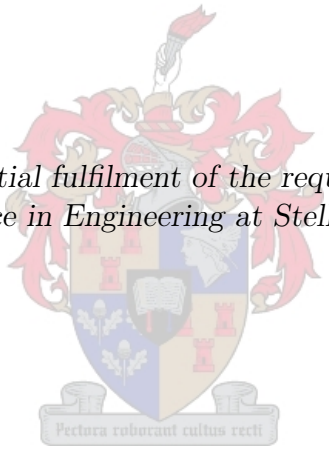
Comparative Evaluation of Video Watermarking Techniques in the Uncompressed Domain

by

Rudolph Hendrik van Huyssteen



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering at Stellenbosch University

Supervisors:

Prof. G-J. van Rooyen
Department of Electrical and Electronic Engineering
University of Stellenbosch

Dr D. Jarnikov
Department of Mathematics and Computer Science
Eindhoven University of Technology

December 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2012

Abstract

Electronic watermarking is a method whereby information can be imperceptibly embedded into electronic media, while ideally being robust against common signal manipulations and intentional attacks to remove the embedded watermark. This study evaluates the characteristics of uncompressed video watermarking techniques in terms of visual characteristics, computational complexity and robustness against attacks and signal manipulations.

The foundations of video watermarking are reviewed, followed by a survey of existing video watermarking techniques. Representative techniques from different watermarking categories are identified, implemented and evaluated.

Existing image quality metrics are reviewed and extended to improve their performance when comparing these video watermarking techniques. A new metric for the evaluation of inter frame flicker in video sequences is then developed.

A technique for possibly improving the robustness of the implemented discrete Fourier transform technique against rotation is then proposed. It is also shown that it is possible to reduce the computational complexity of watermarking techniques without affecting the quality of the original content, through a modified watermark embedding method.

Possible future studies are then recommended with regards to further improving watermarking techniques against rotation.

Uittreksel

'n Elektroniese watermerk is 'n metode waardeur inligting onmerkbaar in elektroniese media vasgelê kan word, met die doel dat dit bestand is teen algemene manipulasies en doelbewuste pogings om die watermerk te verwyder. In hierdie navorsing word die eienskappe van onsaamgeperste video watermerktegnieke ondersoek in terme van visuele eienskappe, berekeningskompleksiteit en weerstandigheid teen aanslae en seinmanipulasies.

Die onderbou van video watermerktegnieke word bestudeer, gevolg deur 'n oorsig van reedsbestaande watermerktegnieke. Verteenwoordigende tegnieke vanuit verskillende watermerkkategorieë word geïdentifiseer, geïmplementeer en geëvalueer.

Bestaande metodes vir die evaluering van beeldkwaliteite word bestudeer en uitgebrei om die werkverrigting van die tegnieke te verbeter, spesifiek vir die vergelyking van watermerktegnieke. 'n Nuwe stelsel vir die evaluering van tussenraampie flikkering in video's word ook ontwikkel.

'n Tegniek vir die moontlike verbetering van die geïmplementeerde diskrete Fourier transform tegniek word voorgestel om die tegniek se bestandheid teen rotasie te verbeter. Daar word ook aangetoon dat dit moontlik is om die berekeningskompleksiteit van watermerktegnieke te verminder, sonder om die kwaliteit van die oorspronklike inhoud te beïnvloed, deur die gebruik van 'n verbeterde watermerkvasleggingsmetode.

Laastens word aanbevelings vir verdere navorsing aangaande die verbetering van watermerktegnieke teen rotasie gemaak.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations:

- My supervisors, Prof. Gert-Jan van Rooyen and Dr Dmitri Jarnikov for their guidance and motivation;
- my family and friends for their continued support and understanding;
- my colleagues in the MIH Media Lab for making this project extra enjoyable;
- the MIH Media Lab for funding this research project; and
- Irdeto, for supporting my research through collaboration.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms and Abbreviations

A/D	Analog to Digital
D/A	Digital to Analog
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FDFTT	Fast Discrete Fourier Transform Technique
HVS	Human Visual System
JAWS	Just Another Watermarking System
JND	Just Noticeable Difference
Mbit/s	Megabit Per Second
MSE	Mean Squared Error
MTF	Modulation Transfer Function
NQM	Noise Quality Measure
PSNR	Peak Signal to Noise Ratio
QIM	Quantisation Index Modulation
RGB	Red-Green-Blue
SD	Spatial Domain
SSIM	Structural SIMilarity
SVD	Singular Value Decomposition
VQM	Video Quality Metric
WMS	Watermark Minimum Segment

Notation

Symbol	Definition	Dimension
$\mathbf{A}$	Unwatermarked video frame	$X \times Y$
$\widetilde{\mathbf{A}}$	Watermarked video frame	$X \times Y$
$\mathbf{b}$	Binary message	$M \times 1$
$\mathbf{b}'$	Binary message processed for embedding	
$\mathbf{b}''$	Pair-wise version of $\mathbf{b}'$ used in the DFT technique	
$\mathbf{b}'''$	Difference between pair-wise elements in $\mathbf{b}'$, used in the DFT technique	
$\hat{\mathbf{b}}$	Extracted binary message	
b_i	i^{th} bit of message $\mathbf{b}$	$[-1, +1]$
$\mathbf{B}$	Watermark obtained by attacker	$X \times Y$
$d(x, y, z)$	Frequency spectrum of video block D	
$d_{\max}, d_{\min}$	Parameters to specify upper and lower temporal frequency limits for watermark embedding by the DFT technique	
$D(x, y, z)$	Unwatermarked, uncompressed video block (luminance frames)	$[0, 1]$
E	Energy measure of DWT coefficients	
$\|\|E\|\|$	L^1-norm of DWT subbands	
$\mathbf{G}$	Family of Gold codes	
$I(x, y, z)$	Function to map cartesian coordinates to a serial index of pixels in a video block	
K	Decomposition depth of DWT	
L	Maximum value that a pixel can assume	
M	Number of bits in $\mathbf{b}$	
n	Index number of pixel in video block	
$\mathbf{p}$	Pseudo-noise sequence	$\eta M \times 1$
r	Rank of matrix	
$r_{\max}, r_{\min}$	Parameters to specify upper and lower spatial frequency limits for watermark embedding by the DFT technique	
$\mathbf{S}$	Pseudo-diagonal matrix containing singular values of $\mathbf{A}$	$X \times Y$
$\mathbf{U}$	Left singular vectors of $\mathbf{A}$	$X \times X$
$\mathbf{V}$	Right singular vectors of $\mathbf{A}$	$Y \times Y$
$W(x, y, z)$	Watermarked, uncompressed video block (luminance frames)	$[0, 1]$
x, y	Pixel coordinates in a video frame	
X, Y	Frame width and height, respectively	
z	Video frame index	

Symbol	Definition	Dimension
Z	Number of frames in video block	
α	Watermark embedding strength	
η	Spread-spectrum spreading factor	
λ	Scaling factor	
σ	Singular values contained in $\mathbf{S}$	
$\sigma_{\min}, \sigma_{\max}$	Singular values with indexes of min and max to be used for watermark embedding by the SVD technique	
$\mathbf{1}^{\eta}$	Vector of ones	$\eta \times 1$

Chapter 1

Introduction

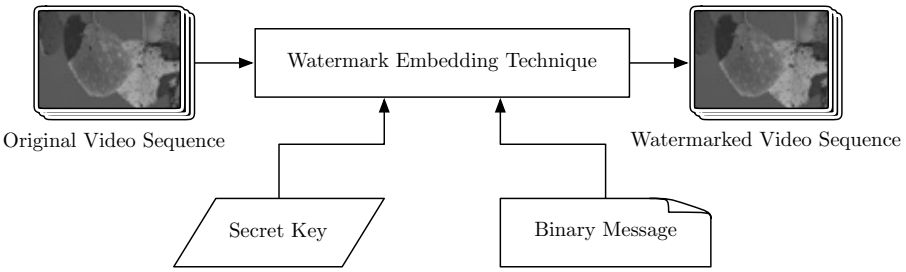
1.1 Prelude to Electronic Watermarking

Electronic watermarking is a method of robustly embedding information into media, which remains intact even after digital to analog and other signal conversions. This is achieved by embedding information imperceptibly into the media content itself, rather than relying on a file header or other techniques to convey information [1].

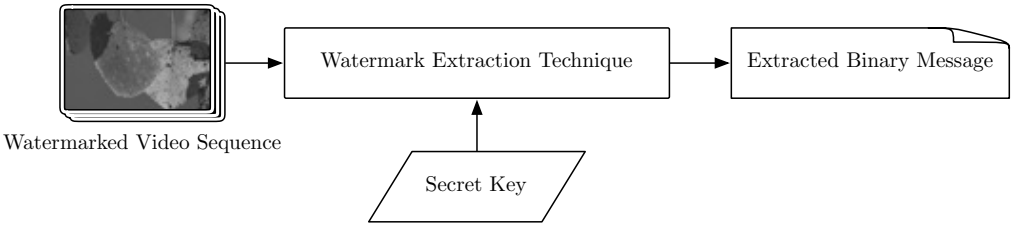
Simplified watermark embedding and extraction processes are shown in Figures 1.1a and 1.1b respectively. A watermark embedding technique is supplied with an unwatermarked video sequence, a secret key and a binary message to embed in order to obtain a watermarked video sequence. This message can then be extracted using the appropriate watermark extraction technique and secret key, as shown in Figure 1.1b.

A well-designed watermark is difficult to remove without degrading the media content itself. If implemented correctly, watermarks can provide copyright protection of content where classic copyright management approaches fail [2]. In theory, even if a user captures video content from a computer screen with a video camera, the watermark will stay intact. Depending on the content of the watermark, this can then be used to trace the user who illegally distributed the content, or to prevent playback on certain devices.

It is important to note that watermarking is usually not directly used for copyright protection, but rather to identify perpetrators and serve as a "last line of defense" in the fight against piracy [3]. If combined with encryption techniques to protect the content in the digital domain, watermarking can help to effectively combat piracy in modern media distribution systems.



(a) Visual representation of video watermark embedding process.



(b) Visual representation of watermark extraction from a watermarked video sequence.

Figure 1.1: Visual representation of a simplified video watermark embedding and extraction processes.

1.2 Motivation

Recent advances in technology have made the distribution of media content easier than ever. Peer-to-peer media applications such as BitTorrent [4] now take up a large share of all traffic on the Internet and can be used to illegally distribute pirated media content [5]. In addition to this, digital video equipment now allow content to be duplicated in quality higher than ever.

In the earlier years of analog media, a loss of quality was introduced with each generation of copied media. This made it desirable to obtain a first generation copy from a vendor in order to experience the best fidelity possible. Digital media can, however, be copied indefinitely without loss of quality and reduces the motivation to obtain a copy from the original media vendor.

In fear of large scale piracy, many content owners are unwilling to make their content available to new Internet-based content distribution systems [2, 6, 7]. To effectively combat piracy, content owners need to find a reliable way to control illegal distribution of their media, while maintaining easy accessibility for legal users [8].

The challenge to effective copyright control lies in the fact that all media content

needs to be eventually decrypted and presented to the user in analog form. While cryptography techniques can help ensure the secure delivery of video content, this protection is lost once the content is decrypted for viewing [7, 9]. It follows that additional methods such as watermarking are required to retain copyright information in digital media after digital-to-analog conversion and other signal operations are applied. Since watermarks are embedded into the media content itself, the embedded information (such as copyright information) ideally remains intact even after the decryption process. Thus, video watermarking in combination with cryptography techniques can play a vital role in the battle against piracy.

1.3 Problem Statement

For a novice wishing to enter into the field of watermarking, a survey of articles found in literature may not be sufficient to gain an in-depth understanding and working knowledge of video watermarking techniques. With multiple watermarking techniques available, differentiating between the various techniques and choosing the most appropriate solution for a specific application can be challenging [10].

While techniques published in the literature usually include test results, few use the same robustness criteria or source material, which makes comparison difficult [11]. Commercial systems may provide benchmark results, but these watermarking systems are usually proprietary and the reader is unable to gain insight into the functioning of the technique. For these reasons, most benchmarking results and survey articles are largely unsuitable for a direct comparison of the physical-layer signal processing techniques used to hide information in watermarking applications.

1.4 Research Objectives

It is possible to group different watermarking techniques into specific categories, based on the mathematical approaches used by the techniques. The aim of this study is to provide an insight into the characteristics of basic video watermarking techniques by comparing the performance of representative techniques from each category against attacks, using the same source content and visual perceptibility criteria. Such a comparison is intended to make it easier to select an appropriate technique for a specific application, and to provide quantitative results for future video watermarking research.

It follows that the objectives for this study are to:

- conduct a taxonomy of existing video watermarking technologies;

- identify representative techniques from different categories of video watermarking techniques;
- evaluate the visual characteristics of each technique;
- evaluate the computational complexity and robustness of each technique; and
- draw conclusions from the results and make recommendations for future research.

1.5 Thesis Statement

This study is based on the following hypothesis:

By implementing and evaluating selected video watermarking techniques, using the same source content and visual quality constraints, the distinct characteristics of each watermarking approach can be identified. Appropriate conclusions can then be drawn and recommendations for future research can be made.

1.6 Contributions

The following contributions are made in this study:

- The existing Structural Similarity Index quality measure is extended for improved performance when comparing watermarking techniques where artefacts caused by techniques differ in terms of visual properties and localisation;
- a new metric is proposed to evaluate interframe flicker in video sequences caused by video watermarking algorithms; and
- recommendations are made to reduce the computational complexity of a discrete Fourier transform watermarking technique while improving the robustness of watermark extraction against rotation of the video frames.

1.7 Thesis Overview

Chapter 2 discusses the foundations of video watermarking research and necessary background which is required by the rest of this study. A survey of existing video watermarking techniques is conducted and the main categories of video watermarking techniques identified.

Chapter 3 describes selected watermarking techniques in terms of appropriate mathematical characteristics, after which the implementation of each technique is discussed.

Chapter 4 provides a visual evaluation of the artefacts caused by the selected video watermarking techniques. Existing image quality metrics are reviewed and the structural similarity index is extended to improve its performance when comparing these video watermarking techniques. A new metric for the evaluation of interframe flicker in video sequences is presented. The developed techniques are then used to evaluate the implemented watermarking techniques and appropriate conclusions are drawn.

Chapter 5 evaluates the computational complexity and robustness of the implemented watermarking techniques. Conclusions are drawn with reference to Chapters 2, 3 and 4.

Chapter 6 recommends a technique for improving the robustness of the implemented discrete Fourier transform technique against rotation without the use of templates to detect and correct geometrical transforms.

Chapter 7 concludes the thesis and provides an overview of results obtained. Conclusions are drawn in terms of optimisation of watermark technique robustness, successful attacks that prevent correct watermark extraction as well as recommendations for future work.

Chapter 2

Background

The background required for this study is now briefly discussed. An overview of electronic watermarking is given, reviewing previous research, applications, requirements, attacks and other considerations that are applicable to electronic watermarking. Colour representation in digital systems is discussed, followed by an overview of properties of the human visual system. The chapter is concluded with an overview of popular video watermarking techniques found in the literature. References are provided throughout this chapter to enable further reading where additional detail is required.

2.1 Electronic Watermarking

Electronic watermarking is a method whereby information can be imperceptibly embedded into electronic media [1]. This embedded information should ideally be robust against common signal manipulations such as the addition of random noise, digital-to-analog conversion, lossy compression or intentional attacks to remove the embedded watermark [12]. The requirements for effective watermarking are further detailed in Section 2.1.4.

2.1.1 Overview of Electronic Watermarking Research

The very first version of the electronic watermark can be traced back to 1954 in the form of a patent filed by Emil Hembrooke of the Muzac Corporation, titled "Identification of sound and like signals" [13]. In this system, a code was embedded into a piece of music to later help determine the origin of the work. The idea was that if the origin of pirated material can be determined, it may discourage piracy. Although the implementation was fairly basic, the idea of using electronic watermarking to determine the origin of pirated material is still in use today.

With the rise of electronic media in the late 1980s and early 1990s, more research was done on watermarking and the number of papers published on the topic of watermarking increased [14, 15]. These early techniques tended to model the watermark as a communication channel, with the host content and any additional distortions treated as noise. As a result, the watermark is embedded without taking the characteristics of the host content into account. By the late 1990s, more advanced techniques such as [16–18] were developed. These techniques modelled watermarking as communication with side information available at the transmitter (watermarking embedding stage) and possibly the receiver (watermark extraction stage) [16]. Most modern watermarking techniques such as [19–26] are known as adaptive watermarking techniques and take properties of the host content and human auditory or vision systems into account to imperceptibly embed watermarks into the host content while maximising robustness. By 2009, the field of image watermarking was considered to be fairly mature, with video, audio and text watermarking research requiring further research [15]. Furthermore, although advances have been made in the field of electronic watermarking research, some researchers [2, 27] believed that watermarking technologies are in general less mature than those used in cryptography, requiring more refinement in techniques and applications.

2.1.2 The Specifics of Video Watermarking

Electronic watermarking can be applied to various typed of media, which include text, image, audio and video content. As the focus of this research project is video watermarking, further discussions will focus mainly on topics related to the watermarking of video content.

The video watermarking category can be divided into two sub-categories, namely

- compressed-domain video watermarking; and
- uncompressed-domain video watermarking.

Compressed-domain video watermarking techniques can embed watermarks into compressed video streams without the need to uncompress and recompress the video stream, which can speed up the watermarking process. Uncompressed-domain techniques, on the other hand, first need to uncompress the video in order to embed a watermark. After the watermark has been embedded, the content needs to be recompressed. While uncompressed-domain video watermarking may be more computationally complex than compressed-domain watermarking, it has the advantage that it can be applied to a large variety of video types. Compressed-domain video

watermarking techniques are usually tailored to a specific type of video compression, such as the techniques detailed in [28–32].

Another way to subdivide watermarking techniques is by blind versus non-blind detection. This refers to whether the watermarking scheme requires the original, unwatermarked data to detect the watermark. Non-blind watermarking techniques require the original unwatermarked content, while blind techniques can extract a watermark without any reference to the original content. Blind detection in general complicates detection and limits the data capacity of the watermarking scheme. Non-blind watermark extraction can be more robust but may require large databases of original content, rendering the technique unpractical for some applications [33, 34].

The rest of this overview mostly focuses on blind uncompressed-domain video watermarking techniques.

2.1.3 Video Watermarking Applications

Since Hembrooke's initial watermarking application in 1954, additional applications for electronic watermarking have been developed. While numerous applications exist, it is possible to group most of these techniques into six main application categories [1, 14, 35–39], which are now discussed in the following paragraphs.

Transaction Tracking

Transaction tracking watermarks are used to track how content was distributed through a system or transmitted between multiple points. A unique identifier is embedded into the media at the time of playback, which can later be extracted. In the case of illegal distribution of the content, it should ideally be possible to identify the source from where the distribution occurred, possibly identifying the misappropriating party [8, 14].

The watermark can be embedded by the playback device itself, but may impose limitations on the sophistication of the technique if the device is resource-constrained. Special attention also needs to be paid to the tamper-resistance of the watermarking algorithm if client-side watermarking is employed.

Alternatively the watermark can be embedded by a server at the time of distribution, but this can place additional load on the server. The authors of [40] suggest a solution to this challenge for audio watermarking by watermarking content server-side and then assembling the watermark on the device itself.

Further watermarking techniques for transaction tracking are detailed in [41–45].

Broadcast Monitoring

Broadcast monitoring enables broadcasters or content owners to track or verify the transmission of media in a broadcast system. The watermarks can automatically be extracted to verify if a commercial has successfully been aired or whether a certain segment of material was used in a broadcast. The content is usually watermarked by the content owner, while detection can be done by a monitoring site in the broadcast chain or a third party at the receiving end.

A real-time watermarking technique named JAWS (Just Another Watermarking System) was proposed by [46] as a professional broadcast surveillance system, with other techniques discussed in [47] and [48].

Copy Control

Copy control aims to disable the duplication of copyrighted material on devices equipped with special watermark detectors. The watermark is used to indicate copy control information, such as COPY_NEVER, COPY_ONCE or COPY_FREELY [9]. By implementing watermark extraction and embedding in devices, the user can be allowed or denied permission to duplicate content.

Several techniques are discussed in [7, 9, 49] which were designed primarily for the purpose of conveying copyright information for media such as digital video discs. In theory, even if a user captures content from the analogue output on a digital video disc player, re-encodes it and attempts to write this to disc with a compliant media burner, permission would be denied.

Content Authentication

Content authentication is a method that attempts to ensure the integrity of media by detecting attempted tampering of the original content. At creation, the content is usually watermarked with a semi-fragile watermark, which is designed to be affected by signal transformations. Tampering with the content should destroy or alter this semi-fragile watermark, which could then be used to determine that the content is not authentic [6]. Techniques that are focused on content authentication are further discussed in [50–52].

Ownership Identification

In this application, watermarks can be used to identify the rightful owner or creator of content [34]. After the original content was watermarked, disputed ownership can be resolved by extracting the original watermark. Resolving rightful ownership can,

however, be challenging as pirates may also embed their own watermark, in which case it can be difficult to determine which is the original watermark. This is known as an ownership deadlock problem and is discussed in detail by [53]. Ownership identification techniques are further discussed in [54] and [55].

Fingerprinting

This category is only included for clarity, as there exist at least two definitions of fingerprinting, each with specific characteristics and applications.

The first definition of media fingerprinting is given by [56] as *"the art, or algorithm, of identifying component characteristics of a source and then reducing it into a fingerprint that can uniquely identify it."* These techniques do not add any additional information to the media, but rather generates a compact signature based on the unique properties of the content [8]. These fingerprints can later be used to uniquely identify content, independent of the format and without the need for metadata [57]. This application falls outside the scope of this research.

The second definition of media fingerprinting is used synonymously with transaction tracking, which was discussed earlier in this section.

2.1.4 Requirements for Effective Watermarking

While the requirements for a watermarking technique can vary according to the intended application, most watermarks share a common set of requirements. Designing a watermark to satisfy all these requirements can be challenging, therefore it may be necessary to reach a compromise between them [6].

A well-designed watermarking technique is not only imperceptible to the observer, but should also provide a high data payload [12]. The watermarking technique also needs to be robust to enable the media to undergo signal conversions and small alterations without destroying the watermark [8]. These alterations can include those caused by normal compression and signal conversions, but also intentional attacks as discussed in Section 2.1.5.

Popular requirements mentioned in the literature [3, 6, 35, 58] are now discussed.

Imperceptibility

The artefacts produced by watermark embedding should not degrade the quality of the original content in such a way that it is perceptible to viewers [34, 39]. As discussed earlier, advanced techniques take the properties of the human visual system into account to achieve robustness while maintaining imperceptibility.

Robustness

It is desirable that a watermark must be robust against attacks to such an extent that the quality of watermarked content must be considerably degraded in order to remove a watermark. The watermark should not only be robust against intentional attacks, but also to standard video manipulations such as cropping, scaling and compression.

The most prominent intentional attacks applicable to video watermarking are discussed in Section 2.1.5.

Data Capacity

The data capacity or payload size of a watermark is an indication of the amount of information that can be embedded with a watermark. The payload size varies with the application, but in general an identifier packet of 64 bits is considered appropriate for most applications [59–61].

Error correction techniques such as Reed-Solomon coding [62] or Turbo coding [63] are often used to improve the robustness of techniques against attacks and other signal manipulations, but may reduce the data capacity of the watermarking technique. It is necessary to determine the number of bits that a watermarking technique can embed and how many useful information bits this results in if error correction techniques are applied.

Another important factor when discussing data capacities is the granularity of the watermark. This defines the minimum amount of data required to embed or extract one unit of watermark information. This is sometimes referred to as the Watermark Minimum Segment (WMS). In the case of video watermarking, the WMS refers to the minimum successive video frames required to embed or detect a watermark. The WMS should not be too long, as a watermark should be extractable from a short piece of video. Furthermore, if too many frames are lost from the WMS, the watermark may not be extracted successfully. By reducing the WMS, one reduces the chance of a watermark being destroyed due to temporal cropping, but the potential data capacity is also reduced [7]. A WMS value between 1 and 10 seconds is considered to be acceptable in most cases [59].

Resistance to Statistical Analysis

If an attacker obtains multiple watermarked video sequences each containing the same watermark, the attacker should not be able to detect or identify the watermark through statistical analysis. This is not only to prevent unauthorised detection, but also removal. If an attacker can detect if a watermark is present and whether it

can be extracted, the removal can be aided by experimenting with attacks until the watermark is no longer detected or suitable for extraction [58, 64]. Attacks through statistical analysis are further discussed in Section 2.1.5.

Security

The watermarking technique should adhere to Kerckhoffs's principle, which states that a crypto-system must be secure even if the attacker knows everything about the system, but does not have the correct key [3]. Therefore, even if an attacker has access to the exact algorithm used for watermarking and knows that a watermark is present, he or she must be unable to detect or decrypt the data in a reasonable amount of time [34]. The key used for watermarking should also be difficult to predict and cryptographically strong [39].

Cascadability

Different watermarks may already be applied to the content by the time the content tracking watermark is inserted. It is desirable for the watermarks to co-exist in media without affecting the performance of the watermarking extraction processes [59].

Low error probability

It is desirable to ensure that a watermark payload can be extracted with high confidence and minimum error. Two types of errors can occur, namely a detection error and payload recovery error [61]. A detection error refers to the case where a watermark is detected, but one is not present (false alarm). The second case is where a watermark exists, but is not detected (false negative). A payload recovery error refers to a case where a watermark is correctly detected, but the payload incorrectly extracted.

The probability of errors need to be small enough to ensure that, if a case of misappropriation is taken to court, the watermark cannot be argued as unreliable by lawyers. In general, a false alarm probability of 10^{-8} is required, while a probability of 10^{-12} is desired in most cases [7, 59].

2.1.5 Attacks on Video Watermarks

A successful attack on a watermarking technique refers to a case where the watermark has been removed or modified to prevent successful extraction, without degrading the quality of the watermarked content significantly [65]. A successful attack on a watermark does not necessarily mean that the content was restored to the original,

unwatermarked state, but instead that the watermark detection and extraction processes were defeated. This can be achieved in one of two ways. Firstly, the attacker can apply manipulations to the media that causes the watermark to not be detected. Alternatively, manipulations can be applied to cause the watermark detection process to be unreliable [58].

A summary of common attacks is shown in Figure 2.1. These can be grouped into two main categories, namely intentional and unintentional attacks. Intentional attacks are deliberate attempts to prevent successful extraction of embedded watermarks. Unintentional attacks, on the other hand, are caused by normal signal conversions and compression that may be introduced in a distribution chain. Normal signal conversion operations to which a video watermark should be robust include:

- analog to digital (A/D) and digital to analog (D/A) conversion;
- scaling and cropping operations;
- aspect-ratio conversion;
- frame rate conversion;
- quantisation;
- noise addition; and
- compression.

Intentional attacks that are common to video watermarking techniques are now discussed.

Geometrical Attacks

In this category of attacks, minor geometric distortions are applied to video frames in an attempt to de-synchronise the watermark extraction process with the embedded watermark. These attacks include simple transforms like rotation, scaling and spatial shifting. Geometrical attacks usually alter every frame of a video sequence, which are referred to as single-frame attacks. Artefacts induced by these attacks can be perceptually negligible, while succeeding in defeating the watermarking scheme [34]. Since blind watermark extraction techniques do not have access to the original content to detect and correct geometrical attacks, these can be more susceptible to geometrical attacks than non-blind extraction techniques [14, 39].

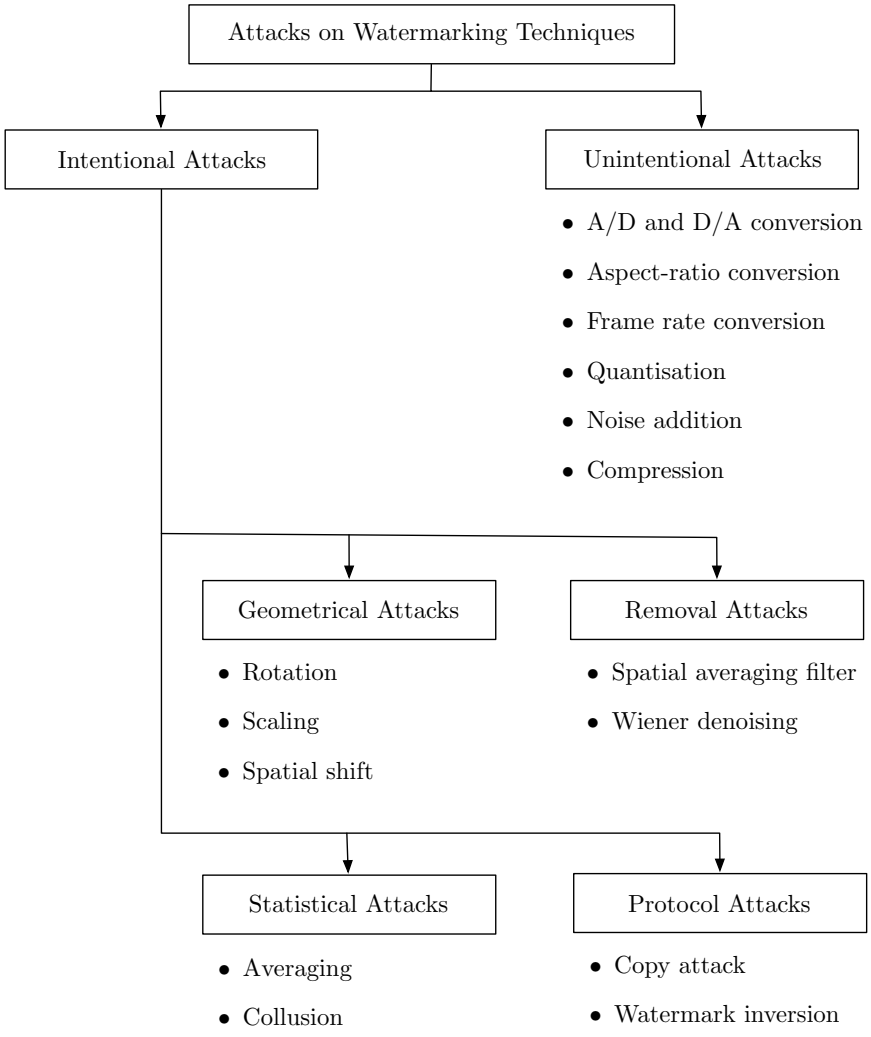


Figure 2.1: A summary of common attacks that may defeat watermarking techniques. These attacks can be grouped into two main categories, namely intentional and unintentional attacks. Intentional attacks are deliberate attempts to prevent successful extraction of embedded watermarks. Unintentional attacks are caused by normal signal conversions and compression that may be introduced in a distribution chain, but which can also be performed intentionally.

Removal Attacks

Removal attacks aim to remove the watermark embedded in the content, usually through filtering. The most basic approach is a spatial averaging filter, while a more advanced denoising technique is the Wiener denoising filter. If the watermark is seen as noise, this filter attempts to restore the watermark content to the unwatermarked state by removing noise, which in this case would remove the watermark. These attacks are generally more effective in video frames that do not contain a high amount of detail, as more aggressive filtering can be applied without affecting the perceived quality of the frame.

Statistical Attacks

Statistical attacks take advantage of the temporal redundancy in video sequences in order to remove the watermark. It is important to note that, although still image watermarking techniques can be applied to video sequences, video watermarking poses some unique challenges not necessarily applicable to image or audio watermarking. Because of the inherently redundant data between frames in a watermarked video sequence, watermarking techniques are susceptible to attacks such as frame averaging, frame swapping and statistical analysis [39].

Two types of statistical attacks, namely averaging and collusion attacks are now discussed.

Averaging attacks: This attack takes advantage of the fact that most consecutive video frames are very similar by simply performing a moving average over a small number of frames in a video. The video content can be seen as changing slowly between frames, whereas the watermark may vary significantly between each frame. It follows that the averaging attack has a higher effect on the watermark than it has on the video content, possibly preventing successful watermark extraction. As expected, this attack works best with static scenes, because blurring effects can be introduced in scenes with fast movement [12].

Collusion attacks: Averaging attacks are not effective in cases where the same watermark is embedded in all frames of a video sequence, as averaging will only blur the video content while the watermark remains intact. In this case, a collusion attack can be more effective [12].

To initiate a collusion attack, an attacker needs access to a number of differently watermarked versions of the same content. There are different approaches to the attack, but the most basic are linear collusion attacks and copy-and-paste collusion

attacks. The linear collusion attack simply involves averaging the different copies into a single copy to defeat the watermark. The copy-and-paste attack consecutively combines different sections from each copy, hoping to corrupt the watermarking scheme by splicing together different watermarks within a single WMS [33]. There are more advanced collusion attacks, such as those discussed in [12], but all rely on having access to more than one differently watermarked copy of the content.

Transaction tracking techniques are particularly susceptible to these attacks, as an attacker can usually easily obtain more than one copy of the same content. It has been shown that in some applications such as the transaction tracking in DivX6, less than 20 copies were required for a successful attack [14].

Protocol attacks

Protocol attacks do not focus on removing or destroying watermarks, but rather on attacking the application for which the watermark is used. An example of this is the copy attack [66], where a watermark is copied from one image to another without any information about the watermarking technology used. It follows that if a scheme is not resistant to the copy attack, watermarks used for verification cannot be trusted, as it may have been copied from another source.

Another example of a protocol attack is the watermark inversion attack, discussed by [53]. Let the original content owner watermark a video frame $\mathbf{A}$ with a watermark $\mathbf{B}$, obtaining the watermarked frame $\widetilde{\mathbf{A}}$. An attacker now attempts to find a watermark $\mathbf{C}$ that is present in both $\mathbf{A}$ and $\widetilde{\mathbf{A}}$. If he succeeds in finding $\mathbf{C}$, he can subtract this from $\widetilde{\mathbf{A}}$ and claim that the result is the original version of the frame. The attacker did not remove the original content owner's watermark, but created confusion over which is the original watermark that indicates rightful ownership.

2.1.6 Other Considerations

Additional aspects to be considered when choosing or designing a watermarking system are now briefly discussed.

Real-time Embedding and Detection

An important consideration when choosing a watermarking scheme is to determine whether real-time embedding and detection is required. Some applications may require a computationally efficient real-time embedding stage, but may be allowed to do offline watermark extraction, which would allow a more complex extraction.

Applications such as broadcast monitoring would, for instance, require real-time watermark detection, but watermark embedding may be performed offline.

Check for Known Data Versus Arbitrary Data Embedding

Depending on the application, the watermarking scheme can be used to embed one of three main types of watermark. Firstly, the watermark may have to convey arbitrary information, such as a unique identification number. Alternatively, it may only have to convey one of a few predefined watermarks such as COPY_NEVER, COPY_ONCE or COPY_FREELY. Lastly, the watermark extractor may simply check for the presence of a certain watermark and return true or false [3].

Host-Adaptive versus Non-Host-Adaptive

Watermarking techniques can be divided into two categories, namely host-adaptive (host dependent) and non-host-adaptive (host independent) techniques.

Non-host-adaptive techniques only take the properties of the human visual system into account. The content of the individual media frames (in this case referred to as the host) is not considered. While these techniques are often less complex to implement, the watermark is often not as robust and imperceptible as host-adaptive techniques.

Host-adaptive techniques, also known as perceptual techniques not only take the properties of the human visual system into account, but also the properties of the frames being watermarked. These techniques often provide higher robustness, while improving imperceptibility. Unless the application is resource constrained and calls for a simple watermarking scheme, it is desired for techniques to be host-adaptive [67].

An example would be if a pure white frame with no detail is watermarked. A host adaptive technique would recognise that the frame does not contain detail and lower the embedding strength. A non-host-adaptive technique will not lower the embedding strength, which may result in the watermark being more visible or easier to detect by attackers.

Close Integration with Cryptography Stage

In a system that employs encryption as well as watermarking, the decryption and watermarking stages need to be closely integrated. It should not be possible for an attacker to obtain access to the content in the stage where it has been decrypted, but not watermarked [68].

Security Key Management

As stated in Section 2.1.4, a properly designed watermarking scheme will be secure as long as the security keys stay secure. Proper key management needs to be implemented and in the final system, the key needs to be conveyed with a higher level of security than the content itself [61].

Legal Considerations

The legal status of watermarking systems needs to be considered for each implementation. In the case of transaction tracking, the correct legal procedures for dealing with misappropriations need to be defined. The legal issues of watermarking systems are outside the scope of this research, but are of importance, as watermarking applications may lead to court proceedings if a misappropriator is found.

2.2 Vision Systems

With the field of video watermarking studied, it is now necessary to examine how images are stored, displayed and perceived. Colour representation is considered, after which properties of the human visual system applicable to video watermarking are detailed.

2.2.1 Colour Representation

Different mathematical representations can be used to represent colour in imaging applications. These different representations are called colour spaces and the reader is referred to [69] for an overview and motivation of these different colour spaces. Two of these, namely the RGB and YC_bC_r colour spaces are now reviewed.

The RGB Colour Space

The Red-Green-Blue (RGB) colour space is a popular choice for representing colour and processing images, as devices often capture and display colour using this colour space [69]. Three primary colours namely red, green and blue are each additively mixed to obtain a desired colour, with equal bandwidth allocated to the representation of each colour channel. This representation is not always desirable, as it is not modelled on the properties of the human visual system, but rather the way in which colour is displayed and captured by devices. The RGB colour space can also be less efficient for use in image processing applications. For instance, to only modify the intensity of a pixel, all three colour pixels need to be modified, meaning that all three

colour components need to be processed to obtain this result. For applications such as watermarking, it is often desired to have direct access to the intensity values in an image, in which case the YC_bC_r colour space is more convenient.

The YUV and YC_bC_r colour spaces

Most watermarking techniques convert content to a luminance and chrominance colour representation before performing watermarking. Two popular colour spaces of this type are YUV and YC_bC_r representations. These two representations are similar and are scaled and offset versions of each other [69]. The YUV representation is often used in composite systems, while the YC_bC_r is used in digital imaging. As watermarking is done in the digital domain in this research, the YC_bC_r colour space is discussed in more detail.

A YC_bC_r representation of an image can be obtained from the RGB representation using:

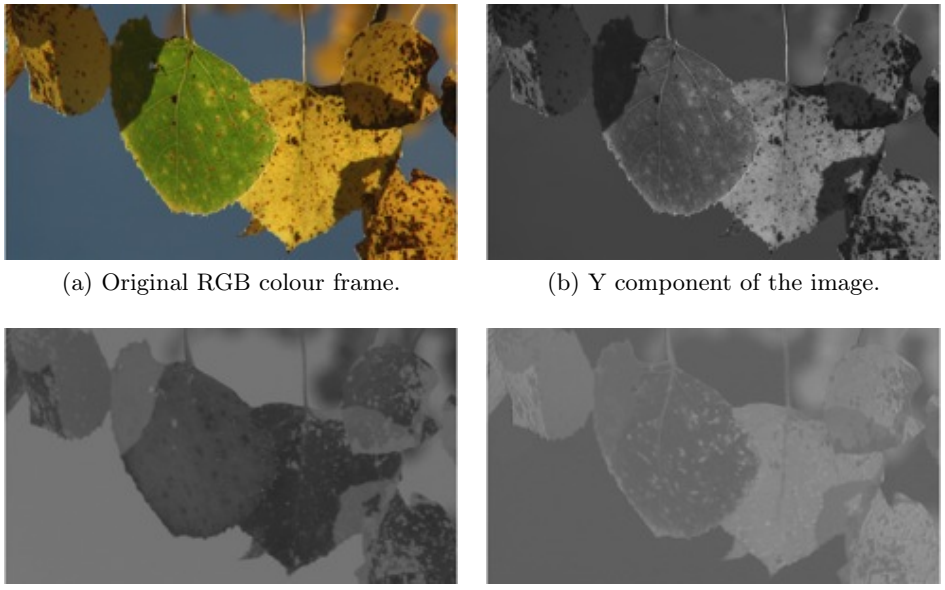
$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{C_b} \\ \mathrm{Cr} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix} \tag{2.1}$$

where R, G and B pixels represent the three colour components of the RGB image.

After this conversion, three component matrices, namely $\mathbf{Y}$, $\mathbf{C_b}$ and $\mathbf{C_r}$, are obtained. The $\mathbf{Y}$ matrix represents the luminance component of the image. This is essentially a greyscale version of the original video and specifies the brightness of each pixel in the video. The $\mathbf{C_b}$ and $\mathbf{C_r}$ components contain colour information of the image. As stated in [15], $\mathbf{C_b}$ can be written as $\mathbf{C_b} = 0.564\,(\mathbf{B} - \mathbf{Y})$ and $\mathbf{C_r}$ as $\mathbf{C_r} = 0.713\,(\mathbf{R} - \mathbf{Y})$. In other words, $\mathbf{C_b}$ represents the difference between the original blue component and the luminance component of the image. This translates to $\mathbf{C_b}$ indicating how much the signal deviates from grey towards blue. Similarly, $\mathbf{C_r}$ indicates the how much the signal deviates from grey towards red. The different components are shown in Figure 2.2. Note how the C_b component shows high amounts of blue for the sky areas, while the C_r components shows higher amounts of red in the leaves.

When using the YC_bC_r colour space, it is possible to use different bandwidths to represent each colour component in the image. This is advantageous, as the human visual system is more sensitive to high frequencies in luminance than in chrominance. It follows that more bandwidth can be allocated to the luminance component and less to the chrominance component. This is known as chroma subsampling and is further discussed in [69, pages 19 – 20].

In terms of watermarking, it is desirable to embed a watermark into the Y com-



(a) Original RGB colour frame. (b) Y component of the image.

(c) C_b component of the image. (d) C_r component of the image.

Figure 2.2: Representation of the different components of a colour image converted from the RGB to YC_bC_r colour space. Note how the C_b component shows high amounts of blue for the sky areas, while the C_r component shows higher amounts of red in the leaves.

ponents of video frames. From the discussion above, it is evident that modifying the Y component of a video frame as a form of an attack should reduce the perceived image quality more than filtering colour components will. This also has the advantage that a watermark can be extracted from a greyscale version of the watermarked material.

Furthermore, by first converting the image data to the YC_bC_r colour space, it is possible to watermark the image by only modifying the luminance component, reducing computational complexity.

2.2.2 The Human Visual System

The Human Visual System (HVS) is a complex mechanism through which humans perceive vision. While this system is challenging to model mathematically [70–72], it is necessary to understand the basic properties of this system when researching watermarking techniques. With this in mind, the most prominent properties of the HVS applicable to watermarking are now detailed.

Frequency Sensitivity

The HVS is not equally sensitive to all frequency components in an image, being less sensitive to noise in high frequency bands of images and also in bands having an orientation of 45° [24]. This is described by the Modulation Transfer Function (MTF) of the human eye in [73], which describes the sensitivity of the human eye to sine wave gratings at different frequencies. By using the MTF, it is possible to determine a threshold for each frequency component in an image at which changes will not be noticeable at a fixed viewing distance. It follows that detail below the threshold can be discarded in compression applications, or be used for watermark embedding in watermarking applications.

Alternatively, the MTF can be used to compare the visual quality of a compressed or watermarked image to that of the original. An example of such a use is the Just Noticeable Difference (JND) map. The example JND map given in [74] is shown in Figure 2.3. Figure 2.3a shows the original image, while Figure 2.3b shows the compressed version. The Sarnoff JND model [75] uses the MTF to generate a JND map to indicate the perceived visual quality for different regions in the image, as shown in Figure 2.3c. Bright areas indicate more perceptible distortions in the compressed version of the image, while dark areas indicate relatively high perceived visual quality. The concept of a JND map is further used in Section 4.2.3 on page 46 of this document.

Luminance sensitivity

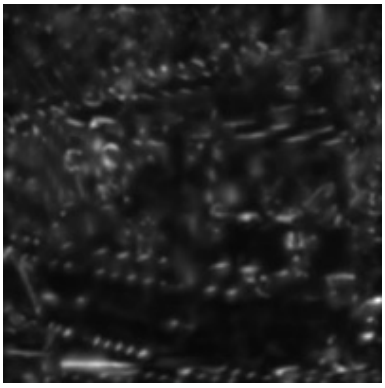
The sensitivity of the HVS to noise is a nonlinear function which is affected by the average background intensity and the intensity of the noise in an image [67]. The HVS is less sensitive to noise in areas of an image with either low or high luminance, while being more sensitive to noise in areas with medium luminance [24].

Contrast masking

Contrast masking is defined in [67] as *the detectability of one signal in the presence of another signal.* This refers to the fact that the eye is less sensitive to noise in areas of an image that are textured, but slightly more sensitive around the edges than the inner parts of these areas [24]. The masking effect is increased when the noise and original image content is of the same frequency [76].

(a) Original image.

(b) Compressed image.



(c) JND map indicating areas of visible distortion.

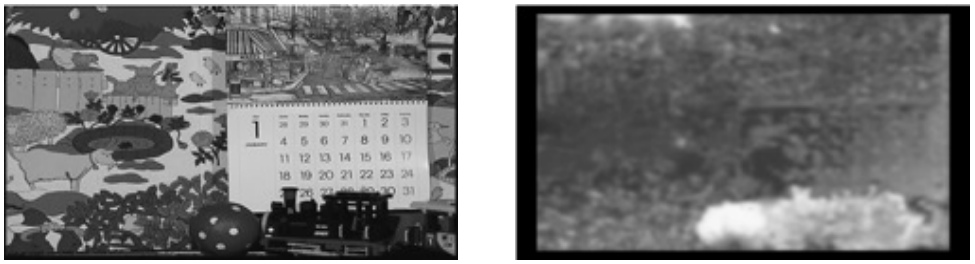
Figure 2.3: An example of a JND map shown in [74] to evaluate the perceived visual quality of a compressed or watermarked image compared to the original version. Bright areas indicate more perceptible distortions in the compressed version of the image, while dark areas indicate relatively high perceived visual quality.

Visual Attention Models

The HVS is more sensitive to noise in areas of an image where the viewer is paying more attention to. For instance, noise in the background of a scene with foreground action may be less noticeable. This effect is further discussed in [77–80] and can be modelled using texture, luminance and motion in video sequences, or using face detection. An example result from the model in [80] is shown in Figure 2.4. Brighter areas indicate regions which more attention will be paid to by the HVS.

Temporal Contrast Threshold

In the case of video watermarking, some properties should be considered that are not necessary when evaluating the quality of still images. An example of this is the



(a) Original image. (b) Result indicating areas of interest.

Figure 2.4: An example result from the model visual attention model in [80]. Brighter areas indicate regions which more attention will be paid to by the HVS.

temporal contrast threshold of the HVS.

When a watermarked video sequence is played back, it is possible that a flickering effect may occur. This is caused when large intensity changes to pixels in the same position in successive frames are made. The point where a change in pixel intensities between frames becomes perceptible is known as the temporal contrast threshold of the HVS and is discussed in [81] and [82].

2.2.3 Overview of Video Watermarking Techniques

Surveys of video watermarking techniques were conducted in [35, 37, 38, 83–85] providing a brief overview of techniques, while [39] is a longer publication providing more detail on different techniques. The watermarking techniques found in the literature can mostly be grouped into five main categories which are now reviewed.

Spatial Domain Watermarking

Spatial Domain (SD) or spread-spectrum techniques refer to a method of watermark embedding and extraction that is performed in the spatial domain, without the need to apply mathematical transforms on the original content. The watermarks are usually encoded to form a noise-like sequence and then added to the original content, while extraction is usually performed with a correlation-based receiver.

Since no mathematical transforms are required, these techniques are relatively computationally efficient. This is advantageous in real-time applications or where resources available for embedding are limited.

Examples of such techniques in the literature include [46, 86–89]. The broadcast monitoring scheme, JAWS [46], mentioned under the watermarking application section in this chapter is an SD technique, to keep computational complexity of the technique low. The technique in [89] is further discussed in Section 3.2.1 on page 27

of this thesis.

Discrete Fourier Transform Watermarking

Discrete Fourier Transform (DFT) techniques take advantage of properties of the DFT to gain robustness against attacks such as spatial and temporal shifts. In order to embed the watermark, a DFT is performed on the original content after which the watermark is embedded by modifying elements in the frequency domain. After the watermark is embedded, an inverse DFT is performed to obtain the watermarked content.

DFT-based video watermarking techniques include [90–93]. The SD technique originally discussed in [91] was accompanied by a proposal to rather apply watermarking using the DFT specifically to obtain robustness against geometric distortions. The technique in [93] is further discussed in Section 3.3.2 on page 31 of this thesis.

Singular Value Decomposition Watermarking

The Singular Value Decomposition (SVD) is a technique that can be used in image compression techniques, but can also be applied to watermarking. The SVD is performed, after which the singular values are usually modified to embed the watermark. A pseudo-inverse SVD is then applied to obtain the original content.

Techniques that use the SVD include [94–98]. The SVD can be used on its own for watermarking, but is also often used in hybrid techniques such as [94] which combines the SVD and the discrete cosine transform. The SVD is relatively computationally complex, but by applying it in hybrid techniques it may not be necessary to perform an SVD on the entire image, lowering the computational complexity. SVD video watermarking techniques seem to only have gained popularity after 2006, compared to other techniques that were pioneered in the late 1990s. This can possibly be attributed to the computational complexity of the SVD which may have prohibited the use in video applications when computing power was limited. The technique in [98] is further discussed in Section 3.4.2 on page 34.

Discrete Wavelet Transform Watermarking

Discrete Wavelet Transform (DWT) watermarking can be used to embed watermarks in areas of a video where it is less likely to cause perceptible distortions. The DWT decomposes video frames into different resolution components, which are then modified to embed the watermark. An inverse DWT is performed to obtain the watermarked content.

Examples of DWT techniques in the literature include [26, 99–105]. The technique detailed in [105] is discussed in more detail in Section 3.5.1 on page 35.

Discrete Cosine Transform Watermarking

Discrete Cosine Transform (DCT) techniques are often used to watermark compressed video streams. DCT coefficients in video streams can be modified without having to first uncompress the video or compress it again after watermarking.

DCT watermarking techniques found in the literature include [21, 31, 32, 89, 106–108] and operate in the compressed video domain. As the focus of the research is the uncompressed video domain, these are excluded from further evaluation.

Other Watermarking Techniques

There exists several other video watermarking categories found in the literature, such as moment-based watermarking [109, 110] and techniques using principal component analysis [96, 111]. Moment-based techniques are of interest because they provide robustness against rotation, but can be computationally complex.

These categories of watermarking techniques are less prominent in the literature and were excluded from further evaluation.

2.3 Chapter Summary

In this chapter, the foundations of video watermarking research were discussed. Various applications of video watermarking were discussed, with references to techniques in the literature designed for each. General requirements for video watermarking were then defined and extended to include robustness against common attacks, which were briefly summarised.

The discussion then turned to how images are represented by digital systems, with references to in-depth explanations on the topic. The properties of the HVS were then discussed and it was noted that a modern video watermarking technique should ideally exploit these properties in order to increase robustness against attacks.

The chapter was then concluded by a literature review of video watermarking techniques. References were provided to video watermarking surveys, after which the main categories of video watermarking techniques were discussed.

After reading this chapter, the reader should have a broad understanding of video watermarking techniques, including the main categories of watermarking techniques and important properties of the HVS.

Chapter 3

Watermarking Implementations

The selection process for representative watermarking techniques from each category in Section 2.2.3 is now discussed. Mathematical principles of each category are explained, after which the implementation of each chosen technique is examined in more detail.

3.1 Technique Selection and Implementation

3.1.1 Selection of Techniques

In order to compare the characteristics of each watermarking category, a representative technique from each was selected, which:

1. operates in the uncompressed domain for embedding and extraction;
2. is able to extract the watermark without the original video; and
3. does not apply error correction.

Most practical techniques employ some form of error correction coding such as [62] or [63] to improve robustness against attacks. Methods to detect and reverse geometrical transformations can also be employed to potentially improve robustness against geometric transforms, as discussed in [12, 31, 87, 93, 112, 113]. In order to evaluate the underlying watermarking techniques, any error correction or geometric correction methods specified in techniques were discarded and only the basic embedding and extraction stages of the techniques were implemented. The representative techniques chosen for evaluation are identified in Table 3.1.

Table 3.1: Representative watermarking techniques chosen for evaluation.

Category	Title of publication
SD	Watermarking of Uncompressed and Compressed Video [89]
DFT	Robust 3D DFT Video Watermarking [93]
SVD	SVD Based Blind Video Watermarking Algorithm [98]
DWT	A Digital Watermark Method Using the Wavelet Transform for Video Data [105]

3.1.2 Implementation Background

The inputs, outputs and indexing used to implement the watermarking techniques are now defined.

In each case the watermark to be embedded is an M-bit polar-encoded binary message $\mathbf{b} = (b_1, b_2, b_3, \ldots, b_M)^\top$ with $b_i \in \{-1, 1\}$, $0 < i \leq M$, where $b_i = +1$ represents a binary 1, and $b_i = -1$ represents a binary 0. This message is embedded into the luminance frames in a block of uncompressed video frames $D(x, y, z)$ to obtain a block of watermarked video frames $W(x, y, z)$, with $1 \leq x \leq X$, $1 \leq y \leq Y$ and $1 \leq z \leq Z$. X and Y represent the width and height of the video frames in pixels respectively while Z specifies the number of frames contained in the video block D.

A function to map Cartesian coordinates to a serial index of pixels in a video block is defined as follows:

$$n = I(x, y, z) = (z - 1)XY + (y - 1)X + x. \tag{3.1}$$

The inverse of (3.1) is also defined as a function taking an index n as argument, and rendering a 3-tuple:

$$(x, y, z) = I^{-1}(n) = (I_x^{-1}(n), I_y^{-1}(n), I_z^{-1}(n)). \tag{3.2}$$

The implementation of the selected watermarking techniques are now discussed. The source code of these implementations can be viewed online at [114].

3.2 Spatial Domain Watermarking

3.2.1 Implementation of the Spatial Domain Watermarking Technique

The SD watermarking technique in [89] uses a spread-spectrum approach to add the watermark signal to video frames without the need to perform mathematical transforms on the host video data. Spread-spectrum techniques represent a narrowband

signal (the binary message) in such a way that it is suitable for transmission over a wideband channel with interference (the host video D).

Embedding

The embedding process for a single video frame using the SD technique is summarised in Figure 3.1. The basic idea is to repeat or chip each bit in the watermark message to be embedded in such a way that the chipped watermark message contains the same number of elements as the original video block. This signal is shown in part (b) of Figure 3.1 and is called a chipped watermark signal. This chipped signal is then multiplied with a frame filled with pseudorandom noise (c) to obtain the watermark signal (d) to be added to the original video frame. The watermark signal in (d) is then amplified by a factor α and added to the original video frame to obtain the watermarked video frame (e). A more formal mathematical explanation now follows.

Each frame of video block D is first line-scanned to represent the complete block in a one-dimensional form. The watermark sequence to be embedded, $\mathbf{b}$, is then spread by a large factor η to obtain $\mathbf{b}' = \mathbf{b} \otimes \mathbf{1}^{\eta}$, where $\otimes$ represents the Kronecker product, and $\mathbf{1}^{\eta}$ is a $1 \times \eta$ vector of ones. The resulting $\mathbf{b}'$ is then a vector with each -1 or $+1$ in the original message $\mathbf{b}$ repeated η times. The spreading factor η can be calculated such that the line-scanned video block and the chipped watermark signal are of the same length. The sequence $\mathbf{b}'$ is then amplified with an amplitude factor α to control embedding strength, and multiplied element-wise with a pseudo-noise sequence p to realise frequency spreading.

The function representing the watermarked video block can then be defined as

$$W(x, y, z) = D(x, y, z) + \alpha \mathbf{b}'_{I(x,y,z)} p_{I(x,y,z)} \tag{3.3}$$

where $\mathbf{b}'$ is indexed in such a way as to spread the vector over the entire video block.

Extraction

To begin extraction, each watermarked frame from the watermarked video block W is first high-pass filtered to obtain $\widetilde{W}$, which is then line-scanned as in the embedding stage. This line-scanned signal is then demodulated by multiplying it with the same pseudo-noise sequence p that was used in the embedding stage. The correlation for each bit is then calculated using a correlation receiver. This is achieved by summing the bits over a window of length η for each embedded bit. If the correlation is

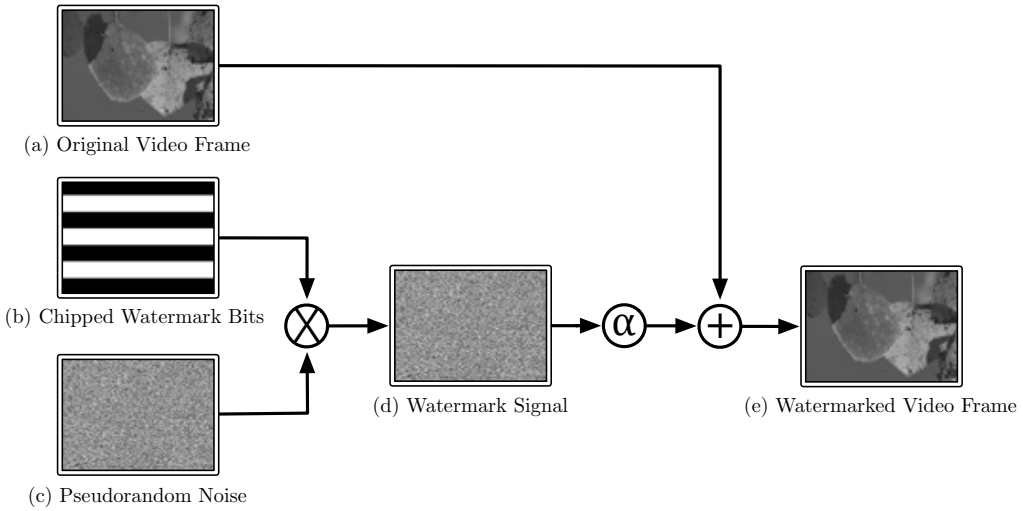


Figure 3.1: A visual summary of the embedding process for the SD technique. For demonstration purposes, only a single frame is shown here, but the process is spread across each video frame in D in order to embed $\mathbf{b}$ into the whole video block.

positive, a binary 1 is detected. If the correlation is negative, a binary 0 is detected. This process is repeated M times until the complete watermark is extracted.

The watermark extraction process can therefore be summarised as

$$\hat{b}_j = \text{sgn}\left[\sum_{n=(j-1)\eta+1}^{j\eta} p_n \widetilde{W}(I_x^{-1}(n), I_y^{-1}(n), I_z^{-1}(n))\right] \tag{3.4}$$

with $1 \leq j \leq M$.

3.3 Discrete Fourier Transform Watermarking

3.3.1 Fundamentals of the Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a common computational tool that can be used for various applications including linear filtering and spectrum analysis. For most imaging applications, a multi-dimensional DFT is required. Fortunately, the techniques used for a one-dimensional DFT can easily be extended to perform a multi-dimensional DFT. To aid further discussion, a brief overview of the DFT is now given. For consistency with the rest of this text, an uppercase variable refers to a spatial domain representation, while a lowercase variable refers to frequency-domain representation.

A one-dimensional, N-point DFT is defined in [115] as

$$h(k) = \sum_{n=0}^{N-1} H(n) e^{-j2\pi kn/N}, \quad k = 0, 1, 2, \ldots, N-1. \tag{3.5}$$

To apply a multi-dimensional DFT, this process is simply repeated along each dimension required. If a three-dimensional DFT of a video block D was required, this could be obtained using

$$d\left(k_x, k_y, k_z\right) = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \sum_{z=0}^{Z-1} D\left(x, y, z\right) e^{-j2\pi xk_x/N_x - j2\pi yk_y/N_y - j2\pi zk_z/N_z} \tag{3.6}$$

with k_x, k_y and k_z defined as in (3.5) for each dimension.

The inverse DFT can also be extended for multiple dimensions. The one-dimensional inverse DFT is defined as

$$H(n) = \frac{1}{N} \sum_{k=0}^{N-1} h(k) e^{j2\pi kn/N}, \quad n = 0, 1, 2, \ldots, N-1 \tag{3.7}$$

and can be extended to three dimensions as

$$D(x, y, z) = \frac{1}{N_x N_y N_z} \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_z=0}^{N_z-1} d\left(k_x, k_y, k_z\right) e^{j2\pi k_x x/N_x + j2\pi k_y y/N_y + j2\pi k_z z/N_z}. \tag{3.8}$$

Note that in (3.6), k_x and k_y represent the spatial frequencies of frames, while k_z represents the temporal frequencies in the video block [93].

The properties of interest of the DFT are now briefly discussed. The manner in which the DFT responds to rotation, scaling and translation is important for video watermarking applications. These are stipulated in [93] and are repeated here for clarity.

When rotation is applied to each two-dimensional video frame in a video block D, the resulting DFT spectrum is rotated by the same amount. If D is rotated by θ degrees around the z-axis, it follows that

$$d(k_x \cos\theta - k_y \sin\theta, k_x \sin\theta + k_y \cos\theta, k_z) \leftrightarrow D(x \cos\theta - y \sin\theta, x \sin\theta + y \cos\theta, z). \tag{3.9}$$

If D is scaled by a scaling factor λ, inverse scaling results in the frequency spectrum:

$$d\left(\frac{k_x}{\lambda_x}, \frac{k_y}{\lambda_y}, \frac{k_z}{\lambda_z}\right) \leftrightarrow D\left(\lambda_x x, \lambda_y y, \lambda_z z\right). \tag{3.10}$$

Another important property of the DFT is that a shift in the spatial domain causes a linear phase shift in the frequency domain. When a shift of a, b and c pixels is applied to the spatial domain in the x, y and z dimensions respectively, it follows that

$$d\left(k_x, k_y, k_z\right) \exp\left[-j\left(ak_x + bk_y + ck_z\right)\right] \longleftrightarrow D(x + a, y + b, z + c). \tag{3.11}$$

With the basics of the DFT discussed, the DFT watermarking technique can now be examined.

3.3.2 Implementation of the DFT Watermarking Technique

The chosen DFT technique [93] uses the three-dimensional DFT to embed spread-spectrum information into the block of video frames D. The technique includes a method to detect spatial transformations such as rotation and aspect ratio change that was not implemented, as discussed in Section 3.1.1.

Embedding

The message to be embedded, $\mathbf{b}$, is firstly represented in a spread-spectrum form. A family of Gold codes [116], $\mathbf{G}$, is generated with a Gold code for each bit in $\mathbf{b}$. $\mathbf{G}$ is an $N \times M$ matrix, where the i^{th} column is a Gold code for the bit b_i. $\mathbf{G}$ is also encoded in polar form, with each element $G_i \in \{-1, 1\}$. The spread spectrum signal to be embedded is therefore obtained as follows:

$$\mathbf{b}' = \mathbf{G}\mathbf{b}. \tag{3.12}$$

A three-dimensional DFT is now performed on the block of frames D and a frequency shift is performed to position the zero-frequency component in the middle of the resulting frequency spectrum. Elements to be modified are selected in such a way as to embed the watermark in the mid-range frequency bands in both the spatial and temporal dimensions. This is achieved by selecting elements in the frequency domain to form cylindrical annuli, as shown in Figure 3.2. Elements to be selected in the x and y dimensions are specified by $r_{\min}$ and $r_{\max}$, while elements to be selected in the temporal dimension are specified by $d_{\min}$ and $d_{\max}$. The values of $d_{\min}$ and $d_{\max}$ indicate how many components in the temporal dimension to exclude. Setting $d_{\min}$ and $d_{\max}$ to 0 will embed the watermark in all temporal frequencies of the video block.

In order for the result of the inverse DFT to be real, the symmetry of the frequency spectrum needs to be maintained. To achieve this, changes made to elements

in the first annulus is mirrored in the second annulus, ensuring that symmetry is maintained.

Further, negative magnitudes in the frequency domain will also result in a non-real result after the inverse DFT. Recall that we modify the magnitudes of the elements in the frequency domain, therefore it needs to be ensured that changes made to the elements do not result in negative magnitudes. Because $\mathbf{b}'$ can contain negative values, each element $\mathbf{b}_i$ is replaced with the pair

$$\mathbf{b}_i'' = \begin{cases} (\mathbf{b}_i', 0) & \text{if } \mathbf{b}_i' \geq 0 \\ (0, \mathbf{b}_i') & \text{if } \mathbf{b}_i' < 0 \end{cases}. \tag{3.13}$$

The values of $\mathbf{b}''$ are then scaled with an amplitude scaling factor α, and added to the magnitudes of the selected elements in the spectrum in such a way as to keep the values symmetrical around the origin of the spectrum. An inverse frequency shift is then performed, followed by an inverse DFT to obtain the watermarked video block W.

The DFT technique is host-adaptive in the sense that it embeds the watermark bits by adding or subtracting values to or from the magnitude of frequency components, maintaining the phase of the frequency components. The amounts which are added or subtracted are, however, not affected by the original magnitudes of the frequency components being modified.

Extraction

A DFT is performed on the video block W, after which a frequency shift is performed to obtain the frequency spectrum of W. The frequency spectrum elements used for extraction are selected in the same manner as during the embedding stage. Recall that the location of each coefficient pair, $\mathbf{b}_i''$, is known. The values of these are retrieved for each bit to be extracted and the difference between each of these coefficient pairs are computed in order to obtain $\mathbf{b}'''$. The embedded message $\hat{\mathbf{b}}$ is then extracted as $\hat{b_i} = \text{sgn}(b_i''')$ with $1 \leq i \leq M$.

3.4 Singular Value Decomposition Watermarking

3.4.1 Fundamentals of the Singular Value Decomposition

Singular Value Decomposition (SVD) is a linear algebraic technique which optimally decomposes matrices to represent a maximum amount of signal energy in as few

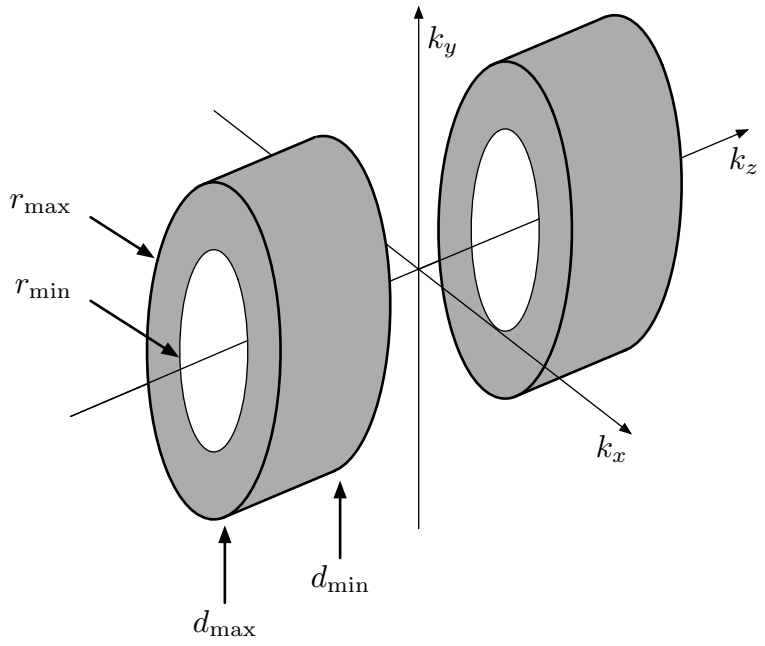


Figure 3.2: A visual representation of the elements selected for embedding in the frequency domain. By selecting the elements in this fashion the watermark is embedded in the middle-frequencies of the video. Elements to selected in the x and y dimensions are specified by $r_{\min}$ and $r_{\max}$, while elements to be selected in the temporal dimension are specified by $d_{\min}$ and $d_{\max}$.

coefficients as possible [117].

As stated in [118], for any real $X \times Y$ matrix $\mathbf{A}$ of rank r there exists a real factorisation

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} \tag{3.14}$$

where $\mathbf{U}$ and $\mathbf{V}$ are square orthogonal matrices with dimensions of $X \times X$ and $Y \times Y$, respectively. The columns of $\mathbf{U}$ are called left singular vectors of $\mathbf{A}$, while the columns of $\mathbf{V}$ are called the right singular vectors of $\mathbf{A}$.

$\mathbf{S}$ is a pseudo-diagonal $X \times Y$ matrix

$$\mathbf{S} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \sigma_r & 0 \end{bmatrix} \tag{3.15}$$

containing the singular values of $\mathbf{A}$ in descending order of magnitude

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r. \tag{3.16}$$

A can also be decomposed as the sum of rank 1 matrices

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j u_j v_j^\top \tag{3.17}$$

where r is the number of non-zero singular values in **S** while u_j and v_j are the column vectors of **U** and **V** respectively. Each iteration is called a rank approximation of **A**. If **A** represents an $X \times Y$ image, the first few rank approximations contain the most signal energy, with following approximations adding detail to the image [119]. Truncating the number of rank approximations used to represent an image is an example of how the SVD can be used for image compression. Refer to [120] for an overview of the SVD and its applications.

3.4.2 The SVD Watermarking Technique

The SVD watermarking technique in [98] employs the properties of the SVD to obtain robustness against small rotations, resizing and flipping of the video frames.

Embedding

The SVD is applied to each video frame **A** of video block D. For any frame, this results in three matrices, **U**, **S** and **V** such that $\mathbf{A} = \mathbf{USV}^\top$. The singular values of **A** are then modified to embed the watermark.

As shown in (3.17), modifying singular values with a lower index j would cause changes to coarser rank approximations of **A**, which may increase the perceptibility of the watermark. Modifying singular values with higher indices would result in modifying detail in the image, being potentially less perceptible, but more easily affected by noise and small changes to the image. For these reasons, indices of the largest and smallest singular values to be used for watermarking, $\sigma_{\min}$ and $\sigma_{\max}$, are selected to embed to watermark in the midrange frequencies of the image. The embedding strength is controlled by an amplitude scaling factor α. In practice, $\sigma_{\min}$ is specified and $\sigma_{\max}$ is automatically calculated depending on the number of bits that need to be embedded in the video frame.

The watermark message b, containing M information bits, is now embedded by modifying the selected singular values of **A**. Recall that the singular values in **S** are arranged in decreasing order of magnitude. When modifying these singular values, it is important to do so in such a way that does not change the magnitudes to such

an extent that the ordering of these values will change. To achieve this, the singular values are modified using

$$\sigma_j^{'} = 0.5\left[(\sigma_{j-1} + \sigma_{j+1}) + \alpha b_i\left(\sigma_{j-1} - \sigma_{i+1}\right)\right] \tag{3.18}$$

with $1 \leq i \leq M$ and $j = \sigma_{\min} + 2(i-1)$ to obtain $\mathbf{S}'$.

After the singular values have been modified, the watermarked frame $\mathbf{A}^{'}$ is obtained using $\mathbf{A}^{'} = \mathbf{U}\mathbf{S}^{'}\mathbf{V}^{\top}$. This process is repeated for each video frame in block D to obtain W.

The above method is based on the concept of changing the value of σ_j to either be closer to σ_{j-1} or σ_{j+1}, depending on whether a 1 or -1 is embedded. The extraction stage then uses the same technique to determine whether a 1 or -1 was embedded using the three singular values.

Extraction

The watermark is extracted by applying the SVD to each watermarked frame in video block W. The extracted watermark message $\hat{b}$ is then extracted by concatenating the watermark information bits extracted from the modified singular values in each watermarked video frame. The watermark information bits are extracted using

$$\hat{b}_i = \begin{cases} -1 & \text{if} \quad \sigma_i > 0.5(\sigma_{i-1}^{'} + \sigma_{i+1}^{'}) \\ 1 & \text{if} \quad \sigma_i < 0.5(\sigma_{i-1}^{'} + \sigma_{i+1}^{'}) \end{cases} \tag{3.19}$$

with $1 \leq i \leq M$ and $j = \sigma_{\min} + 2(i-1)$.

3.5 Discrete Wavelet Transform Watermarking

3.5.1 Fundamentals of the Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) can be used to obtain a multi-resolution representation of digital images for analysis. It has also been shown that the human vision system decomposes a retinal image into various spatially oriented frequency channels in a way similar to the DWT [121]. Because of these properties, the DWT can be used to identify areas in an image where a viewer is less likely no notice image distortions. The DWT also exhibits good spatio-frequency localisation properties where, if a DWT coefficient is modified, only the region of the image containing the specific frequency of the modified coefficient is affected [24]. This allows control of the area in an image where a watermark will be embedded, compared to other techniques that may affect the complete video frame.

A single-level wavelet decomposition decomposes a video frame into three multi-resolution representations, namely LH_1, HL_1, HH_1 and a single multi-resolution approximation named LL_1. These are also called subbands. The LL_1 subband contains a low frequency approximation of the original image. The three remaining subbands, LH_1, HL_1 and HH_1, contain edge detail in the horizontal, vertical and diagonal directions, respectively. This decomposition process can be repeated as many times as required, by each time further decomposing the LL_n approximation. As with the DFT, the one-dimensional DWT can be extended for multiple dimensions by simply applying the one-dimensional process to each dimension in a separable fashion.

The image is decomposed into a set of frequency channels of constant bandwidth on a logarithmic scale. In other words, the image is decomposed into different scale levels, where required levels are then further decomposed with a resolution suited to the level [122]. Unlike the Fourier transform, which has a fixed spatial and frequency resolution, the resolution of the wavelet transform varies with level of the DWT. Through this approach, we obtain a hierarchical multi-resolution representation of the image which is well-suited for interpreting the information contained in the image [123]. A decomposition obtained through a three-level DWT is shown in Figure 3.3. As the level of the DWT increases, the spatial resolution of the subbands decrease.

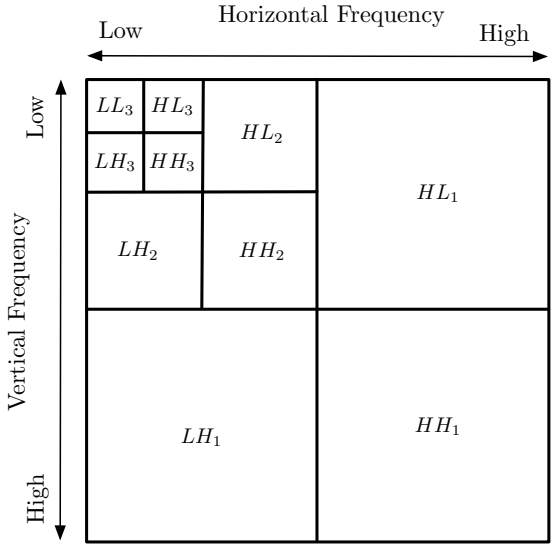


Figure 3.3: A decomposition obtained through a three-level DWT. As the level of the DWT increases, the spatial resolution of the subbands decrease.

Figure 3.4 depicts the steps involved in a single-level two-dimensional DWT. We

start by convolving the rows on LL_n with a specified decomposition low-pass and high-pass filters in two separate stages. In both cases, the columns of the result are downsampled by two, retaining the even indexed columns. The two results are then again filtered and downsampled, in this case retaining even indexed rows.

The multi-resolution approximation LL_{n+1} can then be further decomposed if required, where a K-level DWT would represent a result from repeating this filtering and downsampling process K times [124].

The inverse of these filtering and downsampling steps can be applied to obtain the original video frame from the multi-resolution representations.

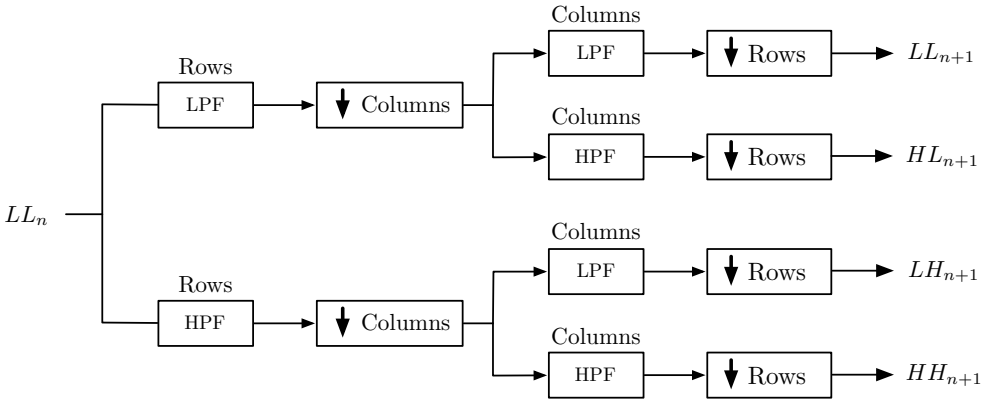


Figure 3.4: A summary of the steps involved in a single-level two-dimensional DWT. The process is started by convolving the rows of LL_n with specified decomposition low-pass and high-pass filters in two separate stages. In both cases, the columns of the result are downsampled by two, retaining the even indexed columns. The two results are then again filtered and downsampled, in this case retaining even indexed rows.

The fundamentals of the DWT applicable to video watermarking were now discussed. Refer to [125] and [126] for a detailed discussion regarding the DWT and wavelets.

3.5.2 The DWT Watermarking Technique

The DWT technique described in [105] allows the embedding of the watermark into textured areas of the image and excludes non-detailed areas.

Embedding

We start by applying a K-level two-dimensional DWT to each video frame in video block D. K is chosen to be a suitable value for the video frame dimensions and desired computational complexity. Larger values of K result in the K^{th} multi-resolution approximation containing less detail of the video frame, which will in turn cause the technique to modify larger areas of the video frame as it embeds the watermark. This potentially decreases the chances of the coefficients being modified by image transformations, as these subbands are less affected by small changes in the image.

The coefficients in each subband are now indexed such that $B_n(r_j)$ refers to a coefficient with index r in a subband obtained from an n^{th} level DWT decomposition. The manner in which the subbands are indexed is not of importance.

The index r_j is the j^{th} number generated by a pseudorandom number generator $\mathtt{F}(\cdot)$. To generate a pseudorandom number, $\mathtt{F}(\cdot)$ is initially seeded with the value 1. Thereafter, each successive number is generated by seeding $\mathtt{F}(\cdot)$ with the previously generated pseudorandom number such that $r_j = \mathtt{F}(r_{j-1})$ and $1 \leq r_j \leq \left(X/2^K\right)\left(Y/2^K\right)$. In this case, the result of $\left(X/2^K\right)\left(Y/2^K\right)$ calculates the number of coefficients contained in each subband after K DWT decompositions. E is then calculated as

$$E = \sum_{n=K-1}^{K} \left\{ \left\| B_n^{LH}(r_j) \right\| + \left\| B_n^{HL}(r_j) \right\| \right\} \tag{3.20}$$

where $\|B\|$ represents the L^1-norm of the coefficients in subband LH_n and HL_n and $1 \leq n \leq K$. If E is larger than a threshold T, a single bit is embedded into the subband $B_N^{LL}(r_j)$ using a controlled quantisation process. If $E \leq T$, then r_j is incremented by 1 and used to seed $\mathtt{F}(\cdot)$ for calculation of the next pseudorandom index. Equation 3.20 is then again applied and the result compared to the threshold T. This process is repeated until a suitable coefficient is found for embedding, or until all possible coefficients were evaluated. If no suitable coefficients were found, the watermark bit is not embedded.

The quantisation process used to embed a watermark information bit can be summarised in the following steps. The mean value Y_{rj} of the wavelet coefficients of $B_N^{LL}(r_j)$ is firstly calculated. This is then followed by the calculation of q_j as:

$$q_j = \text{round}\left(\frac{B_N^{LL}(r_j)}{\alpha}\right) \tag{3.21}$$

where $\text{round}(\cdot)$ represents a function rounding a value to the nearest integer and α denotes the quantisation step size which controls the embedding intensity.

An integer l is then computed such that

$$q_j \in [2l, 2(l+1)) \tag{3.22}$$

and q_j is modified to obtain q'_j according to the watermark information bit b_i to be embedded such that

$$q'_j = \begin{cases} 2l & \text{if} \quad b_i = -1 \\ 2l+1 & \text{if} \quad b_i = 1 \end{cases} \tag{3.23}$$

As a final step, we compute $Y'_{rj} = \alpha q'_j$ and $\delta_{rj} = Y'_{rj} - Y_{rj}$. δ_{ri} is then added to the wavelet coefficients of $B_N^{LL}(r_j)$.

The above process is repeated M times until all the bits of b have been embedded. After the wavelet coefficients have been modified to contain the watermark information, the inverse DWT is applied in order to obtain the watermarked video frame. This process is repeated for all frames in video block V to obtain watermarked video block W.

Extraction

The watermarked video frame is again decomposed K times, as was done in the embedding stage. The pseudorandom number generator $\mathtt{F}(\cdot)$ is seeded in the same manner as in the embedding stage to obtain a pseudorandom index r_j. Using (3.20), we determine E. If $E \leq T$, then r_j is incremented by 1 and used to seed $\mathtt{F}(\cdot)$ for calculation of the next pseudorandom index. If $E \geq T$, the value of watermark bit embedded in the coefficient $B_N^{LL}(r_j)$ is extracted.

We begin the extraction process for a single bit by again calculating the mean U_{rj} of the wavelet coefficients in $B_N^{LL}(r_j)$. The value of q_j is then again computed using (3.21). The embedded bit is then extracted as

$$\hat{b_i} = q'_j \text{mod}(2). \tag{3.24}$$

The binary result is then converted to polar coding.

The above process is repeated until the complete watermark has been extracted.

3.6 Chapter Summary

This chapter provided the reader with an understanding of the basic workings of each technique, along with the mathematical properties related to the technique. The selection and implementation of the watermarking techniques to be evaluated were explained, after which the various watermarking techniques were described and explained in terms of appropriate mathematical characteristics.

Chapter 4

Visual Evaluation of Watermarking Techniques

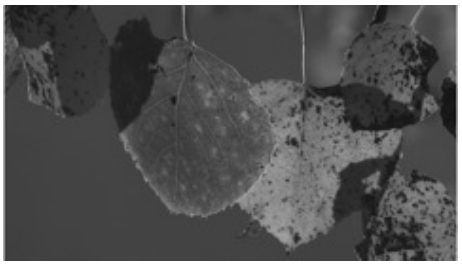
In order to evaluate the perceived visual quality of videos watermarked by techniques described in Chapter 3, the use of quantitative measures is required. In this chapter, the spatial and temporal quality of watermarked video sequences are evaluated. An existing spatial quality metric is extended to provide improved performance for this application. A novel method for evaluating temporal interframe flicker is also developed as a suitable temporal quality metric. Finally, the characteristics of the different watermarking artefacts caused by each watermarking technique are discussed.

4.1 Comparison of Artefacts Caused by Watermarking Techniques

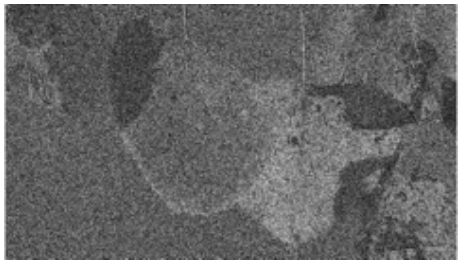
In order to illustrate the nature of the distortions caused by each technique, the frame shown in Figure 4.1 was watermarked with a high intensity using each technique. As expected, the distortions caused by the SD and DFT techniques have a uniform noise-like appearance. This is because white noise is added in the case of the SD technique, while noise is added to the mid-frequencies of the image in the case of the DFT technique.

The SVD causes block-like artefacts as the singular values of the image are modified. Modifying lower rank singular values causes larger blocks, decreasing in size as the rank of the modified singular value increases.

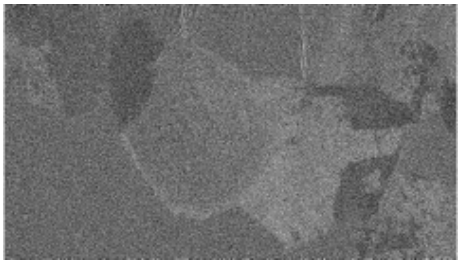
Artefacts caused by the DWT technique are localised distortions appearing in areas containing high texture detail. Placing the distortions in texture decreases the



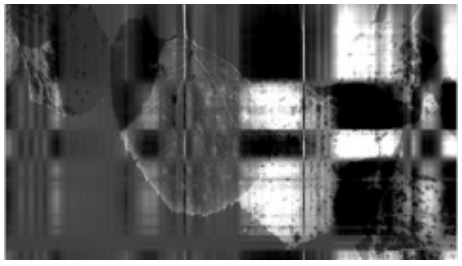
(a) Original frame.



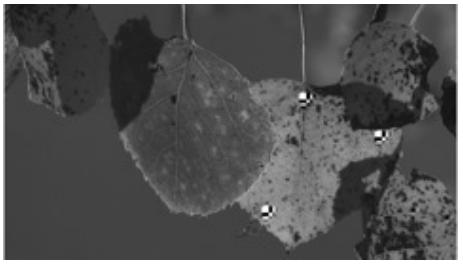
(b) Frame watermarked with SD technique using a high intensity setting.



(c) Frame watermarked with DFT technique using a high intensity setting.



(d) Frame watermarked with SVD technique using a high intensity setting.



(e) Frame watermarked with DWT technique using a high intensity setting.

Figure 4.1: Visual comparison of artefacts caused by each watermarking technique. Frames are watermarked with each technique using a high intensity for demonstration purposes.

visibility of the distortions.

With an understanding of the nature of the distortions caused by each watermarking technique, methods of quantifying the perceived visual quality of watermarked frames are now developed.

4.2 Spatial Image Quality Metric

Spatial quality refers to the perceived visual quality of a single video frame. Numerous spatial quality metrics such as the Peak Signal to Noise Ratio (PSNR), Video Quality Metric (VQM) [127], Noise Quality Measure (NQM) [128] and Structural SIMilarity (SSIM) index [129] exist. The performance of these are compared in

[130–133].

The PSNR is a popular image quality metric and is now reviewed in conjunction with the SSIM, which will be extended for use as a spatial quality metric for the implemented video watermarking techniques.

4.2.1 The Mean Squared Error and Peak Signal to Noise Ratio

The Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) are widely used in the literature as spatial quality metrics for images [11, 15, 129, 132, 134–137]. The MSE is defined as

$$\text{MSE} = \frac{1}{XY} \sum_{i=1}^{X} \sum_{j=i}^{Y} [\mathbf{A}(i,j) - \mathbf{B}(i,j)]^2 \tag{4.1}$$

where X and Y represent the width and height of the image in pixels and $\mathbf{A}$ and $\mathbf{B}$ represent the two images to be compared, respectively.

The PSNR is defined as the logarithmic ratio between the MSE and the maximum value L that a pixel in an image can assume. This can be expressed as

$$\text{PSNR} = 10 \log_{10} \frac{L^2}{\text{MSE}}. \tag{4.2}$$

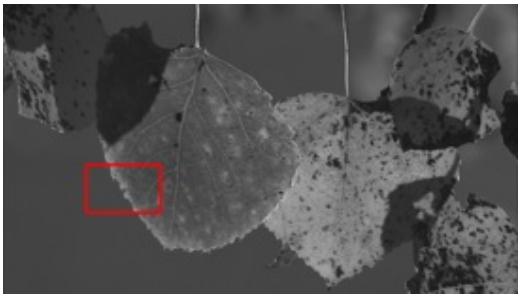
Both the MSE and PSNR calculate the difference between pixels in two images, squares the result and computes the average of this operation for the complete image in order to estimate the average distortion between two images. The PSNR represents the MSE as a logarithmic ratio between the maximum distortion that can be caused in an image and the MSE. This PSNR representation is helpful when comparing images with different dynamic ranges.

The PSNR is simple to implement and understand, but has been shown to correlate poorly with subjective quality testing results [70, 138, 139]. It has also been shown that the PSNR is unreliable when comparing different video compression codecs [134]. Since each video codec may produce visual artefacts with different visual characteristics, obtaining a single number representing the visual quality which is consistent for each artifact can be challenging. Comparing different video watermarking techniques poses the same challenge because, as for different video codecs, each technique can also cause visual artefacts with different properties.

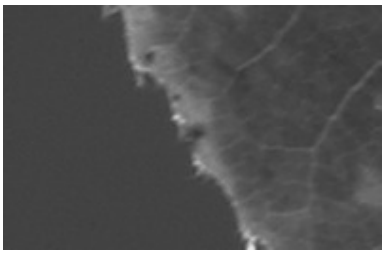
The problematic performance of the PSNR when comparing different visual artefacts is illustrated in Figures 4.2c and 4.2d. The parameters of the SD and DWT techniques were adjusted to obtain a PSNR of 40.09 dB in both cases, but artefacts due to the DWT technique are more perceptible than for the SD technique at this

PSNR. This is because the DWT modifies only a small portion of the image, while the SD technique modifies a large section of the image. While the artefacts caused by the DWT technique will produce a high local error, this local maximum has a small effect when the mean error is taken over the whole image. Refer to [140] for a comprehensive overview of the uses and limitations of the MSE and PSNR.

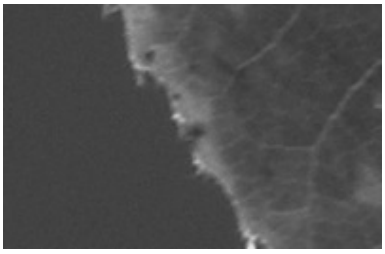
Because of the discussed limitations, it was necessary to find a more suitable image quality metric to compare different video watermarking techniques.



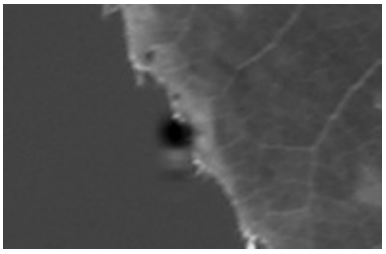
(a) Original video frame.



(b) Detail view of the original video frame.



(c) Video frame watermarked with the SD technique to obtain a PSNR of 40.09 dB.



(d) Video frame watermarked with the DWT technique to obtain a PSNR of 40.09 dB.

Figure 4.2: Visual quality comparison of video frames watermarked with different techniques to each obtain a PSNR of 40.09 dB. Note that, although both watermarked images obtain the same PSNR, noticeable distortion can be seen in the image watermarked with the DWT technique.

4.2.2 The Structural Similarity Index

The Structural SIMilarity index (SSIM) is an alternative visual quality metric introduced by [129]. The SSIM has been shown to be more effective at estimating the perceptual quality of images than the PSNR [132, 140] and was evaluated as a potential visual quality metric for this project.

The SSIM uses luminance, contrast and structure comparison functions to estimate the perceived quality of an image. This correlates well with the properties of the HVS discussed in Section 2.2.2. The result varies from 1.0 to -1.0, where 1 represents perfect quality and -1 very noticeable distortion. The three comparison stages of the SSIM, as defined in [129], are now considered.

Luminance Comparison

Assume the two images to be compared are two-dimensional matrices $\mathbf{A}$ and $\mathbf{B}$. The mean intensity of each image, μ, is computed as

$$\mu_A = \frac{1}{XY}\sum_{i=1}^{X}\sum_{j=i}^{Y}\mathbf{A}(i,j)$$

$$\mu_B = \frac{1}{XY}\sum_{i=1}^{X}\sum_{j=i}^{Y}\mathbf{B}(i,j) \tag{4.3}$$

where X and Y are the width and height, respectively, of the images in pixels. The luminance comparison function is then defined as

$$l(\mathbf{A},\mathbf{B}) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1} \tag{4.4}$$

where C_1 is calculated as $C_1 = K_1L^2$ in order to avoid instability when $\mu_x^2 + \mu_y^2$ is close to 0. K_1 is a chosen as a small constant and L is the maximum value any pixel in the image can assume. The authors of [129] recommend using a value of 0.01 for K_1 and reports that the SSIM is largely unaffected by changes to this value.

Contrast Comparison

The unbiased standard deviation of each image, σ_A and σ_B, are used as an estimate of contrast. The contrast comparison function is given by

$$c(\mathbf{A},\mathbf{B}) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2} \tag{4.5}$$

with $C_2 = (K_2 L)^2$ and K_2 chosen as 0.03 by the authors of [129]. It was again reported that the technique is fairly insensitive to small variations in the value of K_2.

Structure Comparison

For structure comparison, the mean intensity is subtracted from the pixels in each original image. Each image is then divided by its standard deviation so the two images being compared each have unit standard deviation. For simplicity, we represent $\mathbf{A}$ and $\mathbf{B}$ in one-dimensional forms a and b. The correlation coefficient σ_{AB} is now defined as

$$\sigma_{AB} = \frac{1}{XY - 1} \sum_{i=1}^{XY} \left(\frac{a_i - \mu_A}{\sigma_A} \right) \left(\frac{b_i - \mu_B}{\sigma_B} \right) \tag{4.6}$$

and C_3 as a small constant $C_2/2$. The structure comparison function is then given as

$$s(\mathbf{A}, \mathbf{B}) = \frac{\sigma_{AB} + C_2/2}{\sigma_A \sigma_B + C_2/2}. \tag{4.7}$$

Combining the Comparison Functions

The results from Equations (4.4), (4.5) and (4.7) are then combined to obtain the SSIM as

$$\text{SSIM}(\mathbf{A}, \mathbf{B}) = [l(\mathbf{A}, \mathbf{B})]^\alpha \cdot [c(\mathbf{A}, \mathbf{B})]^\beta \cdot [s(\mathbf{A}, \mathbf{B})]^\gamma \tag{4.8}$$

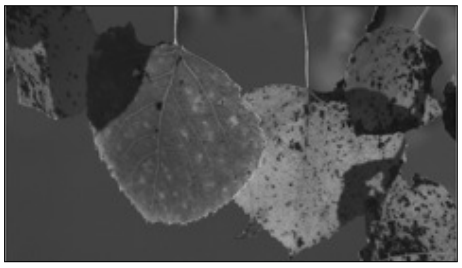
where α, β and γ are used to adjust the weight of each component in the final measurement. Setting $\alpha = \beta = \gamma = 1$, (4.8) simplifies to

$$\text{SSIM}(\mathbf{A}, \mathbf{B}) = \frac{(2\mu_A \mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}. \tag{4.9}$$

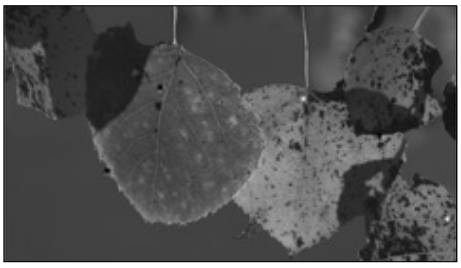
4.2.3 Implementing and Extending the SSIM

Standard Implementation

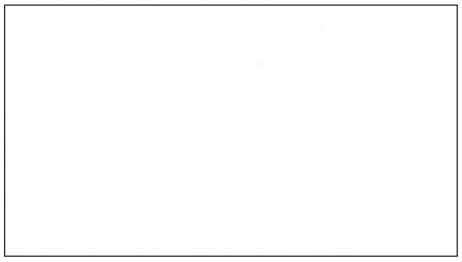
A MATLAB implementation of the SSIM, provided by the authors of [129] at [141], was used to evaluate the SSIM as a potential spatial quality metric. This implementation applies pre-processing and then computes the local SSIM for multiple areas of an image using a sliding window approach. This results in an "SSIM map" indicating the different quality areas in between the images, similar to the JND map discussed in Section 2.2.2 and shown in Figure 2.3 on page 22. The procedure to obtain an SSIM map can be summarised by the following steps:



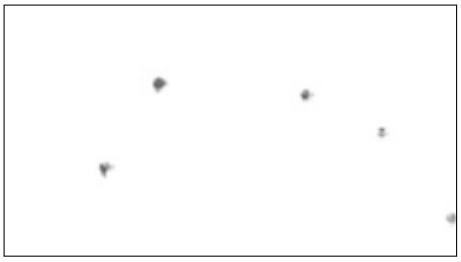
(a) Frame watermarked with the SD technique to obtain a PSNR of 40.09dB.



(b) Frame watermarked with the DWT technique to obtain a PSNR of 40.09dB.



(c) SSIM map for watermarked frame shown in (a).



(d) SSIM map for watermarked frame shown in (b).

Figure 4.3: SSIM maps for the same watermarked images used in Section 4.2.1. Black borders were added to the SSIM maps for visibility. Perfect white areas represent a score of 1.0 which indicates perfect quality, while pure black areas represent a score of -1.0 and indicates severe distortion. Note that the SSIM map successfully identifies areas of distortion in the frame watermarked with the DWT technique.

1. Apply an $F \times F$ averaging filter to both images to be compared;
2. downsample both images by a factor F; and
3. compute the local SSIM with a sliding window to obtain an SSIM map.

The scaling factor F depends on the viewing distance, but the recommended factor is given by $F = \texttt{round}\,(Y/256)$, where Y represents the image height in pixels and the $\texttt{round}\,(\cdot)$ function rounds a value to the nearest integer. The parameters used for the SSIM evaluation are shown in Table A.1 on page 136. SSIM maps obtained for watermarked video sequences are shown in Figure 4.3. Note that the SSIM map successfully identifies areas of distortion in the frame watermarked with the DWT technique.

Extending the SSIM through Pooling

While the SSIM maps in Figure 4.3 successfully identified areas with noticeable distortion where the PSNR did not, it is desirable to obtain a single number representing

the visual quality of each frame for use in objective visual comparisons.

The standard method to obtain such a number from an SSIM map is to calculate the Mean SSIM (MSSIM) value of the SSIM map. If two images **A** and **B** were compared to obtain an SSIM map $\mathbf{C_{AB}}$, the MSSIM would be obtained using

$$\text{MSSIM}(\mathbf{C_{AB}}) = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y}\mathbf{C}_{xy} \tag{4.10}$$

where X and Y represent the width and height respectively of the SSIM map in pixels.

As was the case for the PSNR, the mean value of all the pixels in the SSIM map does not represent highly localised distortions accurately. A localised distortion will receive a low SSIM score while the majority of the image will receive a high score. When calculating the mean SSIM over the whole image, the effect of the localised distortion on the MSSIM is very small, resulting in a high SSIM score. The HVS is sensitive to these localised distortions and will degrade the perceived image quality of the image although the distortion is only in a small region of an image [142].

To mitigate this problem, a so-called *pooling stage* was used to obtain a quality score from the SSIM map of each frame. The idea of pooling is that certain values in a quality map should be weighted heavier than others when calculating a final quality score. Different pooling methods are discussed in [142–144] and pooling was shown to increase the performance of existing metrics, including the SSIM.

In [143] the lowest 6% of values were weighted heavily when calculating the MSSIM while the authors of [145] used the lowest 5%. It was also stated in [143] that the exact weight and percentage of values weighted can be adjusted to suit the specific distortion being evaluated.

Since the distortions caused by the DWT watermarking technique are highly localised, standard image quality metrics were found to be the most inaccurate when estimating the perceived visual quality of frames watermarked with this technique. The artefacts caused by the DWT technique are clusters of approximately 40×40 pixels when watermarking a video frame with a resolution of 1920×1080 pixels. Because these small areas are perceptible, as was demonstrated in Figure 4.1, it is desirable to assign a heavy weight to these pixels during the pooling process. It was decided to pool an area of 40×40 pixels from the original video frame when calculating the MSSIM, which represents the most noticeable distortions in the video frame. Since the SSIM map is a downscaled version of the original image, this results in pooling a 10×10 area of pixels from the SSIM map. When pooling such a small number of pixels, it is desired to assign a heavy weight to these low values. If a

heavy weight is assigned to these low-valued pixels, the lowest 100 values can simply be used and higher values discarded without affecting the MSSIM score significantly. It follows that the lowest 100 values were selected from the SSIM map of a frame, and the pooled MSSIM value was calculated from these.

A simple pooling process similar to those used in [143, 145] was used. Given the SSIM map, the lowest 100 pixel values are selected, after which the mean value is computed.

After implementing the pooling stage, it was possible to obtain a number representative of the perceived image quality, even when distortions were highly localised. Informal subjective visual evaluation suggested that this technique is an effective solution when a simple image quality metric is required. While this technique appears to provide better performance than existing simple image metrics, further subjective tests should be conducted to optimise the performance of this technique.

SSIM scores given in the remaining part of this study refers to an SSIM score obtained using the pooled SSIM metric.

4.3 Proposed Method for Evaluation Interframe Flicker

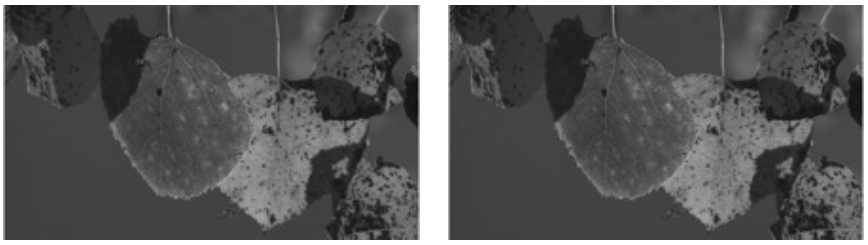
As discussed in Section 2.2.2 on page 20, large changes in pixel intensity values between successive frames can cause a flickering effect when a video sequence is viewed. An example of such a case is shown in Figure 4.4, where a video block of two frames was watermarked with the SD and SVD techniques, respectively, to obtain a mean SSIM value of 0 for each watermarked video block. Comparing Figures 4.4b and 4.4c it is noted that, although the SSIM value is equally low for each sequence, the SVD technique can potentially cause a more prominent flickering effect when the video sequence is viewed. This is because the pixel intensity changes are grouped into blocks of similar intensity changes, whereas the SD technique modifies each pixel independently. This effect needs to be taken into consideration when evaluating interframe flicker.

The aim of this section is to develop a simple and effective method of estimating interframe flicker in video sequences.

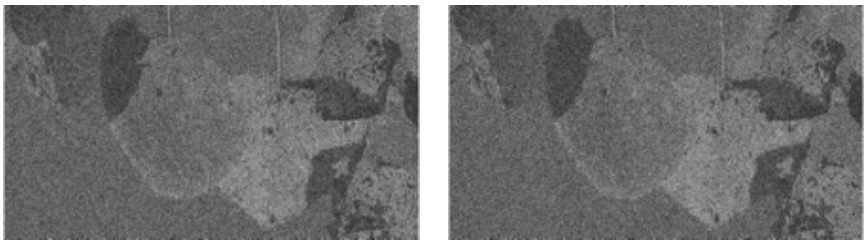
4.3.1 Basic Evaluation

A basic method was developed to calculate the changes caused on a frame-to-frame basis by watermarking techniques.

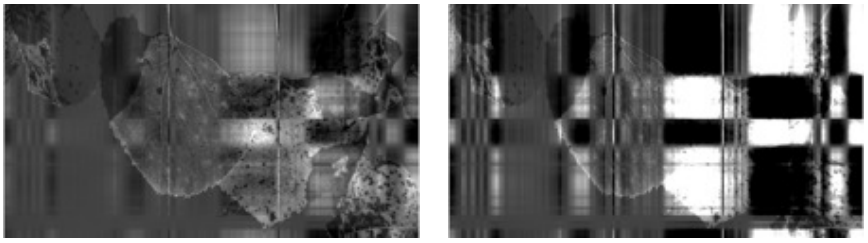
Consider a block of watermarked frames, W, and a block of the corresponding unwatermarked frames, D. For evaluation purposes, two frames were watermarked



(a) Original consecutive frames.



(b) Two consecutive frames watermarked with the SD technique.



(c) Two consecutive frames watermarked with the SVD technique

Figure 4.4: Comparison of interframe flicker caused by different watermarking techniques.

with both the SD and SVD techniques to obtain W_{SD} and W_{SVD}, respectively. In an attempt to evaluate interframe flicker, we calculate $\delta(x, y, z)$ which represents the pixel intensity change between frames that was caused by watermarking

$$\delta(x,y,z) = \underbrace{|D(x,y,z) - D(x,y,z+1)|}_{\text{(a)}} - \underbrace{|W(x,y,z) - W(x,y,z+1)|}_{\text{(b)}} \tag{4.11}$$

with $1 \leq x \leq X$, $1 \leq y \leq Y$ and $1 \leq z < Z$. X and Y represent the width and height of the video frames in pixels, respectively. Z represents the comparison step index, which defines the two frames in the video block to be compared.

Part (a) of (4.11) represents the change between unwatermarked frames in the video block, while part (b) represents the change between the watermarked frames. By subtracting (b) from (a), the interframe change caused by watermarking is obtained. The results of each step in (4.11) are shown in Figure 4.5, enhanced for

visibility. It is observed that this method successfully obtains a difference map indicating the interframe changes caused by watermarking only.

When attempting to obtain a single value representing interframe flicker, caution needs to be exercised. Simply summing the pixel values of δ to obtain a number, F, representing interframe flicker with

$$F = \sum_{x=1}^{X}\sum_{y=1}^{Y}\delta(x, y, 1) \tag{4.12}$$

can be misleading. Recall from Figure 4.4 that the SVD technique causes more interframe flicker than the SD technique. However, the result of (4.12) for the SD technique is a large number while the SVD technique obtains a lower flicker value. Informal tests indicated that the opposite is perceived by the HVS.

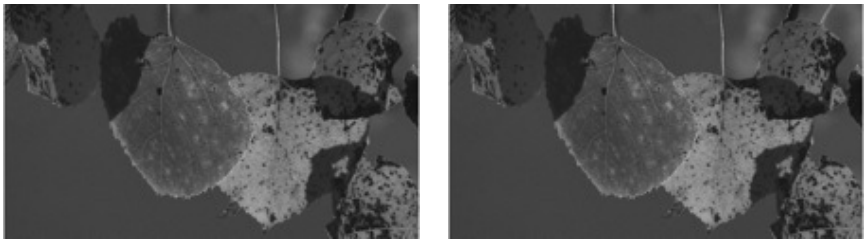
The cause of this problem becomes clear when δ is evaluated using histograms. The histograms for δ_{SD} and δ_{SVD} are shown in Figure 4.6.

The histograms in Figure 4.6 indicate that the SD technique causes a high number of low-intensity changes to the image, while the SVD technique causes changes in a larger intensity range, with the number of pixels modified decreasing as the intensity of the modifications increase. Although a small number of pixels is modified with high intensity by the SVD, these changes are more visible than the high number of low intensity changes caused by the SD technique [142]. The summation process in (4.12) incorrectly produces a high flicker score when a high number of low intensity changes are made, while a small number of high intensity changes results in a lower flicker score.

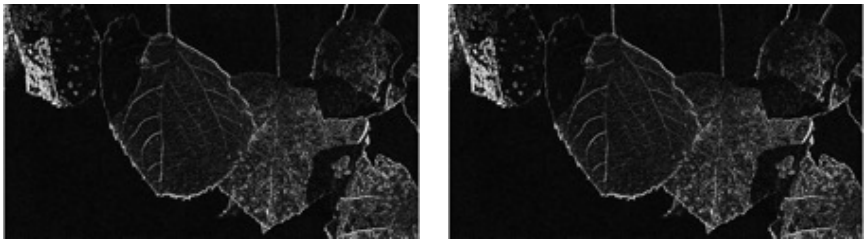
4.3.2 Extending the Basic Method

The basic method developed in Section 4.3.1 is now extended to more effectively evaluate interframe flicker caused by watermarking. This involves identifying areas where pixels are grouped together, creating a corresponding image mask, and then only summing the values identified by the image mask. The concept of applying a mask to exclude certain results from calculation is similar to the attention map discussed in Section 2.2.2 on page 20 and shown in Figure 2.4 on page 23. The proposed method of obtaining this mask to be applied is now detailed.

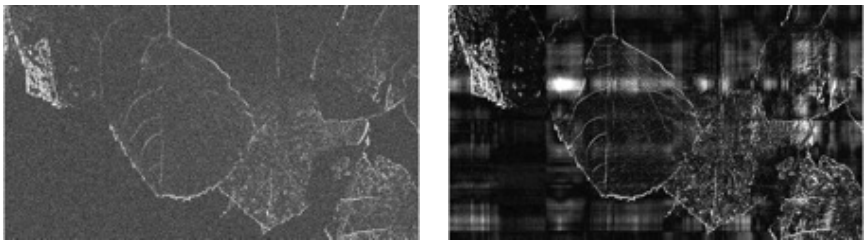
We begin the mask creation process by again using (4.11) to obtain δ. Edge detection is then performed on each frame on δ, followed by a two-dimensional averaging filter on each frame to obtain δ'. This edge detection and smoothing process identifies the outlines of pixel clusters that represent high intensity changes, while ignoring more noisy areas.



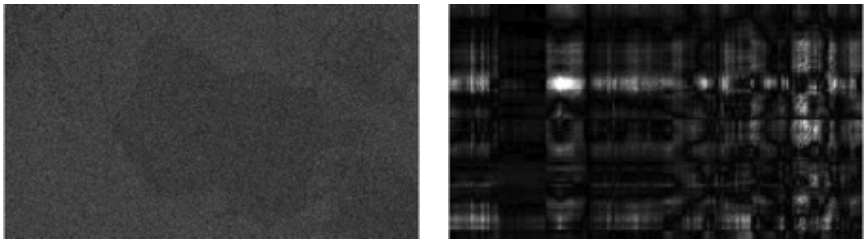
(a) Frames watermarked with the SD and SVD techniques, respectively.



(b) Frame difference computed from (a) in Equation 4.11. For visibility, all pixel values were multiplied by ten.



(c) Frame difference caused by watermarking, computed from (b) in Equation 4.11. For visibility, all pixel values were multiplied by ten.



(d) Difference caused by watermarking between two frames for the SD and SVD technique respectively, obtained from Equation 4.11. For visibility, all pixel values were multiplied by ten.

Figure 4.5: Results obtained from using (4.11) to evaluate the interframe flicker caused by watermarking. It is seen that this method successfully obtains a difference map indicating the interframe changes caused by watermarking only.

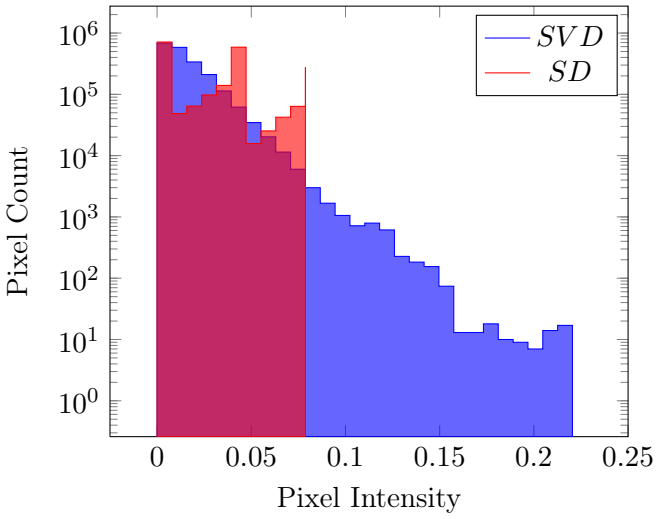


Figure 4.6: Histogram showing the intensities of interframe pixel value changes for the SD and SVD techniques.

The Matlab function `imfill` is then used to fill the insides of the identified clusters, in the image. The `imfill` function is used with the `holes` mode specified. This fills the insides of dark areas in an image which are surrounded by lighter pixels in such a way that it cannot be reached when the background of the image is filled from the edges of the frame.

In order to remove excess noise caused by the filling operation, a two-dimensional median filter is applied to δ' after the filling operation, to obtain δ''. δ'' is then converted to a binary image by setting any value larger than 0 to 1 with

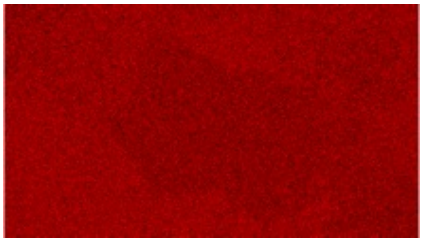
$$\delta'' = \begin{cases} 0 & \text{if} \quad \delta'' = 0 \\ 1 & \text{if} \quad \delta'' > 0 \end{cases}. \tag{4.13}$$

The mask δ'' is applied to the original image containing the interframe difference, δ, after which the pixel values in each frame are summed:

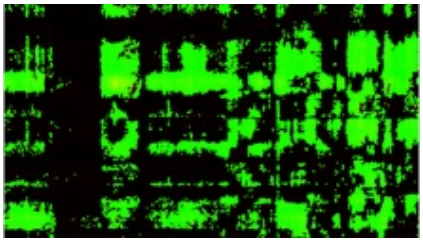
$$\Delta(Z) = \sum_{i=1}^{X}\sum_{j=1}^{Y}\delta(i,j,Z)\,\delta''(i,j,Z).$$

After the masking and summation operation we obtain Δ, which is a value representing the interframe flicker between two frames in a video block. The process can be summarised by the following steps:

1. Obtain interframe flicker, δ;
2. detect edges of pixel clusters;



(a) Mask created for flicker caused by the SD technique.



(b) Mask created for flicker caused by the SVD technique.

Figure 4.7: Results of the proposed mask creation process. Red areas indicate areas that were not selected for summation, while areas in green indicate areas that were selected. Note that the method successfully chooses the areas of grouped pixels caused by the SVD technique, while it does not select any pixels from the changes caused by the SD technique.

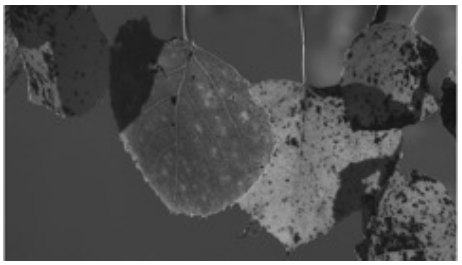
3. apply an averaging filter;
4. fill the insides of pixel clusters using `imfill`;
5. apply a median filter;
6. convert the result to a binary mask; and
7. sum the pixel values of the results.

The masks created by the process is shown in Figure 4.7. Red regions indicate areas that were not selected for summation, while regions in green indicate areas that were selected. Note that the method successfully chooses the areas of grouped pixels caused by the SVD technique, while it does not select any pixels from the changes caused by the SD technique. Informal subjective tests indicated that this technique provides a fair representation of the perceived interframe flicker. Further subjective testing can be done to further optimise the performance of this technique.

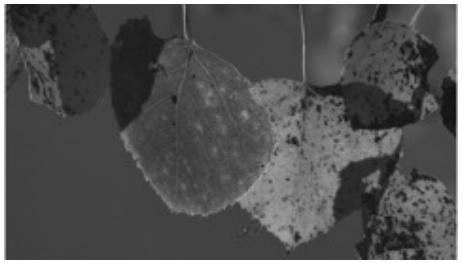
The parameters used for the evaluation in this project are shown in Table A.2 on page 136.

4.4 Visual Comparison of Techniques

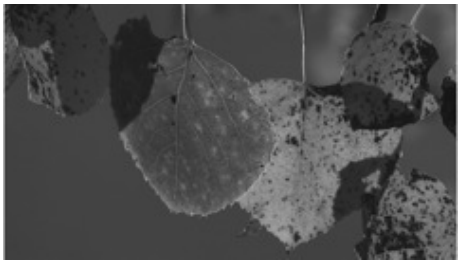
For a fair comparison of the artefacts caused by each watermarking technique, it was desirable that the video sequences watermarked by the various techniques be of similar visual quality. To achieve this, the visual distortion caused by each technique was objectively evaluated using the techniques developed in the previous sections of this chapter. The parameters of each technique were adjusted to obtain an MSSIM



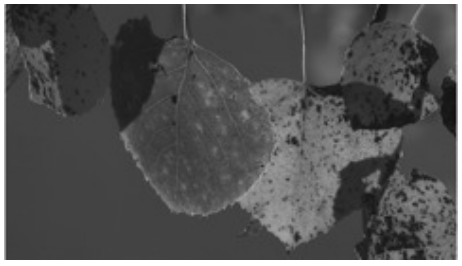
(a) Original frame.



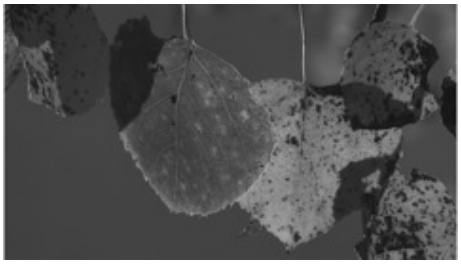
(b) Frame watermarked with the SD technique.



(c) Frame watermarked with the DFT technique.



(d) Frame watermarked with the SVD technique.



(e) Frame watermarked with the DWT technique.

Figure 4.8: Frames watermarked to obtain an SSIM value of 0.999 in each case.

value of 0.999 over two video sequences. The parameters used for embedding the watermarks are shown in Table A.4 on page 137. The artefacts caused by each watermarking technique are now examined in the following sections.

4.4.1 Spatial Comparison of Techniques

The distortions caused by each technique at high watermarking intensities were evaluated in Section 4.1. The two sequences were next imperceptibly watermarked with each technique to obtain a mean SSIM score of approximately 0.999 across the sequences. The results are shown in Figure 4.8.

In order to evaluate the artefacts caused by each technique at very low amplitudes, the difference between the watermarked frame and unwatermaked frame was calculated in each case and multiplied for visibility. Mathematically this can be

expressed as

$$\phi(x, y, z) = 200|U(x, y, z) - W(x, y, z)| \tag{4.14}$$

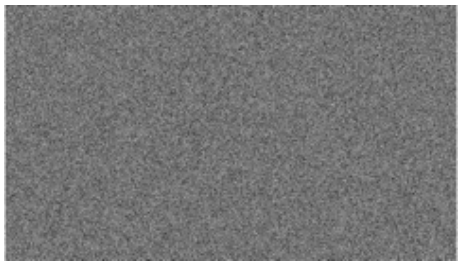
with $1 \leq x \leq X$, $1 \leq y \leq Y$ and z the index of the frame being evaluated. The results are shown in Figure 4.9, where a pure black area represents no change and pure white represents a large change in pixel intensity. The histograms of each of the difference frames in Figure 4.9 are shown in Figure 4.10.

The white noise added by the SD technique is clearly seen in Figure 4.9a. Through inspection of this frame alone, one is not able to identify the nature of the original frame. This is because the technique is not content aware and it will embed noise of the same intensity independent of the content of the original frame. In contrast to this, the nature of the original frame can be identified from the frame watermarked by the DFT technique, shown in Figure 4.9b. The DFT technique is host-adaptive, and makes larger changes in the image where mid-frequencies appear in the frame. Furthermore, the technique also makes larger changes to moving objects, in this case the leaves moving in the wind.

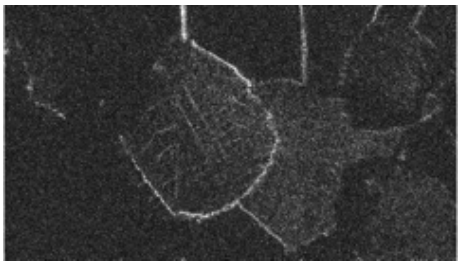
In the case of the SVD technique, the original frame is again not identifiable from Figure 4.9c, which was watermarked by the SVD technique. Figure 4.9d shows the result from the DWT technique and also conveys little information about the original content in the video. It is however possible to say that there are areas with high texture details in the areas where the white spots appear in the frame.

Histograms of the visual artefacts are shown in Figure 4.10, while SSIM maps for the watermarked frames are shown in Figure 4.11. Comparing these figures provides further insights into the natures of the techniques. Note in Figure 4.10a that the SD technique changes a relatively high number of pixels uniformly distributed between two intensity values. Recall that this is done independently of the host content. In the SSIM map of the SD technique in Figure 4.11a, the noise is less visible in the textured areas of the image, which in this case contains the leaves. A content-aware technique would realise that a larger embedding strength can be used in the textured areas of the image. An example of this is the DFT technique, shown in Figures 4.10b and 4.11b. Note that the edges of the leaves are less pronounced in the SSIM map. This is because the DFT technique makes larger changes to moving objects and edges. In the histogram in Figure 4.10b, note that, by using this approach, the DFT technique is able to make higher intensity changes to the frame while maintaining the same imperceptibility as the other techniques.

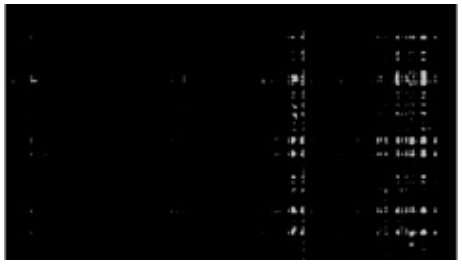
A potential weakness of the SVD technique becomes evident from Figure 4.10c, a large number of pixels are changed by a small amount, with the number of pixels modified decreasing rapidly as the intensity increases. While at first inspection of



(a) Differences between a watermarked and unwatermaked frame for the SD technique.



(b) Differences between a watermarked and unwatermaked frame for the DFT technique.



(c) Differences between a watermarked and unwatermaked frame for the SVD technique.

(d) Differences between a watermarked and unwatermaked frame for the DWT technique.

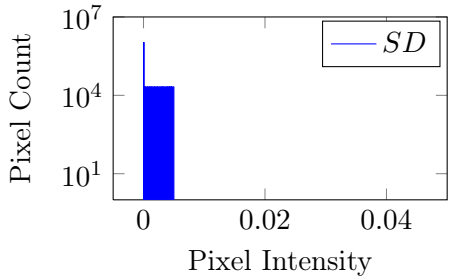
Figure 4.9: Illustration of differences between the watermarked frames and unwatermaked frames for each watermarking technique. For visibility, the results were multiplied by 200 in each case.

the histogram, it might appear that this watermark would be less perceptible than the other techniques, this is not the case. The fact that the artefacts caused by the SVD technique are grouped together in lines or blocks, make them visible even at very low amplitudes. While only small portions of the image are modified, these artefacts will become visible if the amplitude of the watermark is increased.

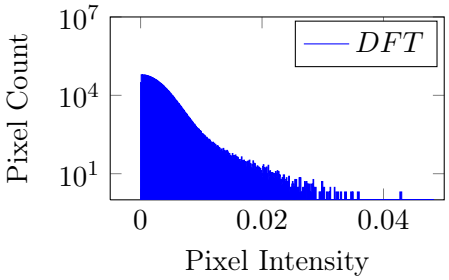
While the DWT also only modifies small portions of the image, as seen in Figures 4.10d and 4.11d, it is able to use higher intensities than the SVD technique while maintaining low perceptibility. This follows from the fact that the DWT technique selects areas of the image with high texture detail to embed the watermark, where the eye is less sensitive to these distortions.

4.4.2 Temporal Comparison of Techniques

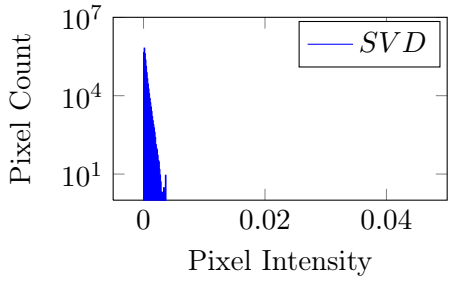
Video adds the temporal dimension which is not present in image watermarking. While this added dimension provides more data in which to embed a watermark, interframe quality and flicker are important considerations. The temporal stability of the techniques were compared in two ways. The SSIM of each frame was firstly computed and graphed. As a final evaluation technique, the interframe flicker was



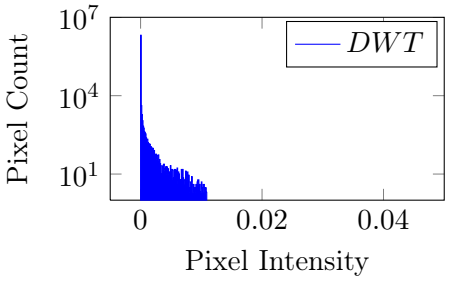
(a) Histogram for results from the SD technique.



(b) Histogram for results from the DFT technique.



(c) Histogram for results from the SVD technique.



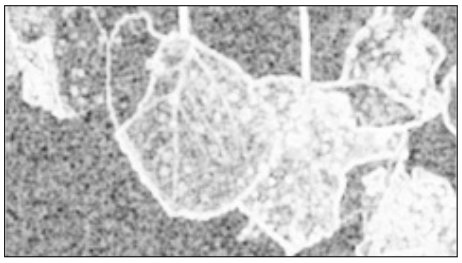
(d) Histogram for results from the DWT technique.

Figure 4.10: Histograms of artefacts caused by each watermarking technique. From these it is possible to evaluate the number of pixels that are modified by each technique, and at which intensity.

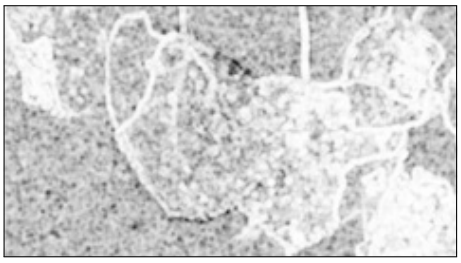
computed and graphed across complete watermarked video blocks using the flicker evaluation technique developed in Section 4.3.

Figure 4.12 shows the SSIM score obtained across the two video sequences for each technique. Note that the SD and DFT techniques in Figures 4.12a and 4.12b are the most stable, with the SSIM value varying only slightly as the contents of the frames change. The DWT technique in Figure 4.12d is slightly less stable, and the SVD in Figure 4.12c is the most unstable with the SSIM varying as the contents of the frames change. Instability makes it difficult to ensure that all frames are watermarked imperceptibly and it may be necessary to obtain parameters specific to the video content in a scene. The DFT and SD techniques are advantageous in this regard, since it is not necessary to change the parameters of the technique as the contents of the frames change.

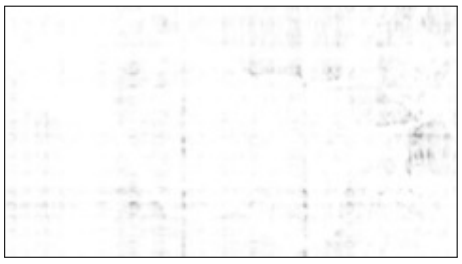
The interframe flicker for watermarked video sequences was calculated using the technique developed in Section 4.3 on page 49 and the results are shown in Figure 4.13. The SD and DWT techniques are stable and show low interframe flicker. Note that the SVD technique exhibits low flicker for the first video sequence, with



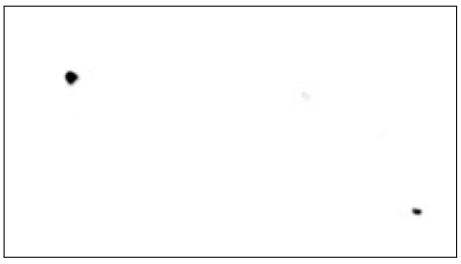
(a) SSIM map for frame watermarked with the SD technique.



(b) SSIM map for frame watermarked with the DFT technique.



(c) SSIM map for frame watermarked with the SVD technique.



(d) SSIM map for frame watermarked with the DWT technique.

Figure 4.11: SSIM maps for frames watermarked with each watermarking technique. All values were multiplied by 200 for visibility.

more flicker for the next, making it once again unstable and difficult to obtain parameters suitable for watermarking a long sequence of frames. The DFT technique also displayed a moderate amount of flicker, but is slightly more predictable than the SVD technique.

The nature of the interframe flicker when watermarking the two video sequences is shown in Figures 4.14 and 4.15. The first scene is `Aspen_8bit.avi` with moving leaves. Note that the DFT technique in Figures 4.14c and 4.15c makes changes to moving objects such as the leaves and scrolling text. This is why it scores higher interframe flicker values, but embedding in moving parts of the sequence is desirable as it is less perceptible to the human eye. It is once again seen in Figures 4.14b and 4.15b that the SD technique is not affected by the image content. The SVD technique exhibits low flicker in Figure 4.14d and a high flicker in Figure 4.15d. The DWT technique in Figures 4.14e and 4.15e causes only small interframe flicker and follows the textured areas in the scene.

Table 4.1: Summary of spatial visual artefacts produced by different watermarking techniques.

Technique	Nature of distortion	
	Appearance	Location
SD	White noise	Whole frame
DFT	Medium frequency noise	More intense in moving areas and mid-frequency sections
SVD	Blocks of intensity changes	Localised
DWT	Small, bullet-hole-like distortions	Highly localised

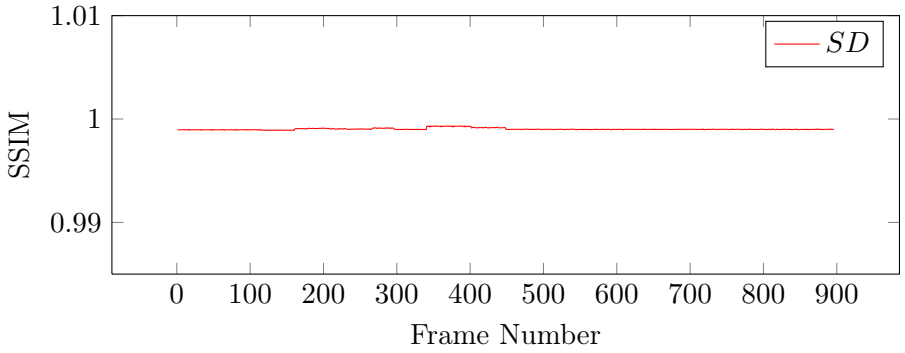
Table 4.2: Summary of temporal and spatial stability properties of artefacts caused by watermarking techniques.

Technique	Stability		Flicker
	Spatial	Temporal	
SD	High	High	Low
DFT	Medium	Medium	Medium
SVD	Low	Low	Low/High mix
DWT	Medium	High	Low

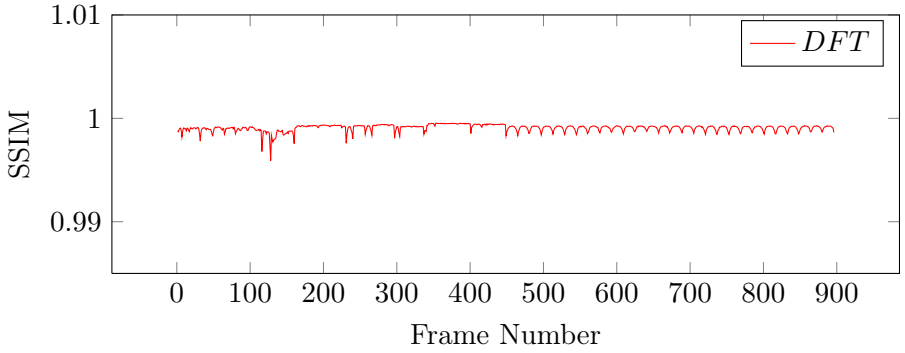
4.5 Chapter Summary

In this chapter, existing spatial domain techniques were evaluated and found to be insufficient as visual quality metrics when comparing these watermarking implementations. The SSIM was extended using pooling to improve the performance of the SSIM when comparing these different watermarking techniques. A novel temporal flicker metric was also developed to easily evaluate interframe flicker in watermarked video sequences.

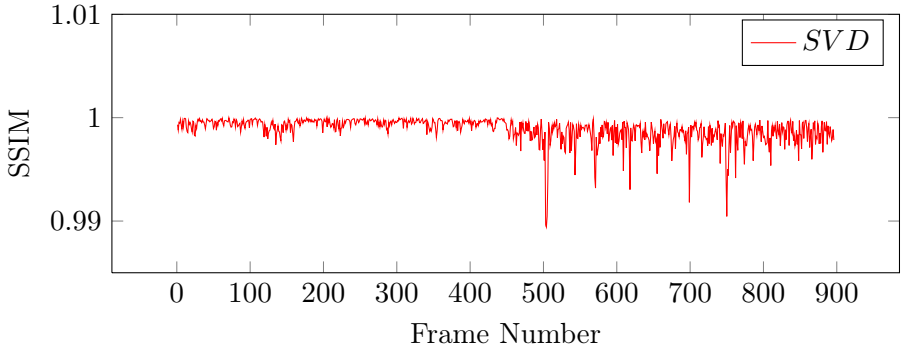
Using the techniques developed, parameters were computed to achieve the same perceived visual quality for sequences watermarked using the different watermarking techniques. The visual artefacts produced by each watermarking technique were then evaluated. The results from the evaluation are summarised in Tables 4.1 and 4.2.



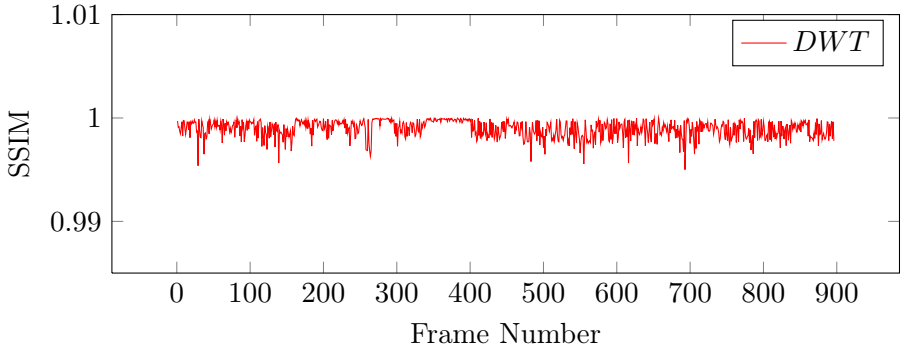
(a) SSIM values of frames watermarked with the SD technique.



(b) SSIM values of frames watermarked with the DFT technique.

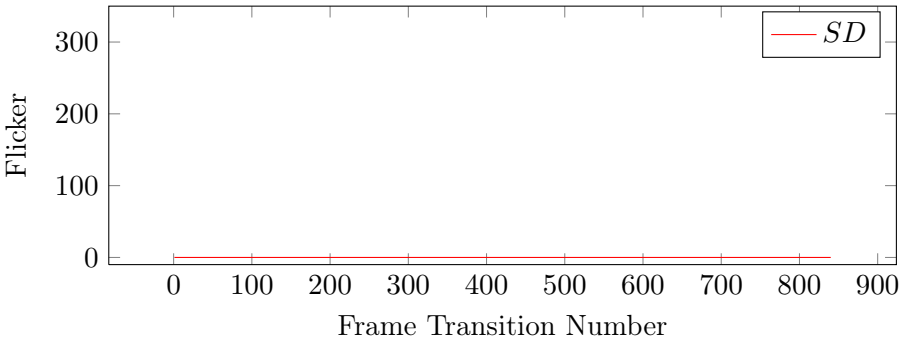


(c) SSIM values of frames watermarked with the SVD technique.

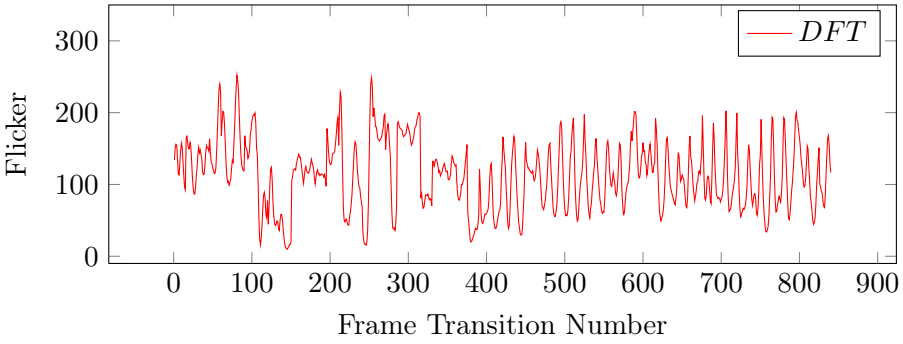


(d) SSIM values of frames watermarked with the DWT technique.

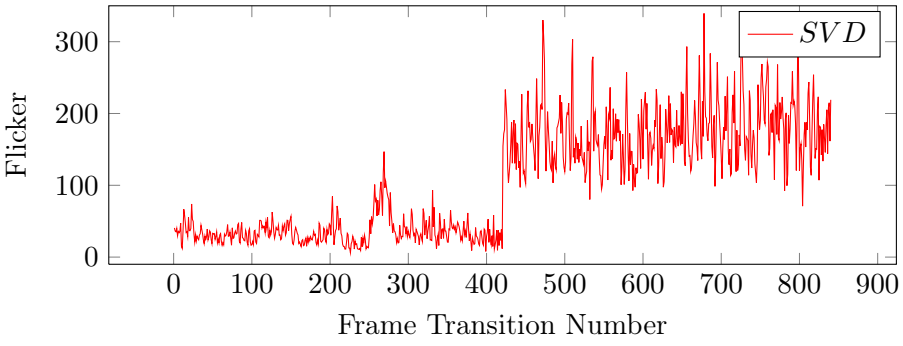
Figure 4.12: SSIM values computed across complete watermarked video block for each technique in order to compare temporal stability of the techniques.



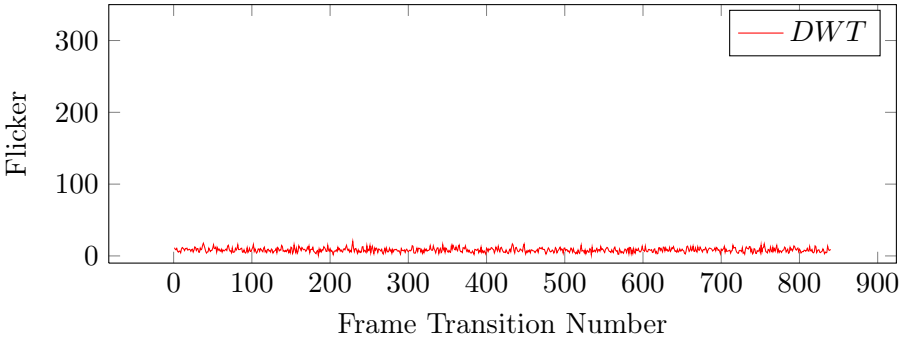
(a) Interframe flicker values between frames watermarked with the SD technique.



(b) Interframe flicker values between frames watermarked with the DFT technique.

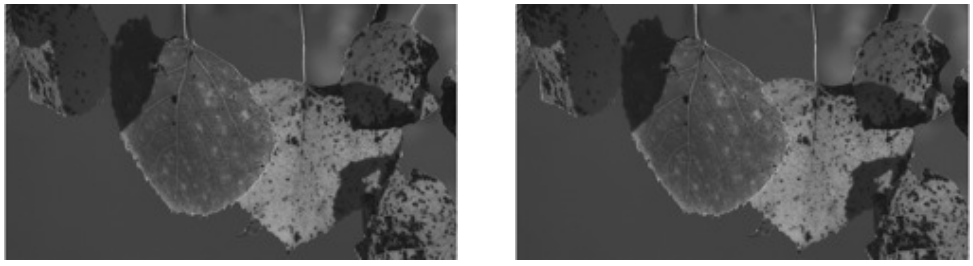


(c) Interframe flicker values between frames watermarked with the SVD technique.

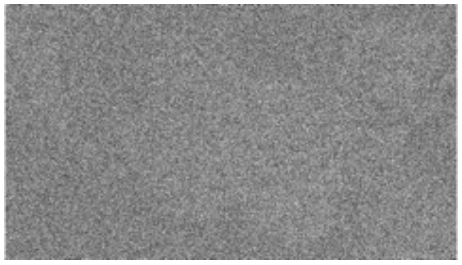


(d) Interframe flicker values between frames watermarked with the DWT technique.

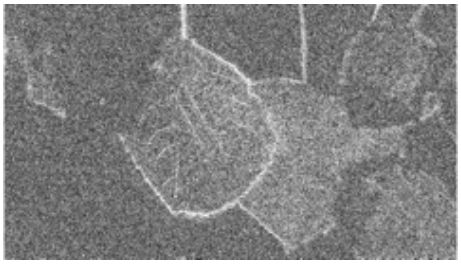
Figure 4.13: Interframe flicker values for frames watermarked to obtain an SSIM value of 0.999 in each case.



(a) Original consecutive frames

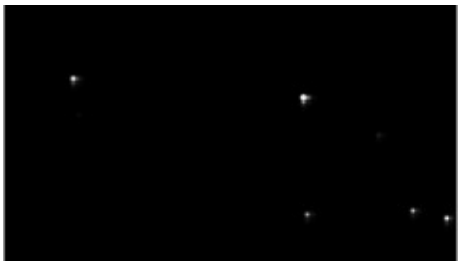


(b) Two consecutive frames watermarked with the SD technique



(c) Two consecutive frames watermarked with the DFT technique

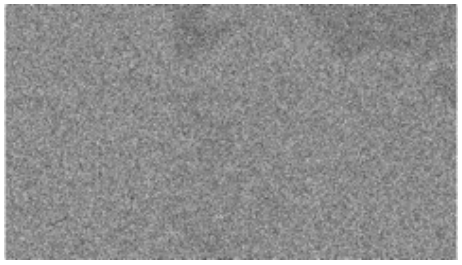
(d) Two consecutive frames watermarked with the SVD technique



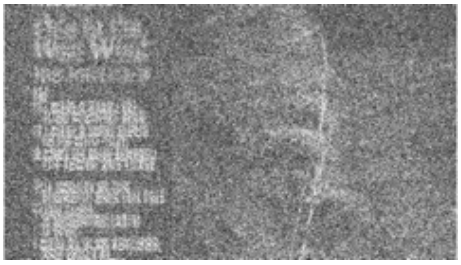
(e) Two consecutive frames watermarked with the DWT technique

Figure 4.14: Nature of the interframe flicker when watermarking the video sequence `Aspen_8bit.avi`.

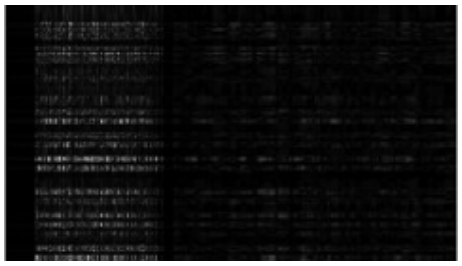
(a) Original consecutive frames



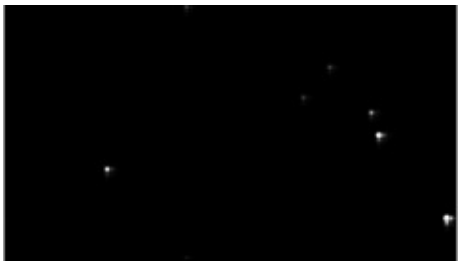
(b) Two consecutive frames watermarked with the SD technique



(c) Two consecutive frames watermarked with the DFT technique



(d) Two consecutive frames watermarked with the SVD technique



(e) Two consecutive frames watermarked with the DWT technique

Figure 4.15: Nature of the interframe flicker when watermarking the video sequence `WestwindEasy_8bit.avi`.

Chapter 5

Performance Evaluation of Techniques

The performance of the implemented watermarking techniques are now evaluated. Performance parameters evaluated include computational complexity, robustness against normal signal transformations as well as robustness against intentional attacks. Before discussing the results obtained from the evaluation, the configuration of the testing procedure used is briefly outlined.

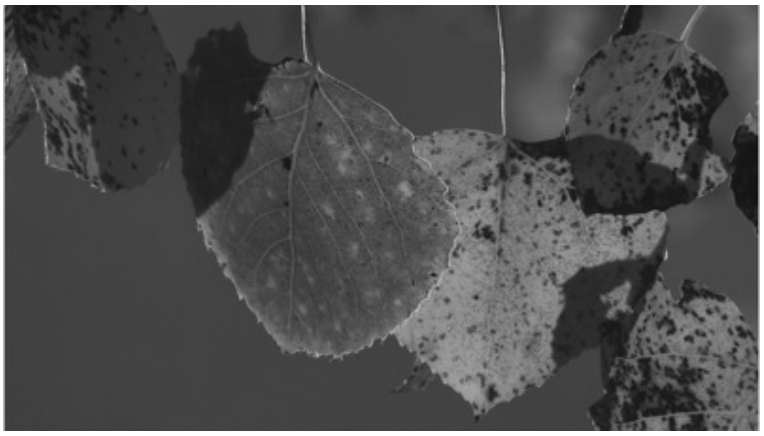
5.1 Testing Procedure and Configuration

5.1.1 Choice of Attacks

A number of common signal manipulations were chosen against which to evaluate the selected watermarking techniques. These were chosen to represent common basic attacks and signal manipulations experienced in the media distribution chain. Most of these attacks were discussed in Section 2.1.5 on page 12 and include rotation, denoising and cropping of the video frames, while common signal manipulations include quantisation and resizing. For further reading, refer to [12] and [146] for overviews of common methods to affect the successful extraction of watermarks.

5.1.2 Test Configuration

The evaluation system and the respective watermarking techniques were implemented in Mathworks MATLAB R2011a on the hardware defined in Table A.3 on page 137. Test sequences used were retrieved from the Xiph.org Test Media Repository [147]. Sequences used were `Apen_8bit.avi` and `WestwindEasy_8bit.avi`. Each sequence was 19 seconds in length with a frame rate of 30 frames per second,



(a) Frame 61 of `Aspen_8bit.avi`.

(b) Frame 61 of `WestwindEasy_8bit.avi`.

Figure 5.1: Example unwatermarked video frames from the two test sequences that were used for performance evaluation of watermarking techniques.

resulting in 570 frames per sequence. Each had a resolution of 1920×1080 pixels with 8-bit 4:2:2 chroma sub-sampling.

As per the instructions provided by [147], the first and final two seconds of each sequence were discarded resulting in 510 frames per sequence to be used for testing. Frames 61 to 509 of each sequence were grouped into video blocks of 16 frames each and converted to the YC_bC_r colour space. Eighty information bits were then embedded into the luminance (Y) component of each video block using the watermarking parameters in Table A.4 on page 137. This resulted in five bits effectively being embedded per frame of the video block. Unwatermarked video frames from `Aspen_8bit.avi` and `WestwindEasy_8bit.avi` are shown in Figures 5.1a and 5.1b respectively. Compressed versions of these video files can be viewed at [114] for evaluation purposes.

5.1.3 Bit Error Rate

Various attacks were applied to each watermarked video block, the watermark extracted and the bit error rate calculated. The bit error rate of the extracted watermark was calculated in each case as

$$\text{BER} = \frac{\text{Number of bits incorrectly extracted}}{\text{Number of bits extracted}}. \tag{5.1}$$

Since reversal of bit polarity is trivial to detect, the BER was adjusted with

$$\text{BER} = \begin{cases} \text{BER} & \text{if} \quad \text{BER} \leq 0.5 \\ 1 - \text{BER} & \text{if} \quad \text{BER} > 0.5 \end{cases}. \tag{5.2}$$

The resulting BER was recorded for each attack with varying parameters.

5.2 Computational Complexity Evaluation

The importance of the computational complexity of a watermarking technique varies between applications, but in general, techniques with low computational complexities are desirable [3]. The computational complexity of the different watermarking techniques were estimated by measuring the elapsed time for the embedding and extraction stage of each technique. The time required to execute each mathematical transform required by the different techniques were also evaluated. By comparing the execution times of the watermarking techniques to those of only the mathematical transforms required by each technique, it is possible to see whether more time is spent performing the transform or whether it is spent performing other operations in the watermarking technique implementation. Although the current implementations may not be optimally efficient, it does provide an approximate indication of computational complexity.

Method

To compare the execution speed of different software implementations, it is recommended to use the Matlab stopwatch timer functions, `tic` and `toc` [148]. These were used to measure the elapsed time to execute the watermarking implementations, using the hardware configuration shown in Table A.3 on page 137. No background tasks were run by the user during the tests.

The necessary video frames were already loaded into memory and converted to the $\text{YC}_\text{b}\text{C}_\text{r}$ colour space before the timer was started. Each embedding and extraction function was called and only the time to execute the function was taken into account.

To evaluate the computational complexities of the mathematical transforms required by the techniques, only the transforms were applied to video frames and the execution time noted in each case.

As an example, the DFT embedding and extraction functions were used to embed and extract a watermark from a video block containing 16 video frames and the execution time noted. To evaluate the computational complexity of only the DFT itself, a DFT was applied to 16 video frames and the execution time was noted in each case. Each evaluation was run fifteen times, after which the mean value was calculated for each technique.

Results

The computational complexities of the embedding and extraction phases of each technique are shown in Figure 5.2. Figure 5.3 illustrates the time required to apply only the mathematical transforms required by each technique on the block of video frames, without any further operations.

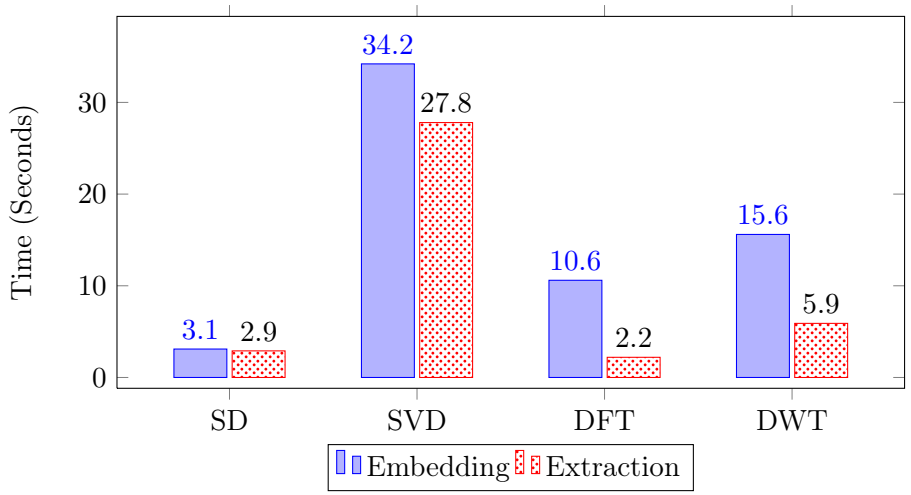


Figure 5.2: Comparison of the time required to watermark 16 video frames with 80 bits of information, using different techniques.

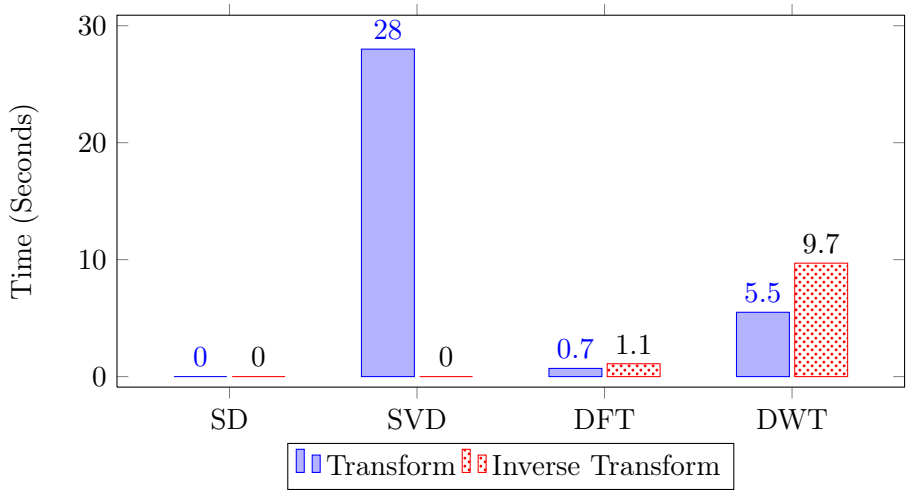


Figure 5.3: Comparison of the time required to apply only the mathematical transforms required by each watermarking technique on a block of 16 video frames, without any further operations.

Discussion

It is seen that the implementation of the SVD watermarking technique is computationally relatively complex, while the DFT technique is relatively efficient. The SD technique, which requires no transform, is the most efficient.

Comparing Figures 5.2 and 5.3, it is noted that the DFT as a transform alone is efficient, but the DFT watermarking technique requires other operations which slightly reduces the computational efficiency of the watermarking technique. The same is found in the case of the SD technique. While the SD watermarking technique does not require mathematical transforms, such as an SVD or DFT, other operations required by the technique add to the computational complexity and execution time of the watermarking technique. In the case of the SVD watermarking technique, note that the majority of the execution time is devoted to performing the SVD itself.

For real-time applications, even though a more efficient DFT technique may be suitable for real-time processing, a minimum latency of 16 frames will still be required, as this technique requires a complete block of video frames to embed the watermark. The other techniques are able to embed the watermark systematically, using one frame at a time to obtain a watermarked video block.

5.3 Spatial Desynchronisation Evaluation

Desynchronisation occurs when the spatial or temporal coordinates of an embedded watermark are changed. When this occurs, the watermark detector may not be able

to successfully detect or extract a watermark, even when one is present [149].

Desynchronisation is not only caused by malicious attacks, but can also occur due to common signal processing during broadcasting or recording by end-user devices. Video frames may also contain borders that do not contain valid picture content, which may get discarded by end-user or transmission systems [127]. Borders may also be lost or added due to conversion of the video for display on systems with different aspect ratio displays [150]. For these reasons it is important for a watermarking technique to be robust against spatial desynchronisation.

5.3.1 Spatial Shifting of Video Frames

Spatial shift of an image can be caused by intentional attacks, or devices such as analog video recorders which cause small shifts in a few pixels of a video frame [146]. In these cases, the watermark extractor may be unable to successfully synchronise with the watermark and be unable to successfully extract the information bits.

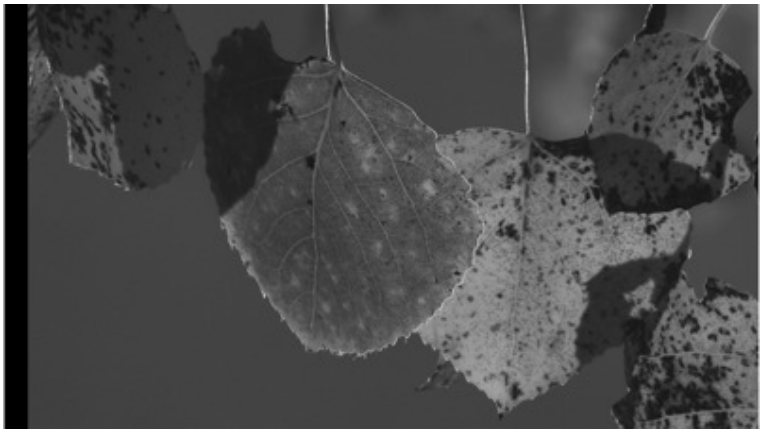
Method

To evaluate spatial desynchronisation robustness, the contents of each video frame were first shifted in the positive x-and y-directions, respectively. This resulted in the contents moving to the right or downwards in the video frame. The watermark was then extracted and the BER calculated in each case. The image was then shifted in both the x and y direction simultaneously, resulting in a diagonal shift of the image in the video frame. The watermark was again extracted and the BER calculated in each case. Blank pixels were replaced with black pixels in all cases.

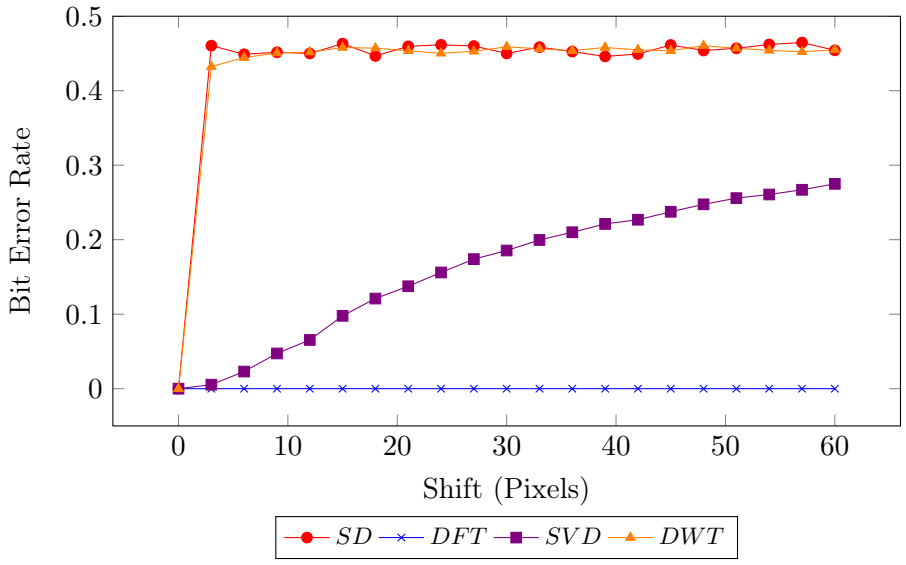
Results

The results of shifting video frames in the positive x and y directions are shown in Figures 5.4 and 5.5 respectively. Each shows an example video frame and a graph indicating the resulting BER for varying amounts of shift. The result of the 60 pixel simultaneous x and y shift is shown in Figure 5.6. Each video frame was shifted by a maximum amount of 60 pixels in each case.

X Shift:



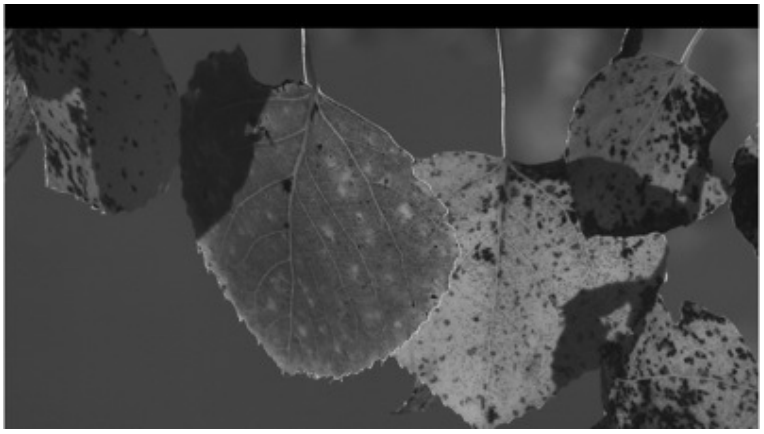
(a) An example of a video frame shifted by 60 pixels in the positive x direction.



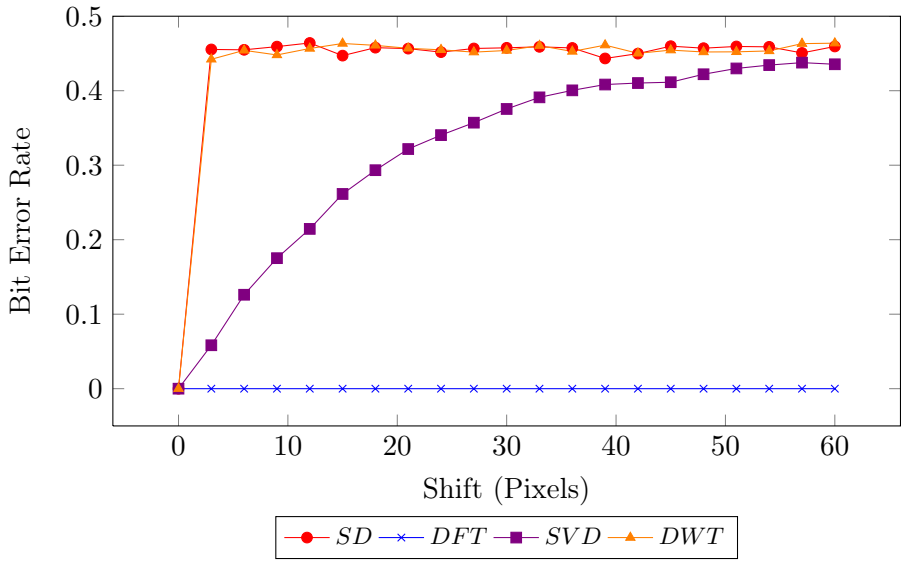
(b) Bit Error Rates of extracted information bits by each watermarking technique for varying amounts of pixel shift in the positive x direction.

Figure 5.4: Results of video frames shifted in the positive x direction.

Y Shift:



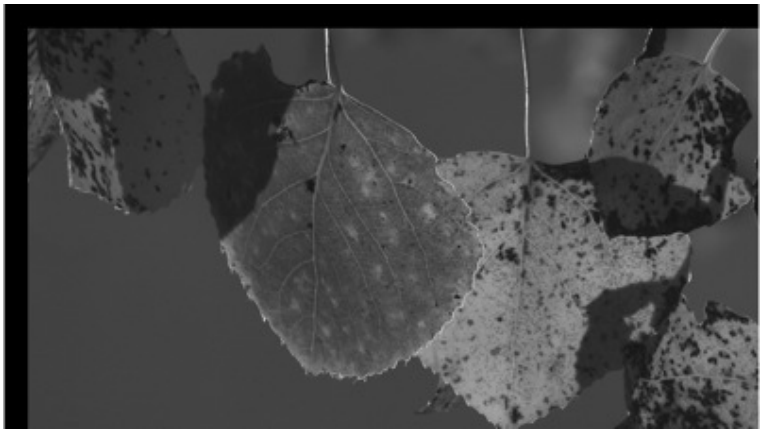
(a) An example of a video frame shifted by 60 pixels in the positive y direction.



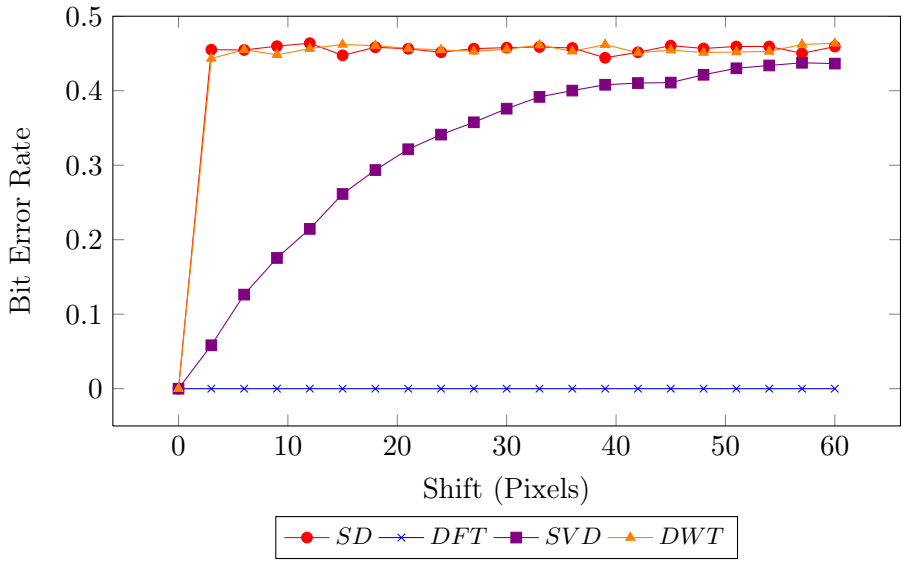
(b) Bit Error Rates of extracted information bits by each watermarking technique for varying amounts of pixel shift in the positive y direction.

Figure 5.5: Results of video frames shifted in the positive y direction.

XY Shift



(a) An example of a video frame shifted by 60 pixels in the positive x and y direction simultaneously.



(b) Bit Error Rates of extracted information bits by each watermarking technique for varying amounts of pixel shift in the positive x and y directions simultaneously.

Figure 5.6: Results of video frames shifted in the positive x and y directions simultaneously.

Discussion

Spatial shifts pose a challenge for the SD and DWT techniques. Correlation-based techniques such as the SD technique are particularly susceptible to desynchronisation because they require near-perfect synchronisation with the watermark signal [151]. The DWT technique also fails, because, although the pseudo-random number generator F generates the same DWT coefficient numbers as during the embedding stage, these coefficients now contain different, possibly unwatermarked, image data. If the coefficient selected contains a high amount of texture detail, this coefficient will still be used for extraction of a watermark bit.

The SVD technique is moderately robust against desynchronisation. This is because the singular values used in the SVD technique each represent rank approximations of the image. Because the watermark is embedded in the first few approximations of the video frame, the spatial shift does not affect these approximations severely. Note that, by embedding data in the smaller singular values, the technique will become more susceptible to spatial shift and other changes in detail.

In contrast to the other techniques, the DFT technique is unaffected by spatial shifts. This can be attributed to the fact that a shift in the spatial domain translates into a phase shift in the frequency domain of the DFT. The watermark is embedded only in the magnitude of the frequency components, which allows the DFT technique to be robust against a spatial shift.

This evaluation also revealed that a spatial shift in the y-direction is almost identically as effective as an attack where pixels are shifted in the x-and y-directions simultaneously.

5.3.2 Rotation

Spatial desynchronisation can also be caused by rotation. Small amounts of rotation may not affect the perceived quality of the video content, but may disable the watermark extraction techniques.

Method

Varying amounts of rotation were applied to each video frame watermarked by each technique, the watermark extracted and the BER of the extracted information bits calculated. Each video frame was rotated around the center of the video frame, otherwise known as on-axis rotation. Off-axis rotation can also occur, where the content is not rotated around the center of the frame. While the implemented watermarking

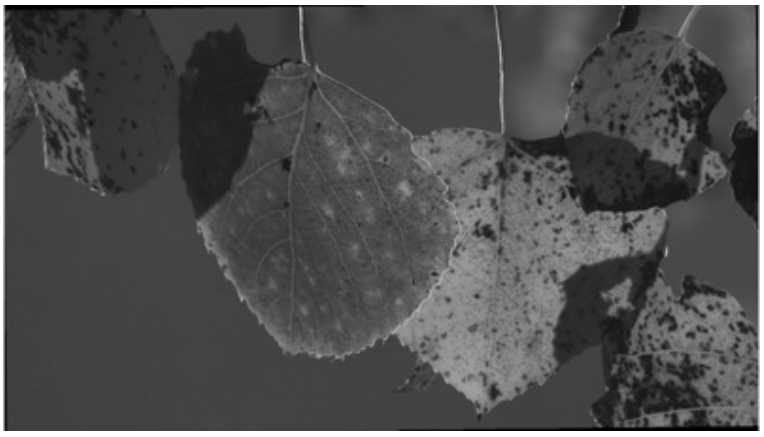
techniques do not claim to be robust against rotation, it is important to evaluate both on-and-off-axis rotation robustness. Off-axis rotation was informally evaluated and was found to yield similar results to on-axis rotation. Since the results are similar, and these watermarking techniques do not claim to be robust against any form of rotation, only on-axis rotation was evaluated in-depth.

Two types of on-axis rotation were evaluated, namely cropped rotation and loose rotation. In the case of cropped rotation the video frame was rotated, after which the resulting images were cropped to fit within the original size of the video frames. In the case of loose rotation the image was rotated and the resulting image resized to fit within the dimensions of the original video frames. No parts of the image were discarded as was done for cropped rotation.

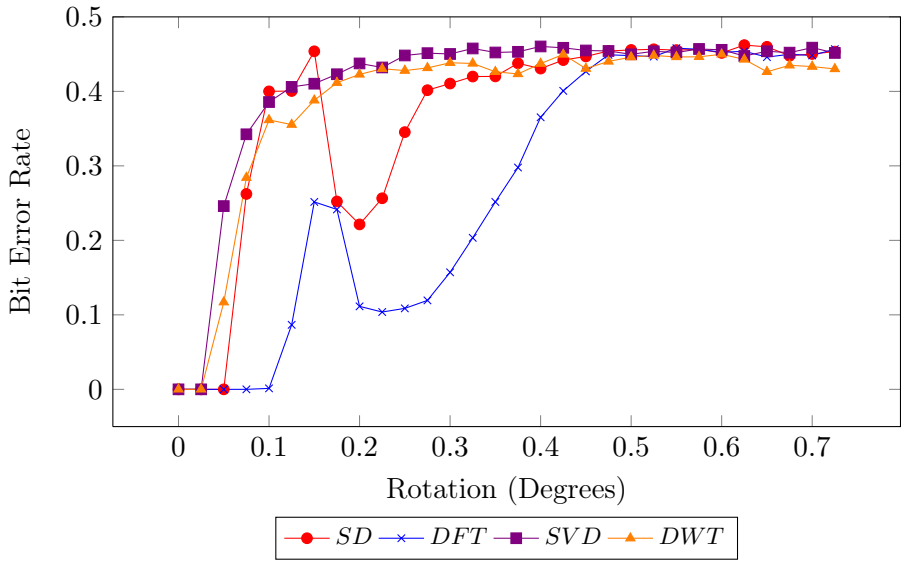
Results

Figures 5.7 and 5.8 show the results of cropped and loose rotation respectively. An example video frame is shown and the BER of the extracted information bits are graphed for varying amounts of rotation.

Cropped Rotation



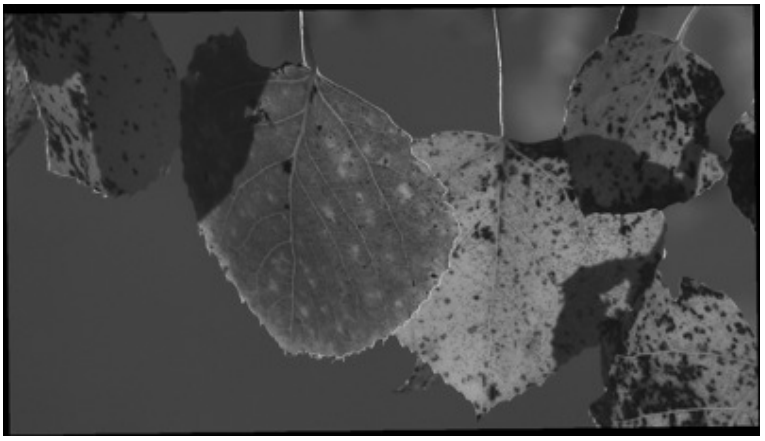
(a) Example of a video frame rotated by 0.7 degrees using cropped rotation.



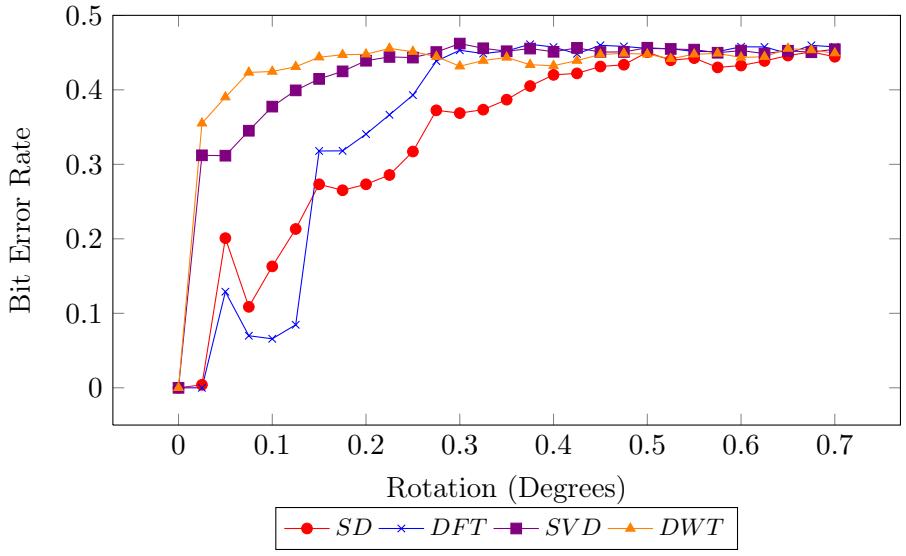
(b) Bit error rates of extracted watermark information bits for small rotations of the video frames using cropped rotation.

Figure 5.7: Resulting BER of the watermark extracted by each technique after performing a cropped rotation as an attack.

Loose Rotation



(a) Example of a video frame rotated by 0.7 degrees using loose rotation.



(b) Bit error rates of extracted watermark information bits for small rotations of the video frames using loose rotation.

Figure 5.8: Resulting BER of the watermark extracted by each technique after performing a loose rotation as an attack.

Discussion

All the watermarking techniques evaluated were found to be susceptible to desynchronisation caused by rotation. As discussed in Section 2.1.5, a successful attack prevents successful watermark extraction while not affecting the perceived quality of the watermarked content. Rotation is a good example of such a case, where the angle of rotation is small enough to not be noticeable by the viewer, but prevents the watermark message from being successfully extracted. Both cropped and loose rotation were effective at preventing successful extraction of the watermark information bits, with loose rotation being slightly more effective. Although the loose rotation approach may be more effective as an attack, it may be argued that a loose rotation is more noticeable by the viewer, whereas cropped rotation with small angles of rotation are not as noticeable.

This test determined that rotation is an effective attack against these watermarking techniques. In both cases, it is unclear why the response of the SD and DFT techniques are not monotonic.

5.3.3 Left-Right and Upside-Down Flipping of Video Frames

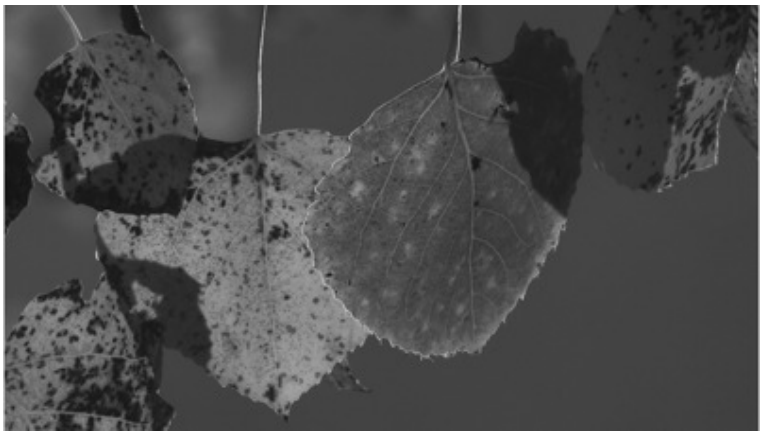
Intentional desynchronisation can also be achieved by flipping the video frames around a horizontal or vertical axis. A flip around the vertical axis causes loss of synchronisation, while it may not be of high importance to the viewer.

Method

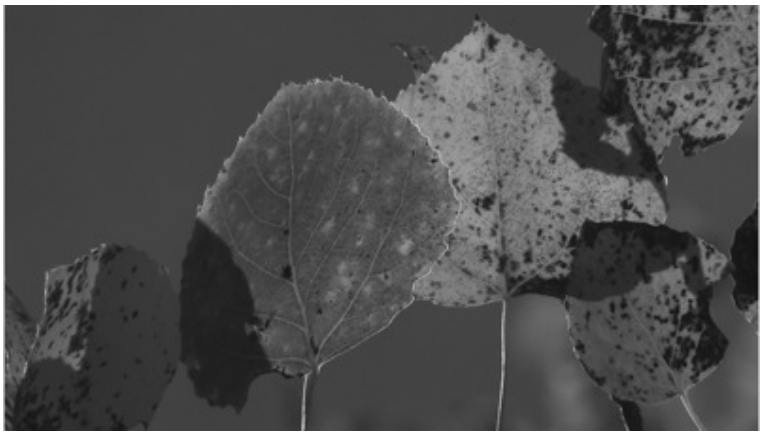
Each frame in the video block was flipped around a vertical and horizontal axis, after which the watermark information bits were extracted and the BER calculated in each case.

Results

Figures 5.9a and 5.9b show examples of video frames flipped around a vertical and horizontal axis respectively. Table 5.1 indicates whether the technique was able to successfully extract the watermark information bits.



(a) Example of a video frame flipped around a vertical axis.



(b) Example of a video frame flipped around a horizontal axis.

Figure 5.9: Examples of video frames flipped around a vertical and horizontal axis.

Table 5.1: Comparison of BER of information bits extracted by each watermarking technique after watermarked video frames were flipped around horizontal and vertical axes. A BER of 0 was seen as successful extraction, while a higher BER was seen as unsuccessful. In all cases, the BER after flipping was either 0 or close to 0.5.

Technique	Flip around vertical axis	Flip around horizontal axis
SD	✗	✗
DFT	✗	✗
SVD	✓	✓
DWT	✗	✗

Discussion

The results of this test indicate that flipping an image around a vertical or horizontal axis prevents successful watermark extraction for all techniques except the SVD technique. In all cases the BER of extracted information bits were either 0 or close to 0.5. Flipping an image around a vertical axis may be an easy and effective attack without affecting the perceived quality of the content. This may prove more effective against automated systems, as a human should be able to identify the flipping and correct it before watermark extraction.

From this test it can be concluded that flipping all video frames around a vertical axis can be an effective attack against automated watermark extraction techniques that are not based on the SVD.

5.4 Temporal Desynchronisation Evaluation

Since only a small number of bits can be embedded in a single video frame while maintaining robustness, it may be necessary to embed the watermark over a block of successive video frames. If this is done, it is important that the watermark extractor can synchronise itself with the video frames. If frame rate conversions are applied to watermarked video content, frame blending may also occur which may affect watermark extraction performance.

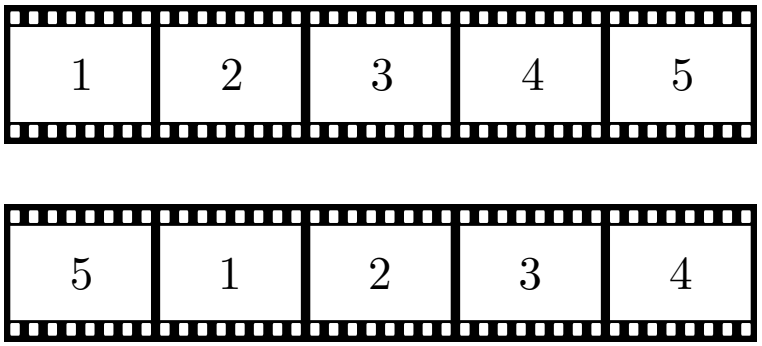
5.4.1 Temporal Shift

Method

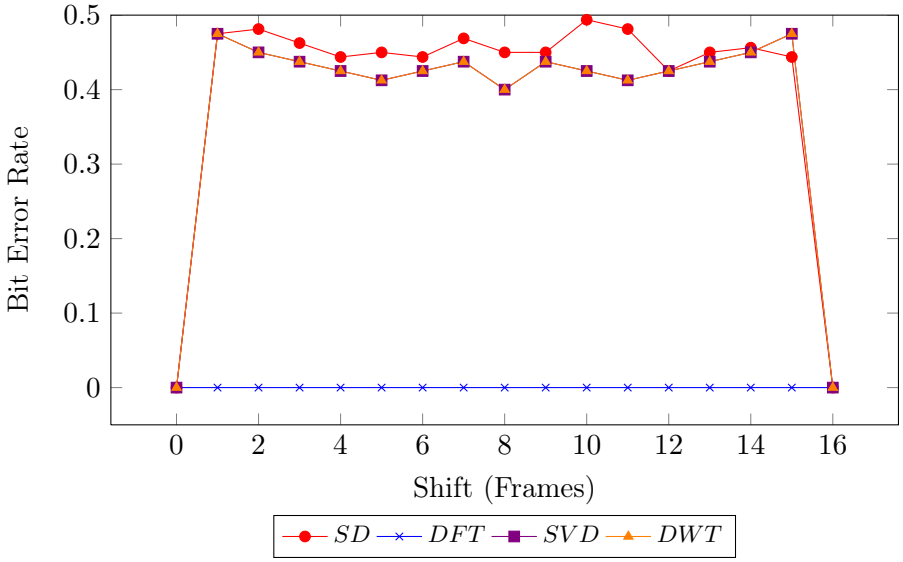
To evaluate temporal shift robustness, frames in each video block were shifted in the positive z-direction and wrapped around to the front of the block, as presented in Figure 5.10a.

Results

The results of a temporal shift are shown in Figure 5.10. An example of the shifting method is shown and the BER of extracted watermark information bits are graphed for varying amounts of shift.



(a) Example of temporally shifting video frames by one frame.



(b) Bit error rates of extracted watermark information bits for varying amounts of temporal shift.

Figure 5.10: Results of temporal shifts applied to a block of video frames. An example of the shifting method is shown and the BER of extracted watermark information bits are graphed for varying amounts of shift.

Discussion

The SD, SVD and DWT techniques require frame-accurate synchronisation in order to successfully extract the watermark. However, in the case of the DFT technique, a shift in the temporal domain causes a linear shift in the frequency domain [93]. If we assume that the same watermark will be embedded in each block of frames, the watermark in W can be seen as a periodic function with the image content seen as noise. A loss of temporal synchronisation in the temporal domain can be seen as a spatial shift, similar to what was observed in Section 5.3.1. This shift will only cause a phase shift in the Fourier coefficients, against which the DFT technique is robust.

Note that although a circular shift was used in this evaluation, if the same watermark is embedded in each video block in a video, the DFT technique is also robust against a linear shift of frames [93].

It follows that the DFT technique is robust against temporal desynchronisation, while all the other techniques require frame-accurate synchronisation.

5.4.2 Frame Blending

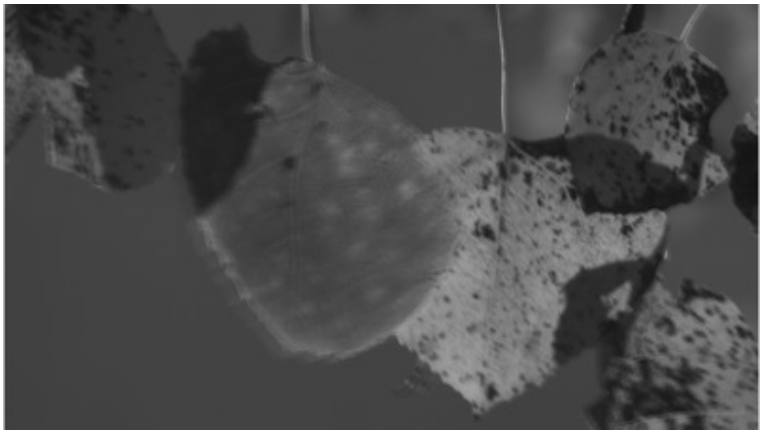
Frame blending can be the result of an intentional attack, or it can occur when frame rate conversions were applied to the watermarked video content. If the video contents are fairly stationary, frame blending can affect the non-stationary watermark while affecting the stationary video content to a lesser extent.

Method

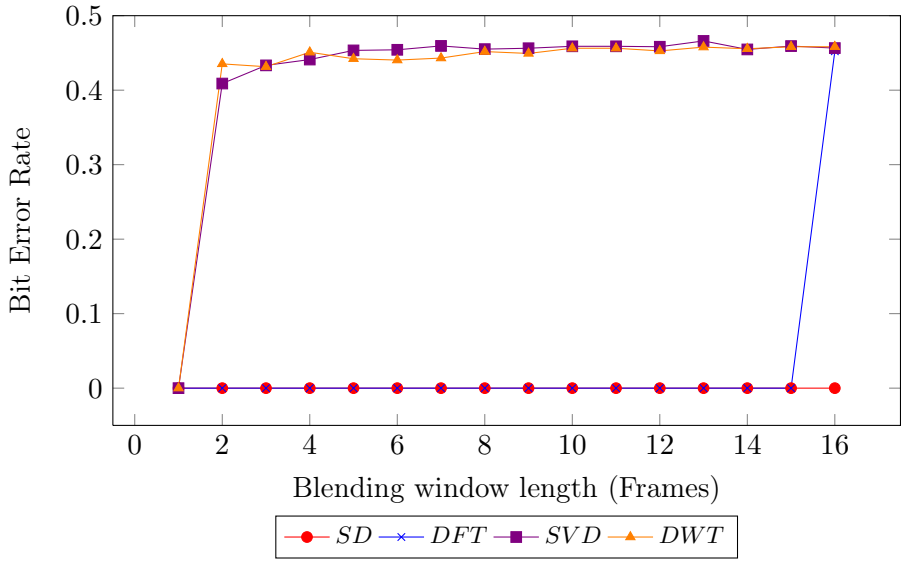
A temporal averaging filter with a varying length was applied on each frame in a video block. This filter calculated the average of a specified number of frames in the temporal dimension in each case. The video block to be evaluated was periodically repeated to create a sufficient number of frames for the averaging operation. The watermark information bits were extracted and the BER calculated in each case.

Results

Figure 5.11a shows the result of a temporal averaging filter with a length of 16 frames applied. Figure 5.11b shows the BER of extracted watermark information bits after frame averaging was applied to watermarked video blocks.



(a) Example of a video frame from a video block with a temporal averaging filter with a length of 16 frames applied.



(b) Bit error rates of extracted watermark information bits after a temporal averaging filter with varying lengths applied to watermarked video blocks.

Figure 5.11: Results of applying a temporal averaging filter with varying lengths to watermarked video blocks.

Discussion

Since the watermark is spread across multiple frames in a watermarked video block, frame blending prevents successful extraction of watermarks in the non-correlation based SVD and DWT techniques. If the complete watermark was embedded in each frame of the video, instead of spreading 80 bits over 16 frames, the techniques would possibly be more robust against frame averaging operations. The SD and DFT techniques are correlation based and are not affected by reasonable amounts of

frame blending.

It follows that the SVD and DFT technique may be affected by frame rate conversions, which may apply frame blending. As an intentional attack, frame blending even with a small temporal window size can defeat the SVD and DWT techniques. If the watermarked content does not contain a high amount of movement, this can be an effective attack.

5.5 Spatial Filtering

If a watermark embedded in a video block is seen as noise, simple denoising methods may be able to remove watermarks without affecting image quality.

5.5.1 Averaging Filter

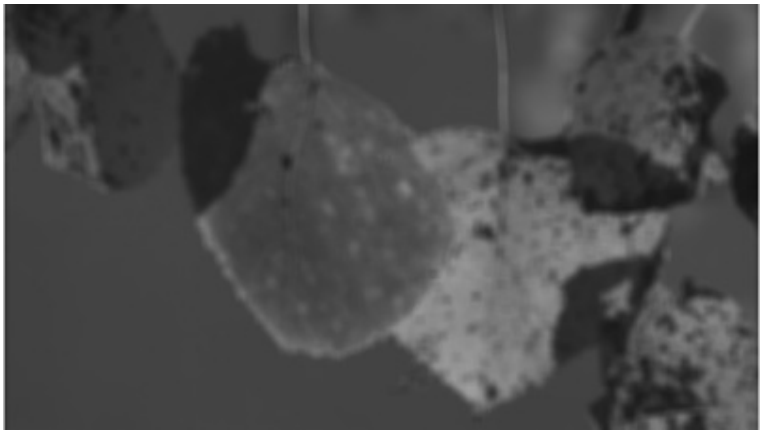
A simple form of spatial noise reduction is a spatial averaging operation, but may also cause a loss of detail in the content [152]. This attack may be effective at removing the embedded watermark without affecting the perceived visual quality of the watermarked content if the windows size of the filter is kept sufficiently small.

Method

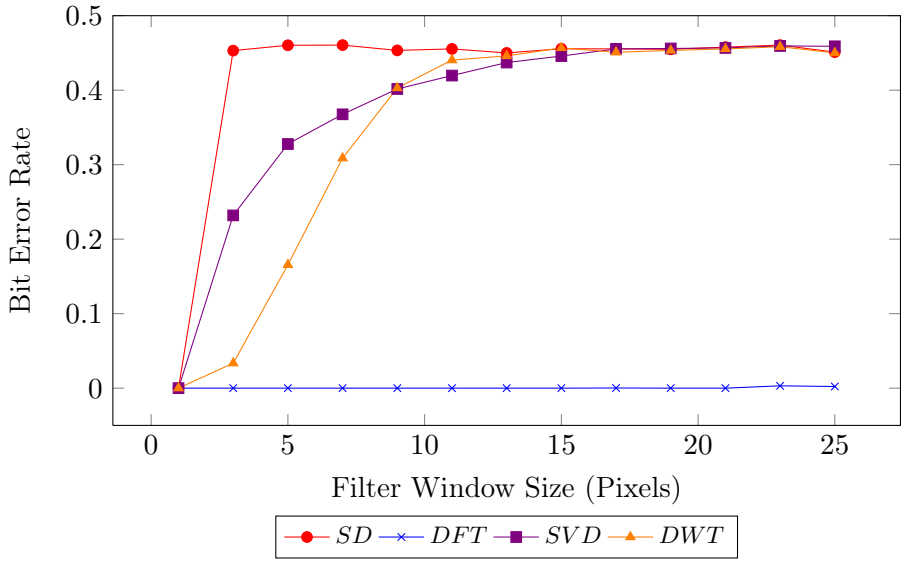
A spatial averaging filter with a square window of varying size was applied to each video frame in the video block. The kernel used for the averaging operation is a square kernel with a size of $n \times n$ pixels. A weight of $1/(n \times n)$ is applied to each pixel inside the kernel when performing the averaging operation. After the averaging operation was applied, the watermark information bits were extracted and the BER calculated for each case.

Results

Figure 5.12a shows an example of a video frame with a spatial averaging filter with a window size of 25×25 pixels applied. Figure 5.12b shows the BER of extracted watermark information bits after the spatial averaging filter was applied to watermarked video blocks.



(a) Example of a video frame after an averaging filter with a window size of 25×25 pixels was applied.



(b) Bit error rates of extracted watermark information bits after applying a spatial averaging filter with varying window sizes.

Figure 5.12: Results of applying a spatial averaging filter applied to a block of watermarked video frames.

Discussion

The investigation revealed that the SD and DWT techniques are defeated by applying an averaging filter with a relatively small window size. The weakness of the DWT technique is that it relies on detail in the image to correctly detect the coefficients to extract the watermark from. The averaging filter removes this detail, causing the technique to fail. The SVD technique is only affected once the filter is large enough to affect the rank approximation that the chosen singular values represent. Since the

DFT technique embeds information only in mid-frequency bands, it is unaffected for reasonable filter window sizes.

Simple spatial filtering can be used to defeat the SVD, DWT and SS techniques, but may result in a loss of detail in the original content.

5.5.2 Adaptive Wiener Denoise

Adaptive Wiener denoising is a filtering method that attempts to remove noise in an image without any prior knowledge about the noise [12]. This more advanced noise reduction technique may be more effective than an averaging filter at removing the watermark while maintaining a relatively high visual quality for the watermarked content.

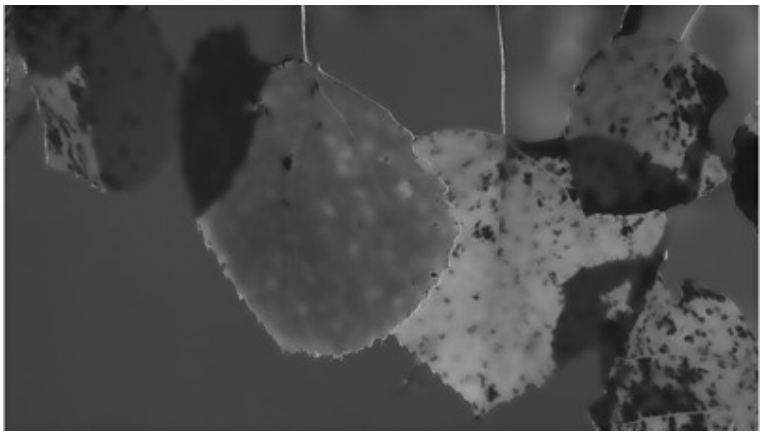
Method

A two-dimensional adaptive noise-removal filtering was applied to each watermarked video frame. The pixel-wise adaptive Wiener denoise function `wiener2` in Matlab was used with a varying window size. As described at by [153] and [154, page 548], the `wiener2` function is applied pixel-wise and estimates the local mean and variance around each pixel and then creates a pixel-wise Wiener filter using these values.

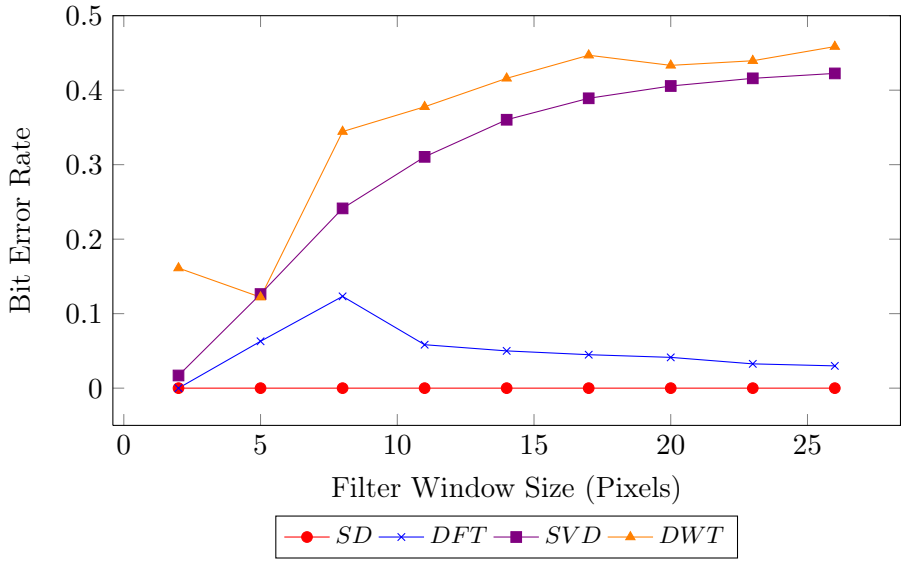
After the adaptive Wiener denoise operation was performed, the watermark information bits were extracted and the BER calculated in each case.

Results

Figure 5.13a shows an example of a video frame with a two-dimensional adaptive Wiener denoising filter with a windows size of 25×25 pixels applied. Figure 5.13b shows the BER of extracted watermarks after Wiener denoising was applied to watermarked video blocks. The SD technique is relatively robust against this form of denoising, as not all components of the embedded signal is removed.



(a) Video frame after a two-dimensional Wiener denoising filter with a window size of 25×25 pixels was applied.



(b) Bit error rates of extracted watermark information bits after an adaptive Wiener denoising filter was applied to the watermarked video blocks.

Figure 5.13: Results of applying an adaptive Wiener denoising filter to watermarked video blocks.

Discussion

It is noted that the SVD and DWT techniques are affected by Wiener denoising, with the DWT technique being the most sensitive. This is because the denoising operation potentially changes the texture information contained in an image, causing the technique to fail. As the window size of the Wiener denoising filter increases, it becomes a smoothing filter, affecting the SVD technique. The SD and DFT techniques are unaffected for reasonable window sizes. It should be noted that a

25×25 pixel window size required to defeat the SVD technique results in considerable loss of detail in the image.

5.6 Cropping and Resizing Evaluation

5.6.1 Cropping

Cropping can be applied to video frames as a form of attack, or by other common video processing operations. In the case of an attack, the central parts of the video frames are often of greatest importance [11], enabling an attacker to crop the image without affecting the perceived quality of the watermarked content significantly.

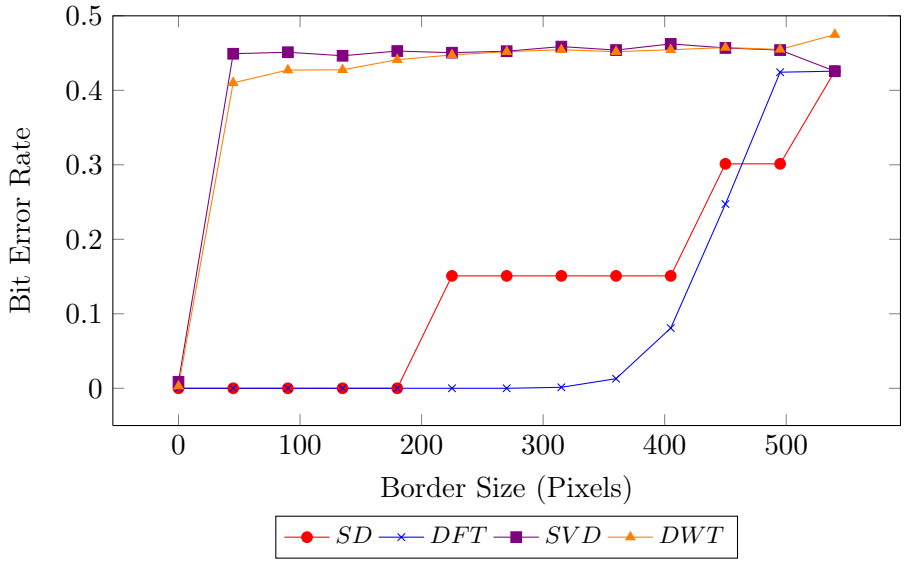
Method

To simulate cropping, the edges of each video frame were filled with black pixels, forming borders with no content around the center part of the image.

Results

Figure 5.14a shows an example of a video frame cropped to remove 500 pixels from each edge of the image. Figure 5.14b shows the BER of extracted watermark information bits after frame cropping was applied to watermarked video blocks.

(a) Example of a video frame cropped to obtain a border of 500 pixels from the edge of each video frame.



(b) Bit error rates of extracted watermark information bits after frame cropping was applied to watermarked video blocks.

Figure 5.14: Results of applying frame cropping to watermarked video blocks.

Discussion

Frame cropping affects the SVD and DWT techniques to a large extent. The cropping affects the rank approximations used by the SVD technique, resulting in incorrect extraction of the watermark. The DWT technique fails because the incorrect coefficients are selected for extraction. Note that the performance of the spread spectrum technique deteriorates in stages. Recall from Section 3.2.1 on page 27 that each bit of the watermark message occupies a horizontal band of the video frame. As these bands are sequentially removed, the information of each embedded bit is lost. The

DFT technique performs well, as the information is embedded in all spatial regions of the image. It only starts deteriorating once enough of the frame has been stripped away in order to cause the correlation receiver to fail.

From these results it can be concluded that cropping is an effective attack against the SVD and DFT techniques, while the SD and DFT techniques require an unreasonable amount of cropping for the watermark to be incorrectly extracted.

5.6.2 Resizing

Resizing is a common signal manipulation that can occur as media is converted for display on different devices, or as an intentional attack.

Method

The video frames were resized to the values shown in Table 5.2b. In order to extract the watermark, the original frame size was obtained in two different ways. The resized video frames were initially resized again to obtain the original video frame size, the watermark extracted and the BER calculated. This process can be summarised as downsampling of the video frames followed by upsampling. As a second test, the resized video frames were padded with black borders to obtain the original video frame size before watermark extraction. The watermarks were then extracted from the padded video frames and the BER calculated in each case. Bicubic resizing was used in all resizing operations.

Results

Figure 5.15 on the next page shows an example of a video frame resized to a size of 320×200 pixels and padded with black to obtain the original image size. Table 5.2a shows the BER of extracted watermarks after frame resizing was applied and the video frame was resized to the original dimensions before watermark extraction. Table 5.2b on page 92 shows the BER of extracted watermarks when the resized video frames were padded with black to obtain the original video frame size before watermark extraction.

Discussion

It is observed that when the content was resized to a smaller version and again resized to the original size, the watermark could be extracted for most reasonable image sizes. However, when the image was resized and padded to the original size, loss of synchronization occurred and the watermark could not be extracted. In the

Figure 5.15: Video frame resized to 320×200 pixels and padded with black borders to obtain the original image size.

case of resizing followed by padding, the relatively high BER is caused in each case by a loss of synchronization rather than the content in the image being removed. Note that downsampling followed by upsampling is similar to the averaging operation discussed in Section 5.5.1.

It follows that for these watermarking techniques, resizing is effective at causing loss of synchronisation and causing watermark extraction to fail rather than removing the watermark. If synchronisation can be re-established, it may be possible to successfully extract the watermark in most of these cases.

5.7 Changes in Pixel Intensity Values

5.7.1 Addition of Random Noise

Method

Uniformly distributed pseudo-random noise was added to the watermarked video frames, the watermark information bits extracted and the BER calculated in each case. The pseudo-random noise was generated with a maximum amplitude A with an offset of $A/2$. This was then added to the watermarked video block, after which any values smaller than 0 or larger than 1 were clipped to be in the valid intensity range for double-precision images in Matlab.

Results

Figure 5.16a shows an example of a video frame with added noise with a maximum amplitude of 1. Figure 5.16b shows the BER of extracted watermarks after noise

Table 5.2: Bit error rates of extracted watermark information bits for video frames resized to various sizes.

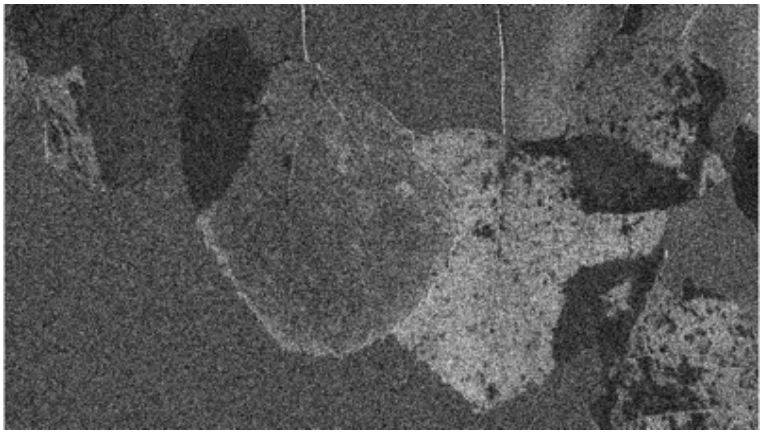
(a) Bit error rates of extracted watermarks for video frame resized to various sizes. Video frames were resized to the original frame size for watermark extraction.

Size	SD	DFT	SVD	DWT
1280×720	0.0000	0.0000	0.0000	0.0000
800×800	0.0000	0.0000	0.0250	0.0083
1024×768	0.0000	0.0000	0.0000	0.0000
640×480	0.0000	0.0000	0.0875	0.0083
500×500	0.0000	0.0000	0.2083	0.0167
320×240	0.0167	0.4125	0.3083	0.1375
200×320	0.0417	0.3917	0.2833	0.1333
100×100	0.3792	0.4292	0.4458	0.4583
50×50	0.4708	0.4375	0.4750	0.4625

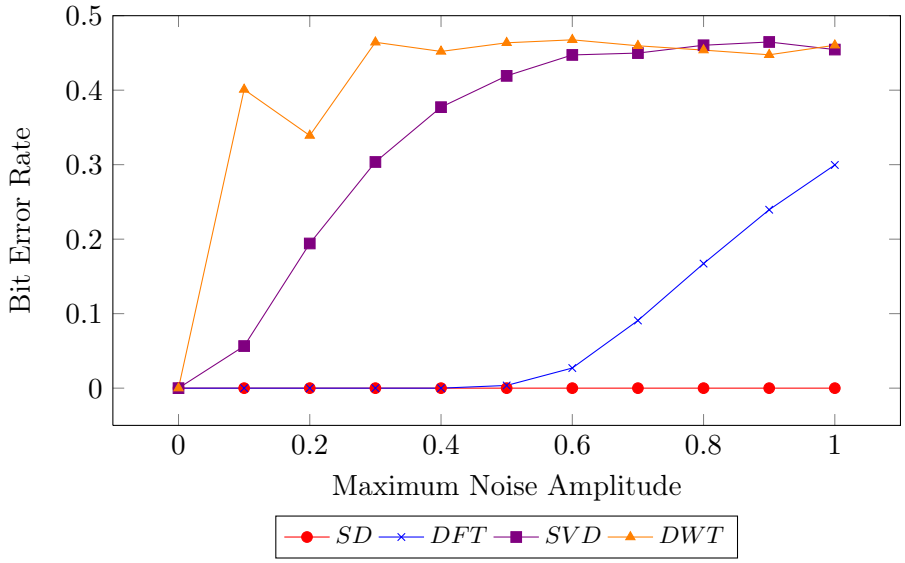
(b) Bit error rates of extracted watermarks for video frame resized to various sizes. Video frames were padded with black borders to obtain the original frame size for watermark extraction.

Size	SD	DFT	SVD	DWT
1280×720	0.4417	0.4625	0.0000	0.4667
800×800	0.4375	0.4667	0.0042	0.4708
1024×768	0.4667	0.4833	0.0000	0.4583
640×480	0.4750	0.4250	0.0417	0.4292
500×500	0.4708	0.4583	0.1083	0.4542
320×240	0.4625	0.4667	0.2292	0.4792
200×320	0.4333	0.4667	0.2375	0.4542
100×100	0.4667	0.4542	0.4333	0.4708
50×50	0.4833	0.4375	0.4583	0.4708

was added to watermarked video blocks. Note that the intensity values of double-precision images in Matlab are in the range $[0, 1]$.



(a) Example of a video frame with noise added with a maximum amplitude of 1.



(b) Bit error rates of extracted watermark information bits after pseudo-random noise was added.

Figure 5.16: Results of the adding pseudo-random noise to watermarked frames.

Discussion

The SD technique performs well, since it uses a correlation detector, which is robust against the addition of random noise that is uncorrelated to the watermark [146]. The DWT is severely affected, since the addition of noise affects the coefficient selection process. Recall that for the DWT technique, a coefficient is randomly selected and E calculated using (3.20) in Section 3.5.2 on page 37. If the coefficient did not contain detail, E was smaller than a threshold T and the coefficient was not selected for embedding. The addition of noise causes coefficients to contain detail

at the extraction stage that was not present at the embedding stage. It follows that coefficients are selected for extraction, although no watermark was embedded in that specific coefficient.

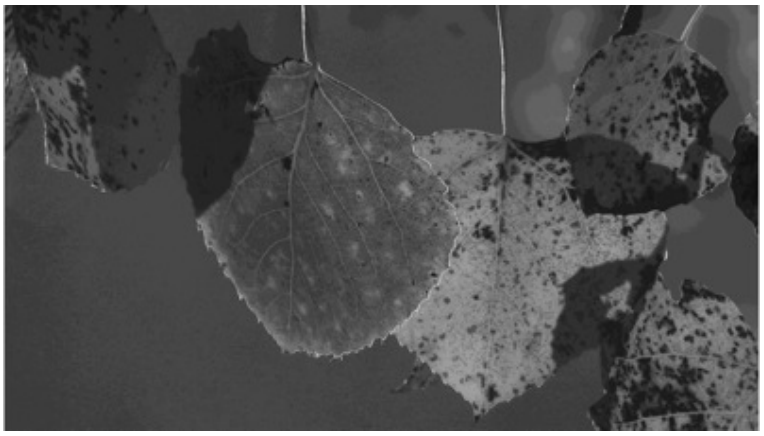
5.7.2 Quantisation

Method

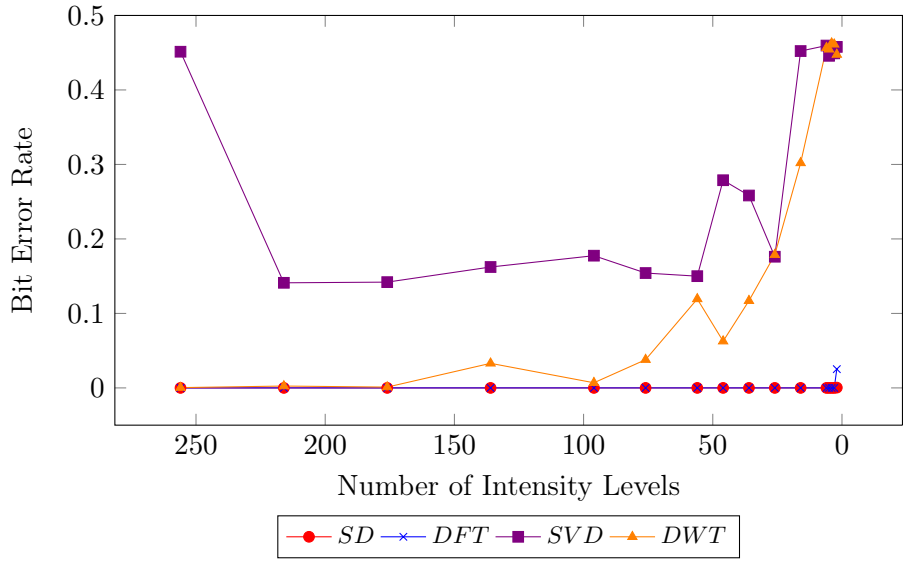
The watermarked video blocks were quantised in each case to contain a varying number of intensity levels. The watermark information bits were then extracted from the quantised video frames and the BER calculated in each case.

Results

Figure 5.17a shows an example of a video frame quantised to contain 20 intensity levels. Figure 5.17b shows the BER of extracted watermarks after the quantisation process was applied to watermarked video blocks.



(a) Example of a video frame quantised to contain 20 intensity levels.



(b) Bit error rates of extracted watermark information bits after quantisation was applied to watermarked video blocks.

Figure 5.17: Results of applying a quantisation operation to watermarked frames.

Discussion

Because the SVD technique causes small intensity changes to the video frames during the watermarking process, these changes can get discarded as the number of quantisation levels decrease. This causes the watermark extraction to fail. The same occurs in the case of the DWT technique, which also fails. Because the SD and DFT techniques distribute the changes over a larger range of pixel intensities, these are largely unaffected by quantisation.

It can be concluded that quantisation is not a serious threat to the SD, DFT and DWT techniques as an unreasonably low number of quantisation levels are required

to prevent successful watermark extraction. The SVD technique is affected to a larger extent, but it should still be possible to retrieve all the watermark bits through the use of error coding.

5.7.3 Pixel Intensity Changes

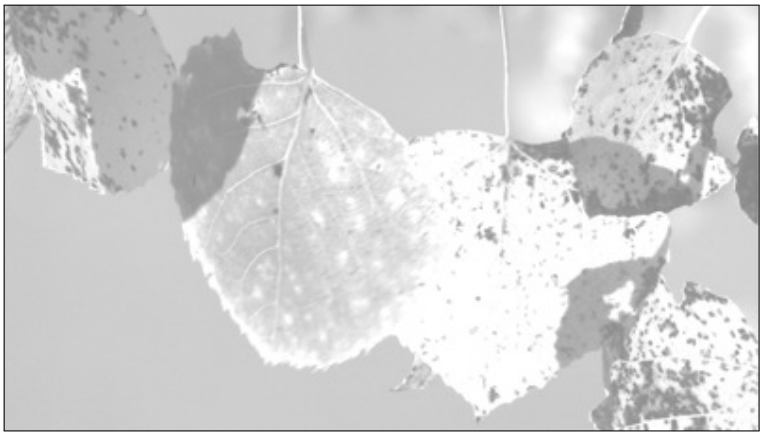
Modifying the pixel intensities of the watermarked luminance video frames translates to a change in brightness of the video content. Slight changes may prevent successful watermark extraction without reducing the perceived visual quality of the watermarked content.

Method

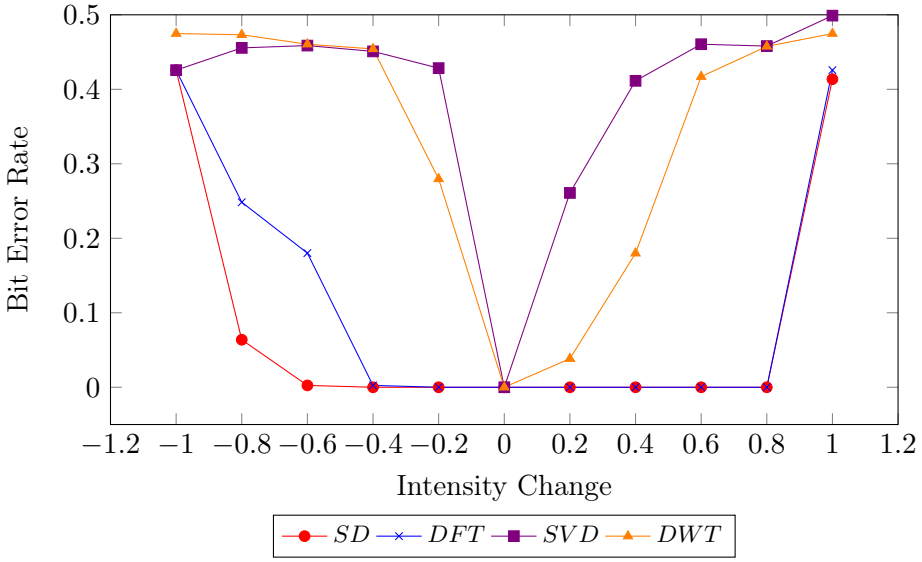
The pixel intensity of each video frame was changed by adding or subtracting a value from each pixel in the video frame. This value was varied from -1 to $+1$ and any values smaller than 0 or larger than 1 were clipped to be in the valid intensity range for double-precision images in Matlab. The watermark was extracted from each video block and the BER calculated in each case.

Results

Figure 5.18a shows an example of a video frame with the brightness of each pixel increased by a value of 0.5. Figure 5.18b shows the BER of extracted watermarks after the brightness process was applied to the watermarked video blocks.



(a) Example of a video frame with the brightness of each pixel increased by 0.5.



(b) Bit error rates of extracted watermark information bits after the intensities of all video pixels were modified.

Figure 5.18: Results of changing the intensity values of watermarked frames.

Discussion

The SD and DFT techniques are largely unaffected by changes in the amplitude of the luminance frames. The DWT was affected, but it can be argued that an unreasonably large change in intensity value is required in order to defeat the watermark. The SVD technique was the most susceptible to changes in brightness of the video frames.

It follows that changes in brightness can be used to defeat the SVD and DWT watermarking techniques, but the change in brightness may be perceivable by the viewer.

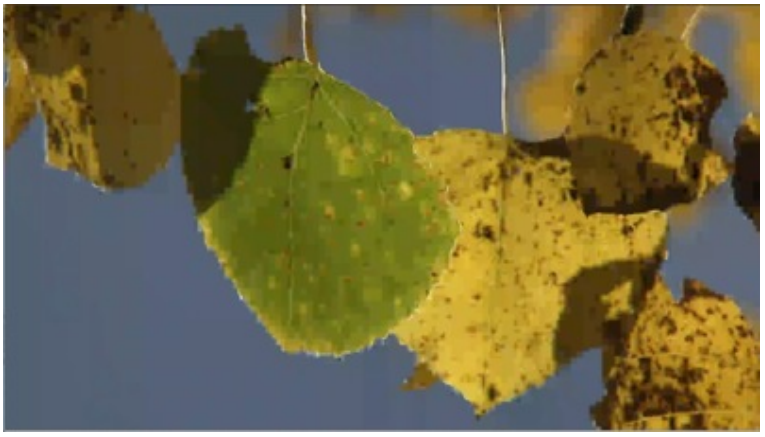
5.8 Compression

Method

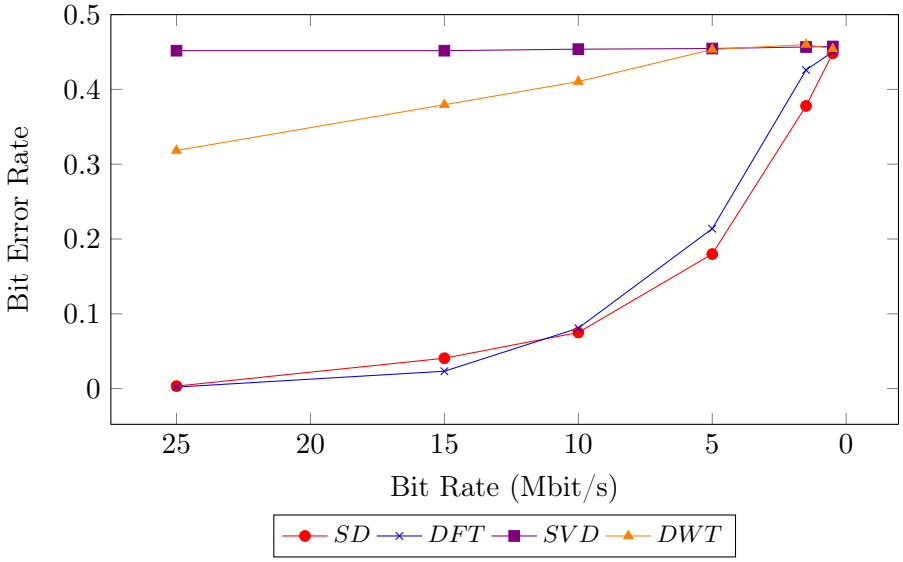
Video sequences were watermarked and saved as 8-bit uncompressed .avi video files, with colour information stored in the BGR24 format in these videos. This resulted in 8 bits allocated to each blue, green and red colour component in the video. The video files were then compressed using H.264 compression at various bit rates, after which the watermark message was extracted from each video and the BER calculated in each case. The video sequences were compressed using an open-source compression utility, Handbrake [155] version 0.9.5, with the parameters specified in Table A.5 on page 137. An overview of the H.264 compression process is provided in [156].

Results:

Figure 5.19a shows the result of H.264 compression at 0.5 MBit/s. Blocking artefacts are observed and motion is not smooth when the video is viewed. Figure 5.19b shows the BER of extracted watermark information bits after compression was applied to video sequences at different bit rates.



(a) Example of a video frame compressed using H.264 compression at a bit rate of 0.5 Mbit/s. Blocking artefacts are seen and motion is not smooth when the video is viewed.



(b) Bit error rates of extracted watermark information bits after H.264 compression was applied at various bit rates.

Figure 5.19: Results of H.264 compression applied to watermarked video sequences.

Discussion

When writing a watermarked sequence to an uncompressed video file, the video frames are firstly converted to the RGB colour space and quantised to 8 bits per colour channel. As was reported in Section 5.7.2, the SVD technique is sensitive to quantisation and the watermark message is incorrectly extracted after the colour space conversion and quantisation. The SD, DFT and DWT techniques are not affected at this stage.

Compression is then applied, which discards statistically and perceptually redundant information from the video file [157] in combination with other techniques to reduce the files size. After compression, the DWT technique is unable to successfully extract the watermark message from a video file encoded at a bit rate 25Mbit/s. Because the DFT technique embeds the watermark message in perceptually relevant frequency bands of the video file, it is more resistant to this effect of compression.

The SD and DFT are relatively robust against compression. The SD technique performs slightly better than the DFT technique against compression, as it embeds the watermark into all frequency bands of the original content. In this case, when high and low frequency components are removed, enough of the watermark signal remains to extract the watermark. By adjusting the spatial and temporal frequency bands in which the DFT technique embeds the watermark, it may be possible to achieve higher robustness against compression.

The SD and DFT techniques fail to successfully extract the watermark message at compressed video bit rates of below 0.5Mb/s, but it should be noted that the video quality is considerably reduced at this bit rate, which may render this a less desirable form of attack.

5.9 Cascadability Evaluation

Cascadability refers to the ability of the watermark technique to embed and detect multiple watermark messages embedded in the same video block. It may be necessary to embed more than one watermark at different parts of the video processing chain, or attackers may embed additional watermarks into the video block in an attempt to disable the original watermark. For these reasons the watermarking techniques should be able to extract watermarks successfully after multiple watermarks were embedded into the same video block [59].

Method

Four watermarked video blocks were created, each using one of the implemented watermarking techniques. Additional watermark messages were then successively embedded into these watermarked video blocks, each time using the same technique that was used to create the original watermarked video block. After each embedding stage, it was attempted to extract the original watermark message from the video block containing multiple watermarks. The BER of the extracted watermark information bits was then calculated for each case.

Results

The results of successively embedding watermark messages using the same watermarking technique into video blocks are displayed in Table 5.3. A ✓ indicates that the technique is cascadable, while a ✗ shows that the technique is not.

Table 5.3: Comparison of the ability of watermarking techniques to extract a watermark when multiple watermarks of the same type are present in a watermarked video block. A ✓ indicates that the technique is cascadable, while a ✗ shows that the technique is not.

Technique	Cascadability	Comment
SD	✓	Multiple watermarks can be embedded and extracted.
DFT	✓	Multiple watermarks can be embedded and extracted.
SVD	✗	Last embedded watermark embedded can be extracted.
DWT	✗	Last embedded watermark embedded can be extracted.

Discussion

The SS and DFT techniques are cascadable in their current form. Multiple watermarks can successfully be embedded and extracted, provided that unique pseudo-random sequences are used for each watermark message. The SVD and DWT techniques are not cascadable in their current form. The SVD technique may be modified to embed different watermarks in different singular values, but this will affect robustness and imperceptibility of the watermarks. Embedding the watermark into smaller singular values will make the watermark more sensitive to changes in detail, while embedding in larger singular values may cause more visible artefacts. The DWT technique can be modified to generate different pseudo-random coefficient indices for embedding and extraction, or different texture thresholds can be used. However, in some video frames there are a limited number of coefficients available with sufficient texture detail for embedding of the watermark and may result in the same coefficients being selected even if the pseudo-random number generator is seeded differently. It follows that the DWT technique can be modified to be cascadable to a certain extent.

5.10 Chapter Summary

The computational complexity and robustness of each watermarking technique were evaluated in this section. By timing the execution time of the implementations, it was found that the SD and DFT techniques are relatively efficient, while the SVD technique requires more time to embed and extract a watermark. Comparing the

execution times to time required to perform only the mathematical transforms by each technique, additional insight was gained into the computational complexity of the techniques. In the cases of the SD and DFT techniques, more time was spent on other operations in the technique implementation than was spent applying the mathematical transform. In the case of the DFT technique, this can be attributed to the extensive process used to select elements in which to embed the watermark. On the other hand, it was seen that, in the case of the SVD technique, most of the time was spent applying the SVD.

Although these implementations can be improved to be more efficient, these results point out that the mathematical transform required is not the only parameter to consider when evaluating the potential computational complexity of a watermarking technique. The technique may also require other operations which can be computationally complex.

The results from the robustness evaluations conducted in this section are summarised in Table 5.4. It was found that numerous effective techniques do not remove the watermark from the content, but desynchronises the watermark with the watermark extraction process and causes the watermark message to be incorrectly extracted. These attacks can be effective in automated systems, but it may be possible through human intervention or registration techniques to detect and undo these attacks. Such an example is rotation or flipping of the watermarked video frames, which may be possible to detect and correct.

Through the evaluation of resizing as an attack, it was also noted that the resizing operation causes desynchronisation rather than removing the embedded watermark.

Basic filtering attacks were evaluated, which aim to remove the watermark without degrading the perceived quality of the watermarked content. While these were relatively successful at removing the watermarks, the quality of the content was degraded in most cases. It may be necessary for an attacker to employ more advanced filtering techniques in order to remove a watermark.

Overall, the DWT technique appeared to be the least robust of the techniques. The SVD exhibited useful properties, such as immunity against flipping of frames around a vertical or horizontal axis, but was susceptible to operations such as quantisation and compression. The SD and DFT techniques performed well against most attacks, but were susceptible to desyncronisation attacks. Although the DFT technique is immune to spatial and temporal shifts, rotation is a relatively easy and effective attack against this technique.

Through the evaluations in this chapter, it became clear that each category of technique has distinct advantages and disadvantages. These results can assist in selecting a watermarking technique for a specific application, as well as to identify

areas for future research.

Table 5.4: Summary of robustness results for each watermarking technique against various attacks. A ✓ indicates that a technique was relatively robust against an attack, while a ✗ indicates that an attack was able to prevent the successful extraction of the watermark message, without degrading the perceived quality of the watermarked content considerably.

Criteria	SD	DFT	SVD	DWT
X Shift	✗	✓	✓	✗
Y Shift	✗	✓	✓	✗
XY Shift	✗	✓	✓	✗
Cropped Rotation	✗	✗	✗	✗
Loose Rotation	✗	✗	✗	✗
Flipping	✗	✗	✓	✗
Temporal Shift	✗	✓	✗	✗
Frame Blending	✓	✓	✗	✗
Spatial Averaging	✗	✓	✗	✓
Wiener denoising	✓	✓	✓	✓
Cropping	✓	✓	✗	✗
Resizing (Downsmampling & Upsampling)	✓	✓	✓	✓
Padded Resizing	✗	✗	✗	✗
Addition of Random Noise	✓	✓	✓	✗
Quantisation	✓	✓	✗	✓
Pixel Intensity Changes	✓	✓	✓	✓
Compression	✓	✓	✗	✗
Cascadability	✓	✓	✗	✗

Chapter 6

Recommendations for Future Work

From the evaluations conducted in Chapters 4 and 5, it was found that the DFT technique performed well in terms of imperceptibility as well as robustness. The technique should ideally, however, be improved in the areas of computational complexity and robustness against rotation. Possible methods for improving the robustness and computational efficiency of the DFT technique are now briefly discussed.

6.1 Existing Rotation Invariant Solutions

The DFT technique in [93] employs a method to detect and correct geometrical transforms applied to the watermarked content before attempting to extract the watermark. This is achieved by embedding a set of anchor points, called a *template*, into the watermarked content. By later examining how these anchor points moved, it is possible to determine and correct geometrical transforms that were applied to the watermarked content. Examples of other techniques employing this approach to obtain rotation invariance include [87, 93, 103] and [112].

Although these template-based watermarking techniques can provide robustness against rotation, they become susceptible to a new form of attack. Instead of the attacker removing the embedded watermark, an attempt can be made to prevent the correct detection of the embedded template. This is known as a *template attack* and is further discussed in [158]. It was shown in [158] that if a watermarking technique relies on a template for successful watermark extraction, it may be possible for an attacker to alter the template in such a way that the watermark message is incorrectly extracted. It this case, the template becomes the weakness of the watermarking technique rather than the watermark extraction process itself [146, 159].

An alternative solution to the rotation attack is proposed in [160], where a circularly symmetric watermark is embedded in the frequency domain. This method can be extended for use in the DFT technique implemented in this thesis, but would require changes to the fundamental watermark embedding and extraction process and can be susceptible to off-axis rotation.

Moment-based techniques such as [109, 110, 161–164] are also presented as a solution against rotation attacks. However, as discussed in [165] and [166], these techniques can be computationally complex and can also cause a loss of quality in the watermarked content.

Rotation invariance can also be achieved through the use of the Fourier-Mellin transform, as was done in [167]. The Fourier-Mellin transform can be implemented by performing a Fourier transform on a log-polar mapping of an image, where the log-polar mapping maps coordinates in the form (x, y) to the form (μ, θ), with $x = e^{\mu} \cos\theta$ and $y = e^{\mu} \sin\theta$ [168]. The technique in [167] applies a Fourier transform to a frame to be watermarked, followed by a log-polar mapping. The magnitudes of the frequency spectrum resulting from the Fourier transform can be used to embed a shift-invariant watermark, as was seen in Section 5.3.1. After applying a Fourier-Mellin transform to this frequency spectrum, a scale and rotation invariant watermark can be embedded [169]. By combining the Fourier transform followed by a Fourier-Mellin transform, a translation, scale and rotation invariant watermark can be embedded.

The drawback of the Fourier-Mellin approach is that an inverse Fourier-Mellin transform needs to be applied, which results in a loss of quality due to interpolation caused by the log-polar mapping [103]. It follows that the use of a Fourier-Mellin transform can cause loss of quality in the watermarked content and potentially be prohibitively computationally complex.

From the discussion above it is noted that rotation invariance can be achieved, but can add to the computational complexity of a technique as well as potentially reduce the quality of the watermarked content. If computational complexity can be reduced while maintaining image fidelity, it may be possible to use the Fourier-Mellin approach in combination with the DFT technique discussed in [93] to obtain a translation, scale and rotation invariant technique.

6.2 Proposed Solution to Reduce Computational Complexity while Maintaining Image Quality

In order to improve rotation robustness, it is desirable to apply a Fourier-Mellin or similar technique to the DFT technique discussed in [93] with reduced computational complexity, without affecting the image quality. Such a solution is now proposed and briefly discussed.

Let $D(x, y, z)$ represent a block of unwatermarked, uncompressed video frames with $1 \leq x \leq X$, $1 \leq y \leq Y$ and $1 \leq z \leq Z$. Here X and Y represent the width and height of the video frames in pixels respectively, while Z specifies the number of frames contained in the video block D.

Most traditional approaches, such as those discussed in this thesis, tend to embed a watermark message using the three steps outlined below.

1. A mathematical transform is applied to D in order to obtain $\hat{D}$. In the case of the DFT technique, this would be a frequency domain representation of D.

2. The elements of $\hat{D}$ are then modified to embed the required watermark message.

3. Finally, a suitable inverse mathematical transform is applied to $\hat{D}$ in order to obtain W, which is the watermarked version of D.

This approach is suitable when watermarking images or lower resolution videos, as a relatively small number of pixels are available to be used for the embedding of a watermark. In the case of the DFT technique, a full-resolution DFT is desired, as this results in a larger number of frequency components being available from the limited number of pixels available, increasing the robustness of the watermark.

In the case of high-resolution videos, however, a larger number of pixels are available to be used for watermarking. After applying a full-resolution DFT, this translates to a higher number of frequency components available to be used for watermarking. Not only may more components be available than what is required for efficient watermarking, but applying a DFT or other suitable mathematical transform to a high resolution video sequence can be unnecessarily computationally complex. It may be desirable to limit the number of elements available for watermarking, which will translate to reduced computational complexity while maintaining acceptable watermark robustness.

An alternative approach is proposed to enable the use of lower resolution mathematical transforms without negatively affecting the visual quality of the watermarked content. Instead of relying on the mathematical transforms to maintain image fidelity, a lower resolution transform can be used and the watermark to be embedded

can be calculated. The watermark that needs to be added is then transformed to the spatial domain, after which it is added to the original content D. The watermark is of lower resolution, while the visual quality of the content is not affected.

A watermark can be embedded into D using the DFT technique through the following steps:

1. Resize D to a smaller size to obtain D'. The size should be chosen to lower computational complexity of the watermarking embedding, while still providing a large enough number of elements to be used for embedding in the frequency domain.

2. Apply a Fourier transform to D' in order to obtain $\hat{D}'$.

3. Embed the watermark into the magnitudes of the frequency components in $\hat{D}'$, as described in Section 3.3.2.

4. Apply an inverse Fourier transform to obtain a watermarked version of $\hat{D}'$ in the spatial domain, named $\hat{D}''$.

5. Calculate the change caused by watermarking in the spatial domain as $\delta D' = D' - \hat{D}''$.

6. Resize $\delta D'$ to match the dimensions of the original content D.

7. Obtain the watermarked content, W, by adding the lower resolution watermark to the original resolution original content using $W = D + \delta D'$.

Watermark extraction is performed by resizing the watermark content to the size that was used to calculate the lower resolution watermark, after which the watermark message can be extracted as described in Section 3.3.2.

For evaluation purposes, a lower resolution watermark was embedded into 16 video frames each with a resolution of 1920×1080 pixels. The lower resolution watermark had a frame size of 540×960 pixels. A watermark message of 80 bits were embedded into 20 consecutive video blocks of 16 frames each. The time required to watermark each block of 16 frames was logged, after which the mean value was calculated. The extraction process was done in a similar fashion, with the results shown in Figure 6.1.

The implementation of this technique in combination with the DFT technique is called the Fast Discrete Fourier Transform (FDFT) technique. A comparison of the time required for the embedding and extraction stages of the original SD and DFT techniques compared to the FDFT technique is shown in Figure 6.1. The FDFT technique executes considerably faster than the original DFT technique. The

FDFT technique closely matches the SD technique on the embedding stage while outperforming it for watermark extraction.

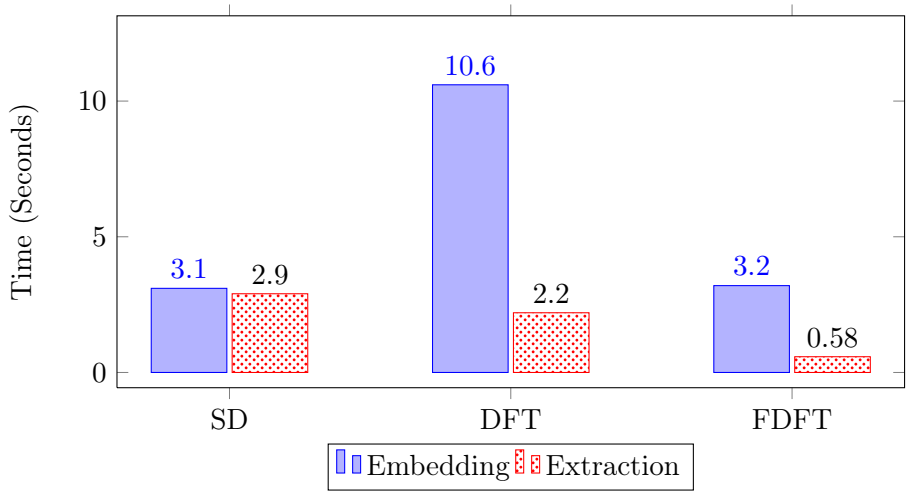


Figure 6.1: Comparison of the time required for the embedding and extraction stages of the original SD and DFT techniques compared to the Fast DFT (FDFT) technique. It was found that the FDFT technique is considerably faster than the original DFT technique. The FDFT closely matches the SD technique on the embedding stage while outperforming it for watermark extraction.

6.3 Chapter Summary

Various methods to obtain rotation invariance in watermarking techniques were briefly discussed. Two problem areas with these techniques were identified, namely computational complexity and the possible degradation of image quality. A new method was proposed to allow a lower resolution watermark to be embedded into a video sequence without affecting the resolution of the original content.

Using this proposed technique, it may be possible to apply solutions such as the Fourier-Mellin approach to the DFT technique. This can result in a computationally efficient technique that is invariant to scale, rotation and translation without the use of templates. The combination of the Fourier-Mellin approach with the FDFT technique should be further investigated.

The FDFT technique can also be used to embed a watermark using resource constrained devices, or to improve the embedding speed of the original DFT technique in general.

Chapter 7

Conclusion

7.1 Summary of Findings

It was hypothesised in Chapter 1 that:

> By implementing and evaluating selected video watermarking techniques, using the same source content and visual quality constraints, the distinct characteristics of each watermarking approach can be identified. Appropriate conclusions can then be drawn and recommendations for future research can be made.

By adhering to the research objectives defined, this hypothesis was proven to hold true. After implementing and evaluating representative watermarking techniques, it was possible to identify distinct characteristics associated with each technique. By comparing these characteristics to the fundamental requirements and applications for watermarking techniques, it is possible to select a potentially suitable watermarking technique for a specific application.

Through the visual and robustness evaluations in Chapters 4 and 5, important visual properties for video watermarking were identified. It was confirmed that an effective watermarking technique should ideally:

- be host-adaptive to ensure maximum robustness with minimum perceptibility; and
- embed the watermark in perceptually relevant areas of videos to prevent removal through attacks and compression.

Further robustness evaluations revealed that:

- successful watermark extraction can effectively be prevented by desynchronising the watermark extraction process with the watermark rather than having to remove the watermark from the content; and that
- rotation is an effective desynchronisation attack against all the watermarking techniques evaluated.

7.2 Summary of Contributions

Although the DFT technique was susceptible to desynchronisation through rotation, it performed well against the majority of attacks. After evaluating solutions to improve robustness against rotation, it was proposed to extend the DFT technique through the use of a Fourier-Mellin transform in combination with a modified method of embedding in order to reduce computational complexity without affecting the visual quality of the watermarked content. The FDFT technique was found to significantly improve the computationally efficiency of the DFT watermarking technique.

The existing SSIM method was also extended for effective visual comparison of watermarking techniques. In addition, a novel method to evaluate temporal flicker caused by video watermarking was proposed. These two contributions formed a valuable toolset to ensure fair comparison of the watermarking techniques evaluated.

7.3 Recommendations for Future Work

The DFT technique in [93] was found to perform well against most attacks, while being relatively computationally efficient. The FDFT technique was then developed to improve the computational efficiency and enabling the use of lower resolution watermarks without affecting the resolution of the original content. This should enable the modification of the technique to employ a Fourier-Mellin based embedding stage to provide robustness against rotation. This possibility should be further investigated, as it could result in a computationally efficient watermarking technique that is robust against scaling, rotation and translation in addition to the attacks evaluated in this thesis.

The pooled SSIM metric and temporal flicker metrics that were developed should be subjectively evaluated to improve the performance of these techniques.

7.4 Significance to Future Research

This research project provided insight into the characteristics of basic video watermarking techniques by comparing the performance of each against simple attacks, with the same source content and visual perceptibility criteria. This comparison can make it easier to select an appropriate technique for a specific application, and to provide quantitative results for future video watermarking research.

In addition, this thesis provides the necessary background to aid a newcomer to the watermarking field to gain a working knowledge of watermarking technologies. The source code for all implemented techniques and newly developed methods for visual evaluation are available at [114] to further help the development and research of new watermarking technologies.

Bibliography

[1] Seitz, J.: *Digital Watermarking for Digital Media*. Information Science Publishing, March 2005. ISBN 9781591405184.
Available at: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-518-4

[2] Piva, A., Bartolini, F. and Barni, M.: Managing copyright in open networks. *IEEE Internet Computing*, vol. 6, no. 3, pp. 18–26, May 2002. ISSN 1089-7801.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1003126

[3] Hartung, F. and Kutter, M.: Multimedia watermarking techniques. *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, July 1999. ISSN 00189219.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771066

[4] BitTorrent Inc.: BitTorrent. 2012.
Available at: http://www.bittorrent.com/

[5] Van der Schaar, M., Chou, P. and Corporation, E.: *Multimedia over IP and Wireless Networks: Compression, Networking, and Systems*. June 2003. Elsevier, 2007. ISBN 0120884801.
Available at: http://www.lavoisier.fr/notice/fr332943.html

[6] Mintzer, F., Braudaway, G. and Yeung, M.: Effective and ineffective digital watermarks. In: *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 9–12. IEEE Comput. Soc, 1997. ISBN 0-8186-8183-7.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=631957

[7] Maes, M., Kalker, T., Linnartz, J.-P., Talstra, J., Depovere, F. and Haitsma, J.: Digital watermarking for DVD video copy protection. *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 47–57, December 2000. ISSN 10535888.

Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=879338`

[8] Milano, D.: Content Control : Digital Watermarking and Fingerprinting. Tech. Rep., Harmonic Inc., 2008.
Available at: `http://www.rhozet.com/whitepapers/Fingerprinting_and_Watermarking.pdf`

[9] Bloom, J., Cox, I., Kalker, T., Linnartz, J.-P., Miller, M. and Traw, C.: Copy protection for DVD video. *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1267–1276, July 1999. ISSN 00189219.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771077`

[10] Petitcolas, F.: Watermarking schemes evaluation. *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 58–64, 2000. ISSN 10535888.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=879339http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=879339`

[11] Kutter, M.: Fair benchmark for image watermarking systems. In: *Proceedings of SPIE*, vol. 3657, pp. 226–239. SPIE, 1999. ISSN 0277786X.
Available at: `http://link.aip.org/link/?PSI/3657/226/1&Agg=doi`

[12] Deguillaume, F.: Countermeasures for unintentional and intentional video watermarking attacks. In: *Proceedings of SPIE*, vol. 3971, pp. 346–357. SPIE, 2000. ISSN 0277786X.
Available at: `http://link.aip.org/link/?PSI/3971/346/1&Agg=doi`

[13] Hembrooke, E.F.: Identification of Sound and Like Signals, United States Patent 3004104. 1961.

[14] Cox, I. and Miller, M.: Electronic watermarking: the first 50 years. In: *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 225–230. IEEE, 2001. ISBN 0-7803-7025-2.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=962738`

[15] Cvejic, N., Drajic, D. and Seppänen, T.: *Recent Advances in Multimedia Signal Processing and Communications*, vol. 231 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-02899-1.

Available at: http://www.springerlink.com/index/10.1007/978-3-642-02900-4

[16] Cox, I., Miller, M. and McKellips, A.: Watermarking as communications with side information. *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999. ISSN 00189219.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=771068http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771068

[17] Chou, J., Sandeep Pradhan, S. and Ramchandran, K.: On the duality between distributed source coding and data hiding. In: *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers (Cat. No.CH37020)*, vol. 2, pp. 1503–1507. IEEE, 1999. ISBN 0-7803-5700-0.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=832000

[18] Chen, B. and Wornell, G.: An information-theoretic approach to the design of robust digital watermarking systems. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 4, pp. 2061–2064, 1999.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=758336

[19] Abdulfetah, A., Sun, X. and Yang, H.: Robust Adaptive Video Watermarking Scheme using Visual models in DWT domain. *Information Technology Journal*, vol. 9, no. 7, pp. 1409–1414, July 2010. ISSN 18125638.
Available at: http://www.scialert.net/abstract/?doi=itj.2010.1409.1414

[20] Koz, A. and Alatan, A.: Oblivious Spatio-Temporal Watermarking of Digital Video by Exploiting the Human Visual System. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 326–337, March 2008. ISSN 1051-8215.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4449114

[21] Biswas, S., Das, S. and Petriu, E.: An Adaptive Compressed MPEG-2 Video Watermarking Scheme. *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1853–1861, October 2005. ISSN 0018-9456.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1514634

[22] Agrawal, V.: *Perceptual Watermarking of Digital Video Using the Variable Temporal Length 3D-DCT*. Ph.D. thesis, Indian Institute of Technology, Kanpur, 2007.
Available at: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.4715&rep=rep1&type=pdf`

[23] Lei, B.Y. and Lo, K.T.: A blind and robust watermarking scheme for H.264 video. In: *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 421–424. IEEE, July 2010. ISBN 978-1-4244-8224-5.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5581966`

[24] Barni, M., Bartolini, F. and Piva, A.: Improved wavelet-based watermarking through pixel-wise masking. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 5, pp. 783–91, January 2001. ISSN 1057-7149.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/18249667`

[25] Daren, H., Jiufen, L., Jiwu, H. and Hongmei, L.: A DWT-based image watermarking algorithm. In: *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, vol. 1, pp. 313–316. IEEE, 2001. ISBN 0-7695-1198-8.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1237719`

[26] Swanson, M. and Tewfik, A.: Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 540–550, May 1998. ISSN 07338716.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=668976`

[27] Hunter, P.: Combating video piracy. *Network Security*, vol. 2004, no. 2, pp. 18–19, February 2004. ISSN 13534858.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S135348580400039X`

[28] Wang, Y. and Pearmain, A.: Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain. *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1536–43, June 2006. ISSN 1057-7149.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/16764278`

[29] Langelaar, G.C., Lagendijk, R.L. and Biemond, J.: Real-Time Labeling of MPEG-2 Compressed Video. *Journal of Visual Communication and Image Representation*, vol. 9, no. 4, pp. 256–270, December 1998. ISSN 10473203.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S1047320398903972`

[30] Hartung, F. and Girod, B.: Digital watermarking of MPEG-2 coded video in the bitstream domain. In: *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2621–2624. IEEE Comput. Soc. Press, 1997. ISBN 0-8186-7919-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=595326`

[31] Alattar, A., Lin, E. and Celik, M.: Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 787–800, August 2003. ISSN 1051-8215.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1227608`

[32] Noorkami, M. and Mersereau, R.: Compressed-domain video watermarking for H.264. In: *IEEE International Conference on Image Processing 2005*, pp. II–890. IEEE, 2005. ISBN 0-7803-9134-9.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1530199`

[33] Wu, M., Trappe, W., Wang, Z. and Liu, K.: Collusion-Resistant Fingerprinting for Multimedia. *Signal Processing Magazine, IEEE*, vol. 21, no. 2, pp. 15–27, 2005. ISSN 1053-5888.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1276103`

[34] Swanson, M., Kobayashi, M. and a.H. Tewfik: Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, June 1998. ISSN 00189219.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=687830`

[35] Deshpande, N., Rajurkar, A. and Manthalkar, R.: Review of Robust Video Watermarking Algorithms. *International Journal of Computer Science and*

Information Security, vol. 7, no. 3, pp. 237–246, 2010.
Available at: `http://arxiv.org/pdf/1004.1770`

[36] Potdar, V., Han, S. and Chang, E.: A survey of digital image watermarking techniques. In: *INDIN 2005. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pp. 709–716. IEEE, 2005. ISBN 0-7803-9094-6.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1560462`

[37] Paul, R.T.: Review of Robust Video Watermarking Techniques. *IJCA Special Issue on Computational Science - New Dimensions & Perspectives*, vol. 3, pp. 90–95, 2011.
Available at: `http://www.ijcaonline.org/nccse/number3/SPE169T.pdf`

[38] Bhattacharya, S., Chattopadhyay, T. and Pal, A.: A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H.264/AVC. In: *2006 IEEE International Symposium on Consumer Electronics*, pp. 1–6. IEEE, 2006. ISBN 1-4244-0216-6.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1689458`

[39] Pan, J. and Huang, H.: *Intelligent watermarking techniques*. World Scientific Publishing Co. Pte. Ltd., 2004. ISBN 9812389555.

[40] Jarnikov, D., Lourens, J. and Westerveld, E.: Server-guided watermarking for resource-constrained devices. In: *Consumer Electronics, 2009. ISCE'09. IEEE 13th International Symposium on*, vol. 36, pp. 76–80. IEEE, 2009. ISSN 0018-9316.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5157033`

[41] Wu, M., Trappe, W., Wang, Z. and Liu, K.: Collusion-Resistant Fingerprinting for Multimedia. *Signal Processing Magazine, IEEE*, vol. 21, no. 2, pp. 15–27, March 2004. ISSN 1053-5888.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1276103`

[42] He, S.: Collusion-resistant video fingerprinting for large user group. *Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 697–709, 2007.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4378258`

[43] Kundur, D.: Video Fingerprinting and Encryption Principles for Digital Rights Management. *Proceedings of the IEEE*, vol. 92, no. 6, 2004.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1299167`

[44] Boneh, D. and Shaw, J.: Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998. ISSN 00189448.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=705568`

[45] Shashanka, D. and Bora, P.K.: Collusion Secure Scalable Video Fingerprinting Scheme. In: *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, pp. 641–647. Ieee, December 2007. ISBN 0-7695-3059-1.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4426040`

[46] Kalker, T.: Video watermarking system for broadcast monitoring. In: *Proceedings of SPIE*, vol. 3657, pp. 103–112. SPIE, 1999. ISSN 0277786X.
Available at: `https://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.344661`

[47] Depovere, G., Kalker, T., Haitsma, J., Maes, M., de Strycker, L., Termont, P., Vandewege, J., Langell, A., Alm, C., Norman, P., O'Reilly, G., Howes, B., Vaanholt, H., Hintzen, R., Donnelly, P. and Hudson, A.: The VIVA project: digital watermarking for broadcast monitoring. In: *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 2, pp. 202–205. IEEE, 1999. ISBN 0-7803-5467-2.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=822884`

[48] De Strycker, L., Termont, P., Vandewege, J., Haitsma, J., Kalker, A., Maes, M. and Depovere, G.: Implementation of a real-time digital watermarking process for broadcast monitoring on a TriMedia VLIW processor. *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 147, no. 4, p. 371, 2000. ISSN 1350245X.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=872725http://link.aip.org/link/IVIPEK/v147/i4/p371/s1&Agg=doi`

[49] Miller, M., Cox, I. and Bloom, J.: Watermarking in the real world: an application to DVD. In: *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers (Cat. No.CH37020)*, vol. 2, pp. 1496–1502. IEEE, 1999. ISBN 0-7803-5700-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=831999`

[50] Leung, H.: Chaotic Watermarking for Video Authentication in Surveillance Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 704–709, May 2008. ISSN 1051-8215.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4454234`

[51] Yin, P. and Yu, H.H.: A semi-fragile watermarking system for MPEG video authentication. In: *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. IV–3461–IV–3464. IEEE, May 2002. ISBN 0-7803-7402-9.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5745399`

[52] Thiemert, S.: Using entropy for image and video authentication watermarks. In: *Proceedings of SPIE*, vol. 6072, pp. 607218–607218–10. SPIE, 2006. ISSN 0277786X.
Available at: `http://link.aip.org/link/PSISDG/v6072/i1/p607218/s1&Agg=doi`

[53] Craver, S., Memon, N., Yeo, B.-L. and Yeung, M.: Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 573–586, May 1998. ISSN 07338716.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=668979`

[54] Qiao, L. and Nahrstedt, K.: Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. *Journal of Visual Communication and Image Representation*, vol. 9, no. 3, pp. 194–210, September 1998. ISSN 10473203.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S1047320398903911`

[55] Qiao, L. and Nahrstedt, K.: Watermarking methods for MPEG encoded video: towards resolving rightful ownership. In: *Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No.98TB100241)*, pp. 276–285. IEEE Comput. Soc, 1998. ISBN 0-8186-8557-3.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=693656`

[56] Kudrle, S.: Media Fingerprinting : Managing Content, Security and Quality. Tech. Rep., Miranda Technologies, California, USA.

[57] Cano, P., Batlle, E., Kalker, T. and Haitsma, J.: A Review of Audio Fingerprinting. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 41, no. 3, pp. 271–284, November 2005. ISSN 0922-5773.
Available at: `http://www.springerlink.com/index/10.1007/s11265-005-4151-3`

[58] Boney, L., Tewfik, A. and Hamdy, K.: Digital watermarks for audio signals. *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pp. 473–480, 1996.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=535015`

[59] Mason, A., Salmon, R. and Devlin, J.: User requirements for watermarking in broadcast applications. In: *International Broadcasting Convention (IBC 2000), Amsterdam, The Netherlands*. 2000.
Available at: `http://www.bbc.co.uk/rd/pubs/papers/pdffiles/ibc00ajm.pdf`

[60] Cheveau, L., Goray, E. and Salmon, R.: Watermarking - Summary results of EBU tests. Tech. Rep. March 2001, European Broadcasting Union, 2001.
Available at: `http://tech.ebu.ch/docs/techreview/trev_286-cheveau.pdf`

[61] Barda, J. and Cheveau, L.: Access control and watermarking. *Ebu Technical Review*, vol. 282, no. 282, pp. 1–12, 2000.
Available at: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.8525`

[62] Zhang, L., Cao, Z. and Gao, C.: Application of RS-coded MPSK modulation scenarios to compressed image communication in mobile fading channel. In:

Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152), vol. 3, pp. 1198–1203. IEEE, 2000. ISBN 0-7803-6507-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=886291`

[63] Ambroze, a., Wade, G., Serdean, C., Tomlinson, M., Stander, J. and Borda, M.: Turbo code protection of video watermark channel. *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 148, no. 1, p. 54, 2001. ISSN 1350245X.
Available at: `http://link.aip.org/link/IVIPEK/v148/i1/p54/s1&Agg=doi`

[64] Su, J.K., Hartung, F. and Girod, B.: Digital watermarking of text, image, and video documents. *Computers & Graphics*, vol. 22, no. 6, pp. 687–695, December 1998. ISSN 00978493.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S0097849398000892`

[65] Arnold, M.: Attacks on digital audio watermarks and countermeasures. In: *Proceedings Third International Conference on WEB Delivering of Music*, pp. 55–62. IEEE Comput. Soc, 2003. ISBN 0-7695-1935-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1233874`

[66] Martin, K. and Sviatoslav, V.: The watermark copy attack. *Proc. of the SPIE, Security and Watermarking*, 2000.
Available at: `http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+watermark+copy+attack#8`

[67] Wolfgang, R., Podilchuk, C. and Delp, E.: Perceptual watermarks for digital images and video. *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, July 1999. ISSN 00189219.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771067`

[68] Barda, J. and Cheveau, L.: Eurovision-network security through access control and watermarking. Tech. Rep., European Broadcasting Union, 1999.
Available at: `http://tech.ebu.ch/docs/techreview/trev_281-barda.pdf`

[69] Jack, K.: *Video Demystified: A Handbook for the Digital Engineer*. Fifth edit edn. Elsevier Inc., 2007. ISBN 978-0-7506-8395-1.
Available at: `http://books.google.com/books?hl=en&lr=`

&id=BtCOgrpzbz4C&oi=fnd&pg=PP2&dq=Video+Demystified:+A+Handbook+for+the+Digital+Engineer&ots=W8MWpSbHWy&sig=IOfWnwAaGja1TR2oDdOzN19YpGk

[70] Eskicioglu, A. and Fisher, P.: Image quality measures and their performance. *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, 1995. ISSN 00906778.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=477498

[71] Trussell, H., Saber, E. and Vrhel, M.: Color image processing : Basics and special issue overview. *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 14–22, January 2005. ISSN 1053-5888.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1407711

[72] Jayant, N., Johnston, J. and Safranek, R.: Signal compression based on models of human perception. *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1385–1422, 1993. ISSN 00189219.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=241504

[73] Blakemore, C. and Campbell, F.: On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images. *The Journal of Physiology*, pp. 237–260, 1969.
Available at: http://jp.physoc.org/content/203/1/237.short

[74] Lubin, J., Brill, M. and Crane, R.: Vision model-based assessment of distortion magnitudes in digital video. pp. 1–13, 1996.
Available at: http://www.mpeg.org/MPEG/JND/

[75] Lubin, J. and Fibush, D.: Sarnoff JND Vision Model. 1997.
Available at: http://www.videoclarity.com/PDF/Sarnoffjnd-1.pdf

[76] Watson, A.B.: The cortex transform: Rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 311–327, September 1987. ISSN 0734189X.
Available at: http://linkinghub.elsevier.com/retrieve/pii/S0734189X87801846

[77] Bashinski, H.S. and Bacharach, V.R.: Enhancement of perceptual sensitivity as the result of selectively attending to spatial locations. *Perception & Psy-*

chophysics, vol. 28, no. 3, pp. 241–248, May 1980. ISSN 0031-5117.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/7433002http://www.springerlink.com/index/10.3758/BF03204380`

[78] Lin, W. and Yang, X.: Modeling visual attention's modulatory aftereffects on visual sensitivity and quality evaluation. *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1928–1942, November 2005. ISSN 1057-7149.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1518955`

[79] Solomon, J.a.: The effect of spatial cues on visual sensitivity. *Vision research*, vol. 44, no. 12, pp. 1209–16, June 2004. ISSN 0042-6989.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/15066386`

[80] Liu, A., Verma, M. and Lin, W.: Modeling the masking effect of the human visual system with visual attention model. In: *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)*, 2, pp. 1–5. IEEE, December 2009. ISBN 978-1-4244-4656-8.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5397653`

[81] Virsu, V., Rovamo, J., Laurinen, P. and Näsänen, R.: Temporal contrast sensitivity and cortical magnification. *Vision Research*, vol. 22, no. 9, pp. 1211–1217, January 1982. ISSN 00426989.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/0042698982900876`

[82] Swanson, W.H., Ueno, T., Smith, V.C. and Pokorny, J.: Temporal modulation sensitivity and pulse-detection thresholds for chromatic and luminance perturbations. *Journal of the Optical Society of America A*, vol. 4, no. 10, p. 1992, October 1987. ISSN 1084-7529.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/3430210http://www.opticsinfobase.org/abstract.cfm?URI=josaa-4-10-1992`

[83] Lee, S.-j. and Jung, S.-h.: A survey of watermarking techniques applied to multimedia. In: *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570)*, vol. 1, pp. 272–277. IEEE, 2001. ISBN 0-7803-7090-2.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=931796`

[84] Doërr, G. and Dugelay, J.-L.: A guide tour of video watermarking. *Signal Processing: Image Communication*, vol. 18, no. 4, pp. 263–282, April 2003. ISSN 09235965.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S0923596502001443`

[85] JAYAMALAR, T. and RADHA, V.: Survey on Digital Video Watermarking Techniques and Attacks on Watermarks. *International Journal of Engineering Science and Technology*, vol. 2, no. 12, pp. 6963–6967, 2010.
Available at: `http://www.ijest.info/docs/IJEST10-02-12-050.pdf`

[86] Lancini, R., Mapelli, F. and Tubaro, S.: A robust video watermarking technique in the spatial domain. In: *International Symposium on VIPromCom Video/Image Processing and Multimedia Communications*, June, pp. 251–256. Croatian Soc. Electron. Marine, 2002. ISBN 953-7044-01-7.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1026664`

[87] Niu, X.: Video watermarking resistance to rotation, scaling, and translation. In: *Proceedings of SPIE*, vol. 4675, pp. 512–519. SPIE, 2002. ISSN 0277786X.
Available at: `http://link.aip.org/link/?PSI/4675/512/1&Agg=doi`

[88] Su, K., Kundur, D. and Hatzinakos, D.: A novel approach to collusion resistant video watermarking. In: *Proceedings of SPIE*, vol. 4675, p. 491. Spie, 2002. ISSN 0277786X.
Available at: `http://www.ee.tamu.edu/~deepa/pub/SuKunHatSPIE02.pdf`

[89] Hartung, F. and Girod, B.: Watermarking of uncompressed and compressed video. *Signal Processing*, vol. 66, no. 3, pp. 283–301, May 1998. ISSN 01651684.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S0165168498000115`

[90] Barni, M., Bartolini, F., De Rosa, A. and Piva, A.: Optimum decoding and detection of multiplicative watermarks. *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1118–1123, April 2003. ISSN 1053-587X.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1188753`

[91] Su, J.K.: Fundamental performance limits of power-spectrum condition-compliant watermarks. In: *Proceedings of SPIE*, vol. 3971, pp. 314–325. SPIE,

2000. ISSN 0277786X.
Available at: https://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.384986

[92] Liu, Y. and Zhao, J.: A new video watermarking algorithm based on 1D DFT and Radon transform. *Signal Processing*, vol. 90, no. 2, pp. 626–639, February 2010. ISSN 01651684.
Available at: http://linkinghub.elsevier.com/retrieve/pii/S0165168409003454

[93] Deguillaume, F.: Robust 3D DFT video watermarking. In: *Proceedings of SPIE*, vol. 3657, pp. 113–124. SPIE, 1999. ISSN 0277786X.
Available at: http://link.aip.org/link/?PSI/3657/113/1&Agg=doi

[94] Wang, H.-X., Lu, Z.-M., Sun, S.-H., Simos, T.E. and Psihoyios, G.: A Blind Video Watermarking Algorithm Based on SVD in the DCT Domain. *AIP Conference Proceedings*, vol. 2007, pp. 360–364, 2008.
Available at: http://link.aip.org/link/APCPCS/v1060/i1/p360/s1&Agg=doi

[95] Wu, D., Kong, W., Yang, B. and Niu, X.: A fast SVD based video watermarking algorithm compatible with MPEG2 Standard. *Soft Computing*, vol. 13, no. 4, pp. 375–382, June 2008. ISSN 1432-7643.
Available at: http://www.springerlink.com/index/10.1007/s00500-008-0328-6

[96] Kaufman, J. and Celenk, M.: Digital Video Watermarking using Singular Value Decomposition and 2D Principal Component Analysis. In: *2006 International Conference on Image Processing*, pp. 2561–2564. IEEE, 2006. ISBN 1-4244-0480-0.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4107091

[97] Rajab, L., Al-Khatib, T. and Al-Haj, A.: Video Watermarking Algorithms Using the SVD Transform. *European Journal of Scientific Research*, vol. 30, no. 3, pp. 389–401, 2009.
Available at: http://www.eurojournals.com/ejsr_30_3_05.pdf

[98] Kong, W., Yang, B., Wu, D. and Niu, X.: SVD Based Blind Video Watermarking Algorithm. In: *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, vol. 1, pp. 265–268. IEEE,

2006. ISBN 0-7695-2616-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1691791`

[99] Chan, P. and Lyu, M.: A DWT-Based Digital Video Watermarking Scheme with Error Correcting Code, Information and Communications Security, Lecture Notes in Computer Science. In: Qing, Sihan and Gollman (eds.), *Information and Communications Security*, pp. 202–213. Springer, Berlin / Heidelberg, 2003. ISBN 978-3-540-20150-2.
Available at: `http://www.springerlink.com/index/MU1P1FGKD3UJCVJM.pdf`

[100] Lyu, M. and Chin, R.: A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1638–1649, December 2005. ISSN 1051-8215.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1546010http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1546010`

[101] Shi, Y.: A robust DWT-based video watermarking algorithm. In: *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, vol. 3, pp. III–631–III–634. IEEE, 2002. ISBN 0-7803-7448-7.
Available at: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1010303http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1010303`

[102] Ejima, M. and Miyazaki, A.: A wavelet-based watermarking for digital images and video. In: *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 0, pp. 678–681. IEEE, 2000. ISBN 0-7803-6297-7.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=899545`

[103] Serdean, C., Ambroze, M., Tomlinson, M. and Wade, J.: DWT-based high-capacity blind video watermarking, invariant to geometrical attacks. *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 150, no. 1, p. 51, 2003. ISSN 1350245X.
Available at: `http://link.aip.org/link/IVIPEK/v150/i1/p51/s1&Agg=doi`

[104] Zhu, W., Xiong, Z. and Zhang, Y.-q.: Multiresolution watermarking for images and video. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 545–550, June 1999. ISSN 10518215.

Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=767121

[105] Inoue, H., Miyazaki, A., Araki, T. and Katsura, T.: A digital watermark method using the wavelet transform for video data. In: *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349)*, vol. 4, pp. 247–250. IEEE, 2000. ISBN 0-7803-5471-0.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=779988

[106] Langelaar, G.C. and Lagendijk, R.L.: Optimal differential energy watermarking of DCT encoded images and video. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 1, pp. 148–58, January 2001. ISSN 1057-7149.
Available at: http://www.ncbi.nlm.nih.gov/pubmed/18249605

[107] Marziliano, P. and Ho, A.: A hybrid watermarking scheme for H.264/AVC video. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pp. 865–868 Vol.4. IEEE, 2004. ISBN 0-7695-2128-2.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1333909

[108] Zhang, J., Ho, A.T.S., Qiu, G. and Marziliano, P.: Robust Video Watermarking of H.264/AVC. *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 2, pp. 205–209, February 2007. ISSN 1057-7130.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4100886

[109] Deng, C., Gao, X., Li, X. and Tao, D.: A local Tchebichef moments-based robust image watermarking. *Signal Processing*, vol. 89, no. 8, pp. 1531–1539, August 2009. ISSN 01651684.
Available at: http://linkinghub.elsevier.com/retrieve/pii/S0165168409000474

[110] Farzam, M. and Shirani, S.: A robust multimedia watermarking technique using Zernike transform. In: *2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564)*, vol. 9140, pp. 529–534. IEEE, 2001. ISBN 0-7803-7025-2.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=962787

[111] Mirza, H. and Thai, H.: Digital Video Watermarking Based on RGB Color Channels and Principal Component Analysis. *Knowledge-Based Intelligent Information and*, pp. 125–132, 2010.
Available at: `http://www.springerlink.com/index/WL6R8QK15832XR37.pdf`

[112] Pereira, S., Ruanaidh, J., Deguillaume, F., Csurka, G. and Pun, T.: Template based recovery of Fourier-based watermarks using log-polar and log-log maps. In: *Proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 1, pp. 870–874. IEEE Computer Society, 1999. ISBN 0-7695-0253-9.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=779316`

[113] Pereira, S. and Pun, T.: Fast robust template matching for affine resistant image watermarks. *IEEE Transactions on Image Processing*, vol. 9, pp. 1123—-1129, 1999.
Available at: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.2404`

[114] van Huyssteen, R.H.: Online Resources for the thesis: Comparative Evaluation of Video Watermarking Algorithms in the Uncompressed Domain. 2012.
Available at: `http://ml.sun.ac.za/~hendrikvh/`

[115] John, G. and Dimitris, M.: *Digital Signal Processing: Principles, Algorithms and Applications*. 4th edn. Pearson Education, Inc., 2007. ISBN 013228731.

[116] Gold, R.: Optimal binary sequences for spread spectrum multiplexing (Corresp.). *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619–621, October 1967. ISSN 0018-9448.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1054048`

[117] Chandra, D.: Digital image watermarking using singular value decomposition. In: *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, vol. 3, pp. III–264–III–267. IEEE, 2002. ISBN 0-7803-7523-8.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1187023`

[118] Bensoussan, A. and Lions, J.L. (eds.): *Analysis and Optimization of Systems - Proceedings of the Sixth International Conference on Analysis and Optimization of Systems, Nice, June 19-22, 1984*, vol. 62 of *Lecture Notes in Control*

and Information Sciences. Springer-Verlag, Berlin/Heidelberg, 1984. ISBN 3-540-13551-0.
Available at: http://www.springerlink.com/index/10.1007/BFb0004939

[119] Andrews, H. and Patterson, C.: Singular value decompositions and digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 26–53, February 1976. ISSN 0096-3518.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1162766

[120] Kalman, D.: A Singularly Valuable Decomposition: The SVD of a Matrix. *The College Mathematics Journal*, vol. 27, no. 1, p. 2, January 1996. ISSN 07468342.
Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.113.1193

[121] Mallat, S.: Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 2091–2110, 1989. ISSN 00963518.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=45554

[122] Antonini, M. and Barlaud, M.: Image coding using wavelet transform. *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205—-220, 1992.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=136597

[123] Koenderink, J.J.: The structure of images. *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, August 1984. ISSN 0340-1200.
Available at: http://www.springerlink.com/index/10.1007/BF00336961

[124] Watson, A., Yang, G. and Solomon, J.: Visual thresholds for wavelet quantization error. *SPIE Proceedings*, vol. 2657, pp. 382–392, 1996.
Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.5189&rep=rep1&type=pdf

[125] Daubechies, I.: Ten Lectures on Wavelets. January 1992.
Available at: http://epubs.siam.org/doi/book/10.1137/1.9781611970104

[126] Mallat, S.: *A wavelet tour of signal processing*. Academic Press, 1999. ISBN 012466606X.
Available at: http://books.google.com/books?hl=en&

lr=&id=yW2kut44AsMC&oi=fnd&pg=PR15&dq=A+Wavelet+Tour+Of+Signal+Processing&ots=Fnqv50rd6t&sig=ln0hfi5tkhOUYfoh-Z9rrj2xuIQ

[127] Pinson, M. and Wolf, S.: A New Standardized Method for Objectively Measuring Video Quality. *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, September 2004. ISSN 0018-9316.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1337089

[128] Damera-Venkata, N., Kite, T.D., Geisler, W.S., Evans, B.L. and Bovik, a.C.: Image quality assessment based on a degradation model. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 9, no. 4, pp. 636–50, January 2000. ISSN 1057-7149.
Available at: http://www.ncbi.nlm.nih.gov/pubmed/18255436

[129] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004. ISSN 1057-7149.
Available at: http://www.ncbi.nlm.nih.gov/pubmed/15376593http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1284395

[130] Winkler, S. and Mohandas, P.: The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics. *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, September 2008. ISSN 0018-9316.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4550731

[131] Martínez, J. and Cuenca, P.: Objective video quality metrics: A performance analysis. *Proceedings of the 14th European Signal Processing Conference*, 2006.
Available at: https://investigacion.uclm.es/documentos/fi_1162815334-eusipco06%5Bcamararready2%5D.pdf

[132] Thakur, M.K., Saxena, V. and Gupta, J.P.: A performance analysis of objective video quality metrics for digital video watermarking. In: *2010 3rd International Conference on Computer Science and Information Technology*, pp. 12–17. IEEE, July 2010. ISBN 978-1-4244-5537-9.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5564962

[133] Wang, Y.: Survey of objective video quality measurements. *EMC Corporation Hopkinton, MA*, vol. 1748, pp. 1–7, 2006.
Available at: `ftp://130.215.28.31/pub/techreports/pdf/06-02.pdf`

[134] Huynh-Thu, Q. and Ghanbari, M.: Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, vol. 44, no. 13, p. 800, 2008. ISSN 00135194.
Available at: `http://link.aip.org/link/ELLEAK/v44/i13/p800/s1&Agg=doi`

[135] Eckert, M.P. and Bradley, A.P.: Perceptual quality metrics applied to still image compression. *Signal Processing*, vol. 70, no. 3, pp. 177–200, November 1998. ISSN 01651684.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S0165168498001248`

[136] Wang, Z., Bovik, A.C. and Lu, L.: Why is image quality assessment so difficult? In: *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. IV–3313–IV–3316. IEEE, May 2002. ISBN 0-7803-7402-9.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5745362`

[137] Wang, Z., Lu, L. and Bovik, A.C.: Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004. ISSN 09235965.
Available at: `http://linkinghub.elsevier.com/retrieve/pii/S0923596503000766`

[138] Girod, B.: *Digital images and human vision*. MIT press, Cambridge, MA, USA, 1993. ISBN 0-262-23171-9.
Available at: `http://dl.acm.org/citation.cfm?id=197784`

[139] Teo, P. and Heeger, D.: Perceptual image distortion. In: *Proceedings of 1st International Conference on Image Processing*, vol. 2, pp. 982–986. IEEE Comput. Soc. Press, 1994. ISBN 0-8186-6952-7.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=413502`

[140] Wang, Z. and Bovik, A.: Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, January 2009. ISSN 1053-5888.

Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4775883`

[141] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P.: The SSIM Index for Image Quality Assessment. 2003.
Available at: `http://www.cns.nyu.edu/~lcv/ssim/`

[142] Wang, Z. and Shang, X.: Spatial Pooling Strategies for Perceptual Image Quality Assessment. In: *2006 International Conference on Image Processing*, vol. 1, pp. 2945–2948. IEEE, 2006. ISBN 1-4244-0480-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4107187`

[143] Moorthy, A.K. and Bovik, A.C.: Visual Importance Pooling for Image Quality Assessment. *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 193–201, April 2009. ISSN 1932-4553.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4799310`

[144] Wang, Z. and Li, Q.: Information content weighting for perceptual image quality assessment. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 20, no. 5, pp. 1185–98, May 2011. ISSN 1941-0042.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/21078577`

[145] Wolf, S. and Pinson, M.: Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system. *Proc. SPIE*, 1999.
Available at: `http://www.its.bldrdoc.gov/pub/n3/video/spie99.pdf`

[146] Cox, I. and Linnartz, J.-P.: Some general methods for tampering with watermarks. *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 587–593, May 1998. ISSN 07338716.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=668980`

[147] Xiph.org: Xiph.org Test Media Repository. 2012.
Available at: `http://media.xiph.org/video/derf/`

[148] The MathWorks, I.: Analyzing Your Program' s Performance. 2012.
Available at: `http://www.mathworks.com/help/techdoc/matlab_prog/f8-790895.html`

[149] Lin, E. and Delp, E.: Temporal Synchronization in Video Watermarking. *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3007–3022, October 2004. ISSN 1053-587X.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1337278`

[150] European Broadcasting Union: EBU-Recommendation R95-2008: Safe areas for 16:9 television production. Tech. Rep. September, 2008.
Available at: `http://tech.ebu.ch/docs/r/r095.pdf`

[151] Arnold, M.: Audio watermarking: features, applications and algorithms. In: *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 2, pp. 1013–1016. IEEE, 2000. ISBN 0-7803-6536-4.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=871531`

[152] Gerig, G., Kubler, O., Kikinis, R. and Jolesz, F.a.: Nonlinear anisotropic filtering of MRI data. *IEEE transactions on medical imaging*, vol. 11, no. 2, pp. 221–32, January 1992. ISSN 0278-0062.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/18218376`

[153] The Mathworks Inc: 2-D adaptive noise-removal filtering - MATLAB. 2012.
Available at: `http://www.mathworks.com/help/images/ref/wiener2.html`

[154] Lim, J.A.E.S.: *Two-Dimensional Signal and Image Processing*. Prentice Hall PTR, New Jersey, USA, 1990. ISBN 0139353224.

[155] Handbrake Project: HandBrake. 2012.
Available at: `http://handbrake.fr`

[156] Wiegand, T., Sullivan, G., Bjontegaard, G. and Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003. ISSN 1051-8215.

[157] Haskell, B., Puri, A. and Netravali, A.: *Digital video: an introduction to MPEG-2*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996. ISBN 0412084112.

[158] Herrigel, A., Voloshynovskiy, S. and Rytsar, Y.: The watermark template attack. *Proc. SPIE Security and Watermarking of Multimedia Contents III*, vol.

4314, pp. 394—-405, 2001. ISSN 0277786X.
Available at: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.5902&rep=rep1&type=pdf`

[159] Lin, C.Y., Wu, M., Bloom, J.a., Cox, I.J., Miller, M.L. and Lui, Y.M.: Rotation, scale, and translation resilient watermarking for images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 5, pp. 767–82, January 2001. ISSN 1057-7149.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/18249666`

[160] Solachidis, V. and Pitas, L.: Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 11, pp. 1741–53, January 2001. ISSN 1057-7149.
Available at: `http://www.ncbi.nlm.nih.gov/pubmed/18255515`

[161] Alghoniemy, M. and Tewfik, A.: Image watermarking by moment invariants. *Proceedings 2000 International Conference on Image Processing*, pp. 73–76 vol.2, 2000.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=899229`

[162] Chen, Q.: Robust image watermarking with Zernike moments. In: *Canadian Conference on Electrical and Computer Engineering, 2005.*, May, pp. 1340–1343. IEEE, 2005. ISBN 0-7803-8885-2.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1557225`

[163] Guo-juan, X. and Rang-ding, W.: A Blind Video Watermarking Algorithm Resisting to Rotation Attack. *2009 International Conference on Computer and Communications Security*, pp. 111–114, December 2009.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5380290`

[164] Singhal, N., Lee, Y.-Y., Kim, C.-s. and Lee, S.-u.: Robust Image Watermarking Based on Local Zernike Moments. In: *2007 IEEE 9th Workshop on Multimedia Signal Processing*, x, pp. 401–404. IEEE, 2007. ISBN 978-1-4244-1273-0.
Available at: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4412901`

[165] Liao, S. and Pawlak, M.: On the accuracy of Zernike moments for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

vol. 20, no. 12, pp. 1358–1364, 1998. ISSN 01628828.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=735809http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=735809

[166] Chong, C.-W., Raveendran, P. and Mukundan, R.: A comparative analysis of algorithms for fast computation of Zernike moments. *Pattern Recognition*, vol. 36, no. 3, pp. 731–742, March 2003. ISSN 00313203.
Available at: http://linkinghub.elsevier.com/retrieve/pii/S0031320302000912

[167] O'Ruanaidh, J. and Pun, T.: Rotation, scale and translation invariant digital image watermarking. In: *Proceedings of International Conference on Image Processing*, vol. 1, pp. 536–539. IEEE Comput. Soc, 1997. ISBN 0-8186-8183-7.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=647968

[168] Petitcolas, F., Anderson, R. and Kuhn, M.: Attacks on copyright marking systems. In: *Information Hiding*, vol. 1525 of *Lecture Notes in Computer Science*, pp. 218–238. Springer Berlin / Heidelberg, 1998. ISBN 978-3-540-65386-8.
Available at: http://dx.doi.org/10.1007/3-540-49380-8_16

[169] Kim, H.S. and Lee, H.-k.: Invariant image watermark using zernike moments. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 766–775, August 2003. ISSN 1051-8215.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1227606

Appendix A

Additional Parameters

Table A.1: Parameters used for SSIM evaluation in Section 4.2.3.

Parameter	Value
K_1	0.01
K_2	0.03
F	4
Sliding Window	Rotationally symmetric Gaussian window with size 11 pixels and standard deviation of 1.5 pixels.

Table A.2: Parameters used for interframe flicker metric developed in Section 4.3.

Parameter	Value
Edge detection method	Roberts method using automatic edge threshold.
Averaging filter	Spatial averaging filter with a size of 2×2 pixels. All pixels are equally weighted in the averaging process.
Median filter	Two-dimensional median filter which calculates the median value in a 5×5 neighbourhood around each pixel in the input image.

Table A.3: Hardware configuration used for the evaluation of techniques.

Parameter	Specification
Computer Model	MacBook Pro 8,2
Processor	2.2 GHz Quad-core Intel Core i7-2720QM
Memory	8 gigabytes, 1333MHz DDR3
Storage	Intel 330 Series 120GB solid state drive
Operating System	Mac OS X 10.7.4

Table A.4: Parameters used for the embedding of watermarks.

SD		DFT		SVD		DWT	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
α_{SD}	0.0052	α_{DFT}	8.3	α_{SVD}	0.13	α_{DWT}	31
		r_{min}	0.3	λ_{min}	13	T	100
		r_{max}	0.6				
		d_{min}	2				
		d_{max}	2				

Table A.5: Parameters used for compression of video sequences as discussed in Section 5.8.

Parameter	Value
Compression Codec	H.264
Bit Rate Setting	Constant bit rate
Number of Passes	2
Reference Frames	2
Maximum B-Frames	2
CABAC Entropy Coding	Enabled
8×8 Transform	Disabled
Weighted P-Frames	Enable
Pyramidal B-Frames	Default (Normal)
No DCT Decimation	Disabled
Parameter	Value
Adaptive B-Frames	Default (Fast)
Partition Types	Default (Most)
Adaptive Direct Mode	Default (Spatial)
Trellis	Off
Motion Estimation Method	Default (Hexagon)
Subpixel ME & Mode Decision	6: RD in I/P-frames
x264 Advanced Option String	ref=2:bframes=2:subq=6:mixed-refs=0:weightb=0:8x8dct=0:trellis=0