A collusion attack on digital video watermarks based on the replacement strategy

by

Jonathan Malm

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Information Assurance

Program of Study Committee:
Johnny Wong, Major Professor
Cliff Bergman
Jim Davis
Doug Jacobson

Iowa State University
Ames, Iowa
2003

Graduate College
Iowa State University

This is to certify that the master's thesis of

Jonathan Malm

has met the thesis requirements of Iowa State University

e been redacted for privacy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Digital works such as images, audio and video present security concerns due to their portability and error free reproducibility. Thus, digital work producers are not being properly compensated for copyrighted works that are illegally copied and distributed on the Internet. One solution that has been proposed to solve some of these problems is digital watermarking. Researchers have proposed many different watermarking methods, but for any of these methods to be commercially applicable, they must be secure in the sense of being resilient to all known watermarking attacks. Therefore, the exploration and examination of watermarking attacks must be exhaustive. This paper adds to the knowledge base of known watermarking attacks on digital video. Specifically a type of collusion attack based on the replacement attack strategy is applied and tested against two digital video watermarking schemes. The effectiveness of this attack is measured by evaluating the fidelity of the attacked video as well as the ability of the attack to remove the watermark. This attack will provide yet another quality standard for measuring the effectiveness of watermarking schemes. This standard must be met if watermarking is to be a commercially viable option.

CHAPTER 1. Introduction

1.1 Motivation

Since the advent of digital works there has been an increased interest of digital watermarking. Digital technologies such as streaming audio and video, personal video recorders, DVD, and MPEG compression standards present new security concerns due to their portability and error free reproducibility. One solution that has been proposed to solve some of these problems is digital watermarking. Digital watermarking is the process of hiding data within a digital work, such that the data is imperceptible to the user of the work, difficult to remove without destroying the work, but detectible by applying a specific detection algorithm [4].

Over the past decade, researchers have proposed many watermarking methods that can be used to solve these problems. If these methods are to be commercially applicable, they must be secure in sense of being resilient to all known watermarking attacks. Therefore, an evaluation of the effectiveness of digital watermarking schemes demands an exhaustive exploration of watermarking attacks. It is the purpose of this paper to evaluate the effectiveness of a known attack strategy applied to digital video watermarks.

1.2 Contribution

Most research on digital watermarking has been done on images. Although these watermarks can be readily applied to video by separating the video into frames, digital video watermarking presents a new set of problems and attacks that are not applicable to images. One attack that is exclusive to video is the collusion attack. A collusion attack attempts to create a watermark-free work by using redundant data which is contained not only within individual frames but also between video frames. Digital video watermarks are more susceptible to collusion attacks than image watermarks because of the amount of data contained within a video and the redundancy between frames. The current United States standard, NTSC, plays video at a rate of 29.97 frames per second. [18] This rate provides ample redundancy for a collusion attack within a fraction of a second of video.

This paper will add to the knowledge base of known watermarking attacks on digital video. Specifically a type of collusion attack based on the replacement attack strategy will be applied and tested against digitial video watermarks. The replacement attack was first introduced by Kirovski and Petitcolas in their paper, *Replacement Attack on Arbitrary Watermarking Systems* which implemented a replacement attack on digital audio [13]. This paper will build on the replacement attack by extending it to digital video and testing its effectiveness against two digital video watermarking techniques. The effectiveness of this attack will be measured by evaluating the fidelity of the attacked video as well as the ability of the attack to successfully remove the watermark.

Implementing this attack on digital video instead of digital audio presents several problems not covered by Kirovski and Petitcolas. First, digital audio is a one dimensional signal while video is a three dimensional signal and thus the blocks used in the attack must be adjusted. Also, fidelity concerns differ when working with digital video rather than digital audio.

This attack will provide yet another quality standard for measuring the security of future watermarking schemes. This standard must be met if watermarking is to be a commercially viable option.

1.3 Roadmap

Chapter 2 discusses proposed application areas of watermarks. Chapter 3 defines watermarking by looking at the process and how it may be evaluated. Chapter 4 reviews two video watermarking algorithms and presents different types of attacks on digital video watermarks. Chapter 5 presents the attack applied in this work and how the attack will be evaluated. Chapter 6 presents the results of the attack and a discussion of the results. Chapter 7 sums up the contribution of this paper and presents areas for future work.

CHAPTER 2. Watermarking Applications

Digital watermarking is a new area of research and has only recently begun to attract widespread interest. Papers published on watermarking have gone from a few in the early 90s to over 250 in 1999 [4]. However, traditional paper watermarking has been around since the 13th century [4]. Traditional paper watermarking has many applications today. A prominent example of paper watermarking is the American currency system. If a \$5, \$10 or \$20 bill is held up to the light a faint portrait of a US president is visible on the right side of the bill. This image is a watermark embedded in the bills to prevent counterfeiting.

Parallels can be drawn between this type of watermark and a digital watermark. For example, the watermark contained on a \$20 bill is not visible under normal conditions. It can only been seen when a detection process is applied to the bill. In the case of currency, this process is holding it up to the light [4]. Digital watermarking attempts to be impreceptible as well. This property of unobtrusiveness is a fundamental property of digital watermarking. Under normal circumstances, while viewing or listening to a digital work the watermark should not be detectable by the user. The watermark should only be detectable by applying an extraction algorithm. Money watermarks are also extremely hard to remove without destroying the bill. This robustness property is also desired in digital watermarking. A user should not be able to destroy or remove a digital watermark without destroying the digital work.

Traditional paper watermarks are useful today and serve a clear purpose in protecting content. This type of watermarking has been in use for a long time, but it is only recently that it has been applied to digital works. However, the application of this concept to digital works remains uncertain. Whether, and by what means, digital watermarks can be effectively used in today's digital world is not yet established. There have been many watermarking schemes proposed, but their ultimate efficacy as a means of protecting digital content is not yet proven.

2.1 Copyright Protection and Proof of Ownership

One of major problem that has eluded solution by digital work producers has been copyright protection. Since digital works are merely binary data, it can easily be copied and distributed on the Internet or on physical media without loss of quality [7]. Digital watermarking is proposed to solve this problem by embedding a watermark signal into the digital works such that every time the digital work is copied the watermark will also be copied.

In the past, copyrighted material could be identified under law by simply placing a copyright notice in the form "© *date owner*" somewhere within the content. The same concept can be applied to digital work but this type of protection is easily defeated by simply cropping the photo such that the copyright symbol is no longer visible or by otherwise deleting the notice [4]. In view of this, watermarking has been proposed to solve several areas of copyright protection.

Most watermarks are made to be imperceptual to the user and hard to extract without knowing some information about the watermark. This property makes them effective copy protection tools. Users are able to show their ownership of the work by showing the existence of their watermark within the work in question but unauthorized duplicators are not able to remove the watermarks without destroying the fidelity of the work.

Suppose Alice has a photo that she would like to publish but still protect from being used without her consent. Bob decides he would like to use the photo but does not obtain Alice's consent. In the past Bob would be able to remove the copyright symbol and claim ownership of the photo. Alice would then have to provide some form of original copy such as a negative to prove that she was the original creator [4]. Suppose instead that Alice watermarks her photo before releasing it to the public. When Bob claims the photo is his, Alice is able to refute the argument with digital proof that the photo is hers via the watermark. The above example is one way in which digital watermarking may be used for copyright protection and proof of ownership.

2.2 Transaction Tracking

Transaction tracking is another application of digital watermarking. In this type of watermark application, a unique watermark is placed in each copy of the distributed work [4]. Then, if a work is misused, the owner is able to track down the original malicious user by extracting the watermark and linking the watermark to a specific user. This type of application would be useful for example in tracking persons who illegally distribute copyrighted works over the Internet.

The DivX Corporation applied this technology in its service which offered a pay per view movie scheme [4]. In an attempt to track illegally recorded and distributed movies, DivX players would embed a watermark into the video as it was played. If the video was illegally recorded and pirated the watermark could be extracted from pirated DivX videos and tracked back to the original user [4].

2.3 Copy Control

Copy control attempts to prevent users from making illegal copies of digital works. Traditionally, encryption has been the main way in which to accomplish this task [4]. A "Content Scrambling System" or CSS was implanted on DVD video in order to thwart malicious users from making illegal copies of digital video. This encryption has been defeated by reverse engineering DVD players. Thus DVDs are now easily copied and distributed on the Internet [16].

Digital television producers are also looking for an effective method of copy control. The Federal Communications Commission has mandated that all television broadcasts go digital by the year 2006 [10]. Converting to digital television will benefit consumers by providing a better product and increasing the amount of data that can be sent within the scarce frequency space allocated for digital television. However, as digital television becomes reality, it also becomes clear that it will be easier to copy and redistribute without loss of quality. A strong copy control watermark could help solve the problem of illegal digital television copying and distribution.

Watermarking has been proposed to solve the problem of copy control. By inserting a watermark into digital works, producers are able to control which devices or software are able to read the digital data. For example, suppose a software media player has a built in watermark detector. Before any work is played, the player attempts to detect a watermark within the work. If a watermark is found, then the player decodes the watermark to receive instructions about handling the work. The instructions may tell the player that the work may be copied once or that may be viewable for a certain time period.

The major hurdle to this is convincing producers to manufacture and consumers to buy specific hardware and software that is able to detect watermarks. If there is hardware or software that does not have a watermark detector, the watermark becomes useless. Bloom et al. proposed a solution for this by coupling a mark on the physical media along with the digital watermark embedded into the work [1].

2.4 Content Authentication

This application attempts to preserve the integrity of the digital work. Again, in today's digital world, it is easy to manipulate digital works. Programs like Adobe Photoshop^{TM}and The Gimp make it easy to modify images. Watermarking proposes to solve this problem by inserting a watermark such that any modification of the digital works will destroy the watermark and it will therefore be provable that the work has been altered.

In the past cryptography has been used to achieve integrity of digital works [4]. Digital signatures and hashing have been proven as an effective way to achieve digital integrity. However, cryptography requires that additional information be included along with the digital work. If this additional information is corrupted but the digital work remains intact, then the integrity of the work is lost. Another weakness is the inflexibility of digital signatures or hashes. For example, if a picture undergoes JPEG or GIF compression the digital signature or hash would show that the digital work has been corrupted even though the physical representation of the work is still in tact. A watermark that is able to survive compression without a major change in physical representation of the digital work would be useful in this case.

This would be useful in the area of forensics. Law enforcement could use watermarking to ensure the integrity of digital works used for investigations. This can also be applied to any digital work that is distributed on the Internet.

2.5 Broadcast Monitoring

Broadcast television and radio have become big business in recent years. It is estimated that television broadcasts by companies such as Reuters, CNN, and the Associated Press have a value of over $100,000 per hour [11]. Currently there is no efficient way to track distribution of a broadcast or to verify its delivery. Watermarking has been proposed to protect the distribution of these broadcasts. By embedding a watermark into a digital broadcast, companies hope to solve distribution problems presented by broadcasting.

One problem that broadcast monitoring looks to solve is verifying the delivery of broadcasts. In 1997, it was reported that for over 20 years several major television stations in Japan had been overbooking advertising air time, thus selling advertising they did not in fact deliver [12]. This had gone undetected mainly because there is no way to ensure that advertisements are successfully sent to consumers.

A simple way to do broadcast monitoring is to have human observers watch the broadcast and record what they see [4]. However, this method is inefficient and highly un-scalable. Watermarking, on the other hand, has been proposed to solve these problems in an efficient manner. Watermarking algorithms such as JAWS, Just Another Watermarking System, have been introduced with the specific goal of broadcast monitoring [11]. By inserting imperceptible data into a broadcast, companies will be able to easily monitor what is being broadcast, or more importantly, what is not being broadcast.

An example implementation of this would be if each consumer has a set-top box that is able to detect the watermark in a broadcast. This box can keep statistics about what was watched and what was not watched in an efficient manner. These data can then be sent back to the broadcaster for processing.

CHAPTER 3. Watermarking Principles, Properties and Attacks

This chapter explores previous watermarking work for the purpose of defining digital watermarking. First, the basic idea behind watermarking will be explored with the desired properties of a watermark. Known attacks will then be explored along with an explanation of how to measure the success or failure of these attacks.

3.1 Basics of Digital Watermarking

While there have been many different watermarking techniques proposed, most follow the same basic algorithm. Figure 3.1 shows the steps followed to embed and detect a watermark.

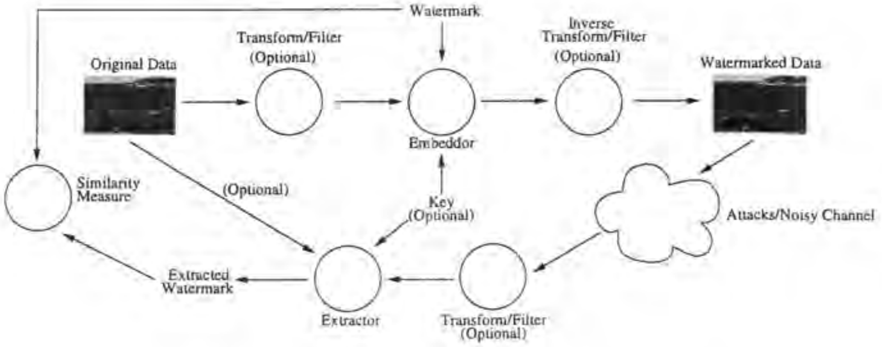


Figure 3.1 Basic watermarking process

3.1.1 Embedding the Watermark

The first step is data manipulation. This manipulation, which is optional, can take many forms. Researchers have proposed spatial transforms such as separation into bit planes and filters along with domain transforms such as the discrete cosine transform (DCT), the discrete wavelet transform (DWT), and the discrete Fourier Transform (DFT) [15, 3, 22]. Other data manipulation techniques, such as region filtering to find regions of interest, have also been proposed.[2, 24]

Once the data have been manipulated so that they are in the correct forms, the watermark is then added to the manipulated data. The basic equation is outlined in Equation 3.1.

$$X^* = X + W \tag{3.1}$$

X^* is the watermarked data , X is the original work and W is the watermark.

An optional key may be used in some embedding algorithms. The use of an optional key may vary in different schemes. Example uses include a pseudo random number generator seed or coordinates that point to specific locations where the watermark is inserted [7, 20]. If necessary, the data are manipulated back to a form which can be physically viewed or listened to by a user.

3.1.2 Detecting the Watermark

Once the watermark has been embedded it must be detectable in order to be useful. It should be noted that once a watermark has been embedded, it is susceptible to attacks or bit errors that are the result of a noisy channel. The detector must be able to extract the watermark despite any attacks or bit errors.

Detection can be separated into two categories, blind and non-blind detection. Non-blind detection is when the watermark is detected using the original digital work whereas blind detection is when the watermark is detected without using the original digital work to extract the watemark. Blind detection can be called a public watermarking system and non-blind detection can be called a private watermarking system [4]. The use of these two detection schemes depends on the watermarking application.

Once the watermark is extracted it must be evaluated to be useful. This evaluation may be a simple decoding of the watermark to get a message or it may be a similarity measure with the original watermark. Decoding a watermark is a straightforward process using the decoding technique. However, evaluation of the comparison can be complex. The way in which this comparision is done varies in different schemes. Whatever scheme is used, there must be some sort of error-rate or similarity level achieved. This measure is then compared to a predetermined threshold that indicates the presence of the watermark in a binary yes or no sense. "Yes" means the watermark is present and "No" means the watermark is not present [9].

Determining this threshold level is a difficult subject that has troubled researchers. The threshold must be set so that it minimizes either false negatives or false positives or both. False positive errors occur when a detector indicates the presence of a watermark in unwatermarked work and false negative errors occur when a detector fails to detect a watermark in a watermarked work [4]. Cox et al. sums up two basic ways in which these error threshold levels can be created for both false postive and false negative errors.

One way to test the system is against random watermarks. Cox et al. pointed out that although this type of modeling is less useful, it is often used because of its simplicity [4]. This type of test looks at the probability of being able to detect a random watermark in a watermarked medium. This type of probability is much easier to compute since the watermark distribution is controlled by the watermarking scheme. This type of test is useful when a large number of watermarks will be inserted into a small number of works. An example of this application would be transaction tracking.

Another way is testing a watermark against random works. Since there are an infinite number of works and the distribution of this is not under the control of the user, this type of probability is very hard to compute. Most researchers have limited their examination to random watermark probabilities because of this fact. This type of test is applicable to copy control applications since a few watermarks would be put into a large number of works.

This paper will use random watermark probabilities when computing threshold levels. Despite the limitations of these probabilities, they are much easier to compute and subsequently have been used in most of the literature on watermarking when creating threshold levels.

3.2 Digital Watermarking Properties

For a watermarking scheme to be successful it must be secure. For purposes of analysis, the concept of security can be defined in terms of three different properties that a secure watermark will possess. A secure watermark is unobtrusive, robust and unambigious. It should be noted that each application of a watermarking scheme will place a different weight on each security requirement to achieve its security goal.

3.2.1 Unobtrusive

The property of unobtrusiveness is important to watermarks and is a vital quality in most watermark applications. Unobtrusiveness means that the watermark should be perceptually invisible.[3] Another term used to describe this is fidelity, which refers to the similarity of the watermarked work to the original work [4]. In most cases it is desirable to have the watermarked work be similar to the original version. This paper will focus on the fidelity of the watermarked work in comparison to the attacked watermarked work.

3.2.2 Robustness

Robustness refers to the ability of the watermarked work to resist attack. A watermarking scheme must be able to extract the watermark despite an attack. Attacks, which will be explored in a subsequent section, can be intentional or unintentional. Watermarking schemes need not be robust to all attacks but rather robust to specific attacks that would hinder the overall application of the watermarking scheme [4].

3.2.3 Unambigious

The watermark should be able to be retrieved unambiguously by the watermarker [3]. This means that the detection should be able to show with certainty that a watermark is within a work. A 100 percent detection rate is obviously desired. However, this is very hard, if not impossible, to achieve especially in the face of attacks. It is therefore desirable to create a scheme that has a high probability of detection. These probabilities can be very hard to compute. This idea will be explored later in this paper.

3.3 Attacks on Watermarks

There are three basic ways to defeat a watermarking system. These ways as defined by Cox et al. are unauthorized embedding, unauthorized detection and unauthorized removal [4].

3.3.1 Embedding Attacks

Embedding attacks involve a malicious user embedding a watermark into a work. If a malicious user is able to embed his or her own or someone else's watermark then he or she may be able to thwart the original use of the watermark.

Craver et al. have explored a form of an embedding attack known as an ambiguity attack [5]. They argue that a watermarking scheme must be non-invertible and non-

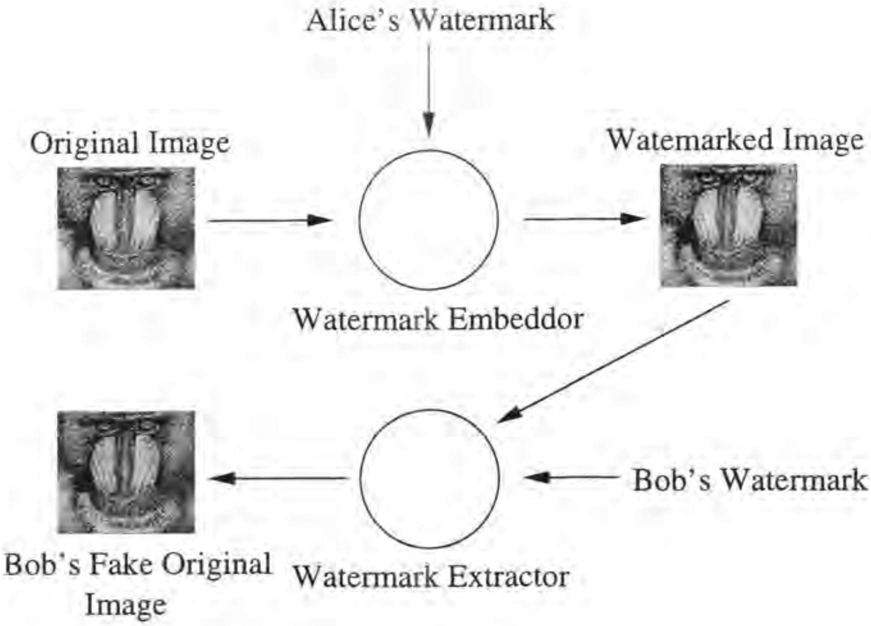


Figure 3.2 Craver et al. ambiguous attack process.

quasi-invertible to be successful in providing rightful ownership. This attack is as follows: Suppose Alice watermarks her photo before releasing it to the public. Bob, again, wants to use the photo and to claim the photo as his own. If the watermarking scheme is invertible then Bob is able to extract a watermark and create a fake original. He can then claim that he owns the original because his watermark is contained in Alice's original and watermarked image. This process is shown in Figure 3.2 [5]. This idea presented by Craver et al. is an important one when considering proof of ownership. If watermarking is to be used in this type of application it must be certain that the algorithm used satisfies the property of being non-invertible and non-quasi-invertible.

Another form of an embedding attack is the copy attack [14]. This attack looks to estimate the watermark from a watermarked work. Once the watermark has been estimated it can be inserted into different works. This attack presents a form of an identity attack. If a watermark is used to identify someone and a malicious user has access to this watermark then it may be embedded in order to wrongfully identify someone. This type of attack could have very serious consequences. Again a watermarking scheme must be resistant to this attack to be commercially viable.

3.3.2 Detection Attacks

Detection attacks are applicable to situations when a user should not be able to detect the presence of a watermark. If a malicious user is able to detect the watermark, he or she may be able to use it in a way that is contrary to the application of the watermark. This type of attack is geared more towards stegonography but can also be applied to watermarks. If a watermark contains information that should be hidden from the user then the user should not be able to detect the watermark.

An example of this would be if a hospital embedded a patient's name and information into an X-ray [4]. In this case the hospital wants only authorized users to be able to detect the watermark to protect the privacy of the patient. If a malicious user is able to extract the watermark then the watermark scheme is defeated in this application.

3.3.3 Removal Attacks

Removal attacks look to alter the watermarked work in such a way that the watermark cannot be detected by the watermark scheme detector. This has been a very popular attack and has received significant attention from researchers. These attacks include intentional and unintentional attacks [4].

Stirmark is a tool that tests image watermarking schemes against many different removal attacks [17]. Stirmark performs a number of different attacks on a watermarked image which attempts to disrupt the watermark detection. Table 3.1 shows the attacks performed by Stirmark, their classification, and their intention.

The two types of attacks executed by Stirmark are geometric and signal processing. Many of the geometric attacks can also be called synchronization attacks. Synchronization attacks try to de-synchronize the work from the detector. One thing to note is that

Table 3.1 Attacks included in Stirmark 3.1.

Attack	Classification	Intention
Cropping	Geometric	Intentional
Flip	Geometric	Intentional
Rotation	Geometric	Intentional
Rotation-Scale	Geometric	Intentional
Random bending	Geometric	Intentional
Linear transformations	Geometric	Intentional
Aspect ratio	Geometric	Unintentional
Scale changes	Geometric	Unintentional
Line removal	Geometric	Intentional
FMLR, sharpening, Gaussian filtering	Signal Processing	Intentional
Color reduction	Signal Processing	Intentional
JPEG compression	Signal Processing	Unintentional

in most cases the watermark is still present and intact within the digital work. However, it cannot be detected due to de-synchronization. These attacks can quite often be overcome by increasing the complexity of the detection algorithm. However, the allowable complexity of the detection algorithm is application dependent [8].

Statistical attacks are another form of removal attack. These attacks attempt to analyze or combine the digital work data in such a way so as to remove the watermark [6]. This attack looks to analyze the data statistically such that the attack is able to create a watermark free digital work. The collusion attack, which will be explored in a subsequent section, is the main type of attack that falls under this category.

CHAPTER 4. Digital Video Watermarking

This chapter explores the specifics of digital video watermarking. The algorithms used for testing in this paper will be presented in addition to an exploration of attacks specifically designed for digital video watermarks. The algorithms presented here are not fully representative of all existing watermarking algorithms. The recent interest in digital watermarking has produced a large number of digital video watermarking algorithms. Instead of an exhaustive explanation of every watermarking algorithm, this paper will focus on two prominent watermarking algorithms.

Throughout this paper it is assumed that all video is in an uncompressed format. Uncompressed video is used rather than compressed video because of ever changing compression standards. Compression standards are constantly being updated and changed. Standards such as mpeg1, mpeg2, mpeg4,mpeg7, AVI and MOV have all been released fairly recently. For this reason it is desirable to study uncompressed video since this will be a constant compared to changing compression standards over time.

4.1 Watermarking Schemes

4.1.1 Frequency Domain Spread Spectrum Watermarking

Spread spectrum watermarking was one of the first schemes presented for digital watermarking. It is a non-blind watermarking scheme that works in the transform domain. It utilizes communication technology concepts and applies them towards watermarking. This scheme works in the frequency domain and thus will be referred to as a frequency domain watermarking scheme throughout this paper.

Spread spectrum watermarking is a relatively simple technique that embeds data into works using spread spectrum technology. Traditionally spread spectrum is used in communication technology where a narrowband signal is transmitted over a much larger bandwidth such that the signal energy in any given frequency is undetectable [3]. In other words, such a small signal is inserted that this is considered to be part of the noise of the signal and is therefore undetectable unless there is a knowledge of where these signals were inserted.

4.1.1.1 Embedding the Watermark

The first step in inserting the watermark into video is grabbing the individual pixel values. For a black and white image these are integer values ranging from 0 to 255. The next step is to transform the image into the frequency domain. Cox et al. achieve this by using the discrete cosine transform (DCT). This can be accomplished by using Equation 4.1 [19].

$$\begin{gathered}
F(u) = \frac{2c(u)}{N} \sum_{m=0}^{N} f(m) \cos \frac{(2m+1)u\pi}{2N}, \quad where \quad u = 0,1,2,\ldots,N-1 \\
where \\
c(u) = \frac{\sqrt{2}}{2}, \quad for \quad u = 0 \\
c(u) = 1, \quad for \quad u = 0,1,2,3,\ldots,N
\end{gathered} \tag{4.1}$$

This is a one-dimensional equation and in order to apply it to a two-dimensional image it must be applied twice. It first is applied to the rows and then to the columns. One thing to note is that the DCT is applied to the whole image at once.

Once the image is transformed into the frequency domain the watermark is then inserted. The first step is to create the watermark. The watermark consists of a vector of n random real numbers that have a Gaussian distribution of $N[0,1]$, a mean of 0 and a variance of 1 [3]. The top n values are then extracted from the frequency transformed domain to get V_n. This is achieved by doing a zig-zag scan of the top left-most values in the DCT matrix, excluding the DC coefficient. The zig-zag scan is similar to that done in JPEG compression.

Once this is done a vector V_n is obtained consisting of the top n values of the frequency domain transformed image and a watermark vector X_n consisting of n random real numbers that have a Gaussian distribution. The watermark is then inserted by using one of the following equations:

$$V_i' = V_i + \alpha X_i \tag{4.2}$$

$$V_i' = V_i(1 + \alpha X_i) \tag{4.3}$$

$$V_i' = V_i(e^{\alpha X_i}) \tag{4.4}$$

In the above equations V_i' is the watermarked vector and α is the scale. The scale can be set to be whatever the user desires with a tradeoff between unobtrusiveness and robustness. Cox et al. suggest using $\alpha = 0.1$. Once this has been done to each of the n elements in V_n, the watermarked vector is reinserted into the image. The image is transformed out of the frequency domain by performing the inverse DCT. Once this is done, the image is watermarked. An example of an image watermarked using this technique can be seen in Figure 4.1.

4.1.1.2 Detecting the Watermark

Once the image is watermarked, the watermark can be extracted in order to prove that the watermark is contained within the image file. In order to do this, the original image, the watermarked image, and the scale factor are needed. Again, the first step is to transform both the original and watermarked image into the frequency domain by performing the DCT on each image. Next, the watermark is extracted by using one of the three equations above and solving for X_i. For example, if the equation

(a) Original image (b) 1 Watermarked image

Figure 4.1 Frequency domain watermarked image

$V_i' = V_i(1 + \alpha X_i)$ were used to watermark the image then to extract the watermark the equation, $X_i^* = (V_i'/V_i - 1)/\alpha$, where X_i^* is the extracted watermark is used.

It is then necessary to compare the extracted vector X^* with the original watermark vector X. This is done using the following equation:

$$Sim(X, X^*) = \frac{X \cdot X^*}{\sqrt{X^* \cdot X^*}} \tag{4.5}$$

Using this similarity test gives a confidence number that shows the probability that the watermark extracted is the same as the original watermark. The confidence level should be roughly $\sqrt{n}$ [3]. This, however, may vary due to quantization when the image is transformed back to the spatial domain.

A random watermark false probability test can be used to find the probability of false positives in order to measure the effectiveness of the scheme. It is expected that the extracted watermark has a distribution of $N[0,1]$. Therefore, the expected variance of the extracted watermark should be $\sum_{i=1}^{n} X_i^{*2} = X^* \cdot X^* = 1$ and thus the extracted

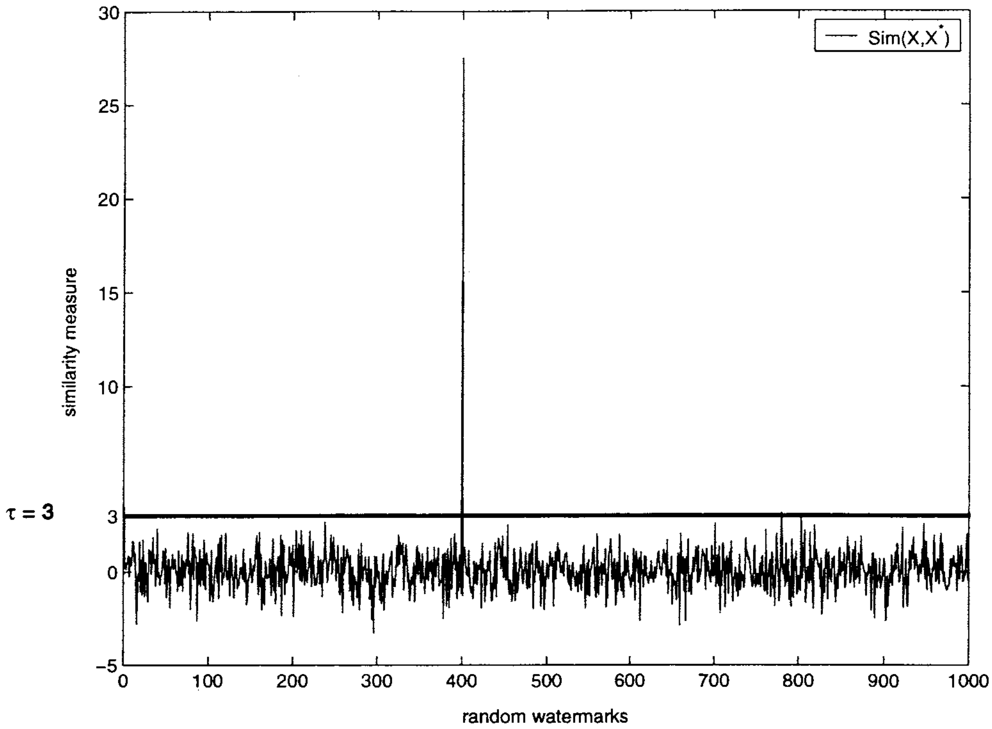


Figure 4.2 Graph of similarity measure for frequency domain watermarkingwhen the extracted watermark is compared to 999 random watermarks and the 1 original watermark.

watermark would have the expected distribution of $N[0, X^* \cdot X^*]$. If the threshold is set to τ, then the probability of $Sim(X, X^*) > \tau$ is the probability of $X^* \cdot X^*$ exceeding its expected mean by more than τ deviations [3]. The central limit theorem tells us that the probability of this can be found using the Equation 4.6 citepprob.

$$\Phi(\tau) = \int_{\tau}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \tag{4.6}$$

For example, if $\tau = 3$ then the probability of $Sim(X, X^*) > 3$ is $\Phi(3) = 0.0013$. This means that there is an approximate probability of 1 in 1000 that there will be a random watermark with a $Sim(X, X^*) > 3$. Figure 4.2 shows a simulation of this. The original watermark is compared with 999 random watermarks. Note that several random watermarks have similarity measures greater than 3.

4.1.2 Spatial Domain Spread Spectrum Watermarking

Hartung and Girod have proposed an alternative spread spectrum watermarking scheme. This algorithm has several distinct differences from the previous method. First, it works in the spatial domain and thus will be referred to as the spatial domain watermarking scheme. No transform is made so the watermark is added directly to individual pixel values. Also, the watermark added to each frame is not the same. The watermark is multiplied by pseudo-noise sequence in order to achieve this non-consistent property.

4.1.2.1 Embedding the Watermark

First, the video is line scanned in order to achieve a single vector of signal data [7]. This is achieved by doing a scan of the first line width wise followed by appending the next line to the first line. This process is outlined in Figure 4.3 [7].

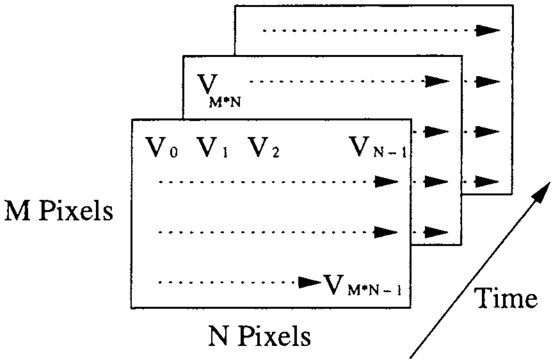


Figure 4.3 Line scan of video

The watermark is a bi-polar set of bits defined by equation 4.7 [7].

$$a_j, \qquad a_j \in \{-1, 1\}, \qquad j \in \mathrm{N} \tag{4.7}$$

Binary data can be used by simply converting a 0 to a -1. This sequence of watermark bits is what is inserted into the video. The signal is then spread by a factor called the chip rate denoted by cr. The chip rate is combined with the watermark sequence as defined in Equation 4.8 [7].

$$b_i = a_j, \qquad j \cdot cr \leq i < (j+1) \cdot cr, \qquad i \in \mathbf{N} \tag{4.8}$$

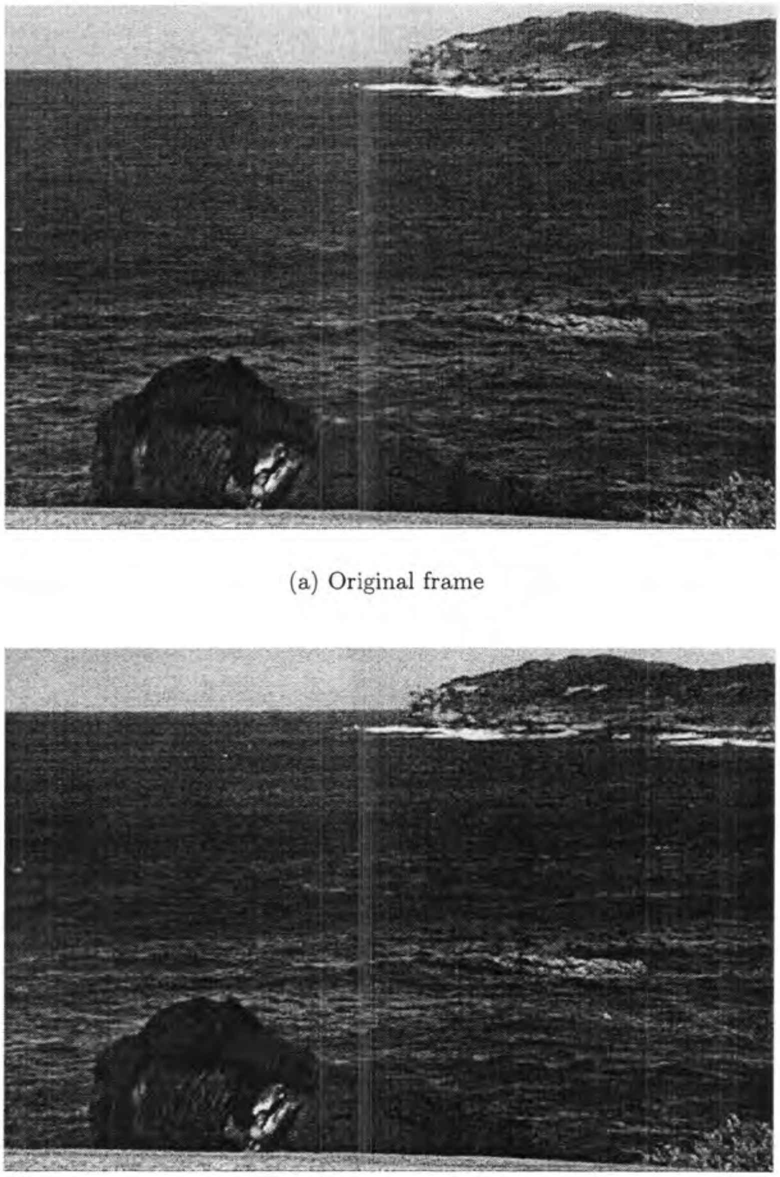
The cr spreads the signal watermark bit over a number of pixels in order to achieve robustness by means of redundancy [7]. This sequence is then multiplied by an amplitude factor $\alpha \geq 0$ and a pseudo random noise sequence p_i. The amplitude factor can be adjusted to different levels. A trade-off between fidelity and robustness must be measured to achieve the right number for α. The higher α is, the lower the fidelity and the higher the robustness and vice versa. The pseudo random noise sequence or psrn is a bi-polar vector of bits that equals the length of the signal vector v_i. The watermark is then added to the original signal by using the Equation 4.9 [7].

$$\tilde{v_i} = v_i + \alpha_i \cdot b_i \cdot p_i, \qquad i \in \mathbf{N} \tag{4.9}$$

The result $\tilde{v_i}$ is then put back into the original video three-dimensional form and the watermark insertion is complete. Figure 4.4 shows an example of a watermarked frame.

4.1.2.2 Detecting the Watermark

The watermark may then be retrieved by using a blind or non-blind method. The blind method involves using a filter to separate the watermark from the video. The authors, Hartung and Girod, suggest using a 3x3 high pass filter. The non-blind method simply subtracts the original signal from the watermark signal in order to extract the watermark. Once the signal has been filtered, either by the original video or some other method, the original watermark bits are then calculated using Equation 4.10.



(a) Original frame

(b) Watermarked frame

Figure 4.4 Spatial domain watermarked frame.

$$sign(s_j) = sign(\sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i^2 \cdot \alpha_i \cdot b_i) = a_j \qquad (4.10)$$

Once again the random watermark false probability test is used to find the probability of false positives. Assuming the watermarks are random, each $\oplus$ comparison in Equation 4.11 has a 50 percent chance of being a 1 when the bits are different and 50% chance of being a 0 when the bits are the same, so one would expect $\sum_{i=1}^{n}(x_i \oplus x_i^*) \approx \frac{n}{2}$ with a variance of $\frac{n}{4}$. Using the central limit theorem we can obtain Equation 4.11 which is used as the similarity measure.

$$Sim(X, X^*) = \left| \frac{\sum_{i=1}^{n}(x_i \oplus x_i^*) - \frac{n}{2}}{\sqrt{\frac{n}{4}}} \right| \qquad (4.11)$$

Under perfect circumstances and using the more reliable non-blind detection there should be no bit errors, which gives a similarity measure of $\sqrt{n}$. However, due to quantization or attacks or both this number will vary. Therefore, it is beneficial to use a random watermark false positive analysis to determine the probability of different thresholds. In other words, the probability of the extracted watermark and a random watermark having a certain similarity level is computed.

Again this similarity rate can be plugged into Equation 4.6. For example suppose $n = 900$ and $Sim(X, X^*) = 3$ which means that 405 out of the 900 bits were different. Using Equation 4.6 gives the probability of a random watermark having a $Sim(X, X^*) > 3$. This is found to be 0.0013. Again, this means that there is an approximate probability of 1 in 1000 that there will be a random watermark with a $Sim(X, X^*) > 3$.

Figure 4.5 shows a simulation of the random watermark false positive probability test. The similarity rate of an extracted watermark is compared to 999 random watermarks and the original watermark. Note that the original watermark clearly stands out above the rest with a $Sim = 30$ and at least one random watermark is greater than 3.

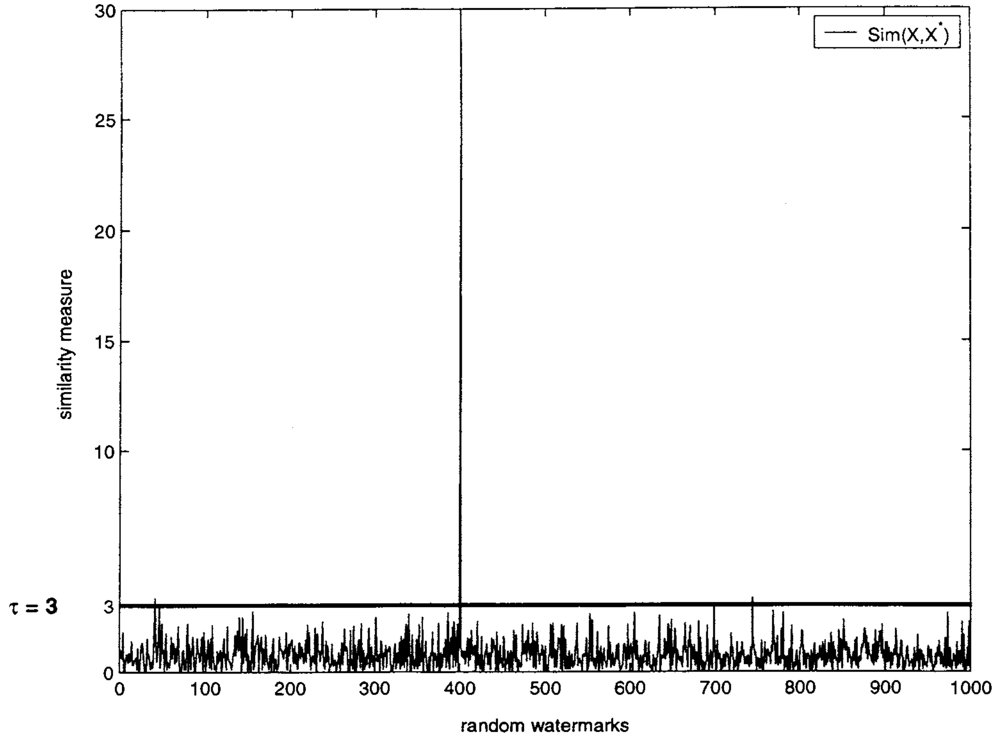


Figure 4.5 Graph of similarity measure for spatial domain watermarking when the extracted watermark is compared to 999 random watermarks and the 1 original watermark.

4.2 Digital Video Watermarking Attacks

As mentioned before, the temporal dimension leaves video watermarking prone to collusion attacks. A collusion attack is when frames of video are analyzed or combined in order to produce a digital work which is free from a watermark [20]. Video contains redundancy between frames since quite often very little changes from frame to frame. Collusion exploits these redundancies in an effort to remove the watermark from the work. Collusion attacks can further be classified into two different categories.

Type 1 collusion attacks exploits temporal redundancy in the watermark [6]. This type of collusion takes visually dissimilar video frames marked with the same watermark and attempts to recreate the watermark [20]. If the watermark can be recreated it can then simply be subtracted from frames in order to remove the watermark.

An example of Type 1 collusion is using a filter to extract a watermark from each frame and then averaging the extracted watermarks. Anisotropic diffusion is a one example of a noise filter that can be used to separate the watermark from the image [23]. Using a filter like this might produce a good estimation of the watermark, but since there is more than one watermark extracted these can be averaged to gain an even more accurate estimation of the original watermark.

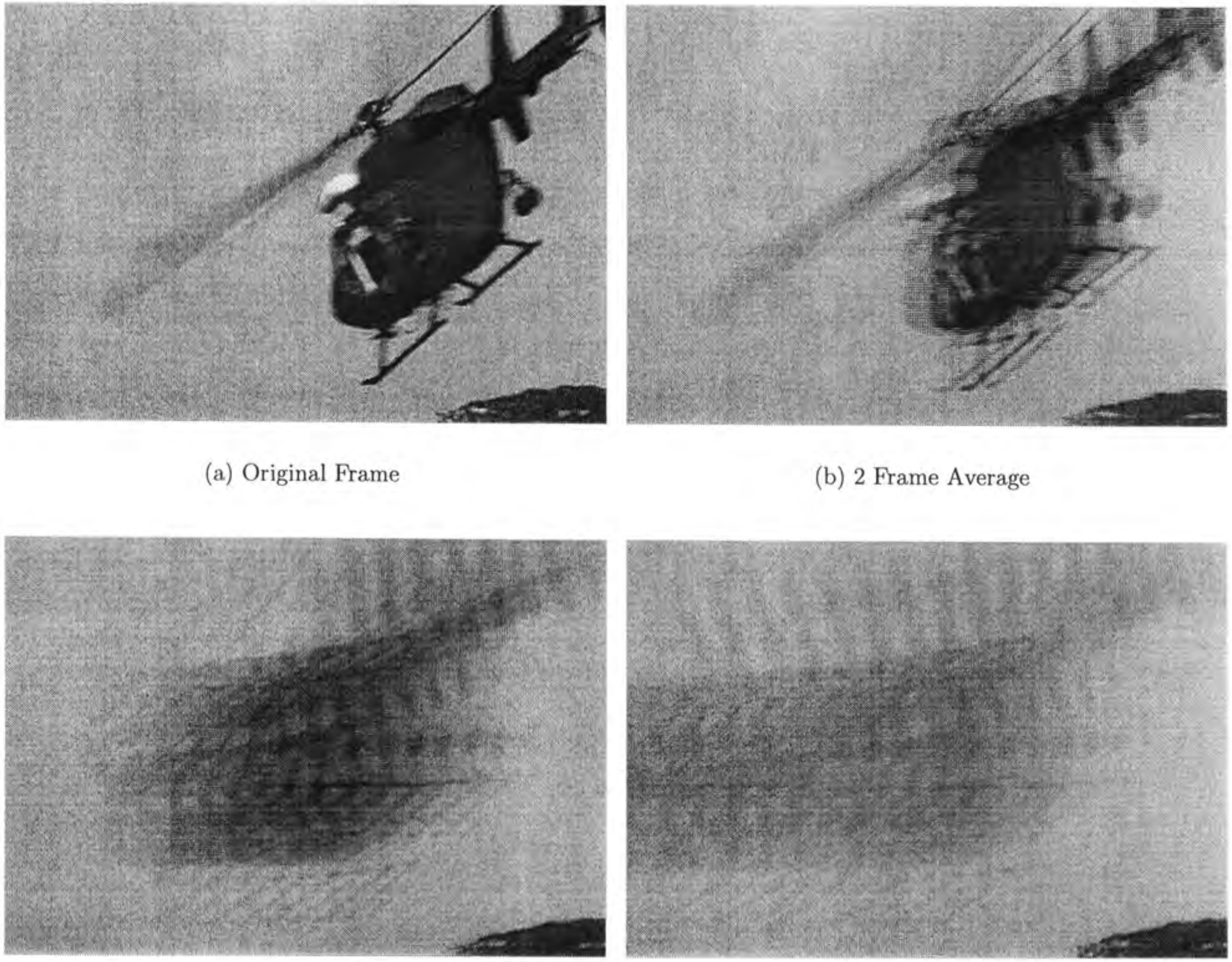
Type 2 collusion attacks exploits temporal redundancy in the video frames [6]. This type of collusion takes visually similar frames marked with a different watermark and attempts to remove the watermark by replacing watermarked data with watermark free or different watermarked data [21].

An example of a Type 2 collusion attack is frame averaging. This attack averages visually similar frames in an attempt to create a watermark free frame. This attack is outlined in equation 4.12.

$$X(i,j,t) = \frac{1}{L}\sum_{k=t} t + LX(i,j,k) \forall i,j \tag{4.12}$$

t is the frame number, L is a the number of frames to average, i and j represent pixel locations and $X(i,j,t)$ represents the frame to be replaced. This attack arithmetically averages pixel values over a certain number of frames to replace a single frame. This attack can be effective in removing a watermark. However, this attack is sensitive to movement between frames. Most video contains some movement from frame to frame. This attack can severely affect the fidelity of the digital work, especially when there is a lot of movement between frames as shown in Figure 4.6

Current Type 2 collusion attacks can be successful at removing the watermark but this is often at the expense of fidelity, especially in high movement video.



(a) Original Frame

(b) 2 Frame Average

(c) 10 Frame Average

(d) 20 Frame Average

Figure 4.6 Average attacked frames.

CHAPTER 5. Attack Methodology

The replacement attack was first proposed by Kirovski and Petitcolas [13]. Their 2002 paper outlines a method for exploiting temporal redundancy in watermarked works. They mention that this attack can be applied to video, but the attack is only applied to audio in the paper. This leaves unanswered questions as to how this attack can be applied to digital video. This chapter explores the replacement attack as applied to digital video watermarks. Two algorithms were created to implement an attack based on the replacement strategy. The algorithms are fully presented as well as an example watermarking scenario in which the attack would be effective.

This attack looks to remove the video watermark by separating each frame into smaller blocks and then replacing these blocks with perceptually similar blocks found within the same frame and temporally close frames. This attack is a Type 2 collusion removal attack. The goal of this attack is to create a watermark free copy of a digital video while preserving the fidelity of the digital video.

Variables for equations used in this attack are defined in Table 5.1.

5.1 The Replacement Algorithm

The first step in the replacement algorithm is to separate the digital work into smaller blocks. There are two types of block separation. The first separation is the replacement blocks. These blocks are non-overlapping blocks that are to be replaced by perceptually similar blocks. The number of blocks are found by Equation 5.1.

Table 5.1 Variable definitions

Variable	Meaning
N	Frame Width
M	Frame Height
F	Number of Frames
m	Size of Blocks
η	Crossover
k	Number of Search Blocks
n	Number of Replacement Blocks
β	Maximum Threshold
α	Minimum Threshold

$$n = M \cdot F(N/m) \tag{5.1}$$

The other type of blocks are search blocks. The difference between search blocks and replacement blocks is the crossover, η. The size of η determines the number of search blocks. Using a smaller value for η will create a large number of search blocks and larger values for η will create a smaller number of search blocks. Equation 5.2 determines the number of search blocks, k. Figure 5.1 shows a visual representation of separation into search blocks. It is hoped that an increase in k will increase the probability of finding a replacement for the replacement blocks.

$$k = M \cdot F\left(\frac{N-m}{\eta} + 1\right) \tag{5.2}$$

Once the replacement and search blocks have been formed, the blocks must then be compared. This is done using the root mean square error or rmse equation shown in Equation 5.3.

$$\phi(B_r, B_s) = \sqrt{\frac{1}{m}\sum_{l=0}^{m-1}[B_r(i_1+l, j_1, t_1) - B_s(i_2+l, j_2, t_2)]^2} \tag{5.3}$$

The value obtained by $\phi(B_1, B_2)$ is then compared to the threshold levels α and β using equation 5.4.

Table 5.2 Replacement algorithm

Input Variable	**Description**
Original Video	This is the watermarked video to be attacked
m	blocksize
β	Maximum Threshold
α	Minimum Threshold
η	crossover value

Output	**Description**
Watermarked Video	The attacked Video
Replacement Count	The number of blocks replaced

Algorithm

```
1. Set Variables
    $rn$ = number of replacement blocks in movie using equation 5.1
    $sn$ = number of search blocks in movie using equation 5.2
2. For every replacement block
    For every search block
        rmse(current replacement block, current search block) use equation 5.3
        if ( $rmse \geq \alpha$ and $rmse \leq \beta$ )
            replace current replacement block with current search block
            Keep replacement block if no suitable search block is found
3. Write attacked video to file
```

Note

The first suitable search block is used for replacement. Each search block may be used an infinite number of times for replacement.

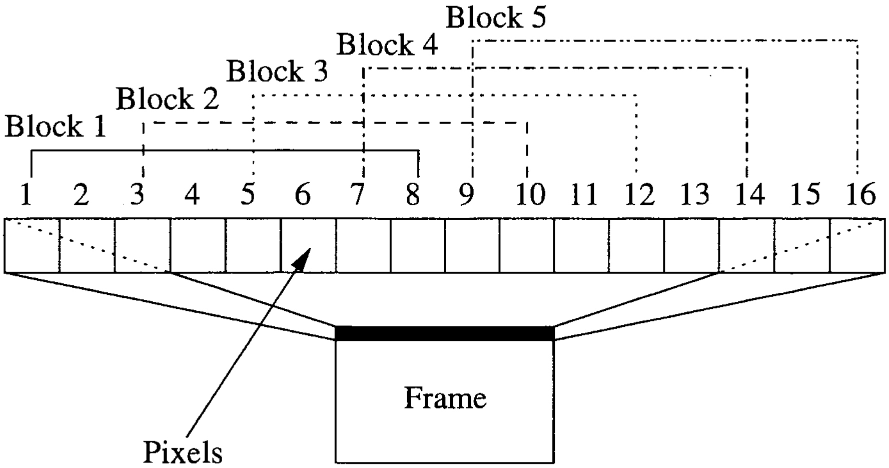


Figure 5.1 Search block separation process

$$\alpha \leq \phi(B_p, B_q) \leq \beta \tag{5.4}$$

If the rmse is between the threshold values then the search block is suitable for replacement. α is used to preserve fidelity and β is used so that the block is not too similar to the original.

This process is repeated for every replacement block. If a suitable block is not found for replacement, then the original block is kept to preserve fidelity.

5.2 Swap Algorithm

The steps in the swap algorithm are similar to the replacement algorithm but there are some key differences. Again the first step is to separate the video into blocks. However, in this algorithm Equation 5.1 is used for both the replacement and search blocks so that the number of search and replacement blocks are the same.

The replacement blocks are compared to each of the search blocks. If the *rmse* falls between the threshold levels then instead of just replacing the replacement block, the blocks are swapped such that the replacement block is put in the place of the search

Table 5.3 Swap algorithm

Input Variable	Description
Original Video	This is the watermarked video to be attacked
m	blocksize
β	Maximum Threshold
α	Minimum Threshold

Output	Description
Watermarked Video	The attacked Video
Replacement Count	The number of blocks replaced

Algorithm

```
1. Set Variables
     $r$ = number of replacement and search blocks blocks in movie using equation 5.1
2. For every replacement block
     For every search block
          rmse(current replacement block, current search block) use equation 5.3
          if ( $rmse \geq \alpha$ and $rmse \leq \beta$ )
               swap the replacement and search block
               record the block numbers so that they are not swapped again
               Keep replacement block if no suitable search block is found
3. Write attacked video to file
```

Note

The algorithm is ran such that suitable search blocks are searched for within the current frame and subsequent frames.

block and search block is put in the place of the replacement block. Once two blocks have been swapped they may not be swapped again.

5.3 Attack Scenario

A transaction tracking digital video watermarking application scenario is established for the sake of analysis as described below. This scenario is used to set the threshold, τ, that indicates the success or failure of an attack. Note that this is not the only scenario in which the attack would be applicable. However, this scenario best fits the random watermark false positive error analysis performed in sections 4.1.1.2 and 4.1.2.2.

Suppose that Alice owns a streaming video service that allows users to stream movies over the Internet to their computers. Alice uses two forms of protection to protect the videos from being illegally copied and distributed. First, the videos are encrypted before they are streamed. The application to play the streaming videos is exclusively distributed by Alice's company so that Alice's application is the only application that is able to decode and play the video. The encryption is used to protect the content while in transit and while resident on the user's machine such that a user would not be able to save the downstream to a physical media and play it on a non-compliant player or program, such as a commercial DVD player or Windows Media Player.

If only encryption is used, the service is still vulnerable to an attack that simply copies the digital work after it has been decrypted. This can be accomplished by grabbing the audio and video after it is decrypted and before it is sent to the sound card and video card. Therefore an additional level of protection is needed. Alice's company decides to use watermarking for the extra layer of protection.

Each video player application provided by Alice's company has a unique watermark that identifies the user to the streaming provider. When a user desires to watch a movie the application sends a request to Alice's video server along with the unique watermark in order to authenticate the user to the server. Alice's server embeds the unique watermark into the video as it is streamed to the user. A recording application may still be used to record the movie, but the recorded video will contain the unique watermark that can be linked back to the original user.

Alice has about one million customers and each customer has a unique watermark. Thus, when an illegally recorded video is found distributed on the Internet it is checked against a database of one million watermarks to find the origin of distribution. Therefore, Alice uses a random watermark false probability test to set the threshold to be used. A threshold of $\tau = 5.5$ is found to have a false positive probability of 1.89 x 10^{-8} using Equation 4.6. If a similarity rate comes up that is above the threshold then the adversary is most certainly identified. Using this threshold gives a very small probability of having a false positive when comparing with one million watermarks. The probability is approximately one in 50 million.

Bob is a user of Alice's service. He wants to record the decrypted video and illegally distribute it on the Internet. He knows that a watermarking scheme is used to deter users from doing this. He also knows that if he distributes a watermarked copy of the video it can be traced back to him. Bob also has knowledge of the watermarking scheme and the threshold used. Therefore, Bob wants to find a way to remove the watermark such that if Alice's company tests the attacked video against Bob's watermark the similarity rate will be less than 5.5.

This logic and scenario is used to analyze the results of the attacks performed on the two different digital video watermarking schemes. The attacks will be deemed successful if a similarity measure that is less than 5.5 is achieved and the video has a suitable level of fidelity.

CHAPTER 6. Results

Simulations were performed using a high movement 5 frame 352 x 512 video sequence taken from a commercially available DVD, *Mission Impossible II.* The original sequence is in true color so it was converted to black and white by separating the frames into hue, saturation and luminance layers with just the luminance layer retained. The video file type used was Audio Video Interlaced (AVI), using no compression.

For the frequency domain watermarking scheme outlined in 4.1.1 the variables are set to $n = 1024$ and $\alpha = 0.1$. Equation 4.3 is used for insertion. The similarity measure is taken for each frame and arithmetically averaged to get the overall similarity measure of the video.

For the spatial domain watermarking outlined in 4.1.2, the variables are set to $cr = 1000$, $n = 901$ and $\alpha = 4$. The watermark extraction uses a non-blind method to get a more accurately extracted watermark.

For the replacement and swap algorithms variable values $m = 8, 16, 32, 64, 128, 256$ & 512, $\alpha = 3$ and $\beta = 10$ were used. The parameter $\eta = \frac{m}{2}$ was used in all replacement simulations.

6.1 Frequency Domain Spread Spectrum Watermarking Results

The results of both the replacement and swap algorithms applied to the frequency domain watermarked video are graphically depicted in Figure 6.1. Using the logic out-

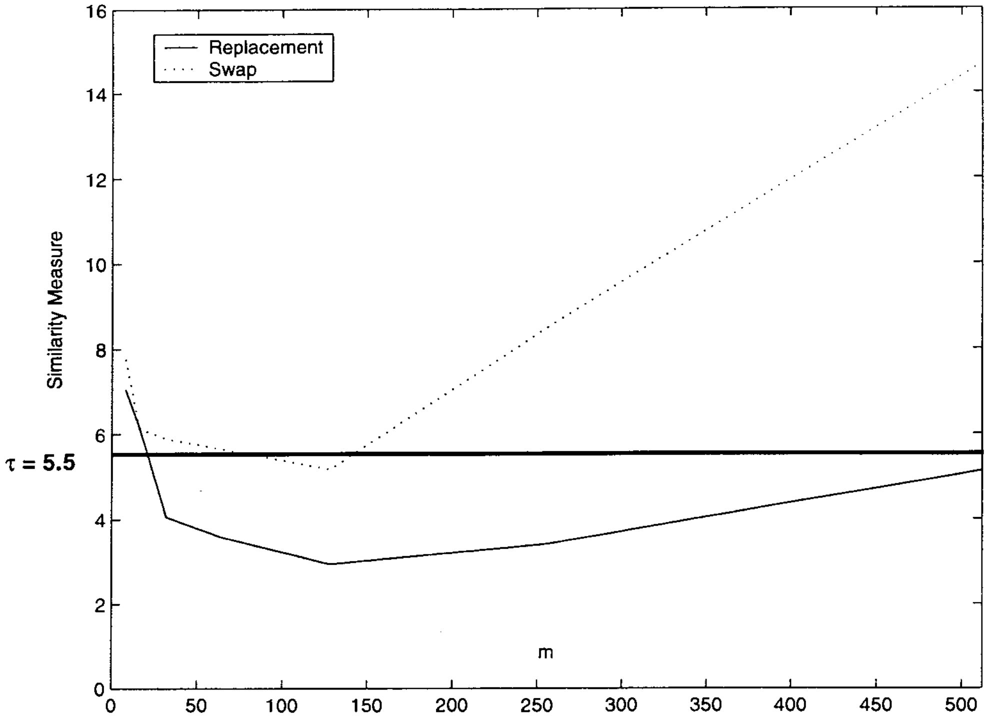


Figure 6.1 Graph of similarity measures for frequency domain watermarking when attacked with different values of m.

lined in section 5.3, a successful watermark extraction threshold is set to $\tau = 5.5$. For the replacement algorithm, $m \geq 32$ shows a successful attack. For the swap algorithm, $m \approx 128$ shows a successful attack. Overall, the replacement algorithm produces a smaller sim rate than the swap algorithm.

The rate of replacement is also compared. These results are shown in Figure 6.2. As expected, the replacement algorithm has an overall higher rate of replacement compared to the swap algorithm.

Overall, the swap algorithm produces a higher fidelity video than the replacement algorithm. Figure 6.3 and Figure 6.4 show the first frame of attacked videos. Figure 6.3 shows frames attacked when $m = 128$. At this block size both algorithms were able to remove the watermark. The replacement algorithm shows obvious lines where the

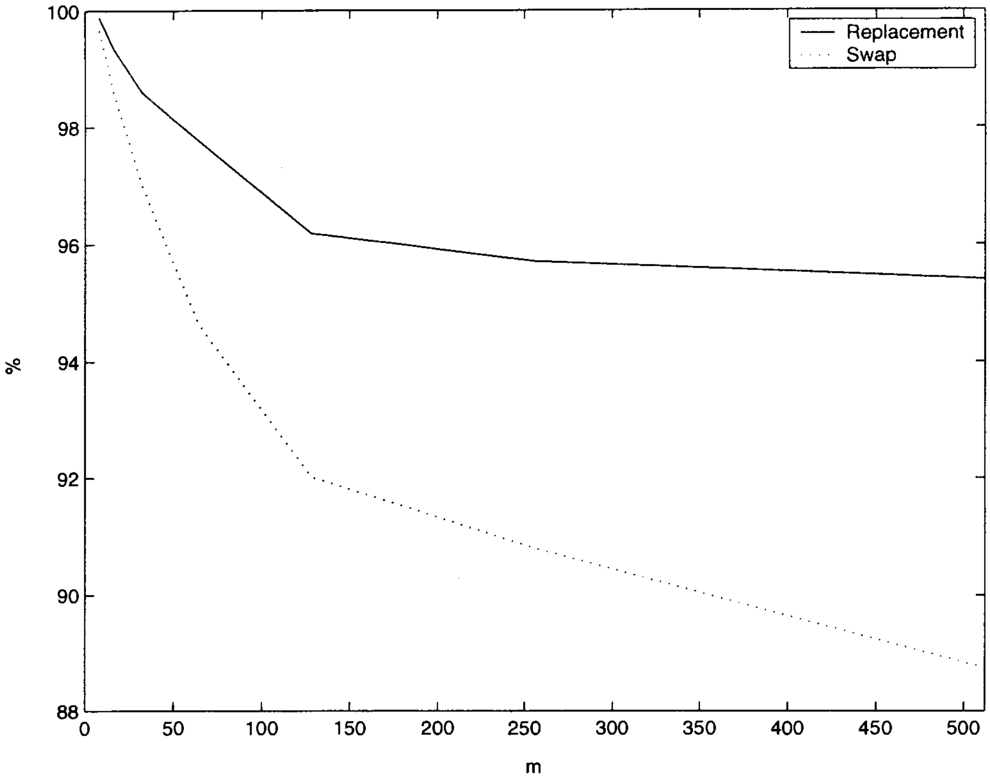
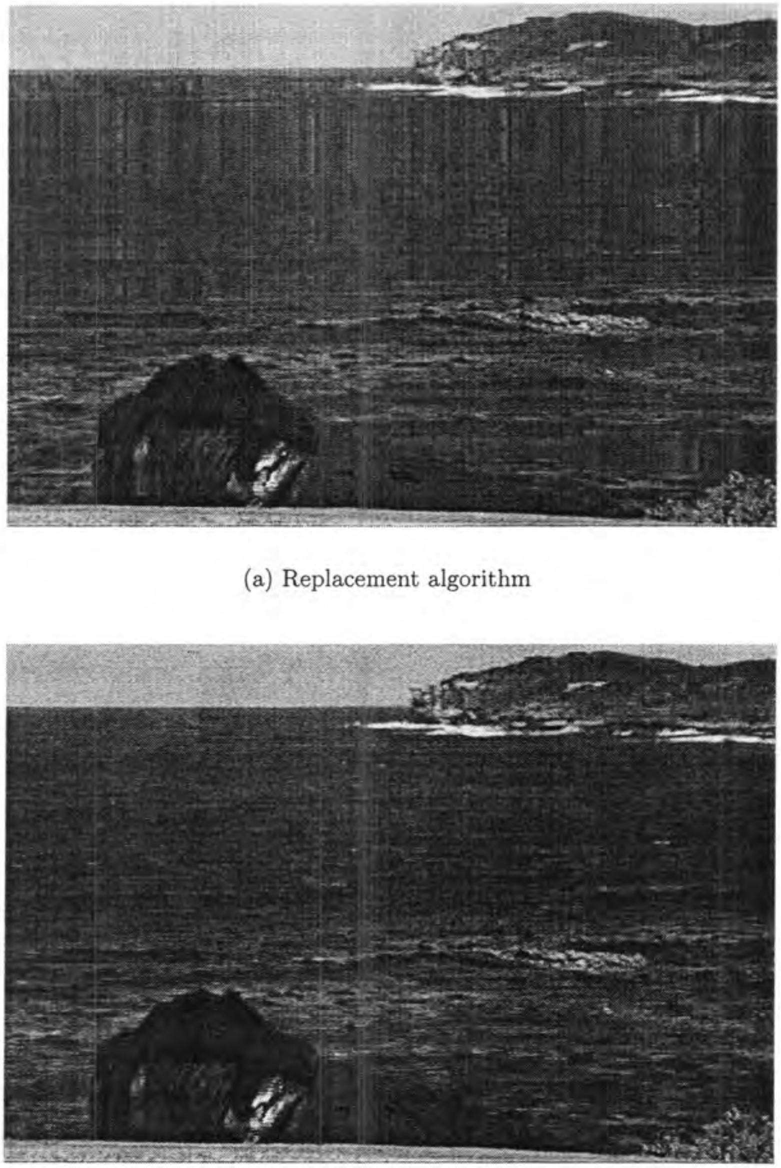


Figure 6.2 Graph of replacement percentages for frequency domain watermarking when attacked with different values of m.

blocks were replaced, whereas the swap algorithm shows fewer visual lines. Figure 6.4 shows frames attacked when $m = 512$. At this block size only the replacement algorithm was successful in removing the watermark. This block size proved to have the highest fidelity for the replacement algorithm. However, lines are still visible and the fidelity of the video is significantly decreased.

6.2 Spatial Domain Spread Spectrum Watermarking Results

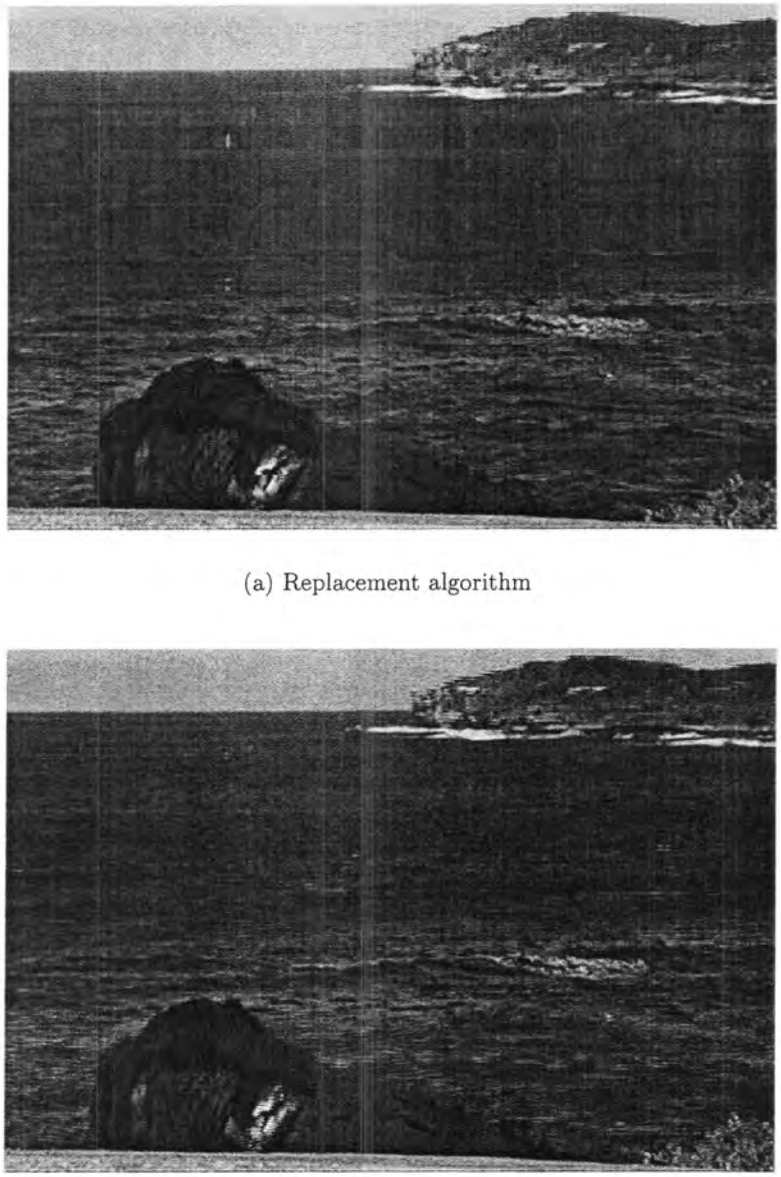
The results of both the replacement and swap algorithms applied to the spatial domain watermarked video are graphically depicted in Figure 6.5. This watermarking scheme is more resistent to the attack than the frequency domain scheme. No value



(a) Replacement algorithm

(b) Swap algorithm

Figure 6.3 This figure compares the 2 frames that have been watermarked using the frequency domain watermarking scheme and attacked using the replacement and swap algorithms using $m = 128$.



(a) Replacement algorithm

(b) Swap algorithm

Figure 6.4 This figure compares the 2 frames that have been watermarked using the frequency domain watermarking scheme and attacked using the replacement and swap algorithms using $m = 512$.

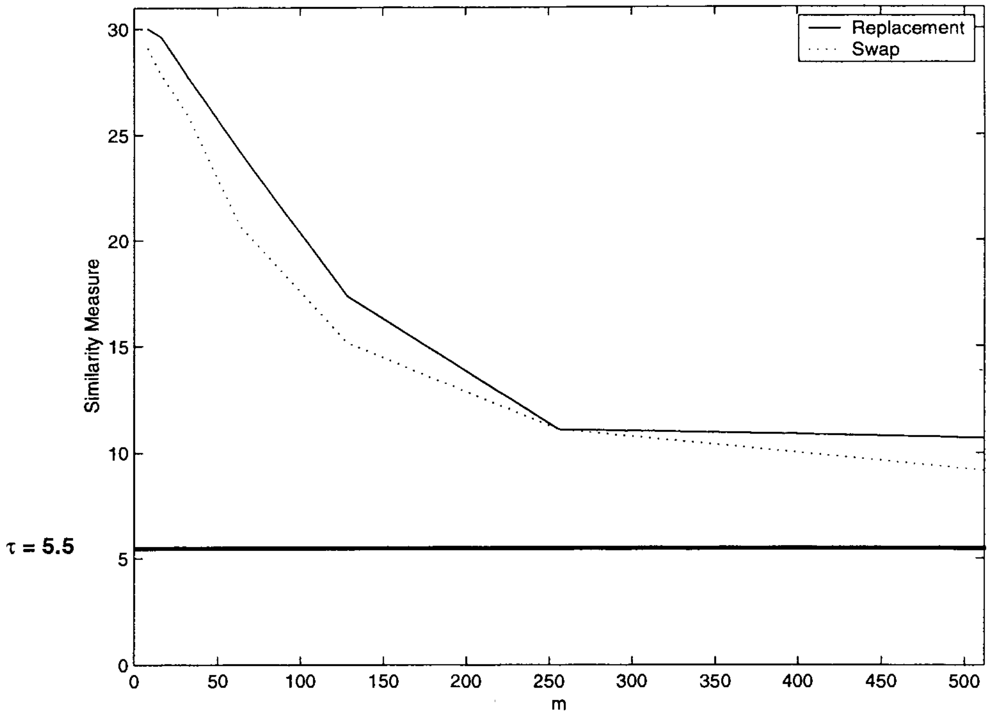


Figure 6.5 Graph of similarity measures for spatial domain watermarking when attacked with different values of m.

of m was found in this simulation to create a similarity measure < 5.5 despite a high percentage of replacement, as shown in Figure 6.6.

Figure 6.7 shows the first frame of the attacked video with $m = 512$. Again, the fidelity is better using the swap algorithm.

6.3 Discussion of Results

Both the replacement and the swap algorithm succeeded in removing the frequency domain watermark. However, neither algorithms were successful in removing the spatial domain watermark.

Both algorithms removed the watermark from the frequency domain scheme but the swap algorithm was able to produce a higher fidelity video sequence. By use of a

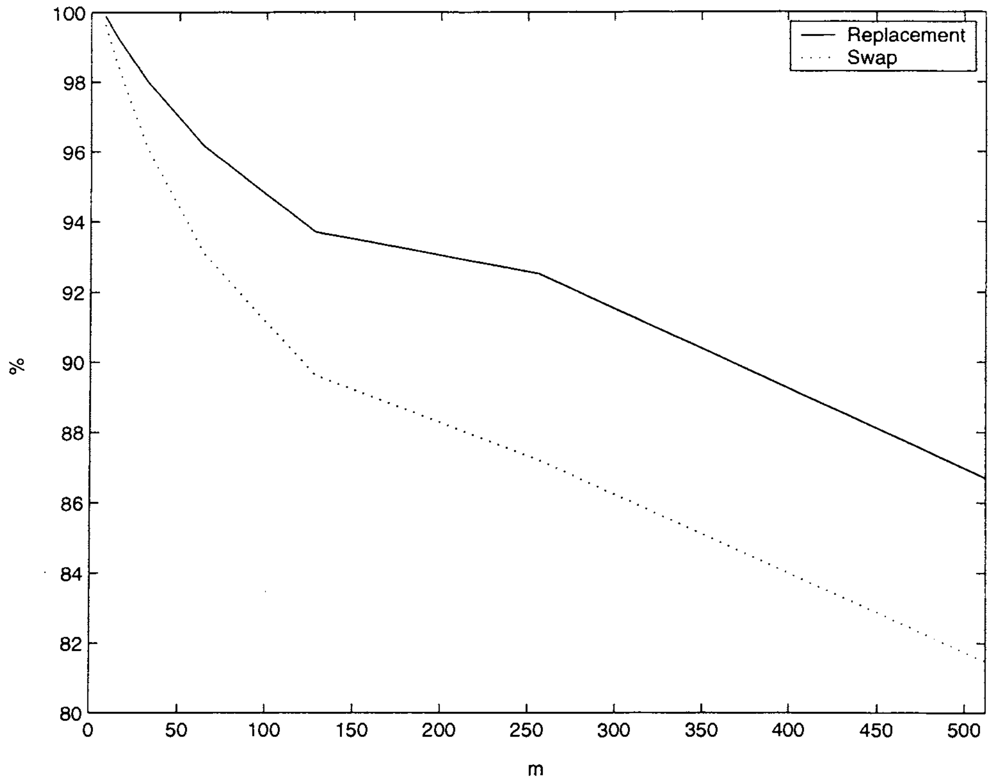
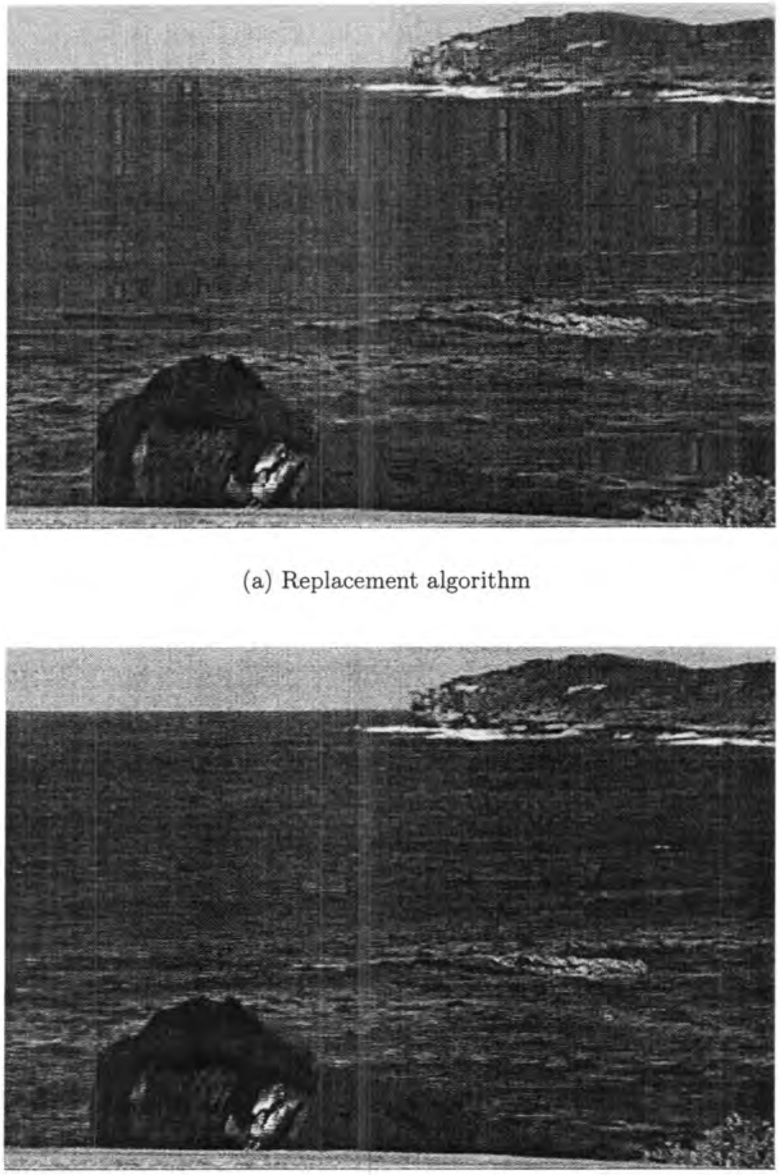


Figure 6.6 Graph of replacement percentages for spatial domain watermarking when attacked with different values of m.

blocksize of $m \approx 128$ it was possible to remove a frequency domain watermark such that the similarity measure was below the threshold and the fidelity of this attacked video was also acceptable. The video was noticeably fuzzy, but it was definitely watchable. Overall, the swap algorithm was a more successful attack because it was able to satisfy both attack requirements.

Neither the replacement algorithm nor the swap algorithm were able to remove the watermark from the spatial domain video watermarking scheme such that the similarity measure was below the threshold. The main reason for this difference is believed to be the redundancy built into the spatial domain watermarking scheme via the chip rate. Even though the similarity measures did not go below the threshold, at some block sizes,



(a) Replacement algorithm

(b) Swap algorithm

Figure 6.7 This figure compares the 2 frames that have been watermarked using the spatial domain watermarking scheme and attacked using the replacement and swap algorithms using $m = 512$.

the attack was able to signicantly decrease similarity measures such that they are very close to the threshold. Further efforts to optimize the attack variables would possibly increase the bit error rate and produce a successful attack. It should also be noted that the example threshold set is not set in stone. Another application of watermarking might set a different higher threshold.

Each of the attack algorithms has advantages and disadvantages. As expected, a larger search space increased the probability of finding a suitable replacement in the replacement algorithm. However, the fidelity of the attacked frames was lower using the replacement algorithm. This is believed to be because search blocks could be used to replace more than one replacement block. When a single search block is similar to a high number of replacement blocks, it is used for replacement a high number of times. This adversely affects the randomness of the image data in the frames and causes lines to appear where blocks were replaced.

Overall, the results show that the spatial domain watermarking scheme is more resilient than the frequency domain watermarking scheme and the most effective algorithm is the swap algorithm. These results are far from a complete exploration of this attack on video watermarking schemes but the data contributed in this paper is sufficient for some general conclusions. First, it appears that the collusion attack based on the replacement strategy as applied to video is a significant attack that compromises the security of video watermarking schemes. The success of the attack calls into question whether such video watermarking schemes can be commercially used. Also, it appears that redundancy is a very important property that a digital video watermarking scheme must possess. The frequency domain scheme does not use redundancy and thus is susceptible to this attack.

CHAPTER 7. Conclusion

This paper introduces two collusion attack algorithms based on the replacement strategy and performed a simulation of the attack against two different watermarking schemes. The attack was shown to be an effective attack on one of the watermarking schemes while the other watermarking scheme was found to be resistant to the attack.

The results of the simulations lead to several conclusions. First, it seems clear that the video watermarking schemes examined are vulnerable to this attack. The attack was able to significantly reduce similarity rates of both watermarking schemes. As a result of this, digital watermarking schemes for video must be improved with this attack in mind if they are to be successful. Lessons can also be learned by exploring why the spatial domain scheme was more resilient than the frequency domain scheme. One of the major differences between the schemes is the redundancy built into the spatial domain scheme. Using the chip rate to spread each bit across many pixels significantly increased the resilience of the scheme. These simulations show that redundancy is a vital property of video watermarking schemes.

Overall, this paper's contributions are twofold. It adds to the knowledge base of known successful attacks and also highlights properties of video watermarking schemes that make them resilient to the attack. This knowledge should be taken into consideration when creating future watermarking schemes.

7.1 Future Work

The work performed here has suggested the following future work may be appropriate.

- **Testing using different values for variables.** The variable values used for testing were not known to be the optimal variables for the attack. Further testing would be beneficial to find optimal thresholds for different block sizes as well as optimal crossover values to get the highest rate of replacement. It would also be interesting to see how changing the watermark scheme variables would affect the schemes resistance to the attacks.

- **Testing against different watermarking schemes.** There are many different video watermarking schemes that this attack could be tested against. A beneficial area of work would be to create a Stirmark-like tool for video watermarks. This tool could implement this attack as well as other digital video watermarking attacks. Creation of such a tool would allow many different schemes to be tested against this and other video specific attacks.

- **Testing against different video sequences.** Since the video sequence used was only five frames long it would be beneficial to look at a much longer video sequence to see how this affects the replacement rate. Also, the video sequence used was a very high movement sequence. It would be interesting to see how the attack affects the fidelity of a low movement video sequence. Another unknown is whether the video size would affect the results of the attack. Would a smaller video height and width decrease the replacement rate because there are less data?

- **Combine algorithms.** It would be beneficial to be able to combine the fidelity produced swap algorithm with the percentage of replacement produced by the replacement algorithm. An algorithm that combines the two ideas might produce some interesting results and probably improve the ability to remove the watermark while maintaining a level of fidelity. This might be accomplished by using the replacement algorithm and keeping a record of each replacement such that no search block could be used twice. It would be interesting to see if such an algorithm would have a high percentage of replacement while maintaining a high level of fidelity.

- **Increase efficiency of algorithms.** Due to the large number of computations that this attack performs, it takes a long time to run, even on a video sequence of five frames. For this attack to be practical as applied to longer video sequences, the amount of computations must be reduced. One way of doing this would be to maintain a record of comparisons already made, so that two blocks are not compared more than once. Another possible way would be to implement a region filter such that only similar regions of blocks would be compared.

- **Different shaped blocks.** Another interesting test would be to determine the effect of different shaped blocks. Linear blocks were used in both the swap and replacement algorithm. It might be beneficial to look at two dimensional square or rectangular blocks. Irregular shaped blocks that outline major figures in a frame might also be beneficial to explore.

- **Combine attacks.** A watermarking scheme must be resistant to all attacks as well as combinations of attacks to be successful. It would be of interest to look at combining this attack with other types of attacks such as those in Stirmark. Also, another attack could be built into this replacement attack. For instance it would be interesting to pool all blocks that were in-between the thresholds and use the idea of an averaging attack to average the blocks together to create a block for replacement.

BIBLIOGRAPHY

[1] Bloom, J. A., Cox, I. J., Kalker, T., Linnartz, J.-P. M. G., Miller, M. L., and Traw, C. B. S. (1999). Copy protection for DVD video. *Proceedings of the IEEE (USA)*, 87(7).

[2] Brisbane, G., Safavi-Naini, R., and Ogunbona, P. (1999, Kuala Lumpur, Malaysia). Region-based watermarking for images. In *International Workshop on Information Security.*

[3] Cox, I., Kilian, J., Leighton, T., and Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

[4] Cox, I., Miller, M., and Bloom, J. (2002). *Digital Watermarking.* Morgan Kaufmann Publishers.

[5] Craver, S., Memon, N., Yeo, B.-L., and Yeung, M. (1998). Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586.

[6] Deguillaume, F., Csurka, G., and Pun, T. (2000). Countermeasures for unintentional and intentional video watermarking attacks. In *IST/SPIE Electronic Imaging2000*, San Jose, CA, USA.

[7] Hartung, F. and Girod, B. (1998). Watermarking of uncompressed and compressed video. *Signal Processing*, 66(3):283–301.

[8] Hartung, F., Su, J., and Girod, B. (1999). Spread spectrum watermarking: Malicious attacks and counterattacks. In *SPIE Conference on Security and Watermarking of Multimedia Contents*, volume 3657, pages 147–158, San Jose, CA. SPIE.

[9] Hernandez, J. R. and Perez-Gonzalez, F. (1999). Statistical analysis of watermarking schemes for copyright protection of images. *Proceedings of the IEEE*, 87(7):1142–1166.

[10] Ho, D. (2003). Digital piracy face-off. *CBS/AP.* Available at: http://www.cbsnews.com.

[11] Kalker, T., Depovere, G., Haitsma, J., and Maes, M. (1999). A video watermarking system for broadcast monitoring. In *SPIE Conference on Security and Watermarking of Multimedia Contents*, volume 3657. SPIE, SPIE.

[12] Kilburn, D. (1997). Dirty linen, dark secrets. *Adweek*, 38(40):35–40.

[13] Kirovski, D. and Petitcolas, F. (2002). Replacement attack on arbitrary watermarking systems. In *ACM Workshop on Digital Rights Management.*

[14] Kutter, M., Voloshynovskiy, S., and Herrigel, A. (2000). The watermark copy attack. In *Electronic Imaging 2000, Security and Watermarking of Multimedia Content II*, volume 3971.

[15] Mobasseri, B. (1999). Exploring cdma for watermarking of digital video. In *SPIE Conference on Security and Watermarking of Multimedia Contents*, volume 3657, pages 96–102. SPIE.

[16] Patrizio, A. (1999). Why the dvd hack was a cinch. *Wired.* November.

[17] Petitcolas, F. A., Anderson, R. J., and Kuhn, M. G. (1998). Attacks on copyright marking systems. In *Information Hiding*, pages 218–238.

[18] Shi, Y. and Sun, H. (2000). *Image and Video Compression for Multimedia Engineering.* CRC Press.

[19] Stearns, S. D. (2003). *Digital Signal Processing with Examples in MATLAB.* CRC Press.

[20] Su, K. (2001). Digital video watermarking principles for resistance to collusion and interpolation attacks. Master's thesis, University of Toronto.

[21] Su, K. and adn Dimitrios Hatzinakos, K. K. (2003). Statistical invisibility for collusion-resistant digital video watermarking - part i: Theoretical considerations. *IEEE Transactions on Multimedia.*

[22] Swanson, M., Zhu, B., and Tewfik, A. (1998). Multiresolution scene-based video watermarking using perceptual models. *EEE Journal on Special Areas in Communications*, 16(4):540–550.

[23] Voloshynovskiy, S., Deguillaume, F., and Pun, T. (September 5-8 2000). Content adaptive watermarking based on a stochastic multiresolution image modeling. In *Tenth European Signal Processing Conference (EUSIPCO'2000)*, Tampere,Finland.

[24] Wang, H., Su, P., and Kuo, C. (1999). Digital image watermarking in regions of interest. In *IST Image Processing Image Quality xImage Capture Systems PICS.*

APPENDIX: Results Tables

Table A.1 Results of the swap algorithm for the frequency domain watermarking scheme

m	**Sim 1**	**Sim 2**	**Sim 3**	**Sim 4**	**Sim 5**	**Average**	**# Replaced**	**Total**	**% Replaced**	Φ
8	9.1038	7.4423	8.7948	5.8356	7.6235	7.7600	112252	112640	99.66	4.22E-15
16	7.6136	4.6054	6.8219	5.4193	6.2222	6.1365	55532	56320	98.60	4.22E-15
32	7.1883	5.9464	6.3052	5.0368	4.8766	5.8907	27320	28160	97.02	1.92E-9
64	5.6731	5.5469	7.1453	4.5683	5.3254	5.6519	13326	14080	94.64	7.93E-7
128	5.7564	5.3770	5.2316	4.6585	4.8330	5.1713	6478	7040	92.02	1.62E-7
256	8.5775	8.6770	8.7886	7.5962	8.8650	8.5009	3196	3520	90.80	0
512	15.3704	16.3232	13.7244	14.6901	13.2247	14.6670	1562	1760	88.75	0

Table A.2 Results of the replacement algorithm for the frequency domain watermarking scheme

m	η	**Sim 1**	**Sim 2**	**Sim 3**	**Sim 4**	**Sim 5**	**Average**	**# Replaced**	**Total**	**% Replaced**	Φ
8	4	7.4660	7.8780	6.569	6.6210	6.6990	7.0467	112509	112640	99.88	9.16E-13
16	8	6.0006	6.9126	6.8450	5.9748	5.1244	6.1715	55956	56320	99.35	3.41E-10
32	16	4.0067	2.8100	3.8316	4.4455	5.1588	4.0625	27767	28160	98.60	2.45E-5
64	32	3.9553	3.5009	3.2772	3.4418	3.7872	3.5925	13723	14080	97.46	1.64E-4
128	64	2.9813	2.2877	3.3809	3.0336	3.0562	2.9480	6772	7040	96.19	1.60E-3
256	128	3.8910	3.4943	2.6690	3.6615	3.3508	3.4133	3369	3520	95.71	3.21E-4
512	256	5.5847	4.5122	4.2831	5.4669	5.7933	5.1280	1679	1760	95.40	1.46E-7

Table A.3 Results of the swap algorithm for the spatial domain watermarking scheme

m	**Incorrect Bits**	**Total Bits**	**# Replaced**	**Total**	**% Replaced**	**Sim**	Φ
8	14	901	112230	112640	99.64	29.0838	0
16	33	901	55376	56320	98.32	27.8178	0
32	62	901	27052	28160	96.07	25.8856	0
64	141	901	13110	14080	93.11	20.6218	0
128	223	901	6310	7040	89.63	15.1582	0
256	284	901	3070	3520	87.22	11.0938	0
512	313	901	1434	1760	81.48	9.1615	0

Table A.4 Results of the replace algorithm for the spatial domain watermarking scheme

m	η	**Incorrect Bits**	**Total Bits**	**# Replaced**	**Total**	**% Replaced**	**Sim**	Φ
8	4	0	901	112519	112640	99.89	30.0166	0
16	8	6	901	55872	56320	99.20	29.6168	0
32	16	62	901	27597	28160	96.07	27.6846	0
64	32	88	901	13539	14080	93.11	24.1532	0
128	64	190	901	6597	7040	93.71	17.3570	0
256	128	284	901	3257	3520	92.53	11.0938	0
512	256	290	901	1526	1760	86.70	10.6940	0

ACKNOWLEDGEMENTS

First off, I would like to give a special thanks to Dr. Wong and Dr. Bergman for their help throughout the writing of this thesis. I really appreciate all of your help and feedback over the past year. I would also like to thank all of the other Information Assurance professors that have taught me so much over the past two years. I have really enjoyed being a part of this program.

I would also like to give a special thanks to my family. Mom and Dad, you have helped me out so much over the years and I hope that one day I will be able to return the favor. Kathryn and Maggie, thank you so much for always being there and I hope that we can all stay close in the upcoming years.

To everyone else in my life I would also like to thank you for supporting me throughout this whole process.