingly more (high to middle frequency) DCT coefficients have to be removed upon embedding of the watermark, which leads to an increasing probability for artifacts to become visible due to the presence of the watermark.

The third parameter is the energy difference D that is enforced to embed a label bit. This parameter determines the number of DCT-coefficients that are discarded. Therefore, it directly influences the visibility and robustness of the label bits. Increasing D increases the probability that artifacts become visible and increases the robustness of the label.

The fourth parameter is the so-called minimal *cut-off index* c_{min}. This value represents the smallest index – in zigzag scanned fashion – of the DCT coefficient that is allowed to be removed from the image data upon embedding the watermark. The smaller c_{min} is chosen, the more robust the watermark becomes but at the same time, image degradations due to removing high frequency DCT coefficients may become apparent. For a given c_{min} there is a certain probability that a label bit cannot be embedded. Consequently, sometimes a *random* information bit will be recovered upon watermark detection, which is denoted as a *label bit error* in this chapter. Clearly, the objective is to make the probability for label bit errors as small as possible.

In order to optimize the performance of the DEW watermark technique, the above mentioned parameters have to be determined. In the previous chapter we have used experimentally determined settings for these parameters. For a given watermark and image or video frame this is, however, an elaborate process. In this chapter, we will show that it is possible to derive an expression for the label bit error probability P_{be} as a function of the parameters n, Q_{jpeg} and c_{min}. The relations that we derive analytically describe the behavior of the watermarking algorithm, and they make it possible to select suitable values for the three parameters (n, Q_{jpeg}, c_{min}), as well as suitable error correcting codes for dealing with label bit errors [Lan99b] and [Lan99c]. Although the expressions in this chapter are derived and validated for JPEG compressed images, they are also directly applicable to MPEG compressed I-frames.

In Section 5.2, we derive an analytical expression for the probability mass function (PMF) of the cut-off indices. In Section 5.3, this PMF is verified with real-world data. After deriving and validating the obtained PMF, we use the PMF to find the probability that a label string cannot be recovered correctly in Section 5.4 and the optimal parameter settings (n, Q_{jpeg}, c_{min}) in Section 5.5. Subsequently in Section 5.6, we experimentally validate the results from Section 5.5. The chapter concludes with a discussion on the DEW watermarking technique and its optimization in Section 5.7.

5.2 Modeling the DEW concept for JPEG compressed video

When operating the *DEW* algorithm, different values for the cut-off index are obtained. Insight in the actually selected cut-off indices is important since the cut-off indices used determine the quality and robustness of the *DEW*. Therefore, in this section we will derive the probability mass function (PMF) for the cut-off index based on a stochastic model for DCT coefficients. This PMF depends only on the parameters Q_{jpeg} and n.

5.2.1 PMF of the cut-off index

The optimal cut-off index varies per label bit that we wish to embed. Therefore, it can be interpreted as a stochastic variable that depends on n, Q_{jpeg}, D, and c_{min}, i.e. $C(n,Q_{jpeg},D,c_{min})$. Mathematically, this gives the following expression for determining C (see Sections 4.2 and 4.3):

$$C(n,Q_{jpeg},D,c_{min}) = \max\{c_{min}, \max\{g \in \{1,63\} | (E_A(g,n,Q_{jpeg}) > D) \wedge (E_B(g,n,Q_{jpeg}) > D)\}\} \tag{5.2.1a}$$

where $$E_A(c,n,Q_{jpeg}) = \sum_{d=0}^{n/2-1} \sum_{i \in S(c)} ([\theta_{i,d}]_{Q_{jpeg}})^2 \tag{5.2.1b}$$

$$S(c) = \{h \in \{1,63\} \mid (h \geq c)\} \tag{5.2.1c}$$

In order to be able to compute the PMF of the cut-off index, we first assume that the energy difference D in Equation 5.2.1a is chosen in the range $[1,D_{max}(Q_{jpeg})]$. Here $D_{max}(Q_{jpeg})$ indicates the maximum of the range of energies defined by Equation 5.2.1b that does *not occur* in quantized DCT blocks because of the JPEG or MPEG compression process.

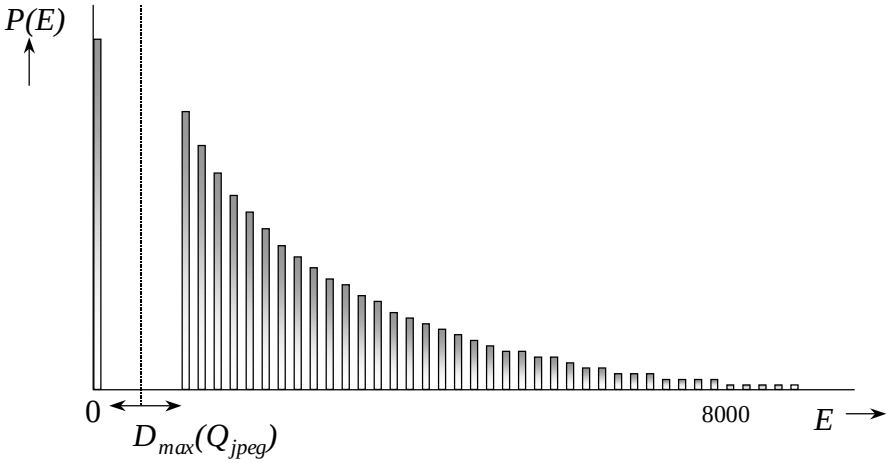


Figure 5.2.1. Energy histogram of $E_{A,B}$ for a wide range of parameters (c,n,Q_{jpeg}).

Figure 5.2.1 illustrates this effect by showing an histogram of the energy $E(c,n,Q_{jpeg})$ for a wide range of values of c, n, and Q_{jpeg}. We notice a clear "gap" in the histogram for smaller energies, because DCT blocks with that small amount of energy can no longer exist after compression.

In general the maximum $D_{max}(Q_{jpeg})$ depends on how heavy the image has been compressed, i.e. it depends on Q_{jpeg}. The smaller Q_{jpeg} is, the larger $D_{max}(Q_{jpeg})$ will be. Mathematically this relation is given by:

$$\begin{aligned} D_{\max}(Q_{jpeg}) &= \left(F(Q_{jpeg}) \min_i(W_i)\right)^2 \\ F(Q_{jpeg}) &= \begin{cases} 50 / Q_{jpeg} & Q_{jpeg} < 50 \\ (100 - Q_{jpeg}) / 50 & Q_{jpeg} \geq 50 \end{cases} \end{aligned} \tag{5.2.2}$$

where $F(Q_{jpeg})$ denotes the coarseness of the quantizer used, and W_i is the i-th element ($i \in [c_{\min},63]$) of the zigzag scanned standard JPEG luminance quantization table [Pen93].

Theorem I:

If the enforced energy difference D is chosen in the range $[1,D_{max}(Q_{jpeg})]$, where $D_{max}(Q_{jpeg})$ is defined by Equation 5.2.2, and if we do not constrain the cut-off index by c_{min}, the PMF of the cut-off index is given by:

$$P[C(n,Q_{jpeg})=c] = P[E(c,n,Q_{jpeg}) \neq 0]^2 - P[E(c+1,n,Q_{jpeg}) \neq 0]^2 \qquad (5.2.3)$$

where $E(c,n,Q_{jpeg})$ is defined in Equation 5.2.1b. Observe that in this theorem $C(n,Q_{jpeg})$ – besides being not constrained by c_{min} – is no longer dependent on D due to the wide range of values in which D can be selected.

Proof:

We first rewrite the definition of the cut-off index in Equation 5.2.1a to avoid the maximum operators as follows:

$$P[C(n,Q_{jpeg},D)=c] = P[\ \{(E_A(c,n,Q_{jpeg})>D) \wedge (E_B(c,n,Q_{jpeg})>D)\} \wedge \{(E_A(c+1,n,Q_{jpeg})<D) \vee (E_B(c+1,n,Q_{jpeg})<D)\}\ ] \qquad (5.2.4)$$

In the following, we will drop the dependencies on n and Q_{jpeg} of the energies for notational simplicity. To calculate Equation 5.2.4 we need to have an expression for probabilities of the form $P[E_A(c)>D]$. As illustrated by Figure 5.2.1, the histogram of $E_A(c)$ is zero for small $E_A(c)$s because the quantization process maps many small DCT coefficients to zero. As a consequence, the energy defined in Equation 5.2.1b is either equal to 0 (for instance for large values of c), or the energy has a value larger than the smallest non-zero squared *quantized* DCT coefficient in the lc-subregion under consideration. This value has been defined as $D_{max}(Q_{jpeg})$ in Equation 5.2.2. Since we always choose the value of D smaller than $D_{max}(Q_{jpeg})$, Equation 5.2.4 can be simplified as:

$$P[C(n,Q_{jpeg})=c] = P[\ \{(E_A(c)\neq 0) \wedge (E_B(c\neq 0)\} \wedge \{(E_A(c+1)=0) \vee (E_B(c+1)=0)\}\ ] \qquad (5.2.5)$$

Due to the random shuffling of the positions of the DCT blocks, we can now assume that $E_A(c)$ and $E_B(c)$ are mutually independent. Following several standard probability manipulations, Equation 5.2.5 can then be rewritten as follows:

$$\begin{aligned} P[C(n)=c] &= P[(E_A(c)\neq 0) \wedge (E_B(c)\neq 0) \wedge (E_A(c+1)=0)] \\ &\quad + P[(E_A(c)\neq 0) \wedge (E_B(c)\neq 0) \wedge (E_B(c+1)=0)] + \\ &\quad - P[(E_A(c)\neq 0) \wedge (E_B(c)\neq 0) \wedge (E_A(c+1)=0) \wedge (E_B(c+1)=0)] \\ &= P[(E_A(c)\neq 0) \wedge (E_A(c+1)=0)]\ P[E_B(c)\neq 0)\ ] \\ &\quad + P[(E_B(c)\neq 0) \wedge (E_B(c+1)=0)]\ P[E_A(c)\neq 0)] \\ &\quad - P[(E_A(c)\neq 0) \wedge (E_A(c+1)=0)]\ P[(E_B(c)\neq 0) \wedge (E_B(c+1)=0)] \end{aligned} \qquad (5.2.6)$$

We first expand the first term of Equation 5.2.6 using conditional probabilities:

$$\begin{aligned} &P[(E_A(c)\neq 0) \wedge (E_A(c+1)=0)] \\ &\quad = 1 - P[(E_A(c+1)=0) \wedge (E_A(c)=0)] - P[(E_A(c+1)\neq 0) \wedge (E_A(c)\neq 0)] \end{aligned}$$

$$\begin{aligned} & - P[(E_A(c+1)\neq 0) \wedge (E_A(c)=0)] \\ = 1 & - P[E_A(c+1)=0 \,/\, E_A(c)=0]\, P[E_A(c)=0] \\ & - P[E_A(c)\neq 0 \,/ E_A(c+1)\neq 0]\, P[E_A(c+1)\neq 0] \\ & - P[E_A(c)=0 \,/\, E_A(c+1)\neq 0]\, P[E_A(c+1)\neq 0] \end{aligned} \qquad (5.2.7)$$

It can directly be seen from the definition in Equation 5.2.1b that $E_A(c)$ is a strictly non-increasing function. Therefore, if there is no energy above cutoff index c, i.e., $E_A(c)=0$, there is also no energy above c+1, i.e. $E_A(c+1)=0$. This yields $P[E_A(c+1)=0 \,/\, E_A(c)=0] = 1$. On the other hand, if there is energy above cutoff index c+1, the same amount of energy or more must be present above cutoff index c, therefore $P[E_A(c)\neq 0 \,/ E_A(c+1)\neq 0] = 1$ and $P[E_A(c)=0 \,/\, E_A(c+1)\neq 0] = 0$. Substitution of these conditional probabilities into Equation 5.2.7 gives the following result:

$$\begin{aligned} P[(E_A(c)\neq 0) \wedge (E_A(c+1)=0)] &= 1 - P[E_A(c)=0] - P[E_A(c+1)\neq 0] \\ &= P[E_A(c)\neq 0] - P[E_A(c+1)\neq 0] \end{aligned} \qquad (5.2.8)$$

A similar approach can be followed to simplify the other terms in Equation 5.2.6. This results in the following expression:

$$\begin{aligned} P[C(n)=c] &= (P[E_A(c)\neq 0] - P[E_A(c+1)\neq 0])\, P[E_B(c)\neq 0] \\ &\quad + (P[E_B(c)\neq 0] - P[E_B(c+1)\neq 0])\, P[E_A(c)\neq 0] \\ &\quad + (P[E_A(c)\neq 0] - P[E_A(c+1)\neq 0])\, (P[E_B(c)\neq 0] - P[E_B(c+1)\neq 0]) \\ &= P[E_A(c)\neq 0]\, P[E_B(c)\neq 0] - P[E_A(c+1)\neq 0]\, P[E_B(c+1)\neq 0] \end{aligned} \qquad (5.2.9)$$

Since the lc-subregions are both build-up from block-shuffled image data, we can assume that the probabilities in Equation 5.2.9 do not depend on the actual lc-subregion for which they are calculated, i.e. $P[E_A(c)\neq 0] = P[E_B(c)\neq 0] = P[E(c)\neq 0]$. Substitution of this equality results in Equation 5.2.3.

5.2.2 Model for the DCT-based energies

Theorem II:

If the PDF of the DCT coefficients is modeled as a generalized Gaussian distribution with shape parameter γ, then the probability that the energy $E_A(c,n,Q_{jpeg})$ is not equal to zero is given by:

$$P[E(c,n,Q_{jpeg})\neq 0] = 1 - \left[\prod_{i=c}^{63}\left\{1 - e^{-(\psi_i Q_i)^\gamma}\cdot\left(\sum_{h=0}^{\gamma^{-1}-1}\frac{(\psi_i Q_i)^{h\cdot\gamma}}{h!}\right)\right\}\right]^{\frac{n}{2}} \qquad (5.2.10)$$

where

$$\gamma^{-1} = 1, 2, 3, \ldots \qquad (5.2.11a)$$

$$\psi_i Q_i = \frac{w_i F(Q_{jpeg})}{2\sigma_i} \sqrt{\frac{(3\cdot\gamma^{-1}-1)!}{(\gamma^{-1}-1)!}} \qquad (5.2.11b)$$

Further, $F(Q_{jpeg})$ denotes the coarseness of the quantizer as defined in Equation 5.2.2, σ_i^2 represents the variance of the i-th DCT-coefficient (in zigzag scanned fashion), and w_i represents the corresponding element of standard JPEG luminance quantization table.

Proof:
The expression for $P[E_A(c) \neq 0]$ can be derived using Equation 5.2.1b. To this end we first need a probability model for the DCT coefficients θ_i. Following literature at this point, we use the generalized Gaussian distribution [Mul93] and [Var89] with shape parameter γ:

$$P(\theta_i) = \xi_i e^{-|\psi_i \cdot \theta_i|^\gamma} \qquad (5.2.12a)$$

where

$$\xi_i = \frac{\psi_i \cdot \gamma}{2(\gamma^{-1}-1)!} \quad \text{and} \quad \psi_i = \frac{1}{\sigma_i}\sqrt{\frac{(3\cdot\gamma^{-1}-1)!}{(\gamma^{-1}-1)!}} \quad \text{for} \quad \gamma^{-1}=1,2,3,\ldots \qquad (5.2.12b)$$

This PDF has zero-mean and variance σ_i^2. Typically, the shape parameter γ takes on values between 0.10 and 0.50. In a more complicated model, the shape parameter could be made dependent on the index of the DCT coefficient. We will, however, use a constant shape parameter for all DCT coefficients. Using Equation 5.2.12 we can now calculate the probability that a DCT coefficient is quantized as zero:

$$P[\hat{\theta}_i = 0] = \int_{-Q_i}^{Q_i} \xi_i \cdot e^{-|\psi_i \cdot \theta_i|^\gamma} d\theta_i = 1 - e^{-(\psi_i Q_i)^\gamma} \cdot \left(\sum_{h=0}^{\gamma^{-1}-1} \frac{(\psi_i Q_i)^{h\cdot\gamma}}{h!} \right) \qquad (5.2.13)$$

where Q_i is the coarseness of the quantizer applied to the DCT coefficients. The probability that $E_A(c,n,Q_{jpeg})$ is equal to zero is now given by the probability that all quantized DCT coefficients with index larger than c in all $n/2$ DCT blocks are equal to zero:

$$P[E(c) = 0] = \left[\prod_{i=c}^{63} P[\hat{\theta}_i = 0] \right]^{n/2} \qquad (5.2.14)$$

Equations 5.2.13 and 5.2.14 use the quantizer parameter Q_i. In JPEG this parameter is determined by the parameter w_i and the function $F(.)$ that depends on the user parameter Q_{jpeg} via Equation 5.2.2. Taking into account that JPEG implements quantization through rounding operations yields:

$$Q_i = ½\ w_i\ F(Q_{jpeg}) \qquad (5.2.15)$$

Combining Equations 5.2.12 - 5.2.15 yields Equation 5.2.10.

5.3 Model validation with real-world data

We validate Theorem I as follows. From a wide range of images we calculated the normalized histogram of $P[E(c,n,Q_{jpeg}) \neq 0]$ as a function of c. As an example we show here the situation of Q_{jpeg}=50 and n=16. Using this histogram Equation 5.2.3 is evaluated to get an estimate of the PMF $P[C(n,Q_{jpeg})=c]$. The resulting PMF is shown in Figure 5.3.1 as the dotted line. Using the same test data, we then directly calculated the histogram of $P[C(n,Q_{jpeg})=c]$ as a function of c. The resulting (normalized) histogram is shown in Figure 5.3.1 as the solid line. It shows that both curves fit well, which validates the correctness of the assumptions made in the derivation of Theorem I.

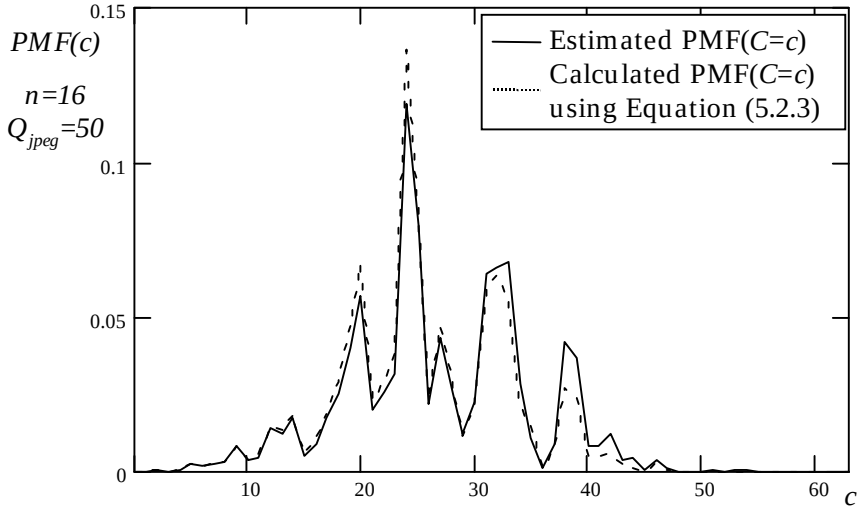


Figure 5.3.1. Probability mass function of the cut-off index $P[C(n,Q_{jpeg})=c]$ as a function of c, calculated as a normalized histogram directly from watermarked images (solid line), and calculated using the derived Theorem I (dotted line).

For the validation of Theorem II, we first need a reasonable estimate of the shape parameter γ and the variance σ_i^2 of the DCT coefficients. In fitting the PDF of the DCT coefficient we concentrated on obtaining a correct fit for the more important low frequency DCT coefficients, and obtained γ=1/7.

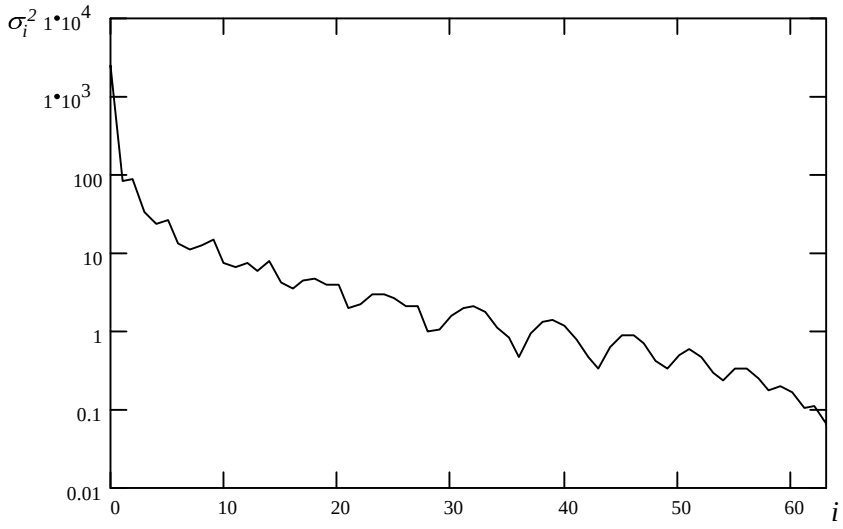
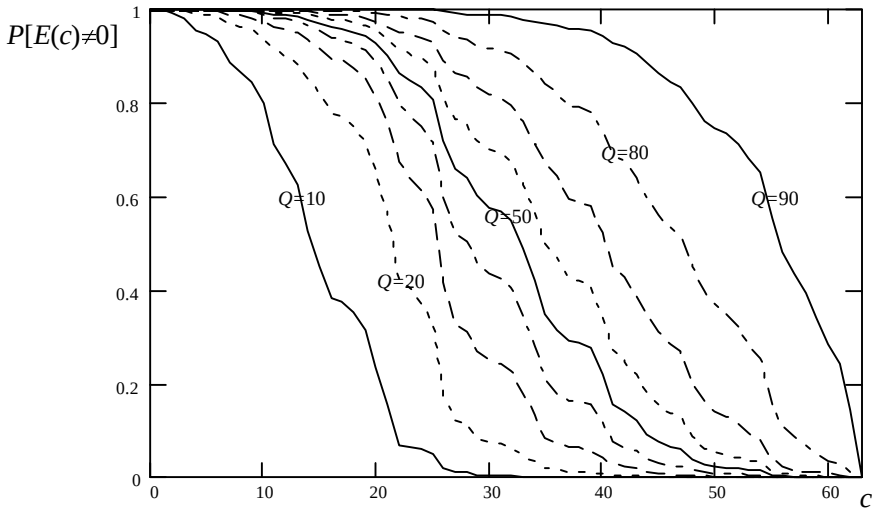
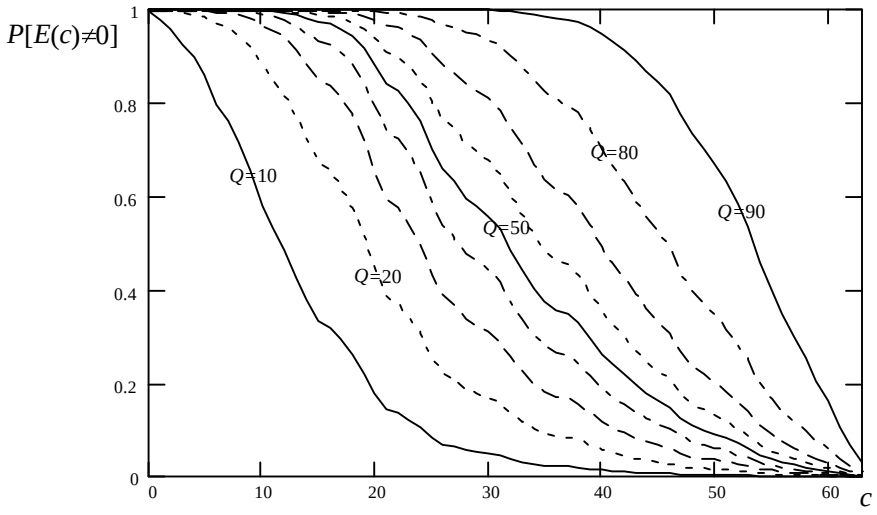


Figure 5.3.2. Measured variances of the unquantized DCT-coefficients as a function of the coefficient number along the zig-zag scan.

The variances of the DCT coefficients were measured over a large set of images, yielding Figure 5.3.2. For the time being, we will use these experimentally determined variances, but later we will replace these with a fitted polynomial function.



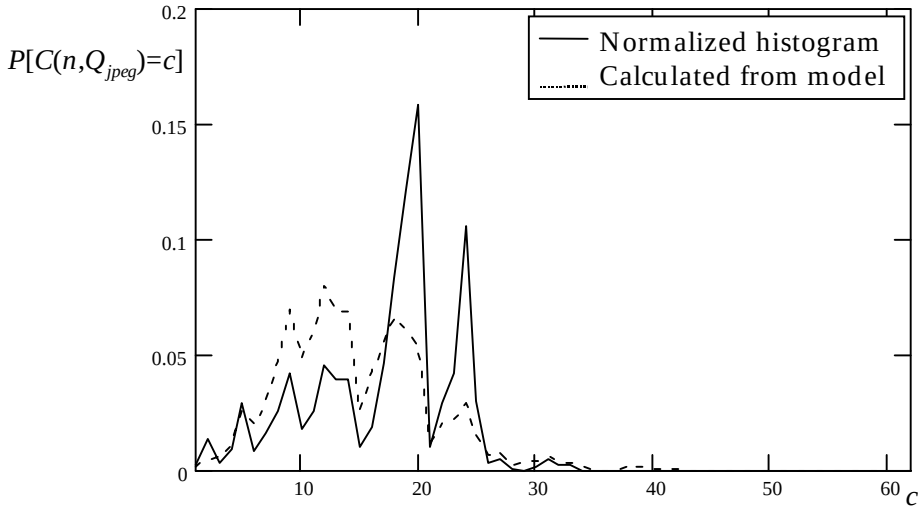
(a) $P[E(c,n,Q_{jpeg})\neq 0]$ calculated as normalized histogram directly from watermarked images



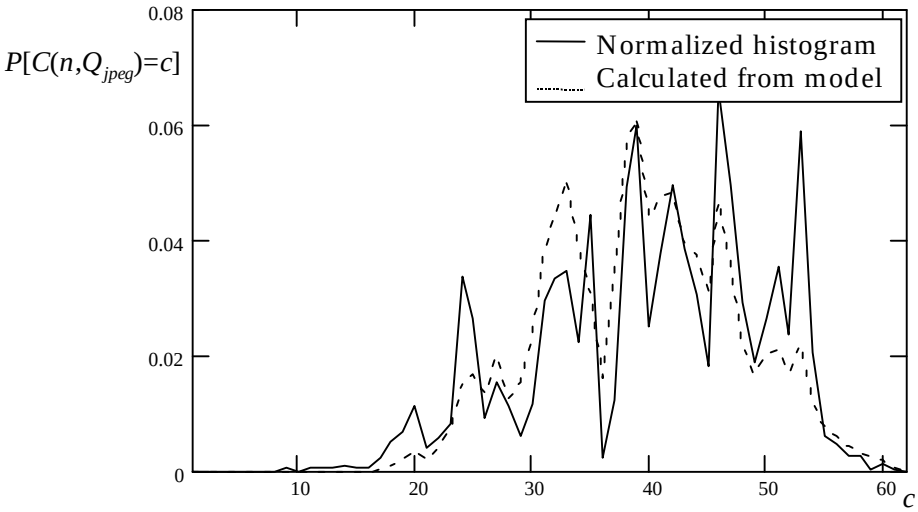
(b) $P[E(c,n,Q_{jpeg})\neq 0]$ calculated using Theorem II

Figure 5.3.3. The probabilities $P[E(c,n,Q_{jpeg})\neq 0]$ as functions of c for n=16.

In Figure 5.3.3a normalized histograms of the energy $E(c,n,Q_{jpeg})\neq 0$ are plotted for n=16 and several values of Q_{jpeg} as a function of c. In Figure 5.3.3b the probabilities $P[E(c,n,Q_{jpeg})\neq 0]$ are shown as calculated with Equation 5.2.10 from Theorem II using the measured variances of the DCT-coefficients. Comparing the Figures 5.3.3a and 5.3.3b, we see that the estimated and calculated probabilities match quite well. There are some minor deviations for very small values of Q_{jpeg} (Q_{jpeg}<15), which is the result of the imperfect model for the DCT coefficients of real image data. We consider these deviations insignificant since they occur only at very high image compression factors. We conclude that the models underlying Theorem II give results for $P[E(c,n,Q_{jpeg})\neq 0]$ that are sufficiently close to the actually observed data.



(a) PMF of $C(n,Q_{jpeg})$ for n=16 and Q_{jpeg}=20



(b) PMF of $C(n,Q_{jpeg})$ for n=16 and Q_{jpeg}=80

Figure 5.3.4. Probability mass function of $C(n,Q_{jpeg})$, calculated as the normalized histogram directly from watermarked image data (solid line), and calculated using Equations 5.2.3 and 5.2.10.

By combining Theorem I and II, we can derive PMFs of the cut-off index as a function of the parameters n and Q_{jpeg} based merely on the variances of the DCT coefficients. To validate the combined theorems we compared the PMFs calculated using the Equations 5.2.3 and 5.2.10 with the normalized histograms directly calculated on a wide range of images. In Figure 5.3.4 two examples of the PMFs are plotted. In these examples, the solid lines represent the normalized histograms of $C(n,Q_{jpeg})$ calculated from watermarked image data, while the dotted lines represent the PMF $P[C(n,Q_{jpeg})=c]$ calculated using Equations 5.2.3 and 5.2.10. The highly varying behavior of these curves as a function of c is mainly due to the zigzag scanning order of the DCT coefficients. We observe that an acceptable fit between the two curves is obtained with some deviations for higher cut-off indices. Since the PMF $P[C(n,Q_{jpeg})=c]$ will be used for calculating the probability of a label bit error, i.e. the probability that the watermarking procedure attempts to select a cut-off index smaller than the minimum allowed values c_{min}, slight deviations at higher values for the cut-off index are not relevant to the objectives of this chapter.

The final step is to use the relation (5.2.3) and (5.2.10) to *analytically* estimate the PMF $P[C(n,Q_{jpeg})=c]$ of the cut-off index for different values of the parameters Q_{jpeg} and n. In this final step we rid ourselves of the erratic behavior of the curves in Figure 5.3.2 and 5.3.4 due to the zigzag scan order of the DCT coefficients by approximating the variances of the DCT coefficients in Figure 5.3.2 by a second order polynomial function. The overall effect of using a polynomial function for the DCT coefficients is the smoothing of the PMF $P[C(n,Q_{jpeg})=c]$.

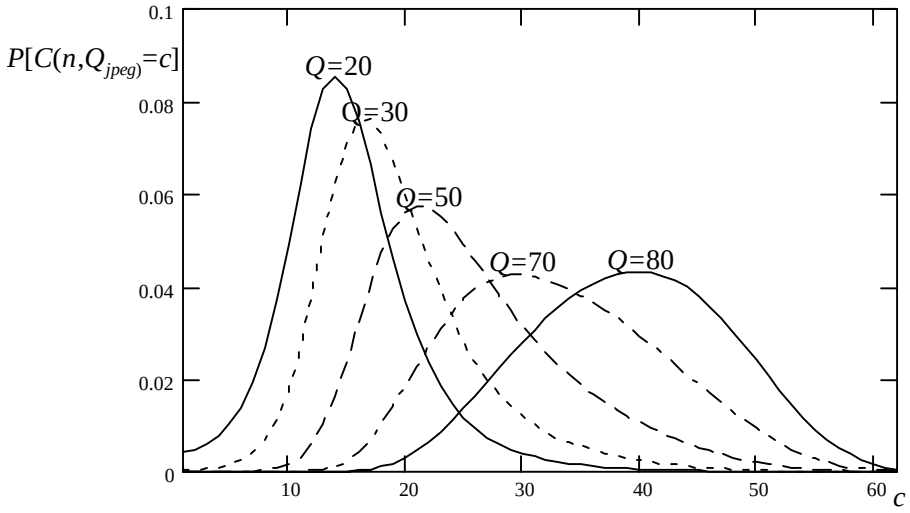


Figure 5.3.5. Analytically calculated PMF $P[C(n,Q_{jpeg})=c]$ using Theorem I and II for various values of Q_{jpeg} and n=16.

In Figures 5.3.5 and 5.3.6**,** the analytically calculated PMFs are shown. These curves are computed using Theorems I and II with only the shape parameter γ and the fitting parameters of the DCT variances as input. In Figure 5.3.5 $P[C(n,Q_{jpeg})=c]$ is shown as a function of Q_{jpeg} keeping n constant, and in Figure 5.3.6 $P[C(n,Q_{jpeg})=c]$ is shown as a function of n keeping Q_{jpeg} constant. It can clearly be seen that decreasing n or Q_{jpeg} leads to an increased probability of lower cut-off indices. This complies with our earlier experiments in Section 4.4.1, which showed that watermarks embedded with small values for n yields visible artifacts due to the removal of high frequency DCT coefficients.

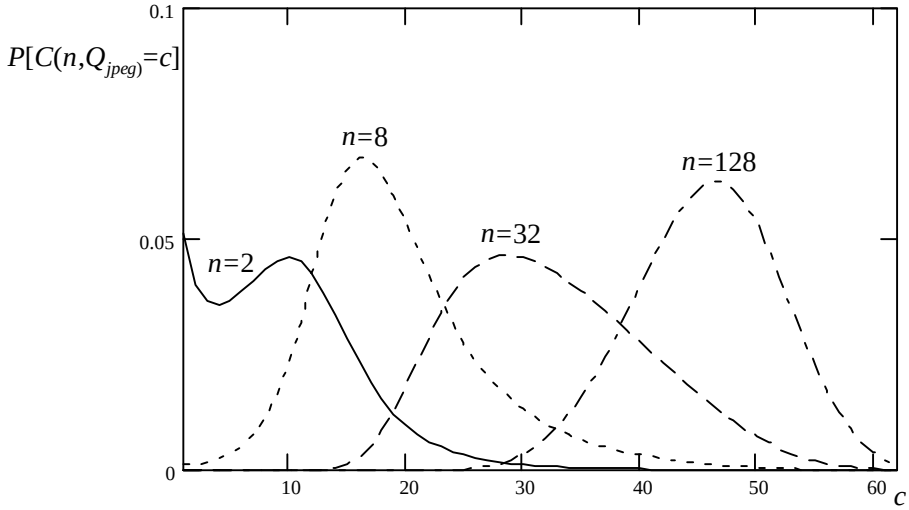


Figure 5.3.6. Analytically calculated PMF $P[C(n,Q_{jpeg})=c]$ using Theorem I and II for various values of n and Q_{jpeg}=50.

5.4 Label error probability

In the analysis of the *DEW* algorithm, we have seen that depending on the parameter settings (n,Q_{jpeg}) certain cut-off indices are more likely than others. In this analysis, however, the selection of the cut-off index by the watermarking algorithm has been carried out irrespective of the visual impact on the image data. In order for the watermark to remain invisible, the cut-off indices are constrained to be larger than a certain minimum c_{min}. Consequently, it may happen in certain lc-regions that a label bit cannot be embedded. This random event is typically the case in lc-(sub)regions that contain insufficient high frequency details.

Using Theorems I and II, we are able to derive the probability that this undesirable situation occurs, and obtain an expression for the *label bit error probability* P_{be} that depends on $Q_{jpeg,}$ n and c_{min}. If a label bit can not be embedded because of the minimally required value of the cut-off index c_{min}, there is a probability of 0.5 that during the extraction phase a random bit is extracted which equals the original label bit. We assume that due to the random shuffling of DCT blocks, the occurrence of a label bit error can be considered as a random event, independent of other label bit errors. The probability that a random error occurs in a label bit, can therefore be computed as follows:

$$P_{be}(n,Q_{jpeg},c_{min}) = 0.5\, P[\, C(n,Q_{jpeg}) < c_{min}] = 0.5 \sum_{c=0}^{c_{\min}} P[C(n,Q_{jpeg}) = c] \quad (5.4.1)$$

Using this relation, we can calculate the label bit error probability for each value of c_{min} as a function of Q_{jpeg} and n. As an example Figure 5.4.1 shows the analytically computed label bit error probability $P_{be}(n,Q_{jpeg},c_{min})$ as a function of Q_{jpeg} and n for c_{min}=3. From this example it is immediately clear that for a given c_{min} certain (Q_{jpeg} , n) combinations must be avoided in practice because they lead to unacceptably high label bit error probabilities.

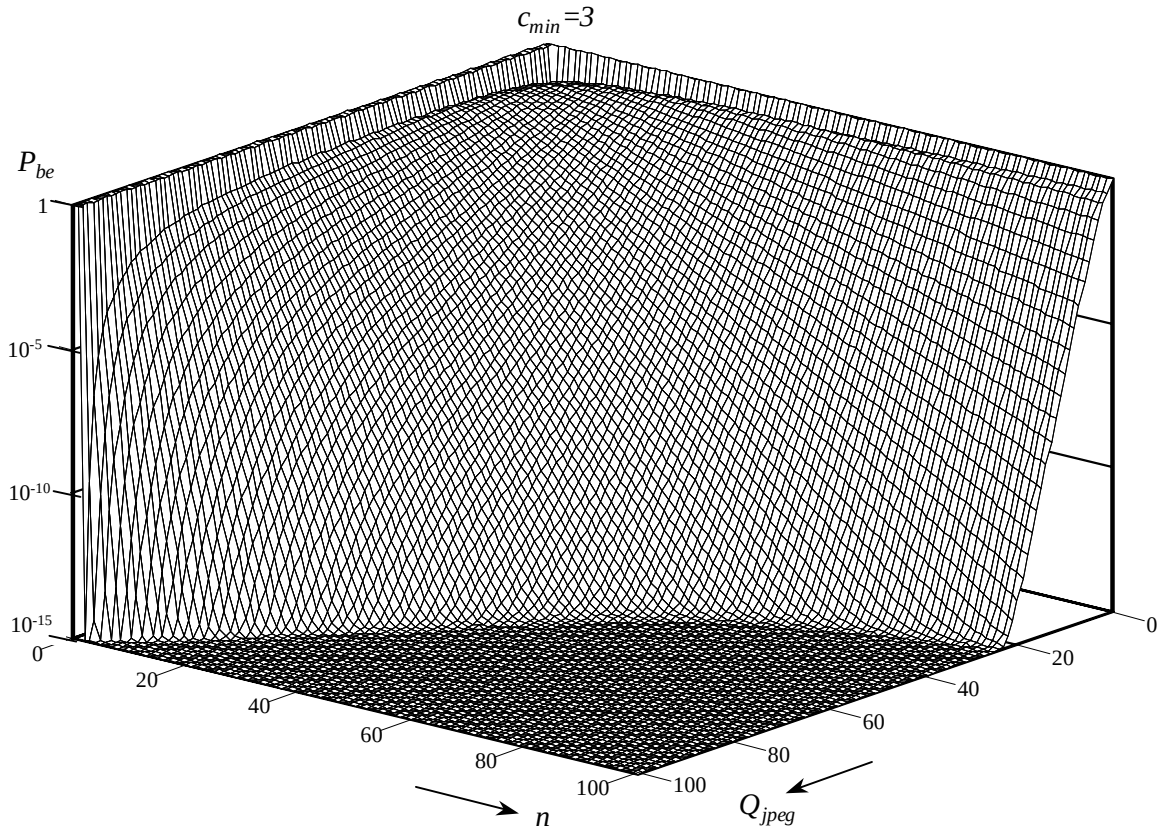


Figure 5.4.1. The bit error probability P_{be} as a function of Q_{jpeg} and n for c_{min}=3.

Using the label *bit* error probability in Equation 5.4.1, we can now derive the *label* error probability P_e, which is here defined as the probability that one or more label *bit* errors occur in the embedded information bit string. Assuming image dimensions of $N \times M$, the number of information bits l that the image can contain is given by

$$l(N, M, n) = \left\lfloor \frac{N \cdot M}{64 \cdot n} \right\rfloor \tag{5.4.2a}$$

with which the label error probability can be calculated as:

$$P_e\,(n, Q_{jpeg}, c_{\min}, N, M) = 1 - (1 - P_{be})^{l(N,M,n)} \tag{5.4.2b}$$

Let us consider one particular numerical example. If, for instance in a broadcast scenario, one incorrect label is accepted per month in a continuous 10 Mbit/s video stream, the label bit error rate should be smaller than 10^{-7}. To select the optimal setting for Q_{jpeg} and n that comply with this label bit error rate, Figure 5.4.2 shows curves of the combinations Q_{jpeg} and n for which P_e equals 10^{-7}. Different curves refer to different values of c_{min}. Further we have assumed the image dimensions $N \times M = 1024 \times 768$.

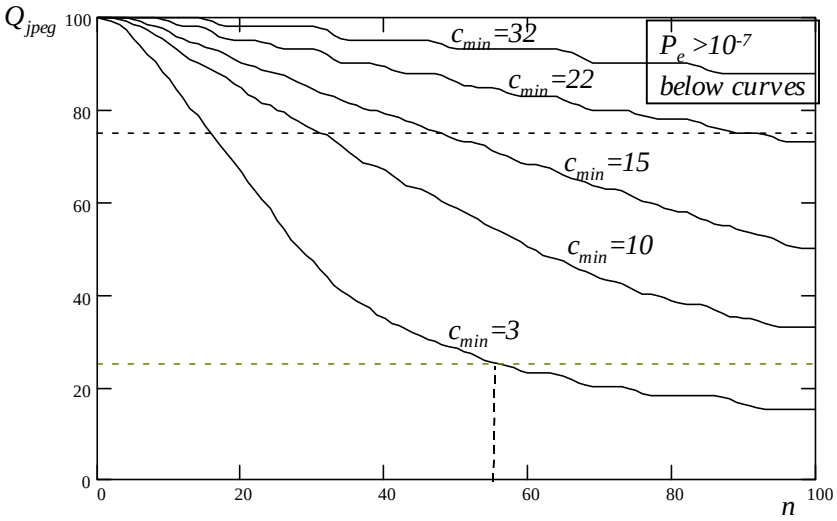


Figure 5.4.2. Combinations of Q_{jpeg} and n for which P_e=10^{-7}.

5.5 Optimal parameter settings

Using results such as the ones shown in Figure 5.4.2, we can now select optimal settings for Q_{jpeg} and n for specific situations. We consider three different cases, namely:

- optimization for re-encoding robustness, number of information bits l, and watermark invisibility;
- optimization for number of information bits l, and watermark invisibility;
- optimization for watermark invisibility.

In all cases the parameter D must be chosen in the range [1, $D_{max}(Q_{jpeg})$] in order for the models in Theorem I and II and the analytical results obtained from these results, to be valid.

If we tune the *DEW* watermark such that it trades-off the re-encoding robustness, number of information bits l, and watermark invisibility, typical choices are to anticipate re-encoding up to JPEG quality factor of Q_{jpeg}=25, and to allow a minimal cut-off index of c_{min}=3. In this case – using Figure 5.4.2 – we need at least n=54 DCT blocks per label bit (which directly determines the number of information bits that can be stored in an image) to achieve the required label error probability of 10^{-7}.

If we require a large label but robustness against re-encoding attacks is not an issue, we can store more than 3 times as many bits in a label with the same label error probability of 10^{-7}. A typical parameter setting would for instance be Q_{jpeg}=75, n=16 and c_{min}=3, as can be seen from Figure 5.4.2.

If visual quality is the most important factor, we need to take the minimal cut-off index sufficiently large. For instance we choose c_{min}=15. Clearly, to obtain the same label bit error probability more DCT blocks per label bit are required since the allowed minimal cut-off index is larger than in the previous example. Using Figure 5.4.2, we find as optimal settings in this case Q_{jpeg}=75 and n=48.

The performance of any watermarking system can be improved by applying error-correcting codes (ECCs). Since we know that the label bit errors occur randomly and independently of other label bit errors, we can compute the probability for *label error* in case an ECC is used that can correct up to *R* label bit errors, namely

$$P_e^{ECC(R)}(n, Q_{jpeg}, c_{\min}, N, M) = 1 - \sum_{j=0}^{R} \binom{l(N,M,n)}{j} P_{be}^{j} (1 - P_{be})^{l(N,M,n)-j} \qquad (5.5.1)$$

with the label bit error probability P_{be} given by Equation 5.4.1.

In Figure 5.5.1 the label error probability $P_e^{ECC(R)}$ is shown as a function of the number of DCT blocks used to embed a single label bit (n) for R=0, 1, 2, Q_{jpeg}=25 and c_{min}=3. We had already found that for a watermark optimized for robustness without error correcting codes, the optimal value of n=54 for a required bit error probability of $P_e<10^{-7}$. From Figure 5.5.1 we see that the same label error probability can be obtained using smaller values of n if we apply error correcting codes

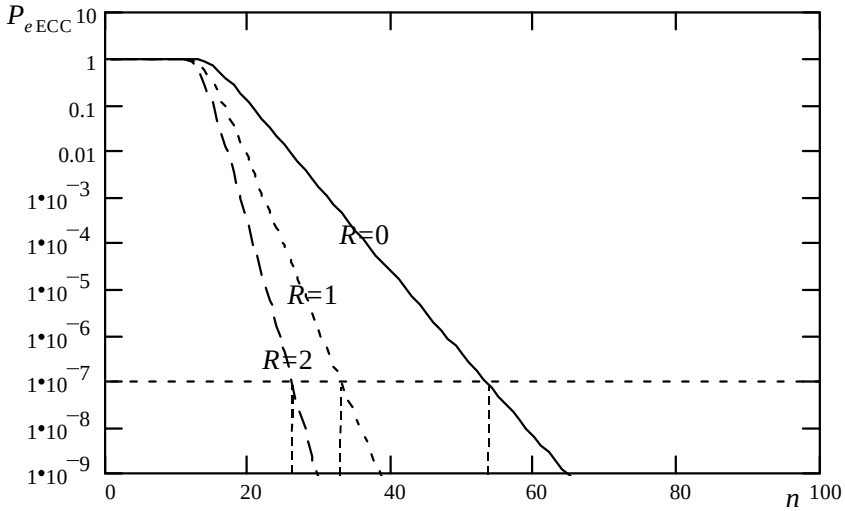


Figure 5.5.1. Label error probability with (R=1,2) and without (R=0) error correcting codes for Q_{jpeg}=25 and c_{min}=3.

For instance, by using an ECC that can correct one error, n can be decreased from 54 to 33. Obviously the use of ECCs introduces some redundant bits. This overhead is however small compared to the increase in capacity due to the use of a smaller value of n. Table 5.5.1 gives some examples of the effective length of labels that can be embedded for $N \times M = 1024 \times 768$. In this table standard BCH codes [Rhe89] are used that can correct one or two errors.

Table 1. Effective number of bits per label that can be embedded into an image of size $N \times M = 1024 \times 768$, with required performance parameters $c_{min}=3$, $Q_{jpeg}=25$ and $P_e^{ECC(R)}<10^{-7}$.

ECC-Type	*R*	*n*	Parity-check bits ECC	Label size corrected for extra parity-check bits
no-ECC	0	54	0	227
BCH (511,502)	1	33	9	363
BCH (511,493)	2	27	18	437

5.6 Experimental results

In this section, we will compare the robustness of labels embedded using settings optimized for maximum label size, namely $c_{min}=3$, $n=16$, $Q_{jpeg}=75$, and $D=25$ with labels embedded using settings optimized for robustness, namely $c_{min}=3$, $n=64$♣, $Q_{jpeg}=25$, and $D=400$. The Lena-image watermarked with the DEW algorithm using settings optimized for maximum label size and the corresponding strongly amplified watermark are presented in Figure 5.6.1. Figure 5.6.2 shows the same images resulting from the DEW algorithm using settings optimized for robustness.

(a) Watermarked image **(b)** Difference $W(x,y)=I-I_W$

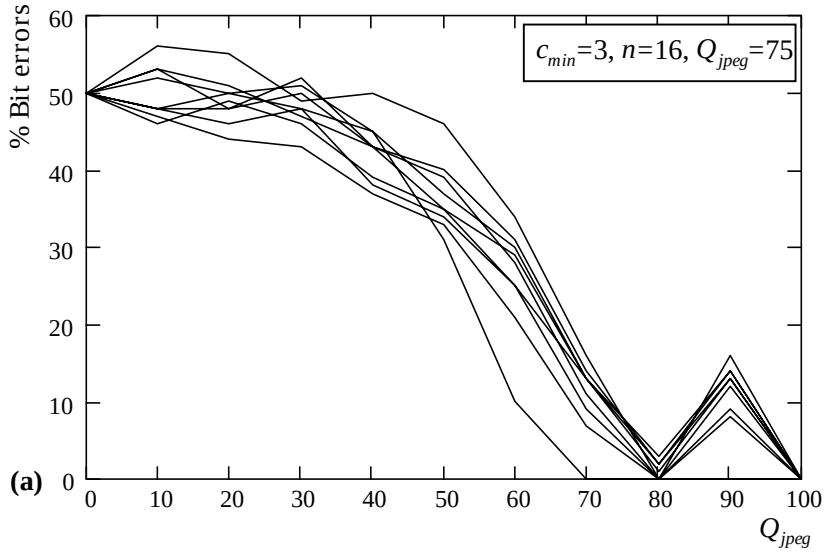
Figure 5.6.1. DEW watermarking using optimal settings for maximum label size.

♣ Our software implementation choices require that $n=16 \cdot k^2$, where k=1,2,3…. We therefore selected $n=64$ instead of the optimal value $n=54$.

(a) Watermarked image **(b)** Difference $W(x,y)=I-I_w$

Figure 5.6.2. DEW watermarking using optimal settings for robustness.

We will first check the robustness against re-encoding. Images are JPEG compressed with quality factor of 100. From these JPEG compressed images two watermarked version are produced, one for each parameter setting. Next, the images are re-encoded using a lower JPEG quality factor. The quality factor of the re-encoding process is made variable. Finally, the watermark is extracted from the re-encoded images and compared bit by bit with the originally embedded watermark. For the labels embedded using settings optimized for maximum label size the extraction parameters D'=40 and Q'_{jpeg}=75 are used. For the labels embedded using settings optimized for robustness the extraction parameters D'=400 and Q'_{jpeg}=80 are chosen. From this experiment, we find the percentages of label bit errors due to re-encoding as a function of the re-encoding quality factor. In Figure 5.6.3 the resulting label bit error curves are shown for nine different images.



(a)

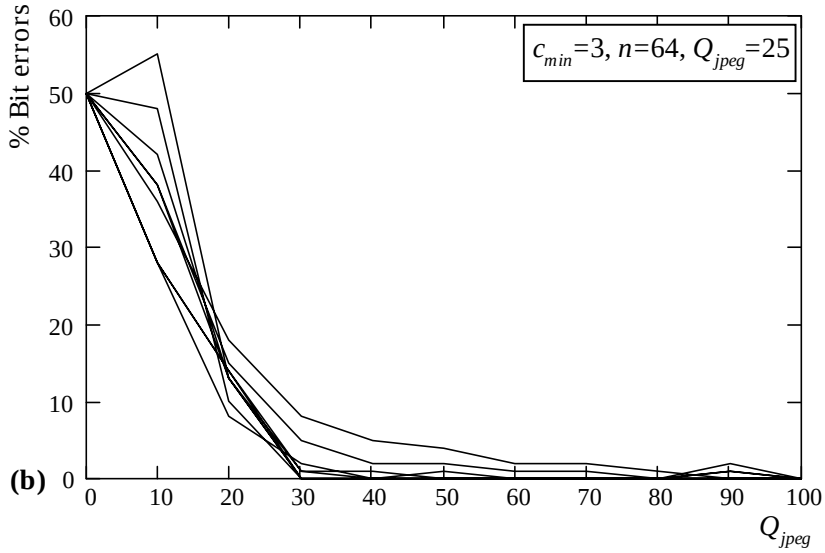
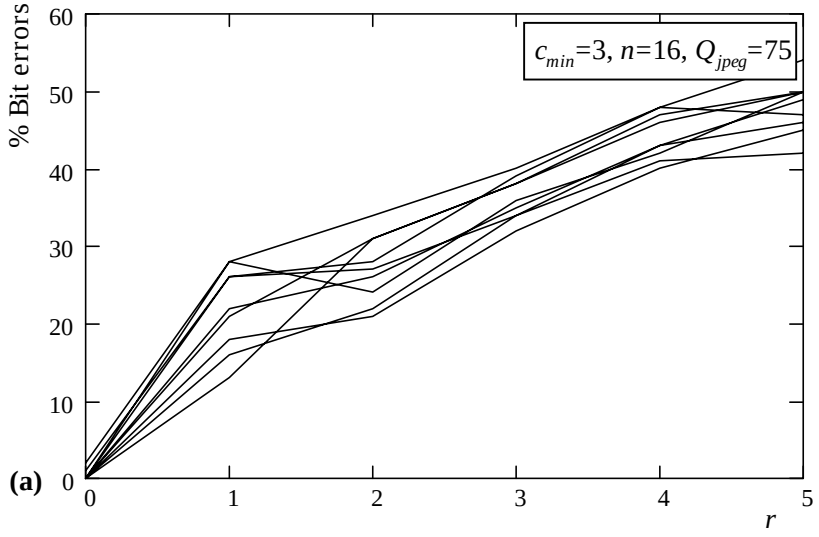


Figure 5.6.3. Percentage bit errors after re-encoding **(a)** using parameter settings optimized for label size; **(b)** parameter settings optimized for robustness.

Although we expect that the percentages label bit errors are very small for JPEG quality factors between 75 and 100 because the parameter Q_{jpeg} is set to 75, we see in Figure 5.6.3a a small increase in bit errors for images re-encoded using a JPEG quality factor of 90. This effect is caused by the two consecutive quantization steps using JPEG quality factors 90 and 75, which are performed before the energy differences are calculated. These quantization steps introduce minor differences in the DCT coefficients. If these minor differences are squared and accumulated over 16 DCT blocks, the energy differences can significantly differ from the originally enforced small energy differences (D=25). This effect can be canceled by omitting the optional quantization step (Q'_{jpeg}=100) during the watermark extraction phase, or by increasing the enforced energy difference D.



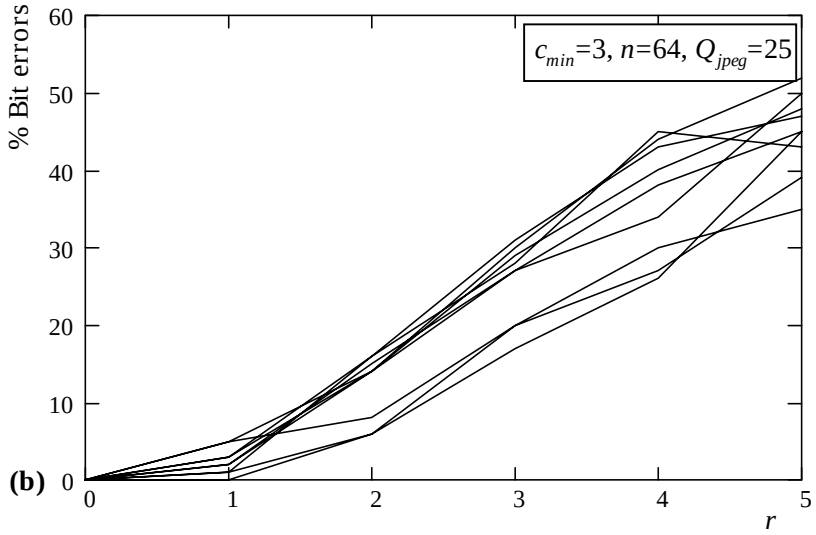


Figure 5.6.4. Percentage bit errors after shifting over *r* pixels **(a)** using parameter settings optimized for label size; **(b)** parameter settings optimized for robustness.

Comparing Figure 5.6.3a (parameter setting optimized for label length using c_{min}=3, n=16, Q_{jpeg}=75, and D=25) and Figure 5.6.3b (parameter setting optimized for label robustness using c_{min}=3, n=64, Q_{jpeg}=25, and D=400), we see an enormous gain in robustness. In Figure 5.6.3b, we see a breakpoint around Q_{jpeg}=25. For higher re-encoding qualities, the percentage label bit errors is below 10%.

In the previous chapter we noticed that the *DEW* watermarking technique is slightly resistant to line shifting. To investigate the effect of the parameter settings optimized for robustness on the resistance to line shifting, we carry out the following experiment. Images are JPEG compressed with a quality factor of 85. These JPEG images are watermarked using the parameter settings optimized for label size or optimized for robustness. Next the images are decompressed, shifted to the right over *r* pixels and re-encoded using the same JPEG quality factor. Finally, a watermark is extracted from these re-encoded images and bit-by-bit compared with the originally embedded watermark. Consequently, we find the percentages bit errors due to line shifting. In Figure 5.6.4 the bit error curves are shown for nine different images. As in the previous experiment, we see an improvement in robustness between Figure 5.6.4a and Figure 5.6.4b. Using the parameter settings optimized for robustness, the *DEW* watermark becomes resistant to line shifts up to 3 pixels.

5.7 Discussion

In this chapter we have derived, experimentally validated, and exploited a statistical model for our DCT-based DEW watermarking algorithm. The performance of the DEW algorithm has been defined as its robustness against re-encoding attacks, the label size, and the visual impact. We have analytically shown how the performance is controlled by three parameters, namely Q_{jpeg}, n and c_{min}. The derived statistical model gives us an expression for the label bit error probability as a function of these three parameters Q_{jpeg}, n and c_{min}. Using this expression, we can optimize a watermark for robustness, size or visibility and add adequate error correcting codes.

The obtained expressions for the probability mass function of the cut-off indices can also be used for other purposes. For instance, with this PMF an estimate can be made for the variance of the watermarking "noise" that is added to an image by the DEW algorithm. This measure, possibly adapted to the human visual perception, can be used to carry out an overall optimization of the watermark embedding procedure using the (perceptually weighted) signal-to-noise-ratio as optimization criterion.

Chapter 6

Benchmarking the DEW Watermarking Algorithm

6.1 Introduction

In literature many watermarking algorithms have been presented in recent years. Most authors claim that the watermark embedded by their algorithm is robust and invisible. However, none of them uses the same robustness criteria and quality measures. Furthermore, the term "robustness" is hard to define and it is even questionable if it can be defined formally. A watermark that is fully resistant to lossy compression techniques may be very vulnerable to a dedicated attack, which may consist of some low complexity processing steps like concatenated filtering. Besides robustness and visibility, the payload and complexity of the embedding and extracting procedure may play an important role. Also the weighting of these performance factors varies significantly for different applications. This makes the comparison of the performance of the different algorithms a difficult task. In spite of this, we attempt in this chapter to derive a fair benchmark for the DEW algorithm by taking into account known attacks from literature and by weighting the performance factors according to the requirements imposed by the application.

In Section 6.2 two watermark benchmarking approaches from literature are discussed. In Section 6.3 two dedicated watermark attacks are presented which can be part of a benchmarking process. The performance of the DEW algorithm is compared to the real-time spread spectrum method of Hartung and Girod [Har98] and the basic spread spectrum method of Smith and Comiskey [Smi96] in Section 6.4. The chapter concludes with a discussion in Section 6.5.

6.2 Benchmarking methods

In literature two watermark benchmarking methods are proposed namely [Fri99a] and [Kut99]. The authors of both methods notice that the robustness is dependent on the payload and the visibility of the watermark. Therefore, to allow a fair comparison between different watermarking schemes, watermarks are embedded in a pre-defined video data set with the highest strength, which does not introduce annoying effects according to a pre-defined visual quality metric. Subsequently processing techniques and attacks are applied to the watermarked data and the percentages bit errors are measured to estimate the performance of the watermarking schemes.

The two benchmarking methods differ in the choice of the payload of the watermark, the visual quality metric and the processing techniques. In [Fri99] the payload of the watermark is fixed to 1 or 60 bits. To evaluate the visual quality of the watermarked video data, the spatial masking model of Girod [Gir89] is used. This model is based on the human visual system and describes the visibility of artefacts around edges and flat areas in video data accurately. The watermark strength is adjusted in such a way that Girod's model indicates less than one percent of pixels with visible changes. Subsequently, the watermarked data is subject to the processing operations listed in Table 6.2.1 and the bit error rate is measured as a function of the corresponding parameters.

Table 6.2.1. List of processing operations to which the robustness of a watermarking method is tested.

Operation	Parameter
JPEG compression	Quality factor
Blurring	Kernel size
Noise addition	Noise amplitude (SNR)
Gamma correction	Gamma exponent
Permutation of pixels	Kernel size
Mosaic filter	Kernel size
Median filter	Kernel size
Histogram equalization	-

The authors [Fri99a] do not claim that this list is exhaustive; other common lossy compression techniques, such as wavelet compression should probably be included.

Using the benchmarking approach described in [Kut99] the payload of the watermark is fixed to 80 bits. To evaluate the visual quality of the watermarked video data, the distortion metric proposed by Van den Branden Lambrecht and Farrell [Bra96] is used. This perceptual quality metric exploits the contrast sensitivity and masking phenomena of the Human Visual System and is based on a multi-channel model of the human spatial vision. The unity for this metric is given in *units above threshold* also referred to as *Just Noticeable Difference* (JND). In [Kut99] this quality metric is normalized using the ITU-R Rec. 500 quality rating [ITU95]. In Table 6.2.2 the ratings and the corresponding visual perception and quality are listed.

Table 6.2.2. ITU-R Rec. 500 quality ratings on a scale from 1 to 5.

Rating	Impairment	Quality
5	Imperceptible	Excellent
4	Perceptible, not annoying	Good
3	Slightly annoying	Fair
2	Annoying	Poor
1	Very annoying	Bad

The ITU-R quality rating Q_{ITU} is computed as follows:

$$Q_{ITU} = \frac{5}{1 + CN + MD} \tag{6.2.1}$$

where *MD* is the measured distortion according to the model of Van den Branden Lambrecht and Farrell and *CN* is a normalization constant. *CN* is usually chosen such that a known reference distortion maps to the corresponding quality rating. The results generated by the model can not be used to determine if for instance an image with quality rating Q_{ITU}=4.5 looks better than an image with quality rating Q_{ITU}=4.6. The results should be interpreted in combination with a threshold: images with quality ratings above Q_{ITU}=4 may only contain perceptible not annoying artefacts.

The watermark strength is adjusted in such a way that the quality rating is at least 4. Subsequently, the watermarked data is subject to a list of processing operations, including lossy JPEG compression, geometric transformations and filters. Most of these processing operations are implemented in one single program called StirMark, which is described in the next section. Instead of applying each processing operation listed in Table 6.2.1 to the watermarked data, only StirMark is applied to the data, which has the same effect as performing the transformations separately with various parameters. Finally, the error rate for the retrieved bits is measured.

6.3 Watermark attacks

6.3.1 Introduction

Watermarks are vulnerable to processing techniques. Therefore, every processing technique that does not significantly impair the perceptual quality of the watermarked data can be considered as an intentional or unintentional watermark attack. In [Har99] the watermark attacks are classified in four groups:

A. "Simple attacks" are conceptually simple attacks that attempt to impair the embedded watermark by manipulations of the whole watermarked data, without an attempt to identify and isolate the watermark. Examples include linear and general non-linear filtering, lossy compression techniques like JPEG and MPEG compression, noise addition, quantization, D/A conversion and gamma correction.

B. "Detection-disabling attacks" are attacks that attempt to break the correlation and to make the recovery of the watermark impossible for the watermark detector, mostly by geometrical distortions like scaling, shifting in spatial or temporal direction, rotation, shearing, cropping and removal or insertion of pixels clusters. A typical property of this type of attacks is that the watermark remains in the attacked data and can still be recovered with increased intelligence of the watermark detector.

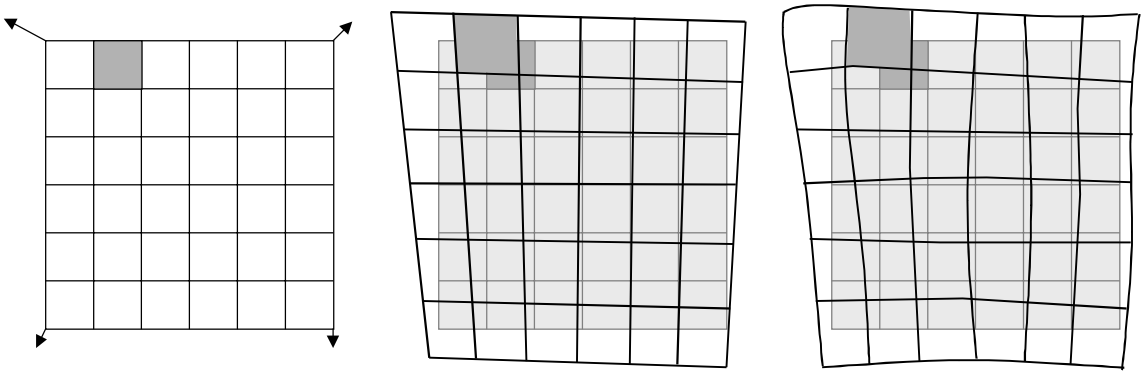
C. "Ambiguity attacks" are attacks that attempt to confuse by producing fake original data or fake watermarked data. This attack is only useful for copyright purposes and therefore outside the scope of this thesis. An example of this attack is the inversion attack described in [Cra96] that attempts to discredit the authority of the watermark by embedding additional watermarks such that it is unclear which was the first watermark and who was the legitimate copyright owner.

D. "Removal attacks" are attacks that attempt to analyze the watermarked data, estimate the watermark or the host data, and separate the watermark from the watermarked data to discard the watermark.

The authors [Har99] note that the distinction between the groups is sometimes vague, since some attacks belong to two or more groups. In Section 6.3.2 the StirMark attack is discussed which belongs to groups A and B. A removal attack on spatial spread spectrum watermarking techniques belonging to group D is presented in Section 6.3.3.

6.3.2 Geometrical transforms

StirMark is a watermark removal attack that is based on the idea that although many watermarking algorithms can survive simple video processing operations, they can not survive combinations of them [Pet98b] and [Pet99]. In its simplest form StirMark emulates a resampling process. It applies minor geometrical distortions by slightly stretching, shearing, shifting and/or rotating an image or video frame by an unnoticeable random amount and then resampling the video data using either bi-linear or Nyquist interpolation. In addition, a transfer function that introduces a small and smooth distributed error into all sample values is applied. This emulates the small non-linear analog/digital converter imperfection typically found in scanners and display devices. In Figure 6.3.1b an example is given of how StirMark resamples the data. The distortions are here exaggerated for viewing purposes. As can be seen the distortion to each pixel is the greatest at the borders of the video data and almost zero at the center.
In addition to this procedure StirMark can also apply global bending to the video data. This results in an additional slight deviation for each pixel, which is greatest at the center of the video data and almost zero at the borders. The bending process is depicted in Figure 6.3.1c. Finally the resulting data is compressed with the lossy JPEG algorithm using a quality factor for medium visual quality.



(a) Original data **(b)** StirMark applied without bending **(c)** StirMark applied with bending

Figure 6.3.1. Exaggerated example of distortions applied by Stirmark.

In Figure 6.3.2b an example is shown of the Lena-image after applying StirMark. Figure 6.3.2c shows the difference between the original image and the StirMarked image. It can be seen that although some pixels are shifted over more than 3 pixels, the image quality is not affected seriously.

(a) Original image **(b)** StirMarked image **(c)** Difference (a)-(b)

Figure 6.3.2. Example of an image after applying StirMark.

The StirMark attack confuses most watermarking schemes available on the market [Pet98b]. Only watermarking schemes with a very low payload can survive this kind of attack.

6.3.3 Watermark estimation

6.3.3.1 Introduction

The spatial spread spectrum watermarking methods described in Chapter 2 basically add a pseudorandom pattern to an image in the spatial domain to embed a watermark. This watermark can be detected by correlating with the same pattern or by applying other statistics to the watermarked image. In this section two attacks are discussed to estimate the pseudorandom spread spectrum watermark from the watermarked image only. If a nearly perfect estimation of the watermark can be found, this estimated watermark can be subtracted from the watermarked image. In this way the watermark is removed without affecting the quality of the image [Lan98b] and [Lan98c].

For our initial experiments we use the basic spread spectrum implementation of Smith and Comiskey [Smi96]. If we apply this method to an image I, a random pattern W consisting of the constants $-k$ and $+k$ is added to obtain the watermarked image $I_w = I+W$, where k is a positive integer value. The watermark energy resides in all frequency bands. Compression and other degradations may remove signal energy from certain parts of the spectrum, but since the energy is distributed all over the spectrum, some of the watermark remains. The random pattern W is uncorrelated with image I, but correlated with I_w:

$$\begin{aligned} &\operatorname{cov}(W, I+W) = \operatorname{var}(W) + \operatorname{cov}(I, W) \approx \operatorname{var}(W) + 0 \\ &\rho(W, I+W) = \frac{\operatorname{cov}(W, I+W)}{\sqrt{\operatorname{var}(W)}\sqrt{\operatorname{var}(I+W)}} \approx \sqrt{\frac{\operatorname{var}(W)}{\operatorname{var}(I+W)}} \\ &\rho(W, I+W) \approx \frac{k}{\sqrt{\operatorname{var}(I+W)}} \end{aligned} \qquad (6.3.1)$$

Evaluation of Equation 6.3.1 for typical images yields that ρ ranges from 0.02 to 0.05. However, if the watermarked images are compressed using the JPEG algorithm or distorted, the approximation in Equation 6.3.1 does not hold. Indeed, the correlation coefficients decrease by a factor 2, while the variance of (I+W) nearly equals the variance of the JPEG compressed version of (I+W).

If an arbitrary random pattern W_x is used, the correlation coefficient will be very small:

$$\begin{aligned} &\operatorname{cov}(W_x, I+W) = \operatorname{cov}(W_x, W) + \operatorname{cov}(W_x, I) \approx 0 + 0 \\ &\rho = \frac{\operatorname{cov}(W_x, I+W)}{\sqrt{\operatorname{var}(W_x)}\sqrt{\operatorname{var}(I+W)}} \approx 0 \end{aligned} \qquad (6.3.2)$$

This holds only if W and W_x are orthogonal and W_x is not correlated with I. Typical values for correlation coefficients between I_W and arbitrary random watermark W_x are a factor 10^2 smaller than $\rho(W,I_W)$.

A simple estimation attack would be to search for all possible random patterns and take the one with the highest correlation value as possible watermark pattern. This approach has several disadvantages. In the first place the search space is huge. Even if the watermark pattern consisting of the integers [-1,1] should meet the requirement that the number of -1s and the number of +1s are equal, more than $4\text{x}10^{306}$ possible patterns have to be checked for a 32x32 pixel watermark. As a first step, we carried out experiments with a genetic algorithm to search the random pattern with the highest correlation coefficient with $I_W = I+W$. In some cases the genetic algorithm found a pattern with a relative high correlation (0.3) with I_W and no correlation with W (10^{-5}). This means that the pattern is adapted to the image contents and not to the watermark. To avoid that the genetic algorithm finds random patterns with higher correlation coefficients than the embedded watermark we must adapt our optimization criterion. From the properties of spread spectrum watermarks we know the following about W:

- $\rho(W, I_W) \in [0.01 .. 0.05]$
- $\rho(W, I) \approx 0$
- W is pseudorandom and has a flat spectrum

If the image is distorted by compression, $\rho(W, I_W)$ is unknown. Too many patterns meet the requirement $\rho(W, I) \approx 0$. The additional information that W is random and has a flat spectrum is also not enough to create a suitable optimization criterion function. If we have several different images with the same watermark on it to our disposal, there are some possibilities (e.g. collusion attacks). A fitness function for the genetic algorithm dependent on all images can be used, or if there are enough images, the average of the images can be taken as estimation of the watermark.

In [Kal98b] and [Lin98] the watermark is estimated by analyzing the watermark detector. However, if different watermarks are used for each image and the watermark detector is not available, we have to follow other approaches that estimate the watermark from the watermarked data only. In [Mae98] an approach is proposed to estimate spatial spread spectrum watermarks by histogram analysis. The results of this approach depend very

much on the content of the images. Watermarks can be estimated quite accurately for images with peaky histograms, however the results for images with a smooth histogram are poor. In the next subsection we propose a watermark estimation approach which is based on non-linear filtering.

6.3.3.2 Watermark estimation by non-linear filtering

In general, a watermark can be regarded as a perceptually invisible enforced distortion in the image. In most cases, this distortion is not correlated to the image contents. If we could apply a nearly perfect image model to the watermarked image $I_W = I+W$, we could predict the image content $\hat{I}$ and find back an estimate of the watermark $\hat{W} = I_W - \hat{I}$. Because perfect image models and perfect noise filters do not exist, $\hat{I}$ will be different from I and $\hat{W}$ will be different from W. Our objective is to separate $I_W = \hat{I} + \hat{W}$ in such a way that the watermark is totally removed from $\hat{I}$ and resides completely in $\hat{W}$ [Lan98b] and [Lan98c]. This means that image contents may remain in the predicted watermark.

An AR-model, linear smoothing filters (3x3 and 5x5), Kuwahara filters [Kuw76] (several sizes), non-linear region based filters and filters based on thresholding in the DCT-domain (coring) are tested to separate I_W in $\hat{I}$ and $\hat{W}$. In some cases, the watermark can be retrieved from both $\hat{I}$ and $\hat{W}$, while $\hat{I}$ has still a reasonable quality and $\hat{W}$ does not contain any image information. In other cases the watermark can only be retrieved from $\hat{W}$, but the quality of $\hat{I}$ is significantly affected and the image contents, especially the edges, remain in $\hat{W}$.

We select some candidates from the separation operations that totally destroy the watermark in $\hat{I}$, $\rho(W, \hat{I}) \approx 0$. From these candidates we select the operation that has the highest correlation coefficient $\rho(\hat{W}, W)$ in a test set of 9 images. In Table 6.3.1 the correlation coefficients for several separation operations are listed.

Table 6.3.1. Correlation coefficients $\rho(\hat{W}, W)$ using different separation operations.

Separation Operation	**$\rho(\hat{W}, W)$**
Misc. Noise Reduction Filters	0.08-0.12
Auto Regressive Model	0.10-0.17
Median 3x3	0.13-0.22

The 3x3 median filter turns out to be the best separation operation and is used for the rough estimation of $\hat{W} = I_W - \text{med}_{3x3}(I_W)$. However, correlation coefficients $\rho(\hat{W}, W)$ between 0.13 and 0.22 are still too low and $\hat{W}$ must be refined further by using information about the watermark properties.

The estimate $\hat{W}$ does still contain edge information. To protect the edges in I_W we limit the range of $\hat{W}$ from [-128..128] to [-2..2] before we subtract $\hat{W}$ from I_W. In Figure 6.3.3 the modulus of the Fourier Transform of the truncated $\hat{W}$ is presented.

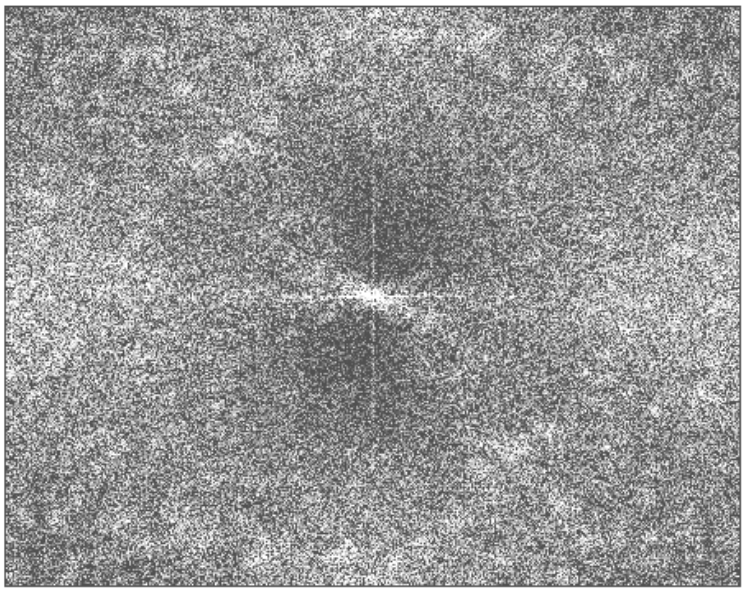


Figure 6.3.3. Power density spectrum of $\hat{W}_{[-2..2]}$.

The horizontal, vertical and diagonal patterns in Figure 6.3.3 clearly indicate that some dominating low frequency components are present in the spectrum. Since a spread spectrum watermark should not contain such dominating components, these come certainly from the image content. To remove these components a 3x3 linear high pass filter is applied to the non-truncated $\hat{W}$. After the filtered $\hat{W}$ is truncated to the range [-2,2] the Fourier spectrum as presented in Figure 6.3.4 is obtained. The correlation coefficients between the high pass filtered $\hat{W}$ and W, $\rho(\hat{W}, W)$, increase now to values around 0.4.

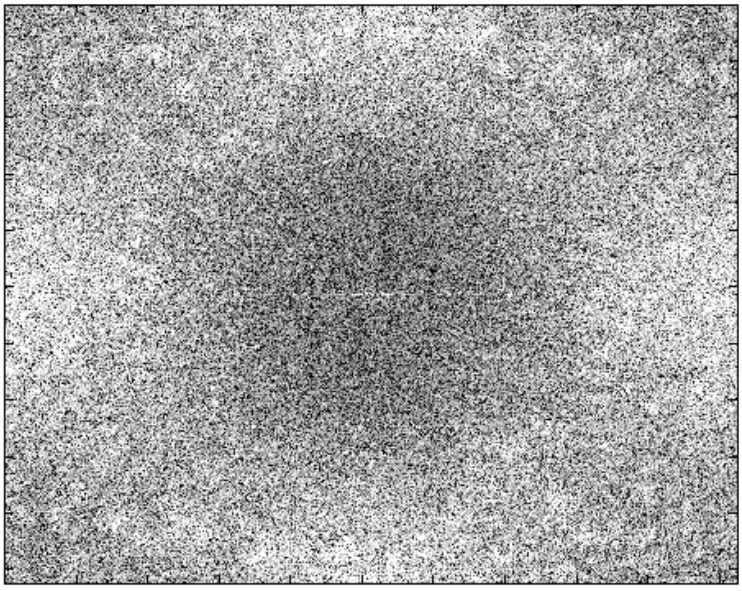


Figure 6.3.4. Power density Spectrum of high-pass$(\hat{W})_{[-2..2]}$.

If the so-found watermark $\hat{W}$ is subtracted from the watermarked image I_w the watermark is not completely removed. This is not surprising, since we are not able to predict the low

frequency components of the watermark. These components are discarded during the high pass filtering stage of $\hat{W}$ or are left in $\hat{I}$ by the median filter. The low frequency components, which can not be estimated properly, give a positive contribution to correlation of the watermark detector, while subtracting the estimated watermark $\hat{W}$, that mainly consists of high frequency components, gives a negative correlation contribution. By amplifying the estimated watermark $\hat{W}$ with a certain gain factor *G* before subtraction, the overall correlation of the watermark detector can be forced to zero. The complete scheme for removing a watermark is represented in Figure 6.3.5.

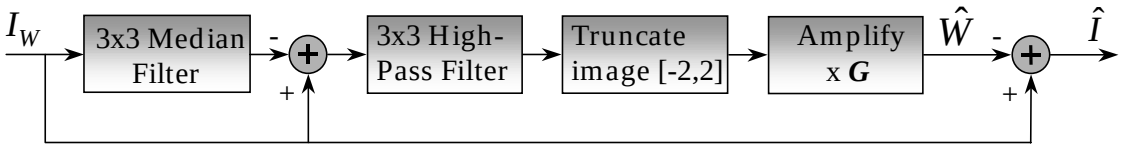


Figure 6.3.5. The complete watermark removing scheme (WRS).

The value of *G* is dependent on the image content and the amount of energy in the embedded watermark. If *G* is chosen too high, the watermark inverts and can still be retrieved from $\hat{I}$ by inverting the image before retrieving the watermark.

The value *G* is experimentally determined. A watermark is added to an image using the method of Smith and Comiskey [Smi96], 32x32 pixels are used to store one bit of watermark information and the watermark carrier consists of the integers {-2,2}. The watermark removing scheme is applied to the watermarked image with several values for *G*. The percentage watermark bit errors is plotted as function of *G* in Figure 6.3.6. If 50% bit errors are made, the watermark is removed, if 100% bit errors are made, the watermark is totally inverted. According to Figure 6.3.6 the gain factor *G* should have a value between 2 and 3 to remove the watermark from this image. The values of the gain factor vary for different kinds of images but are typically in the range from 2 to 3. We therefore fixed the gain factor *G* to 2.5 for all images.

We tested the watermark removing scheme (WRS) represented in Figure 6.3.5 on a set of 9 true color images. Informal subjective tests were performed to determine the quality of the images. Some images hardly contain any textured areas and sharp edges, some contain many sharp edges and much detail, others contain both smooth and textured areas. First, the WRS (*G*=2.5) is applied to the methods of Bender *et al* [Ben95] and Pitas and Kaskalis [Pit95]. The watermarks in the 9 test images are all removed without reducing the quality of the images significantly.

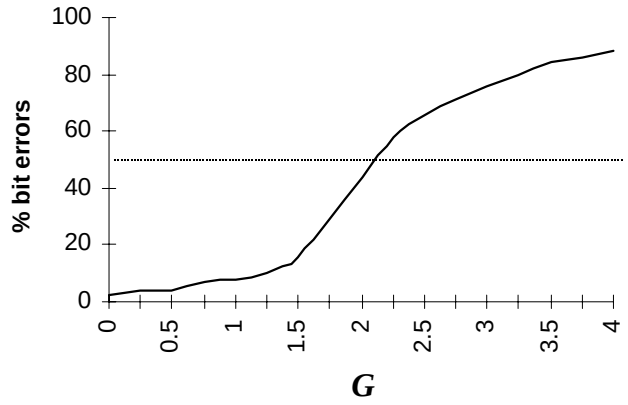


Figure 6.3.6. % bit errors as a function of gain factor *G*.

Subsequently the WRS (*G*=2.5) is applied to the more robust watermarking method of Smith and Comiskey [Smi96]. The watermarks are added using *P* pixels per bit and a gain factor of *k*, where *k*= 1 or 2. If higher gain factors *k* are used the watermark becomes visible. For the values *P*=8x8, 16x16, 32x32, 64x64 and *k*=1,2 the watermarks can be removed without affecting the visual quality significantly. An example is given in Figure 6.3.7. An image is watermarked using the parameters *k*=2, *P*=32x32. To remove the watermark completely (about 44% bit errors) using the JPEG compression algorithm, we have to use a quality factor *Q*=10. The result of this compression operation is presented in Figure 6.3.7a. If we apply the WRS to the watermarked image, the watermark is completely removed (>50% bit errors) and we obtain the image which is shown in Figure 6.3.7b. This image hardly distorted. If the number of pixels per bit *P* is increased further to 128x128 or 256x256, the watermark is fully removed in smooth images, but only partially in textured images.

(a) Removal by JPEG compression **(b)** Removal by the WRS scheme

Figure 6.3.7. Removing a watermark from a watermarked image.

Finally, the WRS (*G*=2.5) is applied to the method of Langelaar *et al* [Lan97a]. This watermarking method determines the gain factor *k* for each watermark bit automatically. Therefore only the number of pixels per bit *P* can be changed. All watermarks added with

this method can be removed for *P*=8x8, 16x16, 32x32. For *P*=64x64, 128x128, … the watermarks are only partially removed. In this case the watermark information is only removed from the smooth regions of the images, but remains in the more textured regions, since the watermark estimate is here not accurate enough.

Some methods (e.g. [Wol96]) first subtract the original image from the watermarked image and apply the watermark retrieval operation on this difference image. However, the WRS also removes the watermarks in this case. Other methods using a similar approach as [Smi96] are not tested, but we expect that such watermarks will be affected in the same way as [Smi96], since they use the same basic principle.

6.4 Benchmarking the DEW algorithm

6.4.1 Introduction

In this section the DEW algorithm is compared to other watermarking methods known from literature. In Section 6.4.2 the performance factors are discussed on which the comparison is based. In Section 6.4.3 the real-time DEW algorithm for MPEG compressed video is compared to the basic spread spectrum technique of Smith and Comiskey [Smi96] that operates on raw video data and to other real-time watermarking algorithms that operate directly on the compressed data. In this comparison the emphasis is on the real-time aspect. This holds for both the watermarking procedures and the watermark removal attacks. The attacks are therefore limited to transcoding operations.

In Section 6.4.4 the DEW algorithm for JPEG compressed and uncompressed still images is compared to the basic spread spectrum method of Smith and Comiskey [Smi96]. Since the latter method is not specially designed for real-time operation on compressed data, the real-time aspect is neglected in this comparison and for the evaluation the guidelines of the benchmarking methods from literature described in Section 6.2 are followed.

6.4.2 Performance factors

To evaluate the performance of the DEW algorithm we have to compare it to other watermarking algorithms with respect to complexity, payload, impact on the visual quality, and robustness. Of these performance factors, the impact on the visual quality is the most important one. A watermark must not introduce annoying effects, otherwise watermarking algorithms will not be accepted as protection techniques by the users, who expect excellent quality of digital data. The weighting of the other performance factors depends heavily on the application of the watermarking method.

As already mentioned in Section 1.4, the focus of this thesis is mainly on the class of watermarking algorithms which can for instance be used in fingerprinting and copy protection systems for home-recording devices for the consumer market. For this class of watermarking algorithms the complexity of the watermark embedding and extraction procedures is an important performance factor for two reasons. On one hand, because the algorithms have to operate in real-time and on the other hand, because the algorithms have to be inexpensive for the use in consumer products.

Another performance factor is the payload of the watermark. For fingerprinting applications and protection of intellectual property rights a label bit-rate of at least 60 bits per second is required to store one identification number similar to the one used for ISBN or ISRC per second [Kut99]. For copy protection purposes, a label bit-rate of one bit per second may be sufficient to control digital VCRs.

The last performance factor is the robustness of the watermark. The robustness is closely related to the payload of a watermark. The robustness can be increased by decreasing the payload and visa versa. In Sections 6.2 and 6.3 an overview has been given of processing techniques to which watermarks are vulnerable. Most of these processing techniques require that the compressed video stream has to be decoded and completely be re-encoded. This is a quite computationally and storage demanding task. The most obvious way to intentionally remove a watermark from a compressed video stream is therefore to circumvent these MPEG decoding and re-encoding steps. This can be done for instance by transcoding the video stream.

6.4.3 Evaluation of the DEW algorithm for MPEG compressed video

To evaluate the DEW algorithm for MPEG compressed video we have to compare it with the real-time watermarking algorithms known from literature as described in Chapter 3. Since the bit domain methods do not survive MPEG decoding and re-encoding, we limit ourselves here to the correlation-based methods described in Section 3.3. Because the method described in [Wu97] decreases the visual quality of the video stream drastically, the method described in [Har98] is the only comparable real-time watermarking method that operates directly on compressed video, while the video bit-rate remains constant.

The authors [Har98] report that the complexity of their watermark embedding process is much lower than the complexity of a decoding process followed by watermarking in the spatial domain and re-encoding and that the complexity is somewhat higher than the complexity of a full MPEG decoding operation. Since the DEW algorithm adds a watermark only by removing DCT-coefficients and no DCT, IDCT or full decoding steps are involved, the complexity of the DEW algorithm is less than half the complexity of a full MPEG decoding operation.

In Figure 6.4.1 an indication is given of the execution times of the following operations on 60 frames of MPEG-2 encoded video. The first bar represents the execution time of a full software MPEG decoding step followed by an MPEG re-encoding step. These steps are necessary if we want to embed a watermark to the compressed video data for instance using the method of [Smi96]. The second bar represents the execution time of a full software MPEG decoding step. This step is required to extract a watermark from the compressed video data for instance using the method of [Smi96]. The third and fourth bars represent the execution times of the fastest software implementation of the correlation-based watermarking algorithm described in Section 3.3 [Har98] and the DEW algorithm. The execution times are normalized such that the execution time of MPEG-2 decoding 60 frames equals to 10.

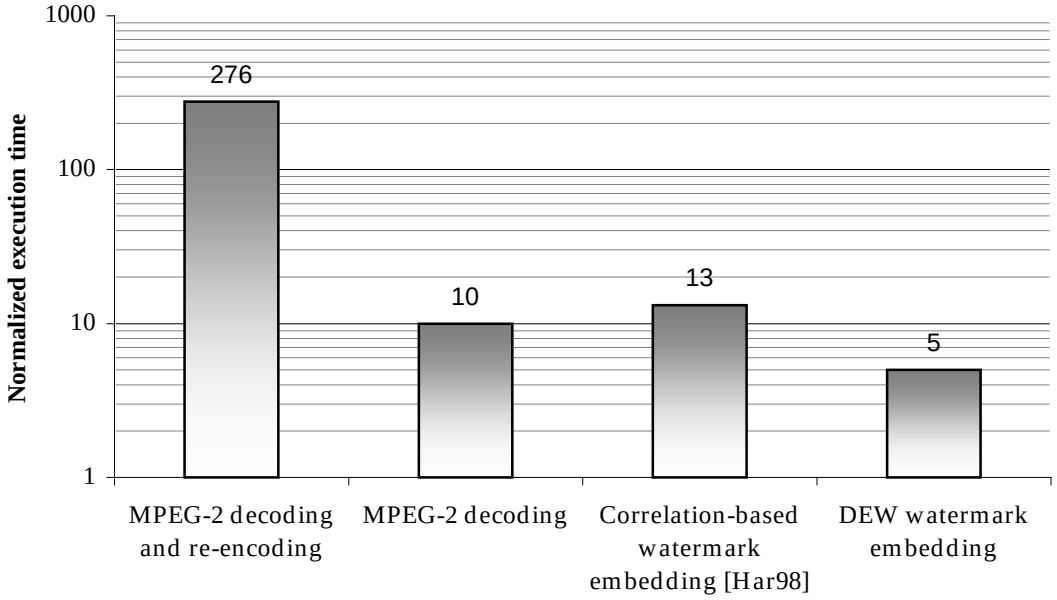


Figure 6.4.1. Normalized execution times of software MPEG-2 re-encoding and decoding operations in comparison to two real-time watermarking techniques.

Concerning the payload of the watermark, the DEW algorithm clearly outperforms the real-time correlation-based method. The authors [Har98] report maximum watermark label bit-rates of only a few bytes per second, while the DEW algorithm has a watermark label bit-rate of up to 52 bytes per second (see Table 4.4.1).

Since no experimental results about robustness against transcoding are reported in literature for the real-time correlation-based method [Har98], we compare the DEW algorithm with the basic spread spectrum method of Smith and Comiskey [Smi96]. Although the real-time method of [Har98] uses the same basic principles as the method of [Smi96], the latter method can embed 100% of the watermark energy instead of 0.5-3% and has a much higher payload, since it is not limited by the constraint that the watermark embedding process must take place in the compressed domain.

To evaluate the resistance to transcoding or re-encoding at a lower bit-rate, the following experiments are performed. The "sheep-sequence" described in Section 3.4.2.1 is MPEG-2 encoded at 8 Mbit/s. This compressed stream is directly watermarked with the DEW algorithm using 3 different parameter settings:

- $n = 32$, $D = 20$, $c_{min}=6$, $D' = 15$, without pre-quantization (0.42kbit/s)
- $n = 64$, $D = 20$, $c_{min}=6$, $D' = 15$, without pre-quantization (0.21kbit/s)
- $n = 64$, $D = 20$, $c_{min}=6$, $D' = 15$, with pre-quantization in the embedding stage (0.21kbit/s)

Pre-quantization means here that, prior to the calculation of the energies (Equation 4.2.1), the DCT-coefficients of MPEG compressed video are pre-quantized using the default

MPEG intra block quantizer matrix [ISO96]. The DCT-coefficients are divided by this matrix, rounded and multiplied by the same matrix.

Next, the "sheep-sequence" encoded at 8Mbit/s is watermarked with the spatial spread spectrum method [Smi96] (Section 2.2.2) by subsequently decoding, watermarking the I-frames and re-encoding the video stream. For the watermarking procedures the following settings are used:

- k=1, P=64x64, without pre-filter in the detector (0.21kbit/s)
- k=1, P=64x64, with pre-filter in the detector (0.21kbit/s)
- k=2, P=64x64, without pre-filter in the detector (0.21kbit/s)
- k=2, P=64x64, with pre-filter in the detector (0.21kbit/s)

As pre-filter a 3x3 edge-enhance filter is applied to the pixels of the I-frames before the correlation is calculated. The convolution kernel of the filter is given by Equation 2.2.5.

Hereafter, the watermarked video sequences are transcoded at different lower bit-rates. The label bit strings are extracted from the transcoded video streams and each label bit string is compared with the originally embedded label bit string. If 50% bit errors are made the label is completely removed. The percentages label bit errors introduced by decreasing the bit-rate are represented in Figure 6.4.2.

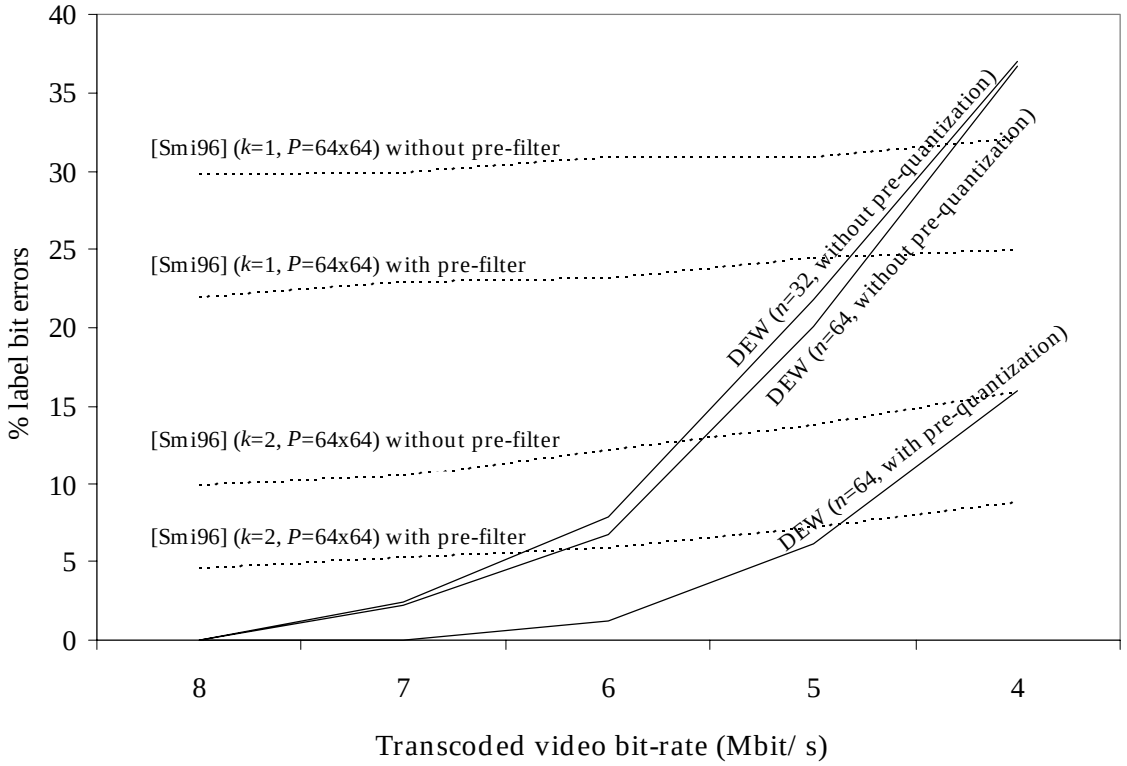


Figure 6.4.2. % Bit errors after transcoding a watermarked 8Mbit/s MPEG-2 sequence at a lower bit-rate.

From this figure several conclusions can be drawn. First, for the DEW algorithm it appears that increasing the number of 8x8 DCT blocks per label bit does not significantly increase the robustness to transcoding. This yields that increasing *n* is only necessary if the watermarking process results in visual artefacts, otherwise it is preferable to choose *n* as low as possible and use error correcting codes to improve the robustness.

Second, the robustness of the DEW algorithm increases drastically if pre-quantization is used during the embedding stage. If we take a closer look at the results of the video stream transcoded to 5Mbit/s and plot the percentages label bit errors of each frame in Figure 6.4.3 instead of the averages over 21 frames from Figure 6.4.2, we see that in some frames still no errors occur after transcoding (frame numbers: 1,2,7,20). However in some other frames the percentage label bit errors is quite high (frame numbers: 12,13). This is due to the fact that for the experiments a fixed pre-quantization level is used for each frame. This is not an optimal solution, since in MPEG coded video streams the quantization levels vary not only temporally but also spatially, dependent on the video bit-rate, video content and buffer space of the encoder. The robustness of the DEW algorithm can therefore be improved further by locally adapting the pre-quantization.

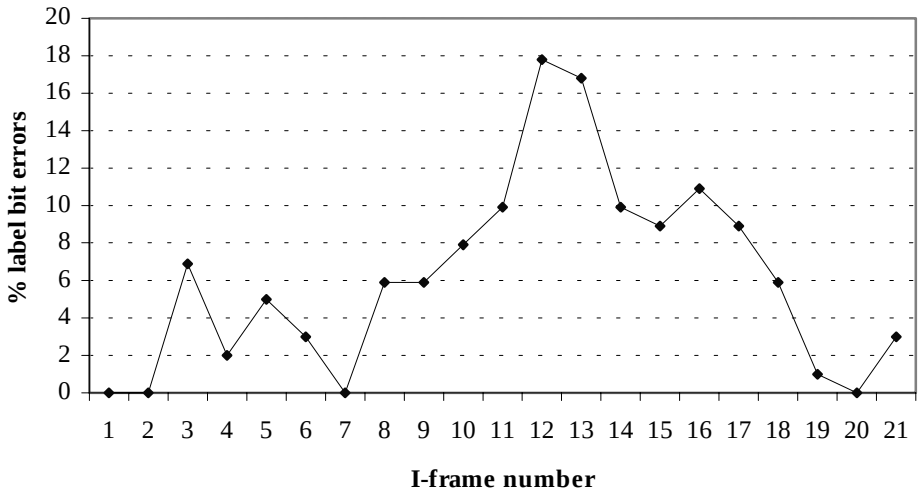


Figure 6.4.3. % Bit errors after transcoding an 8Mbit/s MPEG-2 sequence water-marked using the DEW algorithm (*n*=64, with pre-quantization) at 5Mbit/s.

The third and last conclusion that can be drawn from Figure 6.4.2 is that the DEW algorithm outperforms the correlation-based method [Smi96] with respect to the transcoding attack for bit-rates between 8 and 5 Mbit/s.

Since the real-time correlation-based version described in [Har98] is only capable of embedding 0.5…3% of the total watermark energy which is embedded using [Smi96] due to the bit-rate constraint, it can be expected that this method performs less than the method of [Smi96] and the DEW algorithm concerning the transcoding attack.

6.4.4 Evaluation of the DEW algorithm for still images

To evaluate the DEW algorithm for JPEG compressed and uncompressed still images we compare it to the basic spread spectrum method of Smith and Comiskey [Smi96]. For all experiments in this section the parameter settings optimized for robustness are used for the DEW algorithm, namely c_{min}=3, n=64, Q_{jpeg}=25 and D=400. For the watermark extraction the parameters n=64 and D'=400 are used. Since the detector results are significantly influenced by the pre-quantization stage in the detector, a value for Q'_{jpeg} is chosen out of the set [25, 80, 99] such that the error rate of the detector is minimized. This process can be automated by for instance starting the label bit string with several fixed label bits, so that during the extraction the value Q'_{jpeg} can be chosen that results in the fewest errors in the known label bits.

For all experiments in this section with the method of [Smi96], P=64x64 pixels are used to store each label bit, while the watermark carrier consists of the integers {-2,2} (k=2). This means that the watermarks embedded with both methods have the same payload. Since we noticed in the previous section that pre-filtering significantly improves the performance of the correlation based method [Smi96], we apply a 3x3 edge-enhance filter to the watermarked images before the correlation is calculated. The convolution kernel of the filter is given by Equation 2.2.5.

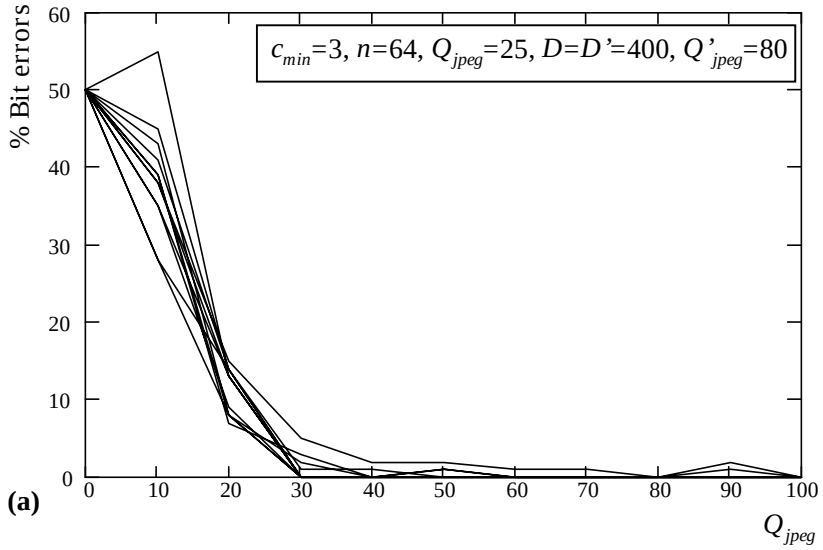
We watermarked a set of twelve images with the two watermarking methods using the parameter settings described above. First we calculate the ITU-R Rec. 500 quality ratings of the watermarked images using the approach described in Section 6.2 (Equation 6.2.1) and test the robustness of the watermarks against the attacks described in Section 6.3. In Table 6.4.1 the results of these experiments are listed for the DEW algorithm. For the StirMark attack version 1.0 is used, using the default parameter settings. In this version only the geometrical distortions are performed as described in Section 6.3.2, the final JPEG compression step is not implemented.

Table 6.4.1. ITU-R Rec. 500 quality ratings and percentages label bit errors for the DEW algorithm after applying the StirMark attack based on geometrical distortions (Q'_{jpeg}=99) and the Watermark Removing Scheme (WRS) based on watermark estimation (Q'_{jpeg}=25).

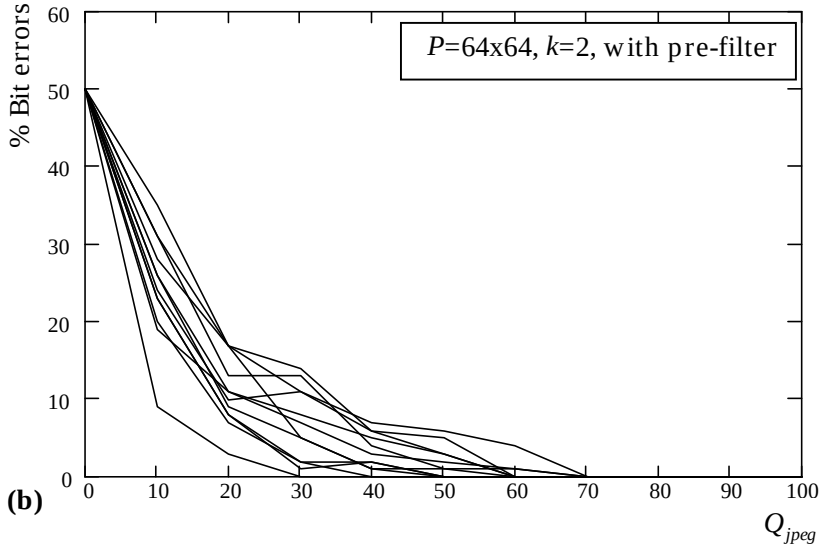
Image name	**Size**	**ITU-R Rec. 500 rating**	**% Label bit errors**	
			StirMark Attack[Pet98b]	**WRS [Lan98b]**
Bike	720x512	4.3	34%	7%
Bridge	720x512	4.5	16%	17%
Butterfly	720x512	4.6	11%	7%
Flower	720x512	4.5	15%	5%
Grand Canyon	720x512	4.4	24%	13%
Lena	512x512	4.6	17%	6%
Parrot	720x512	4.7	28%	8%
Rafter	720x512	4.3	24%	7%
Red Square	720x512	4.6	15%	7%
Sea	720x512	4.4	15%	4%
Temple	720x512	4.6	17%	5%
Tree	720x512	4.3	9%	13%

For the images watermarked with the method of [Smi96] the ITU-R Rec. 500 quality ratings are in the range of 4.7…4.8, the percentages label bit errors for the StirMark attack exceed 40% for all images and the percentages label bit errors for the watermark removal attack by non-linear filtering exceed 30% for all images.
From Table 6.4.1 it can be concluded that the DEW algorithm affects the visual quality marginally more than the correlation-based method, however the ITU-R quality ratings are far above the required minimum of 4. Further it can be concluded that the DEW algorithm clearly outperforms the correlation-based method concerning both watermark removal attacks.

To evaluate the robustness of both algorithms against common simple processing techniques we further tested the robustness against re-encoding, linear and non-linear filtering, noise addition, simple geometrical transformations, gamma correction, dithering and histogram equalization.



(a)



(b)

Figure 6.4.4. Percentages bit errors after re-encoding **(a)** using the DEW algorithm; **(b)** using the correlation based method of [Smi96].

A set of twelve images is watermarked with both watermarking methods. The images are re-encoded using a lower JPEG quality factor. The quality factor of the re-encoding process is made variable. Finally, the watermarks are extracted from the re-encoded images and compared bit by bit with the originally embedded watermarks. From this experiment, we find the percentages of label bit errors due to re-encoding as a function of the re-encoding quality factor. In Figure 6.4.4 the resulting label bit error curves are shown for twelve different images.

As can be seen in Figure 6.4.4 the DEW algorithm is slightly more robust to re-encoding attacks than the correlation based method. To test the robustness against non-linear filtering the test set of twelve images watermarked with both watermarking methods is filtered using a median filter with a kernel size of 3x3. To test the robustness against linear filtering the watermarked images are first filtered with a 3x3 smoothing filter F_{smooth} and subsequently filtered with an edge-enhance filter F_{edge}, where F_{smooth} and F_{edge} are given by the following convolution kernels:

$$F_{smooth} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix} /13 \qquad \text{and} \qquad F_{edge} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix} / 2 \qquad (6.4.1)$$

The percentages label bit errors in the labels extracted from the non-linear and linear filtered images are presented in Table 6.4.2.

Table 6.4.2. Percentages label bit errors for the DEW algorithm (Q'_{jpeg}=99) and the correlation-based method of [Smi96] after applying non-linear and linear filters to the watermarked images.

Image name	**% Label bit errors**			
	Median Filtering 3x3		**Linear Filtering**	
	DEW	**Corr.-based**	**DEW**	**Corr.-based**
Bike	16%	3%	10%	0%
Bridge	9%	8%	0%	0%
Butterfly	15%	3%	0%	0%
Flower	9%	0%	0%	0%
Grand Canyon	18%	4%	2%	0%
Lena	5%	2%	0%	0%
Parrot	22%	3%	1%	0%
Rafter	15%	2%	1%	0%
Red Square	14%	10%	0%	0%
Sea	10%	7%	0%	0%
Temple	13%	8%	0%	0%
Tree	21%	15%	0%	0%

From Table 6.4.2 it appears that both methods are more vulnerable to non-linear filtering than linear filtering. The correlation-based method is slightly more robust to filtering than

the DEW algorithm. The reason for this is that the energy of the DEW algorithm is more or less located in a mid-frequency band, and the energy of the correlation-based method is uniformly distributed over the spectrum. If some frequency bands are affected by filtering operations, there is enough energy left in other frequency bands in the case of the correlation-based method.

Correlation-based methods are quite resistant to uncorrelated additive noise. Experiments show that uniformly distributed noise in the range from -25 to 25 added to images watermarked with the method of [Smi96] does not introduce label bit errors in the extracted labels (0%). To investigate the robustness of the DEW algorithm against additive noise, we add noise to the watermarked images, where the noise amplitude [$-N_a$,N_a] varies between 0 and 25. The results of this experiment are shown in Figure 6.4.5.

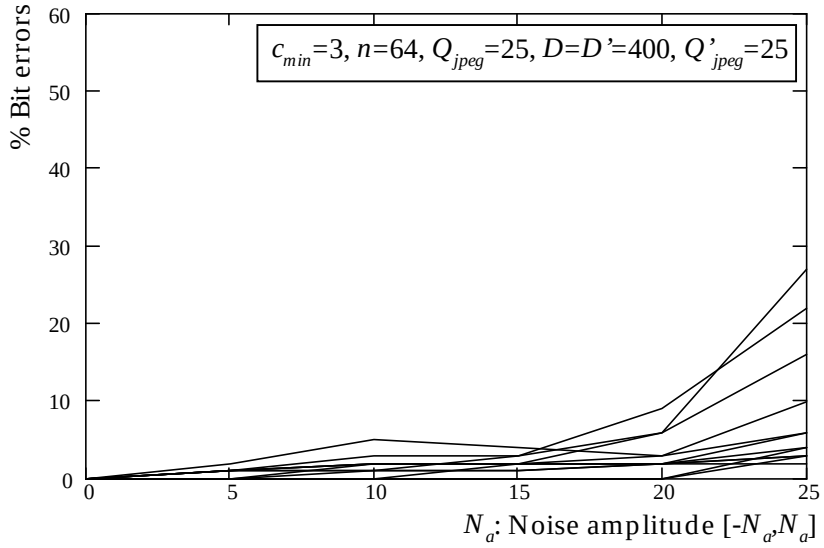


Figure 6.4.5. Percentages label bit errors after extracting labels from images affected with additive noise using the DEW algorithm.

From this figure it appears that the DEW algorithm is also quite insensitive to additive noise.

Robustness against geometrical distortions is very important, since shifting, scaling and rotating are very simple processing operations that hardly introduce visual quality losses. We already tested the robustness of the DEW algorithm against line shifting followed my lossy JPEG compression in Section 5.6 and the resistance to minor geometrical distortions applied by StirMark at the beginning of this section. Nevertheless we perform here some additional experiments to check the robustness against scaling and rotating. We enlarge the watermarked images 1% and crop them to their original size. Next, we rotate the watermarked images 0.5 degree and crop them to their original size. Finally, the watermark labels are extracted and bit-by-bit compared with the originally embedded ones. The percentages label bit errors in the labels extracted from the scaled and rotated images are presented in Table 6.4.3 for the DEW algorithm. It appears that these geometrical transformations, line shifting, scaling and rotating, completely remove the

watermarks embedded by the correlation-based method (percentages bit errors > 40). From Table 6.4.3 and the experiments performed in Section 5.6 it can be concluded that the DEW algorithm clearly outperforms the correlation-based method concerning geometrical transformations.

Table 6.4.3. Percentages label bit errors for the DEW algorithm (Q'_{jpeg}=99) after scaling or rotating and cropping the watermarked images.

Image name	**% Label bit errors**	
	Zoom 1% and crop	**Rotate 0.5° and crop**
Bike	14%	17%
Bridge	3%	5%
Butterfly	0%	7%
Flower	7%	9%
Grand Canyon	17%	6%
Lena	0%	6%
Parrot	11%	10%
Rafter	10%	6%
Red Square	10%	3%
Sea	10%	10%
Temple	7%	8%
Tree	3%	0%

Both the DEW algorithm and the correlation-based method are insensitive to gamma correction and histogram equalization. Even quantization of the color channels from 256 levels to 32, 16 or 8 levels followed by dithering does not affect the watermarks embedded by the DEW algorithm (Q'_{jpeg}=25) or the correlation-based method.

6.5 Discussion

Benchmarking watermarking algorithms is a difficult task. Performance factors like visibility, robustness, payload and complexity have to be taken into account, but the weighting of these factors is application dependent. Furthermore it is questionable if robustness can be defined formally.

In this chapter we discussed two benchmarking approaches for watermarking methods and two dedicated watermark removal attacks. The benchmarking approaches discussed here only give some general guidelines of how watermarking methods can be evaluated. More research and standardization is necessary to derive more sophisticated benchmarking systems. Also the attacks discussed here are just examples to show that robustness against simple standard image processing techniques is not enough to call a watermarking method robust. Other simple processing techniques exist or may be developed, which do not significantly affect the image quality, but can defeat most watermarking schemes.

The attacks presented here can be counterattacked by increasing the complexity of the watermark detectors. But the attacks in their turn can also be improved by taking these changes of the detectors into account. For instance, the watermark removal technique

presented in Section 6.3.3 can be counter attacked by applying a special low-pass pre-filter in the detector [Har99]. However, by replacing the 3x3 high-pass filter in the removal scheme by a filter with a larger kernel and appropriate coefficients this counterattack can be rendered useless. Furthermore attacks can be improved by combining them, for instance, combining a watermark estimation attack with a geometrical transformation attack will defeat any watermarking scheme.

In spite of the problems mentioned above, we evaluated the DEW algorithm in this chapter taking into account the benchmarking approaches and attacks from literature. In comparison to other real-time watermarking algorithms for MPEG compressed video known from literature, we found that the correlation-based method described in [Har98] is the only algorithm that can directly be compared with the DEW algorithm. In this comparison it turned out that the DEW algorithm has only less than half the complexity of this correlation-based method. Furthermore, the payload of the DEW algorithm is up to 25 times higher and the DEW algorithm is more robust against transcoding attacks than the correlation-based methods in the spatial domain. The robustness of the DEW algorithm can even be improved further by making the pre-quantization step variable.

We also compared the DEW algorithm for still images to the basic spread spectrum method of [Smi96], which is not designed for real-time watermarking in the compressed domain.

In this comparison it turned out that the DEW algorithm and the correlation-based method perform equally well concerning the robustness against linear filtering, histogram equalization, gamma correction, dithering and additive noise. The DEW algorithm clearly outperforms the correlation-based method concerning the dedicated watermark removal attacks, geometrical transformations and re-encoding attacks using lossy JPEG compression.

Chapter 7

Discussion

7.1 Reflections

In this thesis we investigated techniques for the real-time embedding of watermarks in and extracting watermarks from compressed image and video data. This class of watermarking techniques is particularly suitable for fingerprinting and copy protection systems in consumer applications.

We noticed that the most efficient way to reduce the complexity of real-time watermarking algorithms is to avoid computationally demanding operations by exploiting the compression format of the video data. The advantage of this approach is that watermarks automatically become video content dependent. Watermarks directly embedded in the compressed domain can only be embedded in the visual important areas, since lossy compression algorithms only encode the video information to which the Human Visual System is most sensitive. A disadvantage of closely following a compression standard and applying the constraint that the size of the compressed video stream may not increase is that conventional watermarking methods either cannot be used or can only add a significantly smaller amount of the watermark energy to the compressed data than they can add to uncompressed data. The distortions caused by watermarks directly applied on a compressed video stream also differ from the distortions caused by watermarks applied on an uncompressed video stream. Due to block-based transformations and motion compensated frame prediction, distortions may spread over blocks and accumulate over the consecutive frames.

Because of the limitations of the conventional watermarking methods, we developed two new watermarking concepts that directly operate on the compressed data stream, namely the least significant bit (LSB) modification concept and the Differential Energy Watermark (DEW) concept.

The LSB concept only replaces fixed or variable length codes in the compressed data stream and is therefore computationally highly efficient. Using this concept extremely high label bit-rates of up to 29kbit/s can be achieved. However, a drawback of the LSB concept is that the watermark embedding and extraction procedures are completely dependent on the data structure of the compressed video stream. Once a compressed video stream watermarked by an LSB-based watermarking method is decompressed, the watermark is lost. Since fully decompressing and re-compressing a video stream is a task

that is computationally quite demanding, this is not really an issue for consumer applications requiring moderate robustness.

For real-time applications that require a higher level of robustness but do not have enough computational power to perform for instance a full MPEG decoding step for watermark detection, we have developed the DEW watermarking concept. For embedding a watermark in or extracting a watermark from a compressed video stream, the DEW algorithm only requires partial decoding steps; it does not require a drift compensation signal or partial video encoding steps. The complexity of the DEW watermarking algorithm is therefore only slightly higher than that of the LSB-based methods, while watermarks embedded with the DEW concept can also be extracted from decoded raw video data, since the watermarks are not dependent on the data structure of the compressed video stream.

Besides the low complexity, the DEW concept also has several other advantages over other watermarking methods. First, it provides a parameter to anticipate to re-encoding attacks. Second, it exhibits some degree of resistance to geometrical distortions. Third, it is directly applicable to video data compressed using coders other than MPEG coders, for instance embedded zero-tree wavelet coders. Fourth, since it exploits the fact that the MPEG encoding algorithm uses a kind of data ordering that complies more or less with the Human Visual System (HVS), it is able to embed perceptually invisible watermarks without explicitly using a model of the HVS. Fifth, it is possible to derive a statistical model for the DEW watermarking algorithm, which gives us an expression for the label bit error probability as a function of the DEW watermark embedding parameters. Using this expression, we can optimize a watermark for robustness, size or visibility and add adequate error correcting codes.

To evaluate the performance of the DEW algorithm we compared it with other real-time and conventional watermarking algorithms. We only found one other real-time correlation-based watermarking algorithm for MPEG compressed video in literature that can directly be compared with the DEW algorithm. In the comparison it turned out that the DEW algorithm has only less than half the complexity of this correlation-based method. Furthermore, the payload of the DEW algorithm is up to 25 times higher and the DEW algorithm is more robust against transcoding attacks than the correlation-based methods in the spatial domain. We also compared the DEW algorithm for still images to a basic spread spectrum method which is not designed for real-time watermarking in the compressed domain. In this comparison it turned out that the DEW algorithm and the correlation-based method perform equally well concerning the robustness against linear filtering, histogram equalization, gamma correction, dithering and additive noise. The DEW algorithm clearly outperforms the correlation-based method concerning the dedicated watermark removal attacks, geometrical transformations and re-encoding attacks using lossy JPEG compression.

7.2 Further extensions

Although the performance results are already very satisfactory, the DEW concept can even be improved further in the following ways. First, we discovered from the statistics that the payload of the watermark can almost be doubled if adequate error correcting codes are used, while the label error probability remains constant. Experiments showed that

decreasing the payload of the watermark by increasing the number of pixels per label bit did not significantly improve the robustness. This yields the conclusion that increasing the number of pixels per label bit is only necessary if the watermarking process results in visual artefacts; otherwise it is preferable to choose this number as low as possible and to use error correcting codes to improve the robustness. Second, it is possible to introduce additional measures to protect the visual quality of the watermarked data beyond to the currently implemented minimum cut-off level. For instance, the energy that has to be discarded in an 8x8 DCT block can be limited to a maximum amount, which is defined by a certain percentage of the total AC-energy present in that particular 8x8 DCT block. Third, experiments also show that the robustness of a DEW watermark for MPEG compressed video can be improved significantly by making the pre-quantization step variable. Fourth, the statistical model can be used to adjust the embedding parameters according to the video data beforehand to minimize the visual impact of the watermark.

Both the LSB concept and the DEW concept are not limited to MPEG/JPEG coded video only. The general ideas behind the concepts can prove their usefulness in the future for compressed audio formats and for new compressed video formats. The general LSB concept of replacing fixed or variable length codes representing audio or video data by a codes with the same size which represent slightly deviating data can be applied to all kinds of compression formats for audio and video data. The process of enforcing energy differences is also not dependent on the compression format, which makes the DEW concept suitable for emerging compression formats. It is possible to introduce minor high-frequency energy differences between two regions in compressed audio or video data by compressing one of the regions a little bit more than the other regions. As an example we showed that the DEW-algorithm for MPEG/JPEG coded video data is directly applicable to embedded zero tree wavelet coded data after minor modification.

7.3 Future research

In comparing the DEW algorithm to other watermarking algorithms we noticed that benchmarking watermarking algorithms is a difficult task. The benchmarking approaches that can be found in literature only give some general guidelines of how watermarking methods can be evaluated. More research and standardization are required to derive more sophisticated benchmarking systems. Performance factors like visibility/audibility, robustness, payload and complexity have to be taken into account. Problems here are the application dependent weighting of these performance factors and the definition of visibility/audibility and robustness. It is even questionable if robustness can be defined formally. The existence of dedicated watermark removal attacks shows that robustness against simple standard video processing techniques is not enough to call a watermarking method robust. Other simple processing techniques exist or may be developed which do not significantly affect the visual quality but can defeat most watermarking schemes.

The existence and further development of sophisticated lossy compression techniques, watermark estimation attacks and geometrical watermark removal attacks significantly limits the bandwidth for watermarking algorithms to reliably transfer watermark information. This is in contrast with the demands of emerging applications which, for instance, require robust watermarks that can store large identification numbers in very short audio samples.

Future research will therefore be aimed at attempting to improve the robustness and payload of the watermarking methods in the following two ways. First, by incorporating human perceptual knowledge in watermark embedding schemes to increase the watermark energy and second, by increasing the complexity of the watermark detectors. Increasing the watermark energy by exploiting Human Perceptual System models only works if lossy compression algorithms do not exploit those models or exploit less sophisticated models. Otherwise the extra watermark energy will simply be discarded by lossy compression schemes. However, it can be expected that a higher performance gain can be achieved by increasing the intelligence of the watermark detector. Experiments already showed significant improvement of matched filtering and exhaustive search techniques in the detectors. Developing watermarking methods that are resistant to a combination of a watermark estimation attack, a geometrical transformation attack and lossy compression schemes will therefore be a major challenge for future research.

Bibliography

[Aka96] Ali N. Akansu, Mark J.T. Smith eds., "Subband and Wavelet Transforms: Design and Applications", Kluwer Academic Publishers, 1996

[And98] Ross J. Anderson, Fabien A.P. Petitcolas, "On the limits of steganography. IEEE Journal of Selected Areas in Communications", 16(4):474-481, May 1998, Special Issue on Copyright & Privacy Protection, ISSN 0733-8716

[Aur95] Tuomas Aura, "Invisible Communication", Proceedings of the HUT Seminar on Network Security '95, Espoo, Finland, November 6, 1995

[Aur96] Tuomas Aura, "Practical invisibility in digital communication", Proceedings of the Workshop on Information Hiding, Cambridge, England, May 1996, Lecture Notes in Computer Science 1174, Springer Verlag 1996

[Bar94] M. Barlaud ed., "Wavelets in Image Communication", Advances in image communication 5, Elsevier, ISBN:0444892818, 1994

[Bar98] F. Bartolini, M. Barni, V. Cappellini, A. Piva, "Mask building for perceptually hiding frequency embedded watermarks", Proceedings of 5th IEEE International Conference on Image Processing ICIP'98, Chicago, Illinois, USA, Vol I, pp. 450-454, 4-7 October 1998

[Bar99] Mauro Barni, Franco Bartolini, Vito Cappellini, Alessandro Lippi, Alessandro Piva, "A DWT-based technique for spatio-frequency masking of digital signatures", Proceedings of the SPIE/IS&T International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, San Jose, CA, January 25 - 27, 1999

[Bas98] Patrick Bas, Jean-Marc Chassery, Franck Davoine, "Self-similarity based image watermarking", IX European Signal Processing Conference, Island of Rhodes, Greece, 8-11 September 1998

[Bas99] Patrick Bas, Jean-Marc Chassery, Franck Davoine, "A Geometrical and Frequential Watermarking Scheme Using Similarities", Proceedings of SPIE Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose (CA), USA, January 1999

[Ben95] W. Bender, D. Gruhl, N. Morimoto, "Techniques for Data Hiding", Proceedings of the SPIE, Vol. 2420 pp 165-173, Storage and retrieval for image and Video Databases III, San Jose CA, USA, 9-10 February 1995

[Bol95] F.M. Boland, J.J.K. Ó Ruanaidh and C. Dautzenberg, "Watermarking Digital Images for Copyright Protection", IEE Int. Conf. on Image Processing and Its Applications, pp 326-330, Edinburgh, Scotland, July 1995

[Bor96a] A.G. Bors, I.Pitas, "Embedding Parametric Digital Signatures in Images", EUSIPCO-96, Trieste, Italy, vol. III, pp. 1701-1704, September 1996

[Bor96b] A.G. Bors, I.Pitas, "Image Watermarking Using DCT Domain Constraints", IEEE International Conference on Image Processing (ICIP'96), Lausanne, Switzerland, vol. III, pp. 231-234, 16-19 September 1996

[Bra96] C.J. van den Branden Lambrecht and J.E. Farrell, "Perceptual quality metric for digitally coded color images", EUSIPCO-96, pp. 1175-1178, Trieste, Italy, September 1996

[Bra97] G.W. Braudaway, "Protecting Publicly-Available Images with an Invisible Watermark", Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Bur98] S.Burgett, E.Koch, J.Zhao, "Copyright Labeling of Digitized Image Data", IEEE Communications Magazine, pp. 94-100, March 1998

[Car95] G. Caronni, "Assuring Ownership Rights for Digital Images", Proceedings of Reliable IT Systems, pp. 251-263, VIS '95, Vieweg Publishing Company, Germany, 1995

[Che99] Brian Chen, Gregory W Wornell, "An Information-Theoretic Approach to the Design of Robust Digital Watermarking Systems", Proceedings ICASSP' 99, Vol 4., Phoenix, Arizona, USA, 15-19 March 1999

[Col99] Dinu Coltuc, Philippe Bolon, "Watermarking by histogram specification", Proceedings of SPIE ELECTRONIC IMAGING '99, Security and Watermarking of Multimedia Contents, January 1999, San Jose (CA), USA

[Cox95] I.J. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", Technical Report 95 - 10, NEC Research Institute, Princeton, NJ, USA, 1995

[Cox96a] I.J. Cox, J. Kilian, T. Leighton and T. Shamoon, "A Secure, Robust Watermark for Multimedia", Preproceedings of Information Hiding, an Isaac Newton Institute Workshop, Univ. of Cambridge, May 1996

[Cox96b] I.J. Cox, J.Kilian, T.Leighton and T.Shamoon, "Secure spread spectrum watermarking for images, audio and video", Proceedings of the 1996 International Conference on Image Processing, vol. 3, pp. 243-246, Lausanne, Switzerland, September 16-19, 1996

[Cox97] Ingemar J. Cox, and Matt L. Miller "A review of watermarking and the importance of perceptual modeling", Proceedings of SPIE Electronic Imaging '97, Storage and Retrieval for Image and Video Databases V, San Jose (CA), USA, February 1997

[Cra96] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung, "Can invisible watermarks resolve rightful ownerships?", IBM Research Report RC 20509, 25 July 1996

[Dav96] Paul Davern, Michael Scott, "Fractal Based Image Steganography", Preproceedings of Information Hiding, An Isaac Newton Institute Workshop, pp 245-256, University of Cambridge, UK, May 1996

[DCM98] U.S. Copyright Office Summary, "The Digital Millennium Copyright Act of 1998", http://lcweb.loc.gov/copyright/legislation/dmca.pdf, December 1998

[Dep98] G. Depovere, T. Kalker, J.-P. Linnartz, "Improved watermark detection using filtering before correlation", Proceedings of 5th IEEE International Conference on Image Processing ICIP'98, Chicago, Illinois, USA, Vol I, pp. 430-434, 4-7 October 1998

[Fle97] D.J. Fleet, D.J. Heeger, "Embedding Invisible Information in Color Images", Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Fri99a] Jiri Fridrich, Miroslav Goljan, "Comparing robustness of watermarking techniques", Electronic Imaging '99, The International Society for Optical Engineering, Security and Watermarking of Multimedia Contents, vol 3657, San Jose, CA, USA, 25-27 January 1999

[Fri99b] Jiri Fridrich, "Robust Bit Extraction From Images", submitted to IEEE ICMCS'99 Conference, Florence, Italy, 7-11 June 1999

[Fri99c] Jiri Fridrich, Miroslav Goljan, "Protection of Digital Images Using Self Embedding", submitted to The Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, March 16, 1999

[Gir87] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences", IEEE Journal on Selected Areas in Communications, Vol. 5, pp. 1140-1154, August 1987

[Gir89] B. Girod, "The information theoretical significance of spatial and temporal masking in video signals", Proceedings of the SPIE Human Vision , Visual Processing, and Digital Display, Vol. 1077, pp. 178-187, 1989

[Gof97] F. Goffin, J.-F. Delaigle, C. De Vleeschouwer, B.Macq and J-.J. Quisquater, "A low cost perceptive digital picture watermarking method", Proceedings of SPIE Electronic Imaging '97, Storage and Retrieval for Image and Video Databases V, San Jose (CA), USA, February 1997

[Han96] A. Hanjalic, G.C. Langelaar, R.L. Lagendijk, M. Ceccarelli, M. Soletic, "Report on Technical Possibilities and Methods for Security of SMASH and for Fast Visual Search on Compressed/encrypted Data", Deliverable #5, AC-018, SMASH, SMS-TUD-648-1, November 1996

[Har96] F. Hartung and B. Girod: "Digital Watermarking of Raw and Compressed Video", Proceedings SPIE 2952: Digital Compression Technologies and Systems for Video Communication, pp 205-213, October 1996 (Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies, Berlin, Germany)

[Har97a] F. Hartung and B. Girod, "Watermarking of MPEG-2 Encoded Video Without Decoding and Re-encoding", Proceedings Multimedia Computing and Networking 1997 (MMCN 97), San Jose, CA, February 1997

[Har97b] F. Hartung and B. Girod, "Digital Watermarking of MPEG-2 Coded Video in the Bitstream Domain", Proceedings ICASSP 97, Volume 4, pp. 2621-2624, Munich, Germany, 21-24 April 1997

[Har97c] F. Hartung and B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video", in: S. Fdida, M. Morganti (eds.), "Multimedia Applications, Services and Techniques - ECMAST '97", Springer Lecture Notes in Computer Science, Vol. 1242, pp. 423-436, Springer, Heidelberg, 1997

[Har98] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", Signal Processing, Vol. 66, no. 3, pp. 283-301, (Special Issue on Copyright Protection and Control, B. Macq and I. Pitas, eds.), May 1998

[Har99] F. Hartung, J.K. Su and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counterattacks", Proceedings of SPIE Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose (CA), USA, January 1999

[Her72] Herodotus, "The Histories (trans. A. de Selincourt), Middlesex, England: Penguin, 1972

[Her98a] Alexander Herrigel, Holger Petersen, Joseph O' Ruanaidh, Thierry Pun, Pereira Shelby, "Copyright Techniques for Digital Images Based On Asymmetric Cryptographic Techniques", Workshop on Information Hiding, Portland, Oregon, USA, April 1998

[Her98b] Alexander Herrigel, Joe J. K. Ó Ruanaidh, Holger Petersen, Shelby Pereira and Thierry Pun, "Secure copyright protection techniques for digital images", In David Aucsmith ed., Information Hiding, pp. 169-190, Vol. 1525 of Lecture Notes in Computer Science, Springer, Berlin, 1998

[Hir96] K. Hirotsugu, "An Image Digital Signature System with ZKIP for the Graph Isomorphism", Proceedings ICIP-96, IEEE International Conference on Image Processing, Volume III pp. 247-250, Lausanne, Switzerland, 16-19 September 1996

[Hsu96] C.-T. Hsu, J.-L. Wu, "Hidden Signatures in Images", Proceedings ICIP-96, IEEE International Conference on Image Processing, Volume III, pp. 223-226, Lausanne, Switzerland, 16-19 September 1996

[IEC958] Digital audio interface, International Standard, IEC 958

[Int97] International Federation of the Phonographic Industry, "Request for Proposals", Embedded Signalling Systems Issue 1.0. 54 Regent Street, London W1R 5PJ, June 1997

[ISO96] ISO/IEC 13818-2:1996(E), "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information", Video International Standard, 1996

[ISO95] Recommendation ITU-R BT.500-7, "Methodology for the subjective assessment of the quality of television pictures", International Telecommunication Union, Broadcasting Service (Television), 1995 BT Series Fascicle, Radiocommunication assembly, Geneva, 1995

[Jac92] A.E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations", IEEE transactions on Image Processing, Vol. 2, No. 1, pp. 18-30, January 1992

[Jai81] Anil K. Jain, "Image Data Compression: A Review", Proceedings IEEE, Vol. 69, no. 3, pp. 349-389, March 1981

[Joh98] Neil F. Johnson, Sushil Jajodia, "Exploring Steganography: Seeing the Unseen", IEEE Computer, Vol. 31, no 2, pp26-34, Februari 1998

[Kah67] D. Kahn, "The Codebreakers", New York: MacMillan, 1967

[Kal98a] Ton Kalker, Jean-Paul Linnartz, Geert Depovere, Maurice Maes, "On the Reliability of Detecting Electronic Watermarks in Digital Images", IX European Signal Processing Conference, Island of Rhodes, Greece, 8-11 September 1998

[Kal98b] Ton Kalker, Jean-Paul Linnartz, Marten van Dijk, "Watermark estimation through detector analysis", Proceedings of 5th IEEE International Conference on Image Processing ICIP'98, Chicago, Illinois, USA, 4-7 October 1998

[Kal99] Ton Kalker, Geert Depovere, Jaap Haitsma, Maurice Maes, "A Video Watermarking System for Broadcast Monitoring", Proceedings of SPIE ELECTRONIC IMAGING '99, Security and Watermarking of Multimedia Contents, January 1999, San Jose (CA), USA

[Kob97] M. Kobayashi, "Digital Watermarking: Historical Roots", IBM Research Report, RT0199, Japan, April 1997

[Koc94] E. Koch, J. Rindfrey, J. Zhao, "Copyright Protection for Multimedia Data", Proceedings of the International Conference on Digital Media and Electronic Publishing, Leeds, UK, 6-8 December 1994

[Koc95] E. Koch, J. Zhao, "Towards Robust and Hidden Image Copyright Labeling", Proceedings IEEE Workshop on Non-lineair Signal and Image Processing, pp. 452-455, Neos Marmaras (Thessaloniki Greece), June, 1995

[Kun97] D. Kundur, D. Hatzinakos, "A robust digital image watermarking scheme using wavelet-based fusion," Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Kun98] D. Kundur, D. Hatzinakos, "Digital watermarking using multi resolution wavelet decomposition", Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, Washington, Vol. 5, pp. 2969-2972, May 1998

[Kut97] M. Kutter, F. Jordan, F. Bossen, "Digital Signature of Color Images using Amplitude Modulation", Proceedings of SPIE Electronic Imaging '97, Storage and Retrieval for Image and Video Databases V, San Jose (CA), USA, February 1997

[Kut98] M. Kutter "Watermarking resistant to rotation, translation and scaling", Proceedings of SPIE, Boston, USA, November 1998

[Kut99] M. Kutter, F.A.P. Petitcolas, "A fair benchmark for image watermarking systems", Electronic Imaging '99. Security and Watermarking of Multimedia Contents, vol 3657, San Jose, CA, USA 25-27 January 1999, The International Society for Optical Engineering

[Kuw76] M. Kuwahara, K. Hachimura, S. Eiho, M. Kinoshita, "Processing of RI-angiocardiographic images", in Digital Processing of Biomedical Images, K. Preston and M. Onoe, Editors, p. 187-203, Plenum Press, New York, 1976

[Lan96a] G.C. Langelaar, J.C.A. van der Lubbe, J. Biemond, "Copy Protection for Multimedia Data based on Labeling Techniques", 17th Symposium on Information Theory in the Benelux, Enschede, The Netherlands, 30-31 May 1996

[Lan96b] G.C. Langelaar, "Feasibility of security concept in hardware", AC-018, SMASH, SMS-TUD-633-1, August 1996

[Lan97a] G.C. Langelaar, J.C.A. van der Lubbe, R.L. Lagendijk, "Robust Labeling Methods for Copy Protection of Images", Proceedings of SPIE Electronic Imaging '97, Storage and Retrieval for Image and Video Databases V, San Jose (CA), USA, February 1997

[Lan97b] G.C. Langelaar, R.L. Lagendijk, J. Biemond "Real-time Labeling Methods for MPEG Compressed Video", 18th Symposium on Information Theory in the Benelux, Veldhoven, The Netherlands, 15-16 May 1997

[Lan98a] G.C. Langelaar, R.L. Lagendijk, J. Biemond, "Real-time Labeling of MPEG-2 Compressed Video", Journal of Visual Communication and Image Representation, Vol 9, No 4, December, pp.256-270, 1998, ISSN 1047-3203

[Lan98b] G.C. Langelaar, R.L. Lagendijk, J. Biemond, "Watermark Removal based on Non-linear Filtering", ASCI'98 Conference, Lommel, Belgium, 9-11 June 1998

[Lan98c] G.C. Langelaar, R.L. Lagendijk, J. Biemond, "Removing Spatial Spread Spectrum Watermarks by Non-linear Filtering", IX European Signal Processing Conference, Island of Rhodes, Greece, 8-11 September 1998

[Lan99a] G.C. Langelaar, "Conditional Access to Television Service", Wireless Communication, the interactive multimedia CD-ROM, 3rd edition 1999, Baltzer Science Publishers, Amsterdam, ISSN 1383 4231

[Lan99b] G.C. Langelaar, R.L. Lagendijk, J. Biemond, "Watermarking by DCT Coefficient Removal: A Statistical Approach to Optimal Parameter Settings", Proceedings of SPIE Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose (CA), USA, January 1999

[Lan99c] G.C. Langelaar, R.L. Lagendijk, "Optimal Differential Energy Watermarking (*DEW*) of DCT Encoded Images and Video", submitted to the IEEE Transactions on Image Processing, 1999

[Lea96] T. Leary, "Cryptology in the 15th and 16th century", Cryptologica v XX, no 3, pp. 223-242, July 1996

[Lin98] J.-P. M.G. Linnartz, M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images", Workshop on Information Hiding, Portland, Oregon, USA, April 1998

[Mac95] B.M. Macq, J.-J. Quisquater, "Cryptology for Digital TV Broadcasting", Proceedings of the IEEE Vol. 83 no. 6, pp. 944-957, June 1995

[Mae98] Maurice Maes, "Twin Peaks: The histogram attack to fixed depth image watermarks", Workshop on Information Hiding, Portland, Oregon, USA, April 1998

[Mul93] F. Muller, "Distribution Shape of Two-Dimensional DCT Coefficients of natural Images", *Electronic Letters*, Vol. 29, no. 22, pp. 1935-1936, 1993

[Ng99] K.S. Ng, L.M. Cheng, "Selective block assignment approach for robust digital image watermarking", Proceedings of the SPIE/IS&T International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, San Jose, CA, January 25 - 27, 1999

[Nik96] N. Nikolaidis and I.Pitas, "Copyright protection of images using robust digital signatures", Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-96), vol. 4, pp. 2168-2171, Atlanta, USA, May 1996

[Pen93] W.B. Pennebakker, J.L. Mitchell, "The JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, New York, 1993

[Per99] Shelby Pereira, Joe J. K. Ó Ruanaidh, Frédéric Deguillaume, Gabriella Csurka and Thierry Pun, Template based recovery of Fourier-based watermarks using log-polar and log-log maps, In IEEE Multimedia Systems 99, International Conference on Multimedia Computing and Systems, Florence, Italy, 7-11 June 1999

[Pet98a] Fabien A.P. Petitcolas and Ross J. Anderson, "Weaknesses of copyright marking systems", Multimedia and Security Workshop at ACM Multimedia '98. Bristol, UK, September 1998

[Pet98b] Fabien A.P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, "Attacks on copyright marking systems", in David Aucsmith (Ed.), Proceedings LNCS 1525, Springer-Verlag, ISBN 3-540-65386-4, pp. 219-239, Information Hiding, Second International Workshop, IH'98, Portland Oregon, USA, April 15-17, 1998

[Pet99] Fabien A.P. Petitcolas and Ross J. Anderson, "Evaluation of copyright marking systems", Proceedings of IEEE Multimedia Systems (ICMCS '99), Florence, Italy, 7-11 June 1999

[Pit95] I. Pitas, T.H. Kaskalis, "Applying signatures on digital images", Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing, pp. 460-463, Neos Marmaras, Greece, 20-22 June 1995

[Pit96a] I. Pitas, "A method for Signature Casting on Digital Images", Proceedings ICIP-96, IEEE International Conference on Image Processing, Volume III pp. 215-218, Lausanne, Switzerland, 16-19 September 1996

[Piv97] A. Piva, M. Barni, F. Bartolini, V. Cappellini, "DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image", Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Pod97] C. Podilchuk, W. Zeng, "Perceptual Watermarking of Still Images", Proceedings of 1997 IEEE First Workshop on Multimedia Signal Processing, pp. 363-368, Princeton, New Jersey, USA, 23-25 June 1997

[Por93] Giovanni Baptista Porta, "De occultis literarum notis", Facsimil of edition from 1593, Cryptography chair Univerisity of Zaragoza, Spain 1996

[Pua96] J. Puate, F. Jordan, "Using fractal compression scheme to embed a digital signature into an image", Proceedings of SPIE Photonics East Symposium, Boston, USA, 18-22 November 1996

[Ren96] J.-L. Renaud, "PC industry could delay DVD", Advanced Television Markets, Issue 47, May 1996

[Rhe89] M.Y. Rhee, "Error Correcting Coding Theory", McGraw-Hill Publishing Company, New York, 1989

[Ron99] P.M.J. Rongen, M.J.J.J.B. Maes, C.W.A.M. van Overveld, "Digital Image Watermarking by Salient Point Modification", Proceedings of SPIE Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose (CA), USA, January 1999

[Rua96a] J.J.K. Ó Ruanaidh, F.M. Boland, O. Sinnen, "Watermarking Digital Images for Copyright Protection", Electronic Imaging and the Visual Arts 1996, Florence, Italy, February 1996

[Rua96b] J.J.K. Ó Ruanaidh, W.J. Dowling, F.M. Boland, "Watermarking Digital Images for Copyright Protection", IEE Proceedings Vision, Image- and Signal Processing, 143(4) pp. 250-256, August 1996

[Rua96c] J.J.K. Ó Ruanaidh, W.J. Dowling, F.M. Boland, "Phase Watermarking of Digital Images", Proceedings of the IEEE International Conference on Image Processing, Volume III pp. 239-242, Lausanne, Switzerland, September 16-19, 1996

[Rua97] Joseph J. K. Ó Ruanaidh, Thierry Pun, "Rotation, scale and translation invariant digital image watermarking" Proceedings of ICIP 97, IEEE International Conference on Image Processing, pp. 536-539, Santa Barbara, CA, October 1997

[Rua98a] Joe J. K. Ó Ruanaidh, Shelby Pereira, "A secure robust digital image watermark" Electronic Imaging: Processing, Printing and Publishing in Colour, SPIE Proceedings, (SPIE/IST/Europto Symposium on Advanced Imaging and Network Technologies), Zürich, Switzerland, May 1998

[Rua98b] Joe J. K. Ó Ruanaidh, Thierry Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking", Signal Processing, Vol. 66, no. 3, pp. 303-317, (Special Issue on Copyright Protection and Control, B. Macq and I. Pitas, eds.), May 1998

[Rup96] S. Rupley, "What's holding up DVD?", PC Magazine, vol. 15, no. 20, pp. 34, November 19, 1996

[Sam91] P. Samuelson, "Legally Speaking: Digital Media and the Law", Communications of ACM, vol. 34, no. 10, pp. 23-28, October 1991

[Sch94] R.G. van Schyndel, A.Z. Tirkel, C.F. Osborne, "A Digital Watermark", Proceedings of the IEEE International Conference on Image Processing, volume 2, pages 86-90, Austin, Texas, USA, November 1994

[Sha49] C.E. Shannon, W.W. Weaver, "The Mathematical Theory of Communications", The University of Illinois Press, Urbana, Illinois, 1949

[Sha93] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", IEEE Transactions on Signal Processing, 41(12), pp. 3445-3462, December 1993

[Smi96] J.R. Smith, B.O. Comiskey, "Modulation and Information Hiding in Images", Preproceedings of Information Hiding, an Isaac Newton Institute Workshop, University of Cambridge, UK, May 1996

[Swa96a] M.D. Swanson, B. Zhu, A.H. Tewfik, "Transparant Robust Image Watermarking", Proceedings of the IEEE International Conference on Image Processing, Volume III pp. 211-214, Lausanne, Switzerland, 16-19 September 1996

[Swa96b] Mitchell D. Swanson, Bin Zhu, Ahmed H. Tewfik, "Robust Data Hiding for Images", 7th IEEE Digital Signal Processing Workshop, pp. 37-40, Loen, Norway, September 1996

[Swa98] Mitchell D. Swanson, Mei Kobayashi, Ahmed H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", Proceedings of the IEEE, 86(6):1064-1087, June 1998

[Tan90] K. Tanaka, Y. Nakamura, K. Matsui, "Embedding Secret Information into a Dithered Multi-level Image", Proceedings of the 1990 IEEE Military Communications Conference, pp. 216-220, September 1990

[Tao97] Bo Tao and Bradley Dickinson, "Adaptive Watermarking in the DCT Domain", IEEE Internatational Conference on Acoustics, Speech and Signal Processing, April 1997

[Tay97] J. Taylor, "DVD Demistified : the Guidebook for DVD-Video and DVD-Rom", McGraw Hill Text, 1997

[Var89] M.K. Varansai and B. Aazhang, "Parametric generalized Gaussian density estimation", J.Acoust.Soc.Amer., vol.86, no. 4, pp. 1404-1415, October 1989

[Vet95] Martin Vetterli, Jelena Kovacevic, "Wavelets and subband coding", Prentice Hall Signal Processing Series, New Jersey, ISBN 0-13-097080-8, 1995

[Voy96] G. Voyatzis and I. Pitas, "Applications of Toral Automorphisms in Image Watermarking", Proceedings ICIP-96, IEEE International Conference on Image Processing, Volume II, pp. 237-240, Lausanne, Switzerland, 16-19 September 1996

[Voy98] G. Voyatzis, N. Nikolaides, I. Pitas, "Digital Watermarking: An Overview", Proceedings of IX European Signal Processing Conference (EUSIPCO), pp. 13-16, Island of Rhodes, Greece, 8-11 September 1998

[Wan95] Brian A. Wandell, "Foundations of Vision", Sinauer Associates, Inc., Sunderland, Massachusetts, USA, 1995

[Wol96] R.B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", Proceedings of the IEEE International Conference on Image Processing, Volume III pp. 219-222, September 16-19, 1996, Lausanne, Switzerland

[Wol97] R.B. Wolfgang and E.J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies," Proceedings of the International Conference on Imaging Science, Systems, and Technology, Las Vegas, USA, June 30 - July 3, 1997

[Wol98] R. B. Wolfgang and E. J. Delp, "Overview of Image Security Techniques with applications in Multimedia Systems," Proceedings of the SPIE Conference on Multimedia Networks: Security, Displays, Terminals, and Gateways, Vol. 3228, pp. 297-308, Dallas, Texas, USA, November 2-5, 1997

[Wol99a] Raymond B. Wolfgang, Edward J. Delp, "Fragile Watermarking Using the VW2D Watermark" Electronic Imaging '99, The International Society for Optical Engineering, Security and Watermarking of Multimedia Contents, Vol 3657, San Jose, CA, USA, 25-27 January 1999

[Wol99b] R. B. Wolfgang, C. I. Podilchuk, E. J. Delp, "Perceptual Watermarks for Digital Images and Video", Proceedings of the SPIE/IS&T International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, San Jose, CA, 25-27 January 1999

[Wol99c] R. B. Wolfgang, C. I. Podilchuk, E. J. Delp, "Perceptual Watermarks for Digital Images and Video", Proceedings of IEEE, May 1999

[Wu97] T.L. Wu, S.F. Wu, "Selective encryption and watermarking of MPEG video", International Conference on Image Science, Systems, and Technology, CISST'97, June 1997

[Xia97] X.-G. Xia, C.G. Boncelet, G.R. Arce, "A Multiresolution Watermark for Digital Images", Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Zen97] W.Zeng, B. Liu, "On resolving Rightful Ownerships of Digital Images by Invisible Watermarks", Proceedings of ICIP 97, IEEE International Conference on Image Processing, Santa Barbara, California, October 1997

[Zha95] J. Zhao, E. Koch, "Embedding Robust Labels into Images for Copyright Protection", Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, Austria, August 21-25 1995

List of Symbols

α	headroom factor
b	label bit
γ	shape parameter of a Gaussian distribution
c	cut-off index
c_{min}	minimum cut-off index
C_{ch}	channel capacity
D	energy difference in the watermark embedding phase
D'	energy difference in the watermark extraction phase
Dr	drift signal
$E_{A,B}$	energy in lc-subregion A or B
F_{edge}	edge-enhance FIR filter
F_H	high frequency bands
F_{HP}	high pass filter
F_M	middle band frequencies
G	gain factor in watermark removal scheme
I	image or video frame
$\hat{I}$	estimate of the image or video frame content
$I(x,y)$	pixels values of I where the spatial location is denoted by the indices (x,y)
$I(u,v)$	frequency transform domain coefficients of I with indices (u,v)
$I_{x,y}(u,v)$	frequency transform domain coefficients with indices (u,v) of block of I where the spatial location is denoted by the indices (x,y)
I_W	watermarked image or video frame
I'_W	possibly watermarked image or video frame
j	label bit index number
k	watermark gain factor
l	label length, number of watermark bits
L	embedded label bit string
L'	extracted label bit string
Msk	masking image
M,N	dimensions of an image or video frame
n	number of blocks/trees per watermark bit
P	number of image pixels per watermark bit
P_{be}	label bit error probability
P_e	label error probability
$P_e^{ECC(R)}$	label error probability after applying an error correcting code that can correct R bits
P_{fp}	false positive probability
P_{fn}	false negative probability

Q	coarseness quantizer
Q_{jpeg}	JPEG quality factor for image compression and watermark embedding
Q'_{jpeg}	JPEG quality factor in the watermark extraction phase
Q_{ITU}	ITU-R Rec. 500 quality rating
RP	pseudorandom pattern
R_{XY}	correlation between X and Y
S	subset of DCT or DWT coefficients
θ	DCT coefficient
$\hat{\theta}$	quantized DCT coefficient
T,T_m	thresholds
w	element of the standard JPEG luminance quantization table
Y	luminance component of an image or video frame
U,V	chrominance components of an image or video frame
W_b	bandwidth
W	watermark pattern
$\hat{W}$	estimated watermark pattern
Z	number of pixels per image or frame

Real-time watermerktechnieken voor gecomprimeerde videodata

Samenvatting

In de afgelopen jaren is het gebruik en de verspreiding van digitale multimedia data explosief toegenomen. PC's met internetaansluitingen hebben de huiskamer veroverd en de verspreiding van multimediadata en -toepassingen sneller en makkelijker gemaakt. Verder wordt analoge audio- en videoapparatuur in de huiskamer langzamerhand vervangen door hun digitale opvolgers.

Hoewel digitale data veel voordelen boven analoge data biedt, zijn de aanbieders van media erg terughoudend in het leveren van digitale media vanwege hun angst voor ongelimiteerde illegale vermenigvuldiging en verspreiding van autheursrechterlijk beschermd materiaal. Door het ontbreken van geschikte beveiligingssystemen is bijvoorbeeld de introductie van de DVD-speler vertraagd. Verscheidene mediabedrijven weigerden DVD-materiaal aan te leveren totdat het kopieerbeveiligingsprobleem opgelost was.

Om in kopieer- en auteursrechtbeveiligingen te voorzien voor digitale audio- en videodata worden twee elkaar aanvullende technieken ontwikkeld, nl. vercijfer- en watemerktechnieken. Vercijfertechnieken kunnen gebruikt worden om digitale data te beschermen gedurende de transmissie van de zender naar de ontvanger. Echter, na ontvangst en ontcijfering is de data niet langer meer beschermd. Watermerktechnieken kunnen hier de vercijfertechnieken aanvullen door een niet waarneembaar geheim signaal (het watermerk) toe te voegen aan de niet vercijferde data. Dit watermerksignaal wordt zodanig toegevoegd dat het niet te verwijderen is zonder ook de audio- of videodata aan te tasten. Het watermerksignaal kan onder andere gebruikt worden om de auteursrechten te beschermen door informatie over de auteur te verstoppen in de data. Het watermerk kan nu gebruikt worden om het eigendomsrecht te bewijzen in een rechtzaak. Een andere interessante toepassing waarvoor het watermerksignaal gebruikt kan worden is het opsporen van de bron van illegale kopiën d.m.v. *fingerprint* technieken. In dit geval voegt de media-aanbieder aan elke kopie van zijn data een watermerk met een serienummer toe dat gerelateerd is aan de identiteit van de koper. Als er nu illegale kopiën aangetroffen worden, bijvoorbeeld op het internet, kan de media-aanbieder gemakkelijk nagaan welke koper inbreuk heeft gemaakt op de koopovereenkomst. Het watermerksignaal kan ook gebruikt worden om digitale opnameapparatuur te sturen door aan te geven of bepaalde data wel of niet opgenomen mag worden. De opnameapparatuur moet dan wel voorzien

zijn van een watermerkdetector. Als andere toepassingen van het watermerksignaal kunnen verder genoemd worden: geautomatiseerde monitoringsystemen voor radio- en TV-uizendingen, data integriteitstesten en het verzenden van geheime boodschappen.

Aan elke watermerktoepassing kunnen andere eisen gesteld worden. De belangrijkste eisen echter die aan de meeste watermerktechnieken gesteld worden zijn dat het watermerk niet waarneembaar is in de data waarin het verborgen wordt, dat het watermerksignaal een redelijke hoeveelheid informatie kan bevatten en dat het watermerksignaal moeilijk of niet te verwijderen is zonder de kwaliteit van de data waarin het verborgen wordt aan te tasten.

In dit proefschrift wordt een uitgebreid overzicht gegeven van verschillende bestaande watermerkmethoden. Maar de nadruk ligt op de specifieke klasse van watermerkmethoden, die geschikt is voor het in real-time aanbrengen van watermerkinformatie in en uitlezen van watermerkinformatie uit gecomprimeerde videodata. Deze klasse van methoden is bijvoorbeeld geschikt voor *fingerprint*- en kopieerbeveiligingssystemen in consumentenopname-apparatuur.

Om als een real-time watermerktechniek voor gecomprimeerde videodata geklassificeerd te worden, moet een watermerktechniek naast de reeds genoemde eisen ook nog voldoen aan de volgende eisen. Er zijn twee redenen waarom de technieken om een watermerk aan te brengen en uit te lezen niet te gecompliceerd mogen zijn: ze moeten in real-time uitgevoerd kunnen worden en ze mogen niet te duur zijn voor gebruik in de consumentenelectronica. Dit betekent dat decompressie van de gecomprimeerde data, het watermerk toevoegen en vervolgens de gewatermerkte data weer comprimeren geen optie is. Het watermerk moet direct aan de gecomprimeerde data toegevoegd kunnen worden. Verder is het belangrijk dat de toevoeging van het watermerk geen gevolgen heeft voor de omvang van de gecomprimeerde data. Als de omvang van MPEG gecomprimeerde data bijvoorbeeld toeneemt, kan de verzending over een kanaal met een vaste bit-rate problemen veroorzaken, kunnen hardware buffers vol raken, of kan de synchronisatie tussen audio en video verstoord worden.

De meest efficiente manier om de complexiteit van real-time watermerkalgoritmen laag te houden is het vermijden van rekenintensieve operaties door handig gebruik te maken van het compressieformaat van de videodata. In dit proefschrift worden twee nieuwe watermerkconcepten voorgesteld die direct toepasbaar zijn op gecomprimeerde videodata, nl. het *Least Significant Bit* (LSB) modificatieconcept en het *Differential Energy Watermark* (DEW) concept. Bij het toevoegen van een watermerk volgens het LSB-concept worden alleen vaste-lengte en variabele-lengte codewoorden in de gecomprimeerde datastroom vervangen door andere codewoorden. Voordelen van dit concept zijn de zeer lage complexiteit en de grote hoeveelheid informatie die het watermerksignaal kan bevatten. Een nadeel van dit concept is dat de procedures om een watermerk toe te voegen en uit te lezen volledig afhankelijk zijn van het voor de videodata gebruikte compressieformaat. Zodra de videodata gedecomprimeerd is, is het watermerk verloren. Dit is geen groot bezwaar voor videostromen voor consumenten toepassingen die geen bescherming van het allerhoogste niveau vereisen, omdat het volledig decomprimeren en opnieuw comprimeren van een videostroom een zeer rekenintensief proces is.

Voor real-time toepassingen die een hoger beschermings niveau vereisen is het DEW-concept ontwikkeld. Het DEW-algorithme voegt een watermerk toe door energie verschillen op te drukken tussen videoregio's. Dit gebeurt door selectief hoge frequentiecomponenten weg te gooien. De DEW-methode toegepast op gecomprimeerde videodata heeft daarom alleen gedeeltelijke decompressie stappen nodig om een watermerk toe te voegen of uit te lezen, gedeeltelijk opnieuw data comprimeren is niet nodig. De complexiteit van het DEW-algoritme is daarom slechts iets groter dan de LSB-gebaseerde methoden. Omdat het DEW-watermerk niet afhankelijk is van het voor de videodata gebruikte compressieformaat, blijft het watermerk aanwezig na het decomprimeren van de videodata.

Het laatste gedeelte van dit proefschrift is gewijd aan de evaluatie van het DEW-concept. Verschillende aanpakken uit de literatuur om watermerkmethoden te evalueren worden besproken en toegepast. Verder worden aanvallen om watermerken te verwijderen uit de literatuur besproken en een nieuwe watermerkaanval voorgesteld.

Acknowledgements

Curriculum Vitae