STEGANALYSIS OF VIDEO SEQUENCES USING COLLUSION SENSITIVITY

A Thesis

by

UDIT BUDHIA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2005

Major Subject: Electrical Engineering

STEGANALYSIS OF VIDEO SEQUENCES USING COLLUSION SENSITIVITY

A Thesis

by

UDIT BUDHIA

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Deepa Kundur
(Chair of Committee)

Narasimha Reddy
(Member)

Don R. Halverson
(Member)

Jennifer Welch
(Member)

Chanan Singh
(Head of Department)

May 2005

Major Subject: Electrical Engineering

ABSTRACT

Steganalysis of Video Sequences Using Collusion Sensitivity. (May 2005)

Udit Budhia, B.E. , Birla Institute of Technology, India

Chair of Advisory Committee: Dr. Deepa Kundur

In this thesis we present an effective steganalysis technique for digital video sequences based on the collusion attack. Steganalysis is the process of detecting with a high probability the presence of covert data in multimedia. Existing algorithms for steganalysis target detecting covert information in still images. When applied directly to video sequences these approaches are suboptimal. In this thesis we present methods that overcome this limitation by using redundant information present in the temporal domain to detect covert messages in the form of Gaussian watermarks. In particular we target the spread spectrum steganography method because of its widespread use. Our gains are achieved by exploiting the collusion attack that has recently been studied in the field of digital video watermarking and more sophisticated pattern recognition tools. Through analysis and simulations we, evaluate the effectiveness of the video steganalysis method based on averaging based collusion scheme. Other forms of collusion attack in the form of weighted linear collusion and block-based collusion schemes have been proposed to improve the detection performance.

The proposed steganalsyis methods were successful in detecting hidden watermarks bearing low SNR with high accuracy. The simulation results also show the improved performance of the proposed temporal based methods over the spatial methods. We conclude that the essence of future video steganalysis techniques lies in the exploitation of the temporal redundancy.

To my parents

ACKNOWLEDGMENTS

I would like to thank my research advisor Dr. Deepa Kundur, for her continual support. This thesis could not have been successfully completed without her guidance. I appreciate the financial aid provided by her during the course of this thesis.

I want to thank all my friends who supported me. I appreciate the constant tips I received from Ranadip Pal throughout the research. I would also like to acknowledge the help provided by Vijay Suryavanshi on my thesis during my internship at Nokia Research Lab in Dallas.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION *

A. Steganography

Steganography is the art of hiding messages in innocuous looking mediums such as text files, audio files, images, video sequences etc. It is different from cryptography where the goal is to convert the message into a form that is not easily comprehensible or deciphered. The main aim of steganography is to hide the very presence of the message by embedding it into a host carrier known as the cover object such that it is not detected. The sender embeds a *secret message* 'm' into the *cover–object* 'c' to obtain a *stego-object* 's' using an embedding scheme and a *secret key* 'K' [1]. A common element shared by steganography and cryptography is that, the security of the underlying methods lie in the secrecy of the embedding and the cryptographic keys, respectively. In other words, the attacker should not be able to detect the presence of the message in the former or be able to decipher the message in the latter without having access to the secret key. As in cryptography, we assume that the details of the embedding algorithm are known to the attacker. (Kerckhoff's Principle [2]).

The existence of steganography has been recorded even in the ancient times where hidden messages were tattooed on the shaven heads of messengers. The messengers were sent across borders once their hair grew and were later shaved again to deliver the message. Much known form of steganography, like sending hidden messages using invisible ink on

---

*Reprinted from pages 210–214, with permission from"Video steganalysis using collusion sensitvity" by U. Budhia and D. Kundur, Proceedings of SPIE: Sensors, Command, Control, Communications and Intelligence(C3I) Technologies for Homeland Security and Homeland Defense, April 2004, vol. 5403.

The journal model is *IEEE Transactions on Automatic Control.*

blank papers or written letters, was used by Great Britain [3]. As we can see, steganography is not restricted to mediums such as text, images, audio, video etc. A form of text steganography used by German spies in World War II taken from [3] is shown below. The following message was sent:

"Apparently neutral's protest is thoroughly discounted and protested. Isman hard hit. Blockade issue affects pretext for embargo on byproducts, ejecting suets and vegetable oils". Taking the second letter from each word the sentence reads,*"Pershing sails from NY June 1"*.

The modern form of steganography is represented in terms of the Prisoner's Problem [4], in which A and B are two inmates, confined to separate cells in a prison and are hatching an escape plan. All communication between them goes through a warden W. In order to exchange messages without arousing any suspicion in the minds of W, they need to pass the information secretly inside a medium that does not draw any attention. The warden may be passive where she just tries to determine whether there is something hidden in the cover object. In this case, the overall goal of steganography is to hide the message in such a way that it is difficult for the third party to distinguish between a cover-object and a stego-object while ensuring accurate covert communication. On the other hand the warden may be active and alter the stego-objects before passing it to B. The following World War II historical example elucidates the actions of an active warden [5]. A telegram originally sent as *"Father is dead"* was changed to *"Father is deceased"*. This prompted a reply, *"Father dead or deceased?"* Thus we can see that apart from sending the message in a way that it does not produce any detectable artifact it should be hidden in a robust manner to survive all perturbations along the path.

B. Steganalysis

The process of detecting the presence of covert communication through innocuous looking multimedia distribution, with high probability is called steganalysis. It is a way of distinguishing between a stego-object and a cover-object. A steganalyst may be passive or active [6]. A steganalyst is said to be passive if his only goal is to detect the presence of a message. He/she may try to identify the embedding method used to hide the messages in the cover medium. However an active steganalyst tries to estimate the hidden message itself. Since finding the true message may be impossible due to secure encryption schemes available in the market, he/she may try to figure out the location or the length of the hidden message or estimate the parameters used in the embedding process (e.g. the strength of the watermark or hidden message in case of spread spectrum steganography [7, 8]).

In this thesis we propose a method to detect the presence of steganographic messages in video data and do not consider estimation of the message. We design a steganalysis method that detects the presence of hidden messages in raw video sequences by taking advantage of the inherent temporal redundancy present in a video sequence. We study the advantages and disadvantages over the current steganalysis methods that can be incorporated for video.

In order to design a passive steganalysis system, one should look for the statistical changes brought about in the cover medium due to embedding. The changes can be quantified and compared to a threshold or to a known database to arrive at a decision. A typical steganalysis system is shown in Figure 1. The attacker or the steganalyst obtains a copy of the host signal from the communication channel. After processing it, he/she measures the statistical change in the host signal due to embedding. The quantified change t is compared against a threshold *thresh* to arrive at a decision of whether there is something hidden or not.

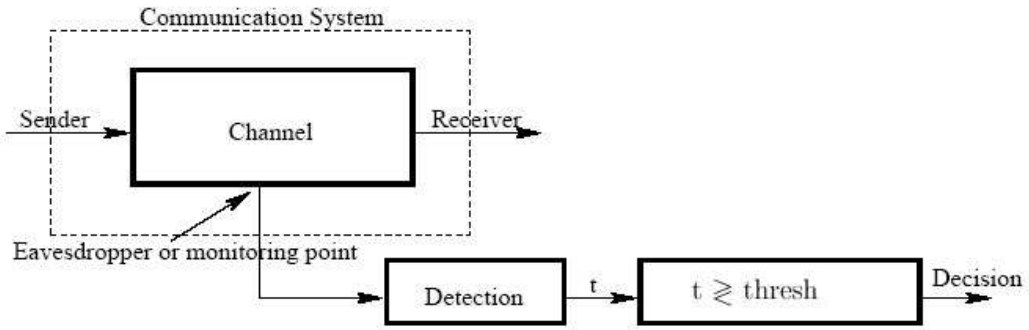


Fig. 1. Steganalysis system.

C. Motivation and Applications

The recent attacks on information systems, cyber-security and cyber-forensics have become a primary concern for both governments and commercial industries. Attackers of information systems can potentially use sophisticated means to hide messages in multimedia for covert communications. Identifying such communications must be automated in order to be able to effectively and practically monitor such behavior [9]. The presence of a temporal domain increases the volume of covert data that can be embedded into a video sequence. Thus from an embedder's point of view, using video sequences as cover-objects is the best choice since the capacity or the amount of covert data that can be carried is very high when compared to other mediums such as text and digital audio.

A number of efficient and reliable techniques have been proposed for still images that can be applied to raw video sequences, but to the best of our knowledge, there have been no steganalysis techniques proposed targeting the characteristics of digital video. This motivates us to develop a video steganalysis scheme that can be used to detect hidden messages.

The video steganalysis is a fundamental problem that has implications for watermark attacks too. The results can be used to design better steganalysis and watermarking methods.

Steganalysis finds its use in a broad area of applications ranging from computer security, cyber-security, cyber-forensics, homeland security, field of watermarking etc. Automated steganalysis techniques can be used to monitor the astronomical amount of Internet data to detect the presence of cover communication. One of the ways to use the Internet to pass covert data apart from using digital media as carrier is through the time stamps of the Internet packets. Steganalysis can be used to stop terrorists from using steganography as a means of covert communication. According to unnamed law officials terrorist organizations are hiding maps and photographs of potential targets, instructions for other terrorists on chat rooms and pornographic sites [10, 11].

Some parties use these sophisticated data hiding methods to pass Trojan content for malicious purposes or to get some information from the receiver without its knowledge. One such instance can be stated from the era of cold war between USA and Russia. The United States security agencies loaded Trojan content in a Control's software built for a gas-pipeline in Russia in 1982. The Trojan ran a test on the pipeline and doubled the pressure causing an explosion equivalent to a nuclear weapon [9, 12]. Detecting Trojan content is yet another application where steganalysis finds its use.

Steganalysis can be used in the field of digital forensics by examiners who look for hidden data or trace of hidden data in digital media [13]. A possible scenario is the distribution of child pornography using digital media as a cover object [14]. Steganalysis softwares will be useful in detecting the presence of such content and can act as a proof in the court of law. It can be used to differentiate between a natural image and digitally made images using graphics application softwares [15]. This finds its use in court cases where the origin of the image (natural, digitally made) is in question. Forensic experts are also hired by

companies to detect steganographic programs on the server that may be constantly sending sensitive information from the company databases.

Steganalysis may help in the design of computer security programs like anti-virus programs. Recently viruses were attached in JPEG images to take advantage of a security flaw in Microsoft's image viewer programs [16]. Steganalysis may be used to detect viruses, spywares, adwares and other malicious programs that may be hidden in digital media and may affect a computer.

Steganalysis and watermarking have a lot of commonality between them. Collusion schemes proposed in this thesis can be used to get an estimate of a watermark in a video sequence. This can be used to authenticate or detect the presence of a watermark in the sequence. Thus we see that steganalysis also finds its use in the field of watermarking.

D. Objectives

The objectives of this thesis are:

1. To propose efficient steganalysis techniques for video sequences that take advantage of the temporal redundancy present in it. We develop a composite method that can be used to detect messages hidden using a variety of embedding schemes that work in the spatial as well as the frequency domain. Most of the current methods assume the knowledge of the embedding scheme and thus are able to achieve higher detection accuracy. However there is a trade off between the detection accuracy and the applicabality of steganalysis to a broad class of embedding algorithms. The inspiration is drawn from a number of currently available steganalysis techniques aimed at detecting hidden messages from a variety of embedding schemes [17, 18, 19, 20].

2. To highlight the limitations of data hiding in video. In this thesis we assert that the chances of detection of hidden messages greatly improve due to the presence of

temporal redundancy in a video sequence. This limits the capacity of the payload that can be successfully embedded in a video sequence without producing statistical artifacts. We show by theoretical arguments and simulations that it is infeasible to hide data in those parts of video that are non-moving or have translational motion. A successful steganalysis algorithm is recognized by its ability to restrict the capacity of hidden messages in the cover medium.

3. To study the relationship between the fields of steganography and watermarking. There are many tools borrowed from watermarking that are used in steganography and vice versa. Through this thesis we want to support the fact that watermarking and steganography complement each other. We use collusion attack–a well studied area in the field of watermarking for our proposed steganalysis method. On the other hand, steganalysis can be used to detect the presence of a watermark.

4. To study the tradeoff between statistical invisibility and robust embedding of hidden messages in a video sequence. Through analysis and simulations we show the lower bounds on the embedding strengths of the hidden message that leads to the failure of the proposed steganalysis method.

5. To design a steganalysis method that can be applied for real-time applications. Most researchers assume the availability of infinite processing power for steganalysis. However this assumption poses serious challenges for real-time applications if the method has a large time complexity. In order to monitor the presence of a steganographic data in a broadcast video scenario a steganalysis method with low time complexity is needed. We propose a method that has a very low memory and processing power requirement and hence can be use for real-time video monitoring.

E. Nomenclature

A steganographic system involves two parties: the *sender* who embeds the secret message in the cover object and the *receiver* who extracts it. Security comes in part from the presence of a secret key K in the system that details how the secret message is embedded and extracted. We assume that K is securely exchanged between the sender and receiver prior to covert communication; this key is specific to the steganography algorithm and can contain information such as how strongly and where in the cover-object the secret information is embedded, and seed information for pseudo-random number generation.

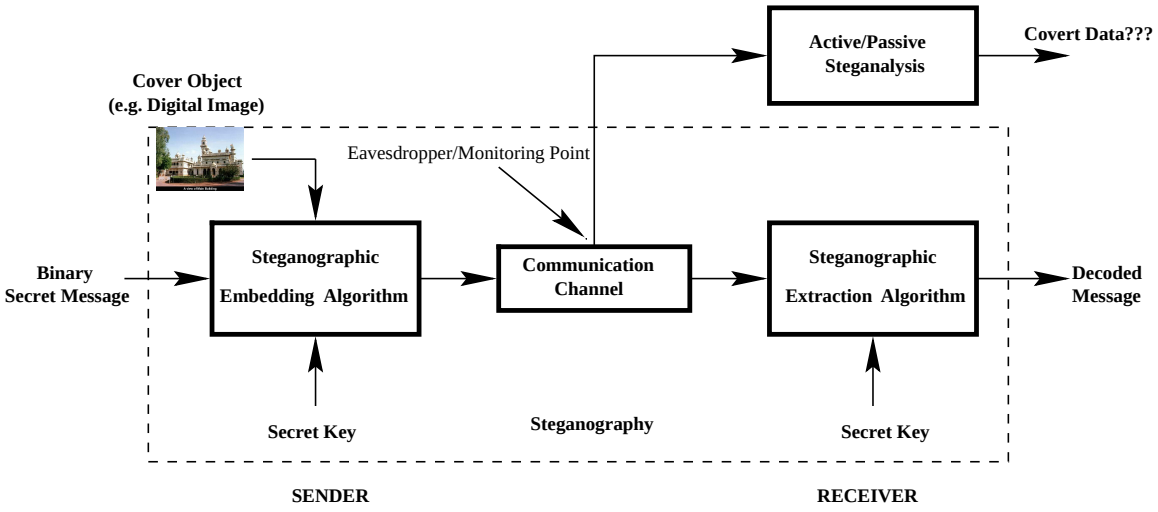


Fig. 2. Steganography and steganalysis. Steganography consists of the process of embedding (by a sender) and extracting (by a receiver) covert information from innocuous messages. Steganalysis is the process of determining from a given message whether or not covert data has been embedded [21].

A typical steganographic system scenario is summarized in Figure 2. The sender takes the "host" video sequence, which represents the *cover-video*, and embeds a secret binary message vector using K to produce a *stego-video* sequence that is perceptually identical to the cover-video. The stego-video is then communicated along a public channel to the

receiver. At the receiver the stego-object and secret key K are used to extract the secret binary message. The public channel may be monitored by an active or a passive steganalyst whose goal is to detect the presence of any covert communication taking place.

The original host video sequence or the cover-object is denoted by $U_k(m,n)$ where $1 \leq k \leq N$ is the frame number and m, n are the row and column indices of the pixels, respectively. The binary secret message is embedded into the host by modulating it into a signal known as the watermark [7] denoted by $W_k(m,n)$. Since the influence of the secret message is carried on to the watermark, we will use the terms hidden message and watermark interchangeably throughout this thesis. Detection of the watermark will imply the presence of hidden information in the medium. For compatibility, the watermark $W_k(m,n)$ is defined over the same domain as the host $U_k(m,n)$. Later on we will ease on this constraint and will look at watermarks embedded in the Discrete Cosine Transform (DCT) domain. The stego-video signal is represented by the commonly used equation [22]:

$$X_k(m,n) = U_k(m,n) + \alpha_k(m,n) \cdot W_k(m,n) \qquad k = 1, 2, 3 \ldots N\,, \tag{1.1}$$

where $\alpha_k(m,n)$ is a scaling factor used to manipulate the strength of the hidden message to trade-off between perceptibility and robustness. In practice, for simplicity α is considered to be constant over all the pixels and frames. So the equation becomes:

$$X_k(m,n) = U_k(m,n) + \alpha \cdot W_k(m,n) \qquad k = 1, 2, 3 \ldots N\,. \tag{1.2}$$

The scaled watermark $\alpha \cdot W_k(m,n)$, in practice, is a function of the binary secret message, secret key K and the host $U_k(m,n)$. The relation between these parameters is decided by the embedding algorithm. In general, every steganographic algorithm can be represented by Equation 1.2, where we first set a value for $\alpha \neq 0$, and let $W_k(m,n) = \frac{X_k(m,n) - U_k(m,n)}{\alpha}$. In order to have a proper reference for effective steganalysis, we must make some assumptions about the embedding method as discussed in the next section.

F. Problem Formulation

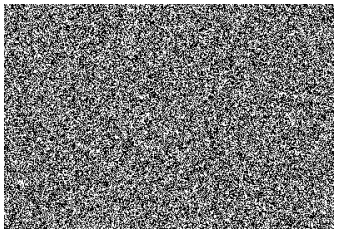
The overall goal of this thesis is to design a steganalysis method for digital video sequences that is more optimum than frame by frame application of previously proposed image methods that do not taken into account the temporal redundancy that can be exploited for higher accuracy detection. We consider this problem by first restricting our video processing to the temporal domain; image methods that work in the orthogonal spatial domain can then be easily incorporated to enhance performance over previously proposed techniques. We focus on steganalysis of spread spectrum-based steganographic methods [7, 8] due to its popularity and influence in the research literature.

In essence, our problem is to develop a decision box that takes a stream of digital video as input and concludes whether or not hidden information is present by using partial information about the embedding algorithm and a model of temporal redundancy in digital video frames; no knowledge of the secret key K, if any is used, is available. In particular, we assume the spread spectrum-based embedding method works by inserting Gaussian watermarks in the spatial or frequency domain of each frame [7, 8]. We therefore make the following necessary assumptions. First, we postulate that the watermarks embedded in each frame $W_k(m, n)$ are independent, have zero mean, and are Gaussian. Second, the sender embeds a watermark into every pixel of each frame of the video sequence; this assumption is valid because to maximize the steganographic capacity, a sender will make use of as much of the host signal as possible for information embedding. There is, however, a trade-off between steganographic security and transmission capacity as we later discuss.

Figure 3 displays the steganographic results for a single image frame to elucidate the concept. Figure 3(a) is the host frame also known as cover-object or cover-video frame, and Figure 3(b) is the stego-object or stego-video frame containing the Gaussian watermark (amplified for visual perceptibility) shown in Figure 3(c) with $\alpha = 5$.

Cover Image

Stego Image



Gaussian Watermark

Fig. 3. Example of steganography in a single image frame. (a) the host or cover-image frame, (b) the watermarked or stego-image frame, (c) the watermark containing the binary secret message.

The figures of merit used to assess success of the algorithm are the probability of false positive detection and the probability of false negative detection defined as follows. The probability of false positive detection is the likelihood of detecting that hidden information is present in a given video sequence when nothing has been embedded (i.e., $\alpha = 0$); that is, a given video signal is declared a stego-video when it is not. The probability of false negative detection is the likelihood of detecting that hidden information is not present when in fact it has been embedded (i.e., $\alpha \neq 0$); that is, a given video signal is declared a cover-video when it is not. A good steganalysis technique should strive to minimize both error probabilities. However, for cyber-security or computer forensic applications, it is imperative that the false negative detection rate be lower. Thus, sacrificing false positive detection for false negative detection may be necessary through the selection of appropriate algorithmic thresholds. Further processing on a video signal flagged by our technique may be optionally conducted for more accurate results. Figure 4 summarizes the basic video steganalysis problem for spread spectrum embedding.

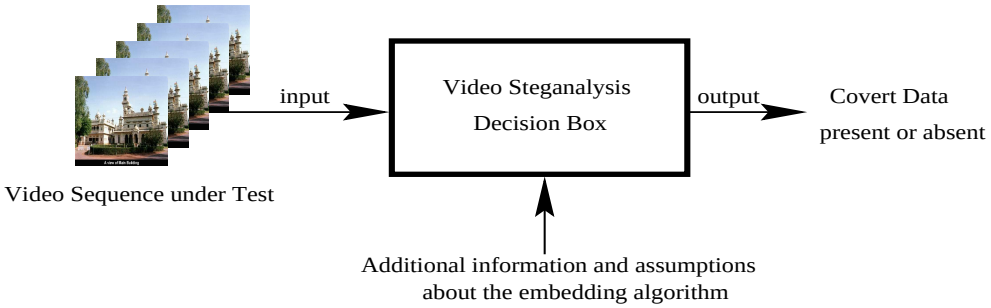


Fig. 4. Video steganalysis problem. The objective is to design an decision box that takes a given video sequence and makes use of partial information about the potential embedding algorithm to decide whether or not hidden information is present in the given media [21].

CHAPTER II

LITERATURE REVIEW

A. Past Work

Much of the research work in the field of steganalysis has been carried out on images. In raw format, video sequences can be considered as a series of still images and the steganalysis methods designed to work for still images can be applied to video sequences. So in this section we look at all the significant steganalysis methods specifically built for still images.

1. Passive Steganalysis

Most of the steganalysis methods developed over time were designed to be passive i.e. the goal was just to detect the presence of hidden messages. Jessica Fridrich, a pioneering researcher developed efficient algorithms to foil steganography schemes based on Least Significant Bit (LSB) embedding. In [23] Fridrich *et al.* propose a method to detect LSB embedding in 24 bit color images by exploiting the fact that the number of unique colored pairs decreases after embedding. The method works reasonably well but has certain constraints since the success of the method is based on the number of unique colored pairs. The authors have pointed out the infeasibility of embedding messages in digital images stored in JPEG format in [24]. The JPEG quantization matrix leaves unique fingerprints in the image. Any deviation from these characteristics signifies the presence of covert data. Other methods such as performing first order statistical analysis in the form of Chi-Square test on Pair of Values has been proposed by Westfield and Pfitzman in [25]. However this strategy fails if the LSB embedding is done at random locations based on some seed. In [26] the authors have shown how first order statistics can be defeated by making sure that the statistics derived from Pair of Values remains same before and after embedding.

The failure of first order statistical techniques led to the development of methods [17, 18, 15, 27] that uses higher order statistics. In this work, Farid and his colleagues designed a blind detection scheme that uses higher order statistics such as mean, variance, skew and kurtosis to measure the disruption of statistical regularity in the wavelet coefficients due to embedding. He uses linear and non-linear classification methods such as Fischer Linear Discrimination Analysis [17] and Support Vector Machine [18] to solve the two class classification problem. The statistics are believed to be rich enough to detect messages using different schemes. In [27] the authors have extended their method to color images and have shown how a reduction from a two class classification problem to a single class can significantly improve the detection capability. This method has better generalization properties and helps in foiling a variety of the embedding schemes.

In [1, 19] the author uses image quality metrics and multivariate regression analysis to detect the presence of covert data in an image. It has been proposed that the distance between a watermarked image and its filtered version is greater than a non-watermarked image and its filtered version. The image quality metrics most sensitive to embedding schemes [28] are chosen to measure the change in distance. The weighted sum of the distance measured from these metrics is calculated and compared to a threshold to detect the hidden messages. In a similar implementation in [19], Avcibas *et al.* use binary similarity measures to calculate disruption of the correlation between the 7^{th} and the 8^{th} bit plane due to LSB embedding. We adopt a similar strategy as proposed in [1] by using temporal filters to get the best estimate of the watermark in each frame and use characteristics of the watermark to detect it.

In [20] Harmsen and Pearlman propose a steganalysis method for all those embedding schemes where the watermark or hidden message can be modeled as an independent additive noise. Their detection scheme exploits the first order statistics of the Histogram Characteristic Function (HCF). They hypothesize that embedding lowers the center of mass

of the HCF due to filtering action of noise added in the form of hidden message. A Bayesian Classifier is used to differentiate between a cover and a stego-image by measuring the Center of Mass and comparing it to a threshold. It is not an efficient scheme since counter measures to compensate for changes in the first order statistics have been proposed. The authors have also proposed a generalized detection scheme where the training is done based on a single class and Mahalanobis distance is used for detection.

The methods proposed for steganalysis in [29, 30] targets wavelet based embedding techniques. This is of particular significance since the current image compression algorithm JPEG2000 is based on wavelets. In [30] the parameters for a Generalized Gaussian Distribution to model the sub-band coefficients in a 3 level wavelet decomposition of an image are calculated. The parameters from the high frequency horizontal, vertical and diagonal regions are fed to a neural network. The neural network is trained using a database of watermarked and non-watermarked images for which the GGD parameters are calculated. The neural-network captures the non-linearity in the decision making process.

Another method that uses wavelet analysis to detect hidden messages in wavelet domain is proposed in [29]. In the proposed method the energy of the wavelet coefficients is calculated by taking the Discrete Fourier Transform of it. The strength of the spikes in the energy curve are measured and compared to a threshold to detect presence of hidden data. Both the methods seem to work reasonably well in the wavelet domain but lack generalization.

2. Active Steganalysis

The aim of active steganalysis techniques is to estimate the hidden message or to find information pertaining to the embedding scheme. (message length, embedding strength etc.)

In [6] Chandramouli suggests a method to estimate the hidden message embedded using spread spectrum principles in two highly correlated stego images. Strong assumptions about two stego images having the same secret message and embedding key are used in the steganalysis method. He shows that the common notion of spread spectrum steganography being robust and secure is wrong. In this thesis we try and break the spread spectrum steganography and support the conception that it is not a good method for steganography. In [31] Trivedi *et al.* propose a method to find the secret key in those digital images which use sequential embedding strategy. The paper focuses on spread spectrum steganography and demonstrates that it leaves a sufficient statistical mark to facilitate active steganalysis.

Fridrich *et al.* have proposed different methods to estimate the length of the messages in digital images for different steganographic algorithms in [32]. It can accurately measure the length of the message in JPEG images using the F5 and the Outguess, in palette based images using EZstego, in raw formats using the LSB embedding schemes. The gain achieved in detecting low capacity payload with high accuracy is at the expense of the loss in generalization capability.

3. Collusion Research

In [22] Su *et al.* have presented a mathematical framework for linear collusion in video sequences and have presented the notion of statistical invisibility. A theoretical proof has been provided to justify that all watermarks embedded in the video sequences can be successfully removed if they are embedded independent of each other or have small correlation with the host sequences. The conditions in which the linear collusion scheme would fail to remove the watermark have been provided. We were inspired by this work since most of the embedding schemes fail to meet these conditions and hence linear collusion scheme could be used to detect the presence of a hidden watermark in video sequences. In [33] Kilian *et al.* have calculated the minimum number of colluders needed to have a successful

collusion scheme for images. Insight into the number of frames needed to collude in a video sequence can be drawn from the authors' findings.

In [34, 35] the authors have used non-linear collusion attack to remove the Gaussian fingerprints embedded in still images. Performance evaluation for various non-linear attacks has been done. The idea is to replace blocks in an image with similar looking blocks from other images thus changing the embedded watermark in the original image. However a non-linear attack will fail to obtain a mark free copy from the watermarked sequence which is needed in our strategy.

Temporal filtering and other intra-frame collusion schemes were implemented in [36] to remove watermarks from video sequence. A new technique for watermark removal using mosaicing was proposed. This is a potential method that could be used for steganalysis to detect the presence of a hidden data. But there are potential limitations to this method since it works well only for panoramic videos.

CHAPTER III

PROPOSED SOLUTION, ANALYSIS AND JUSTIFICATION *

The essence of a steganalysis technique is to quantify the statistical change brought in the cover medium due to embedding. In order to detect the change, a steganalyst may look for the deviation in the characteristics of the cover or probe into the features of the hidden message itself. The pros and cons of each method are discussed below.

Modeling the cover medium limits the steganalysis attack to a narrow class of cover objects that have characteristics of the natural medium. For example in [15] Farid *et al.* extract the characteristics of natural images using wavelet coefficients. The assumption is that any deviation from these characteristics signifies the presence of covert data in an image. There are limitations to this method because images such as medical images, satellite images and digital images (constructed artificially from graphics application softwares) which do not belong to the subset of natural images will always be classified as stego-images.

In order to overcome the above constraint a steganalyst may target the characteristics of the embedded message or changes brought about in the cover due to a particular kind of embedding strategy. This however leads to the loss of generality and the ability of a steganalysis method to detect messages embedded using different steganographic methods.

Due to diversity in the time varying nature of video sequences, we assert that it is impossible to find a well-defined set of features that can differentiate between natural video and stego-video. This leads us to focus on methods that target the characteristics of the hidden message. For spread spectrum steganography, this is specifically in the form of Gaussian watermarks. The detection capability of our proposed steganalysis technique

---

*Reprinted from pages 214–218, with permission from"Video steganalysis using collusion sensitvity" by U. Budhia and D. Kundur, Proceedings of SPIE: Sensors, Command, Control, Communications and Intelligence(C3I) Technologies for Homeland Security and Homeland Defense, April 2004, vol. 5403.

is theoretically limited to spread spectrum steganography but is applicable to embedding either in the spatial or in the frequency domain.

A. Basic Architecture

As discussed above, the spirit of most steganalysis methods is to devise a function that differentiates between the general characteristics of a signal with and without embedding [21]. This function is normally compared implicitly or explicitly to a threshold in order to decide whether or not a given signal Y_k contains hidden information[1]. Much research on image steganalysis has focused on identifying image features that change when steganography algorithms are applied. Researchers have traditionally employed image processing and statistical tool-sets that in some form attempt to estimate a potential "host" $\hat{U}_k = \mathcal{H}[Y_k]$ signal from Y_k. This "host" estimate $\hat{U}_k$ is then compared in some way to Y_k in order to detect if something is hidden. The basic hypothesis is that the deviation of specific characteristics of Y_k and $\hat{U}_k$ will differ if something is embedded in Y_k (i.e., $Y_k = X_k = U_k + \alpha \cdot W_k$) in comparison to when nothing is embedded in Y_k (i.e., $Y_k = U_k$). Pattern classification is often employed to characterize this deviation effectively.

In this thesis, we formulate a novel framework for this problem that employs previous research on digital watermarking attacks. The advantage is that instead of searching libraries of image processing and statistical functions in order to identify potential candidates for steganalysis, we borrow on venerable research in the related field of digital watermarking. Furthermore, our approach is general and can be targeted to identify specific types of steganography by replacing our general blocks with appropriate algorithms.

---

[1]Please note that we have removed the subscripts m, n from our notations for clarity. For the rest of this thesis we will assume that all operations are done on the entire frame unless stated otherwise.

Figure 5 presents our framework. The video sequence under consideration Y_k is passed through a digital watermarking attack block that attempts to estimate the host signal to produce $\hat{U}_k$. This block may assume knowledge of the embedding algorithm (if any is used) to be effective. The estimate of the watermark $\hat{W}_k$, calculated by taking the difference between Y_k and $\hat{U}_k$, is passed through an appropriate pattern classifier. If Y_k is a stego-video then the input to the pattern classifier is a Gaussian watermark signal corrupted by some noise due to filtering(watermarking attack). On the contrary, if Y_k is a video signal without any watermark the estimate $\hat{W}_k$ would simply consist of the noise due to filtering. In an ideal case, if the filter is able to perfectly re-construct the host, the estimate $\hat{W}_k$ will consist of the original Gaussian watermark embedded in case of a watermarked video and will be zero for a non-watermarked video. By employing some a priori information about the embedding algorithm, the distinction between these two cases can be made to detect the presence of covert communication.

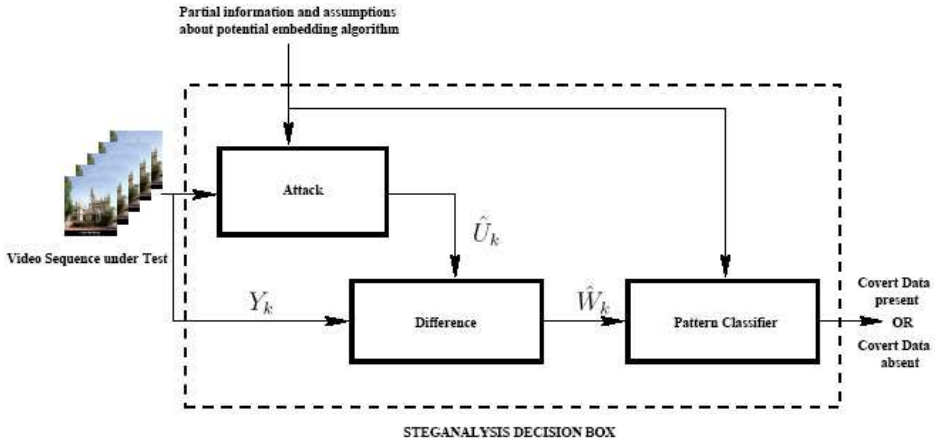


Fig. 5. Proposed framework for steganalysis [21].

Since our goal is, in part, to develop a tool to enhance existing image steganalysis methods, we focus on algorithms for Figure 5 that account for temporal changes in a signal due to embedding. Together, with image steganalysis methods that incorporate spatial information through the use of (weighted) mean and Wiener filters, an improved solution may be produced. We conjecture that the linear collusion attack, used to remove the presence of independent digital watermarks in a sequence of images or video frames is ideal for our problem. First, the attack focuses on temporal correlations between video frames to estimate a "host" video sequence that can be easily incorporated into our framework. Second, much analytic and simulation-based work focuses on this area providing a strong foundation upon which to build a steganalysis method. Finally, the attack is computationally simple making our steganalysis approach practically feasible for real-time applications.

An effective pattern classifier is also developed by incorporating knowledge that the watermark, if any present, is zero mean and Gaussian. The design of the pattern classifier is discussed in the future sections. In the next subsections we dicuss the linear collusion scheme which is used to estimate the host sequence U_k from the received signal Y_k. In particular we discuss the various schemes that can constitute the "attack" block in Figure 5.

B. Collusion Attack

Collusion for digital watermarking and steganography refers to the use of multiple image frames (that may or may not form a video sequence) in order to remove the presence of a watermark in one or more of the image frames. In general, the collusion attack may be linear or nonlinear exploiting the differences and similarities between frames to judiciously reduce the energy of the watermark in comparison to that of the host information. We represent collusion of a sequence of video frames, which produces a resulting frame that has lower watermark content as follows:

$$\hat{X}_k = \mathcal{C}[X_1, X_2, \ldots, X_N] \tag{3.1}$$

where $\hat{X}_k$ is called the colluded result and in this thesis represents the estimate of the kth host frame U_k. $\mathcal{C}$ is the collusion operator that exploits the similarities and differences amongst all or a select subset of watermarked image frames $X_1, X_2, \ldots, X_N$ to produce $\hat{X}_k$. As we discuss, the colluded result $\hat{X}_k$ in general contains significantly less contribution from W_k as compared to X_k. Common forms of the collusion operator $\mathcal{C}$ include taking the pixel-by-pixel maximum, minimum, mean or median over a range of image frames.

Linear collusion is a special case in which $\mathcal{C}$ represents a weighted average operation of select video frames. Intuitively, linear collusion on a sequence of video frames amplifies parts of the frames that are similar and attenuates components that are different. In the next subsection we concentrate on a subset of linear collusion attack where the weights applied to each frame in the collusion attack are equal. This leads to a simple collusion scheme where we take an average over a range of video frames. For the rest of this thesis we refer to the averaging based collusion method as the simple linear collusion scheme. The linear collusion method where the weights are different will be referred to as the weighted collusion scheme.

1. Simple Linear Collusion Scheme

Linear collusion has recently received much attention in the digital video watermarking community [22, 33]. It has been shown analytically that if the linear correlation amongst host video frames U_i for some i differs from that of the watermark frames W_i over the same range of i then the linear collusion scheme based on averaging will be successful in either attenuating or amplifying the presence of the watermark in the resultant frame $\hat{X}_k$ [22].

In this thesis, we focus on the application of spread spectrum steganography on video sequences that in most applications requiring high covert data capacity implies that W_i is independent for each frame. We assume that the motion in the video sequence is "slow" which implies that adjacent video frames are similar. Because of this visual correlation, it is expected that over a neighborhood of i centered at k, the watermarked video frames can be averaged in order to attenuate the presence of the watermark in the k^{th} frame.

Let us assume that we use a sliding window to denote the temporal neighborhood used for frame averaging; this window is assumed to contain visually similar frames. Specifically, we take a window size of $2L + 1$ frames centered at frame k (except toward the beginning and end of the sequence since the window goes outside the range of i) to average the video sequence. Let us formally define the collusion operator $\mathcal{C}_L$ for the simple linear collusion scheme as:

$$\mathcal{C}_L = \frac{1}{2L+1} \sum_{i=k-L}^{k+L} \tag{3.2}$$

The operator represents an averaging over a window of $2L+1$ frames centered over a frame having index k.

The estimate of the kth host frame is given by:

$$\hat{X}_k = \mathcal{C}_L(X_k) \tag{3.3}$$

$$\hat{X}_k = \frac{1}{2L+1} \sum_{i=k-L}^{k+L} X_i \tag{3.4}$$

Equation 3.4 is modified for frames that lie in the beginning and in the end of a video sequence.

$$\hat{X}_k = \begin{cases} \frac{1}{2L+1} \sum_{i=1}^{2L+1} X_i & 1 \leq k \leq L \\ \frac{1}{2L+1} \sum_{i=k-L}^{k+L} X_i & L < k < N-L \\ \frac{1}{2L+1} \sum_{i=N-2L}^{N} X_i & N-L \leq k \leq N \end{cases} \tag{3.5}$$

where k is the frame under consideration to produce $\hat{X}_k$, an estimate of U_k. We next show why we assert that $\hat{X}_k \approx U_k$.

Substituting $X_i = U_i + \alpha.W_i$ for all i from Equation 1.2 into Equation 3.5 we obtain:

$$\hat{X}_k = \frac{1}{2L+1}\sum_i U_i + \frac{\alpha}{2L+1}\sum_i W_i \tag{3.6}$$

where the summations are over the appropriate domains for the various ranges of k shown in Equation 3.5. Since the watermarks W_i are independent and zero mean, the second term of the left hand side of Equation 3.6 approaches zero as L increases. Furthermore, because we assume $U_i \approx U_k$ for all i in the neighborhood of the sliding window centered at k, the first term will dominate resulting in the following approximation:

$$\hat{X}_k \approx \frac{1}{2L+1}\sum_i U_i \tag{3.7}$$

$$\approx \frac{1}{2L+1}\sum_i U_k \tag{3.8}$$

$$\approx U_k \tag{3.9}$$

The effectiveness of $\hat{X}_k$ as an approximation of U_k depends on the value of L in relation to the rate of motion in the video sequence. Through extensive analysis we show that an optimum value of L will lead to the cancelation of the Gaussian watermarks and ensure the assumption that $U_i \approx U_k$ for all i holds true.

If collusion is applied to a given video sequence Y_k that may or may not contain a watermark, we believe that in both cases for slowly varying video and an appropriately selected value of L, the result will be an effective approximation of U_k. Thus if a watermark is embedded in the video, subtracting $\hat{X}_k$ from Y_k gives $Y_k - \hat{X}_k \approx Y_k - U_k = \alpha W_k$, an estimate of the scaled zero mean Gaussian watermark. If no watermark is present in Y_k then the result will be independent of any characteristics such as Gaussianity that we assume for

the watermark. This difference is used by a pattern classifier discussed in the future sections for steganalysis.

In case of a non-watermarked video we have $Y_k = X_k = U_k + \alpha W_k$, where $\alpha = 0$. The estimate of the scaled watermark is denoted by,

$$\begin{aligned}
\hat{W_k} &= Y_k - \hat{X_k} \\
\hat{W_k} &= Y_k - \mathcal{C}_L(X_k) \\
\hat{W_k} &= U_k + \alpha W_k - \mathcal{C}_L(U_k + \alpha W_k) \\
\hat{W_k} &= U_k - \mathcal{C}_L(U_k) \quad \text{Since } \alpha = 0 \text{ for non watermarked sequences} \qquad (3.10) \\
\hat{W_k} &= n_k \quad \text{where } n_k = U_k - \mathcal{C}_L(U_k) \qquad (3.11)
\end{aligned}$$

The residual "noise" from the simple linear collusion scheme is denoted by n_k and is a measure of the invariance of the collusion operator on legitimate non-watermarked data. Ideally we would like $\mathcal{C}_L(U_k) \approx U_k$.

In case of watermarked sequences we have $Y_k = X_k = U_k + \alpha W_k$, the estimate of the scaled watermark is given by,

$$\begin{aligned}
\hat{W_k} &= Y_k - \hat{X_k} \\
\hat{W_k} &= Y_k - \mathcal{C}_L(X_k) \\
\hat{W_k} &= U_k + \alpha W_k - \mathcal{C}_L(U_k + \alpha W_k) \\
\hat{W_k} &= U_k - \mathcal{C}_L(U_k) + \alpha W_k - \alpha \mathcal{C}_L(W_k) \qquad (3.12) \\
&\qquad \text{Since } \mathcal{C}_L(a+b) = \mathcal{C}_L(a) + \mathcal{C}_L(b) \text{ in case of linear collusion} \\
\hat{W_k} &= n_k + W_k^{'} \quad \text{where } W_k^{'} = \alpha(W_k - C(W_k)) \qquad (3.13)
\end{aligned}$$

In the case of the watermarked sequences the estimate of the watermark is the sum of the noise due to collusion attack and a Gaussian signal which bears a very high correlation

with embedded watermark W_k. In case all the host frames are same, n_k will be zero and the estimate of the watermark $\hat{W}_k$ will be the embedded watermark W_k. We can represent the steganalysis in terms of a hypothesis testing problem.

a. Hypothesis Testing

The video steganalysis problem can be mathematically formulated as a hypothesis testing problem.

$$\begin{cases} H_0 : \hat{W}_k = n_k & \textit{k=1,2,...,N} \text{ if watermark is absent} \\ H_1 : \hat{W}_k = n_k + W_k^{'} & \textit{k=1,2,...,N} \text{ if watermark is present} \end{cases}$$

where n_k is the residual noise defined above and $W_k^{'}$ is a Gaussian watermark signal. The aim of steganalysis is to differentiate between the two situations and simultaneously minimize the probability of false positive and false negative. The probability of false negative can be defined as the probability of choosing H_0 when it is actually H_1. Similarly, the probability of false positive can be defined as the probability of choosing H_1 when it is actually H_0.

C. Theoretical Justification and Analysis

The steganalysis method proposed looks at the characteristics of the watermark and uses pattern recognition for finding the hidden messages. Therefore the accuracy of the estimated watermark is related to the accuracy of the hypothesis testing. In this section we study the performance of the collusion scheme, estimate the bounds on the embedding parameters that will lead to the failure of the collusion based steganalysis and find the optimum length for collusion attack.

We make the following assumptions about the video sequences and the watermark frames.

(1) The host frames U_k are assumed to be from a distribution having mean μ and variance σ_u^2.

(2) The correlation model of the host frames follows the first-order Markov model where the correlation between frame U_i and U_j is given by $\rho^{|i-j|}$. Where ρ is the correlation coefficient between any two adjacent frames.

(3) The watermark frames W_k are assumed to be independent from U_k and from each other, and derived from a Gaussian distribution having mean 0 and variance σ_w^2. Since the watermark is embedded with an embedding strength of α the effective variance of the watermark is $\alpha^2\sigma_w^2$.

For slow moving sequence where the scene changes are not drastic we can reasonably make an assumption that the frames have approximately the same mean and variance as stated in Assumption (1). In Assumption (2) we assert that a first order Markov model can be used to model the correlation between various frames of a video sequence. By intuition we know that the correlation between a reference frame and other frames in a video sequence decreases as one moves away from the reference frame. We model this decrease in correlation using the term $\rho^{|i-j|}$, where $|i-j|$ represents the distance between the reference and the other frames in terms of frame index. We note that the term $\rho^{|i-j|}$ decreases with an increase in the distance since, $|\rho| \leq 1$ always holds true.

D. Effectiveness of Simple Linear Collusion Scheme

The effectiveness of the collusion scheme proposed can be studied by looking at the expected Mean Squared Error(MSE) between the estimate of the watermark $\hat{W}_k$ and the embedded watermark αW_k. In order to get the best estimate, the expected MSE should be minimized. In this section we look at conditions where the frame averaging or simple

linear collusion scheme will be successful in extracting the watermark from the original frames.

The mathematical equation to represent the expected MSE between the estimated watermark and the embedded watermark in each frame is given by:

$$\begin{aligned}
\mathbf{E}[(\hat{W}_k - \alpha W_k)^2] &= \mathbf{E}[(Y_k - \hat{X}_k - \alpha W_k)^2] \\
&= \mathbf{E}[(X_k - \hat{X}_k - \alpha W_k)^2] \\
&= \mathbf{E}[(U_k + \alpha W_k - \hat{X}_k - \alpha W_k)^2] \\
&= \mathbf{E}[(U_k - \hat{X}_k)^2] \qquad (3.14) \\
&= \mathbf{E}[(U_k - \hat{U}_k)^2] \qquad (3.15)
\end{aligned}$$

This equation shows that the expected value of the expected MSE between the estimated watermark and the original watermark is the same as the expected value of the MSE between the original host frame and the colluded host frame.

Proposition 1 *Given a sequence of watermarked video frames $X_k, k = 1, 2, \ldots, N$ as defined by Equation 1.2. Under assumptions (1), (2) and (3) the expected MSE between the original watermark and the estimated watermark obtained from collusion attack is given by*

$$\boldsymbol{E}[(\hat{W}_k - \alpha W_k)^2] = \sigma_u^2 \left[\frac{z-1}{z} - \frac{2\rho}{z(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{z(1-\rho)} - \frac{2\rho(1-\rho^z)}{z^2(1-\rho)^2} \right] + \frac{\alpha^2 \sigma_w^2}{z} \quad (3.16)$$

where $z = 2L + 1$.

Proof: See Appendix A.1

In the next proposition we introduce the concept of no-collusion attack. We define the no-collusion attack as the collusion scheme where the number of frames colluded is one. In the trivial case the estimate of the watermark will always be zero irrespective of whether it is a watermarked or a non-watermarked sequence.

Proposition 2 *Under assumptions (1), (2) and (3) the expected MSE between the original watermark and the estimated watermark when there is no collusion is given by*

$$\boldsymbol{E}[(\hat{W}_k - \alpha W_k)^2] = \alpha^2 \sigma_w^2; \tag{3.17}$$

Proof: See Appendix A.2

Since the estimate of the watermark is always zero in case of no collusion the expected MSE between the watermarks is always equal to the variance of the effective watermark embedded i.e. $\alpha^2\sigma_w^2$.

The next proposition helps us in analyzing the success of the collusion attack. We look at the ratio of the variance of the embedded watermark and the variance of the host frame. It is a measure of signal-to-noise ratio(SNR) where the watermark is the signal and the interference comes from the host frames.

Proposition 3 *From Propositions 1 and 2 we obtain the following bound on the ratio of the variance of the host frames σ_u^2 to the effective variance of the embedded watermark $\alpha^2\sigma_w^2$.*

$$\frac{\sigma_u^2}{\alpha^2\sigma_w^2} < \frac{1}{1 - \frac{2\rho}{(z-1)(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{(z-1)(1-\rho)} - \frac{2\rho(1-\rho^z)}{z(z-1)(1-\rho)^2}} \tag{3.18}$$

where $z = 2L + 1$.

Proof: See Appendix A.3

1. Discussion

We arrive at the bounds on the ratio of the variance of host frame to that of the effective variance of the watermark by laying a constraint that the expected MSE between watermarks in case of simple linear collusion is smaller than the expected MSE encountered when there is no collusion at all. There is no additional advantage of using the simple linear collusion

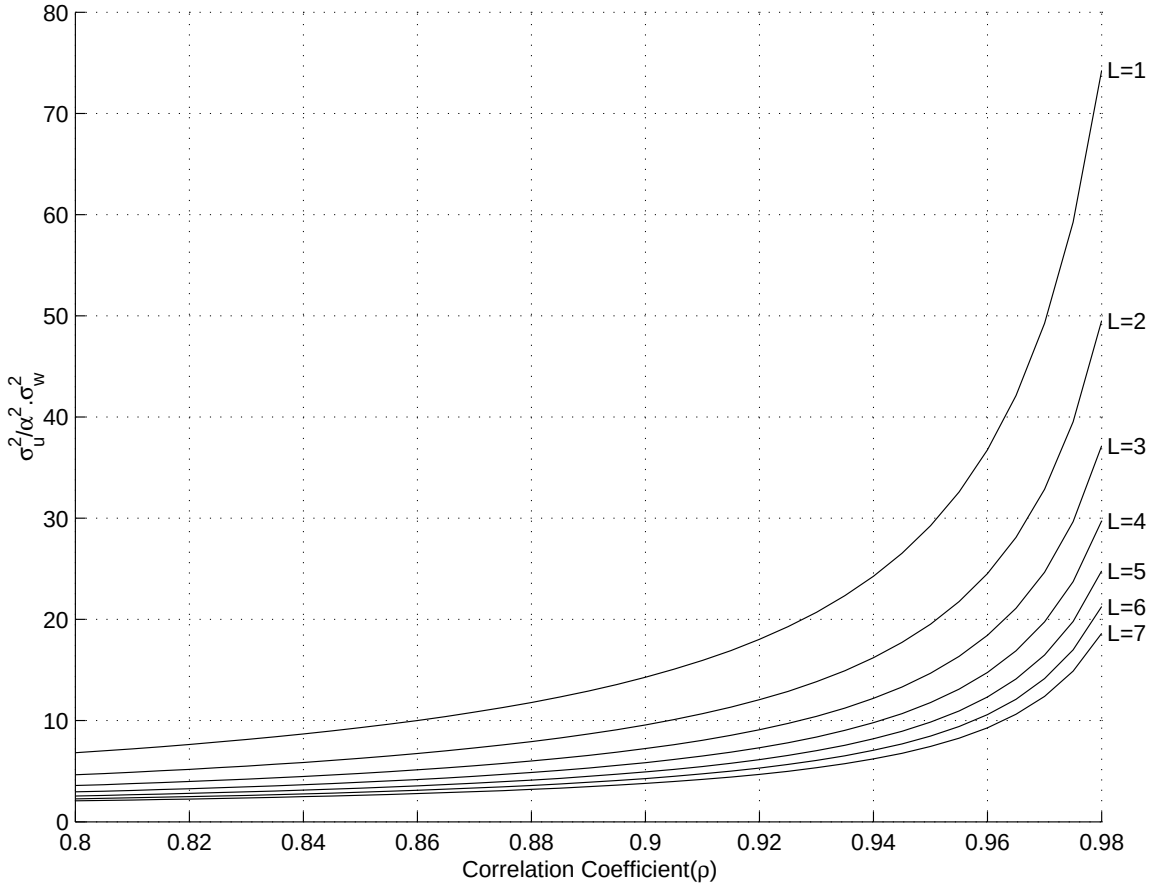


Fig. 6. Upper bound on $\frac{\sigma_u^2}{\alpha^2\sigma_w^2}$.

method if the expected MSE between watermarks in case of this method is larger than that encountered for simple guessing. Therfore we present the conditions where simple linear collusion scheme will be successful in reducing the MSE.

Figure 6 shows the upper-bound on the ratio of the variance of the host frames to the strength of the watermark for various values of L and correlation coefficient ρ as given by Equation 3.18. We would like to recall that the size of the window in case of collusion is given by $z = 2L + 1$. e.g. From Figure 6 we see that for a correlation coefficient of $\rho = 0.94$ between adjacent frames and $L = 4$ the maximum ratio of the variances can

be 10. This means that if the variance of the frame is greater than 10 times the effective variance of the watermark a collusion length of 4 will yield a higher expected MSE than the case when there is no collusion. However we can use a lower value of L to facilitate collusion attack.

The choice of a higher value of L is made to cancel the Gaussian watermarks (Since $\lim_{L\to\infty} \sum_{k=1}^{L} W_k = 0$). However, if the correlation between frames is small an increase in L would increase the residual noise due to collusion. We have shown above that an increase in L will lead to a situation where the expected MSE between watermarks will be greater than the expected MSE in case of no collusion. Hence there is a tradeoff and in order to use collusion to estimate the host frame one will be forced to use a lower value of L.

As the correlation coefficient increases, the upper bound on the ratio increases exponentially. This implies, for a fixed frame variance the ability to collude a watermarked video sequence embedded with lower embedding strengths increases with increase in correlation. Given that the strength of the watermark $\alpha^2\sigma_w^2$ is small in comparison to the strength of the host video σ_u^2 to guarantee imperceptibility, the practical operating range for parameters exists toward the right hand side of Figure 6 (for large ρ).

Although Figure 6 provides us with an idea of when the collusion approach to steganalysis holds promise, it does not, however, give information about the optimal value of L to produce the best estimate of the watermark.

In Figure 7 the expected MSE between the watermarks is plotted as a function of L for various values of ρ in terms of the strength of the watermark $\alpha^2\sigma_w^2$. The variance of the host frames σ_u^2 is assumed to be 10 times the strength of the watermark $\alpha^2\sigma_w^2$. For a SNR of 0.1 we see that if the correlation coefficient is greater than 0.86 we do have a local minimum. The value of L corresponding to the point of local minimum gives the optimum size of the window for collusion attack for a given SNR and correlation coefficient. Intuitively we can see that as the correlation decreases simple linear collusion scheme yields a higher MSE

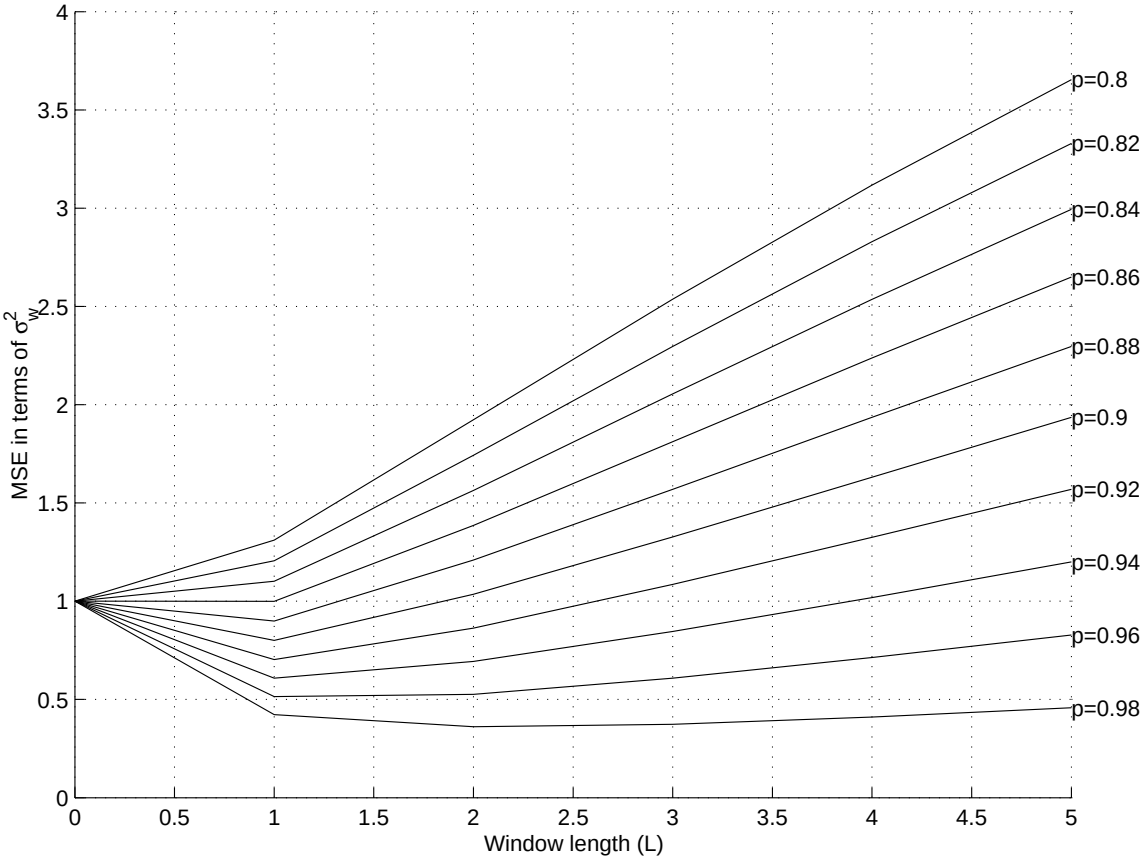


Fig. 7. MSE as a function of collusion length and correlation coefficient.

than the case when we do not have any collusion. The increase in the correlation between the frames results in the increase in the optimum number of frames needed for collusion to minimize the MSE. If there is perfect correlation between the frames ideally we should use infinite number of frames for collusion attack. From the figure we see that the optimum value of L is 1 and 2 for correlation coefficient of 0.94 and 0.98 respectively.

The reader should note that in the case of fast moving video sequences, the simple linear collusion scheme applied to dissimilar frames may not result in a reasonable approximation for U_k. However, in the next subsections we provide a practical alternative to

improve simple linear collusion performance for steganalysis that involves using weighted collusion attack and block based collusion attack.

E. Weighted Collusion Scheme

Linear collusion scheme in the form of frame averaging is sub-optimal since the weights are assumed to be the same for each frame in the collusion attack. A weighted collusion attack may be used to lower the expected MSE between the watermarks. The weighted collusion scheme can be visualized as a low pass filter applied in the temporal domain. The taps of the filters are represented by the weights in the weighted collusion scheme. Equation 3.4 can be modified to represent the weighted collusion scheme in the following way:

$$\hat{X}_k = \sum_{i=k-L}^{k+L} \beta_i \cdot X_i \tag{3.19}$$

However we need to empirically find the weights in order to facilitate the weighted collusion attack.

Proposition 4 *The weights for the weighted collusion scheme as defined in equation 3.19 is given by the following equation:*

$$B = A^{-1}P \tag{3.20}$$

where,

$$P = [1 \quad \sigma_u^2\rho^L \quad \sigma_u^2\rho^{L-1} \ldots \sigma_u^2\rho^L]^T$$

$$B = [\beta_{k-L} \quad \beta_{k-L+1} \ldots \beta_{k+L} \quad \lambda]^T$$

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ \sigma_u^2 + \alpha^2\sigma_w^2 & \sigma_u^2\rho & \dots & \sigma_u^2\rho^{2L} & 1 \\ \sigma_u^2\rho & \sigma_u^2 + \alpha^2\sigma_w^2 & \dots & \sigma_u^2\rho^{2L-1} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_u^2\rho^{2L} & \sigma_u^2\rho^{2L-1} & \dots & \sigma_u^2 + \alpha^2\sigma_w^2 & 1 \end{bmatrix}$$

Proof: See Appendix A.5

The equation suggests that the correlation between frames ρ, the host variance σ_u^2 and the watermark variance $\alpha^2\sigma_w^2$ should be known ahead of time in order to derive the optimal weights for the weighted collusion attack. This is not possible at all times. We can however have a rough estimate of the correlation and the host variance from the test sequence. The approximate weights can be derived by assuming a reasonable value of the variance of the watermarks added.

The other assumption which has been made is that the mean and the variance of each frame in the host video sequence is constant. This may not be true since due to the time-varying nature of the video sequences the mean and variance may vary from frame to frame. So in order to overcome these problems we suggest an adaptive method that can be applied to calculate the weights, and is free of the above constraints.

In the adaptive scheme we embed another Gaussian watermark to the test sequence using the spread spectrum technique as defined in Equation 1.2. The test sequence may or may not contain a hidden message in the form of the original Gaussian watermark. The idea is to find the weights to maximize the correlation between the Gaussian watermark embedded to the test sequence and the estimate of this watermark using the weighted collusion scheme. The weights are found using an iterative search procedure such as gradient descent approach. Once the weights are found, these set of weights are used to estimate the original watermark embedded in the host sequence. We expect that the weights should

work reasonably well in estimating the original watermark in the host sequence. The results using this method are discussed in the next chapter.

F. Block-based Collusion Scheme

In case of fast moving sequences or sequences having non-translational motion the simple collusion scheme or the weighted collusion scheme may be sub-optimal. We recall that the aim of collusion is to produce a watermark free frame from a set of similar watermarked frames. The colluded frame is a close approximation to the host frame. We can imagine each frame to be made of 8x8 blocks and visualize the collusion attack as the collusion of the blocks. We note that in case of simple linear collusion or weighted collusion scheme the blocks that are colluded from different frames may be visually dis-similar and our assumption that all the frames/blocks in the neighborhood of center frame are similar may not hold true. So in order to increase the correlation between the blocks that are colluded we use a block-based similar to MPEG/H.263x coding schemes.

Block based collusion scheme for five frames is shown in Figure 8. The frame corresponding to the center of the window X_k is assumed to be the reference frame. For each block in the reference frame the best match is found in all the other frames (X_{k-2},X_{k-1},X_{k+1},X_{k+2}) in the window. A new set of reconstructed frames (X'_{k-2},X'_{k-1},X'_{k+1},X'_{k+2}) are formed from the matched blocks. The matched blocks are placed in the reconstructed frames at the position corresponding to the reference block in the reference frame. The reconstructed frame corresponding to the reference frame X_k is formed by simply copying the reference frame. The process is repeated for all blocks in the reference frame. Once the reconstructed frames are formed collusion is performed to estimate the host frame U_k. Like before, we perform the collusion operation to estimate all the host frames in the video sequence by shifting the window.

Another insight which is drawn is that the effective embedding data rate that can be achieved in a video sequence can be significantly reduced if a block based collusion attack is used instead of frame based collusion attack. The effective correlation between the blocks will be higher for non moving parts and will help in detecting messages embedded with very low strengths in those areas. Thus from an embedder's point of view he/she can hide the messages only in the moving areas for which a good match cannot be found in the frames under collusion attack.

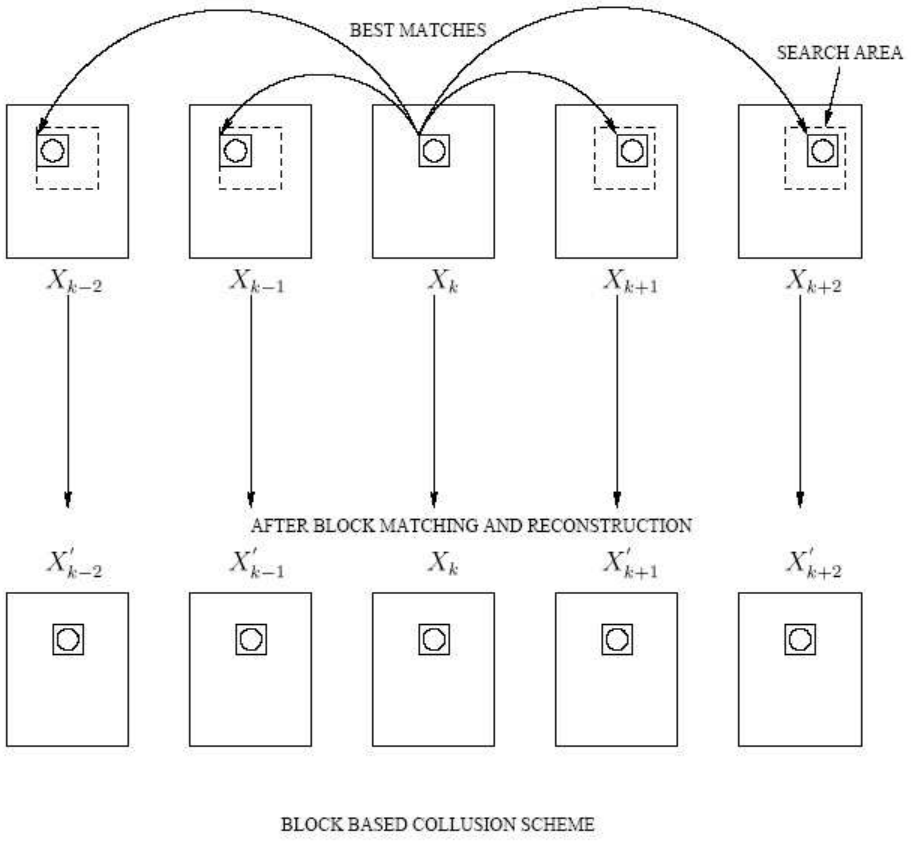


Fig. 8. Block based collusion attack.

In the next section we discuss the ways to implement a pattern classifier. The input to the pattern classifier will be the estimate of the watermark in each frame. The classifier will give a decision to whether there is a message hidden in each frame or not.

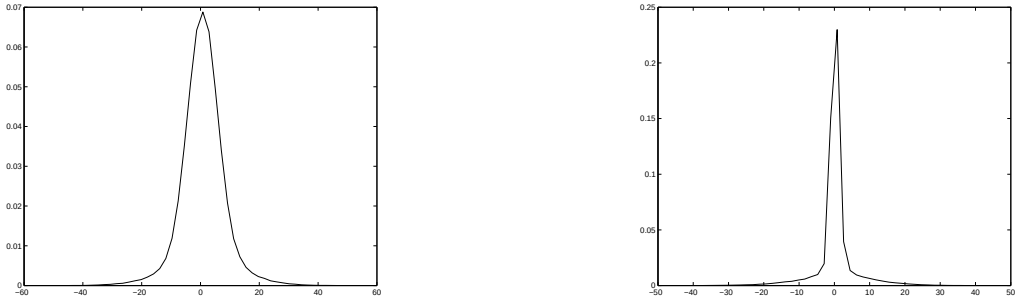
G. Pattern Classifier

A pattern classifier helps in assigning class labels to the objects from one of the underlying classes in the training data. In the perspective of this thesis, the pattern classifier should be able to discriminate between a stego and a cover video based on the input to the classifier, which is the estimate of the watermark in each frame. The two main components of a pattern classifier are the feature extraction and the discriminator. We discuss the design of each of these components in the next subsections.

1. Feature Extraction

Feature extraction is a process of extracting the distinctive features or characteristics from a data set to help the discriminator in distinguishing between different classes. The features extracted from the estimate of the watermark will aid the classifier in detecting the presence of covert data in the video sequence or help in rejecting one of the two hypotheses.

Figure 9 gives an example of the distribution of the estimated watermark $\hat{W}_k$ for a frame from a watermarked and a non-watermarked video sequence. It is clear that there exists a difference between the two cases that can be quantified through statistical features; the case in which no watermark is present results in a distribution that is not Gaussian. Since we assume that steganography occurs through the addition of Gaussian watermarks, we employ features that can measure the level of Gaussianity in a signal. These include kurtosis, entropy and the 25^{th} percentile.



(a) Watermarked sequence (b) Non-watermarked sequence

Fig. 9. Distribution of the watermark estimates for a video sequence (a) with and (b) without steganographic data embedded [21].

a. Kurtosis

Kurtosis [37] is a value that partially measures the "shape" of a distribution. Kurtosis for a Gaussian distribution is 3 and for most of the other distributions it is more than or less than 3 depending on the shape of the distribution. It is defined as

$$Kurtosis = \frac{1}{\sigma^2 N} \sum (x - \mu)^4, \tag{3.21}$$

where σ and μ represent the variance and mean of the distribution. Kurtosis also measures the peakedness of a distribution. A higher value signifies a distribution with higher peak than the normal distribution. We expect the kurtosis of the estimate from the watermarked sequence to have a kurtosis close to 3. The estimate from a non-watermarked sequence should yield a higher kurtosis value owing to is peakness. We can see from Figure 9 that the distribution from the non-watermarked sequence has a curve which is peakier as compared to the other.

Table VII shows the average kurtosis values for the estimates of the watermark over 40 frames for different watermarked and a non-watermarked sequences. We can see that the kurtosis values from non-watermarked sequences are much higher as compared to the watermarked sequences, thus supporting our theory. We note that the kurtosis values from

a watermarked sequence are closer to 3 only for higher embedding strengths. The estimate of the watermark from a watermarked video is given by Equation 3.13 and it shows that the estimate is the sum of residual noise n_k and the Gaussian signal $W_k^{'}$. At lower embedding strength the residual noise masks the Gaussian signal and hence the kurtosis values are higher than expected.

b. Entropy

Entropy [37] helps to determine the degree of "randomness" in a given distribution. For a fixed variance the Gaussian distribution has the maximum entropy. Thus the estimates obtained from the watermarked video sequence should have a higher entropy than those obtained from a non-watermarked sequence since there are a lot of points close to zero. Entropy is given by

$$Entropy = -\sum_{i=1}^{N} (p_X(i) log(p_X(i))), \tag{3.22}$$

where $p_X(i)$ is an estimate of the distribution of $\hat{W}_k$ shown in Figure 9 for a specific test case. In [38] the authors define a good steganographic algorithm as one that can minimize the increase in entropy due to embedding.

We mathematically show that the entropy for the estimates of the watermark in each frame from a non-watermarked and a watermarked sequence are different and hence is a good feature for the classifier. Let us represent the entropy of the estimate of the watermark obtained for non-watermarked sequences as E_0 and the entropy of the estimate of the watermark from a watermarked sequence as E_1. The estimate of the watermark obtained from a watermarked sequence consists of the residual noise encountered due to collusion attack and the Gaussian signal $W_k^{'}$ as shown in equation 3.13. The estimate $W_k^{'}$ is independent of n_k and hence the entropy E_1 can be represented as the sum of E_0 and entropy of $W_k^{'}$.

Proposition 5 *The entropy (E_1) of the estimate of the watermark obtained from a watermarked sequence is greater than the entropy (E_0) of the estimate of the watermark obtained from a non-watermarked sequence in case of simple linear collusion scheme.*
Mathematically,

$$E_1 = E_0 + \frac{1}{2} log(2e\pi\sigma_w^{'2}) \tag{3.23}$$

where $\sigma_w^{'2} = \frac{2L}{2L+1}\alpha^2\sigma_w^2$

Proof: See Appendix A.4

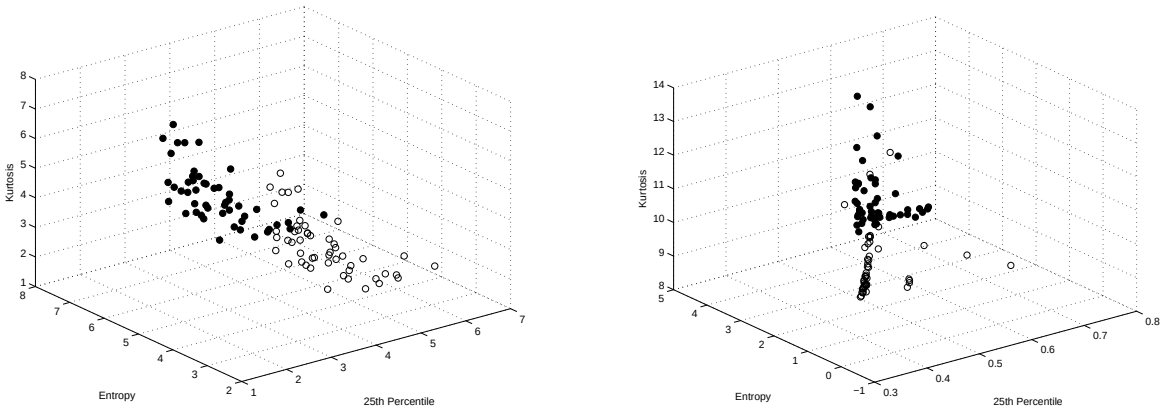
Equation 3.23 suggests that as L increases the difference between E_1 and E_0 is maximized. We would like the difference to be maximized since this is one of the discriminating features used in the classifier. Also, increasing the window length L facilitates the removal of Gaussian watermarks using collusion scheme. However increasing L will also increase n_k, the noise due to collusion attack and increase the expected MSE between the watermarks. Thus we take an optimum value of L that provides enough discriminatory information as well as keeps the noise low.

c. 25^{th} Percentile

The last feature that we consider is the 25^{th} percentile of a given distribution defined as the value above which 25% of the points in the histogram reside. From Figure 9 it is clear that the distribution when a watermark is present is more spread than when no watermark is present resulting in a difference in this percentile value.

Figure 10 represents a *scatter plot* of specific statistical features of $\hat{W}_k$ for different video sequences that do and do not contain steganographic information. The features are estimates of the kurtosis, entropy and 25^{th} percentile of the distribution of $\hat{W}_k$ to form a three-dimensional feature vector that is plotted for different video frames in two different

test video sequences (shown as parts (a) and (b) in the figure). The colored vector points represent the results for different video containing hidden information and the clear points are the results for no hidden information. The separate clustering for the two cases is clear which makes classification possible.



(a) Scatter plot for "Backyard" video sequence.

(b) Scatter plot for "Hotel" video sequence.

Fig. 10. Scatter plots of kurtosis, entropy and 25^{th} percentile feature vectors extracted in each frame for two different test video sequences. The colored and clear points represent the cases with and without a watermark present in the video, respectively [21].

Once the features are extracted, we build a kNN classifier [39, 40]. More sophisticated classifiers using support vector machines and neural networks [40] could have been employed for discrimination, but are higher in complexity without providing significantly improved performance.

2. KNN Classifier

Classifier is an entity that assigns a class or a group to the feature vector extracted from the test data. In other words it labels the test set into one of the underlying groups from the training data. The kNN algorithm classifies the feature vector extracted from the test

data set on the basis of its similarity with the feature vectors from the training set [40]. As the name suggests, the k nearest neighbor algorithm finds the k closest neighbors or feature vectors to the test feature vector in terms of some distance measure (Euclidean in our case) in the training set. It assigns a class on the basis of the class labels that appear most in the k nearest neighbors found. The inputs required for a kNN classifier are the training data, integer k and the metric to measure the closeness. The value of k will be calculated experimentally and the training set is chosen using Cross Validation [39, 40] which is discussed in the next subsection.

a. Training

Training is necessary in a pattern classifier to help the classifier in extracting the important characteristics of all the classes from the data sets for which we know the class labels. The training was done in the following way. We picked up 14 video sequences having different characteristics so that it represents a broad category of video. These sequences were watermarked using the spread spectrum technique as shown in Equation 1.2. The same set of sequences were used to represent the situation where there are no hidden messages embedded by leaving the sequences unmarked. Features were extracted from both of these classes and labeled as Class 1 and 2. Cross validation, a method used to find the best training data from a large set, was used to select the sequences or feature vectors that were used to represent different classes in the final classifier. The idea of cross validation is to pick n random test sequences from this training set and predict the probability of false negative and false positive for the rest of the 14-n sequences. The n sequences act as the training set for the rest 14-n sequences. The process is repeated a number of times to arrive at the training sequences that will minimize the probability of false negative and false positive. This is picked as the final training set and the rest of the sequences are discarded.

Table I summarizes the overall steganalysis method that incorporates linear collusion and classification.

Table I. Simple linear collusion based steganalysis.

- Variable Definitions:

N	Number of Frames
$X_k(,)$	k^{th} frame of the video sequence
$Y_k(,)$	k^{th} received frame of the video sequence
$\hat{X}_k(,)$	colluded version of the k^{th} frame
$\hat{W}_k(,)$	estimate of the watermark in the k^{th} frame
O_k	Output from the pattern classifier for the k^{th} frame
$Coll()$	Averaging based collusion attack on 2L+1 frames as described in Section B
$Patt()$	Pattern Classification on as described in Section G

- Algorithm:

for k={1,2,. . . ,N}

$\hat{X}_k(,) := Coll(X_k(,))$

$\hat{W}_k(,) := Y_k(,) - \hat{X}_k(,)$

$O_k(,) := Patt(\hat{W}_k(,))$

end

CHAPTER IV

RESULTS

The sequences[1] that were chosen for the simulations consist of grayscale video sequences in the raw format. The number of frames in each video sequence was restricted to 40 due to memory constraints in MATLAB. The resolution of the sequences varied for different video sequences. Most of the sequences that were chosen were slow moving video sequences due to the limitations of the proposed algorithm in detecting hidden data in fast moving sequences. These limitations were discussed in the previous chapter. Appendix B.1 contains the description of each sequence that was used for the simulation. We label the sequences from 1 to 27 as shown in Table 1 and will refer to the sequences using these labels for the rest of the thesis.

As discussed in Sections E and F of Chapter I, the messages are embedded in the spatial domain of each video frame to test the performance of our technique. However, the reader should note that our approach to steganalysis will still work if the embedding is done in another linear transform domain such as the discrete cosine transform (DCT). The embedding was done by adding watermarks W_k from a zero-mean Gaussian distribution as presented in Equation 1.2 into every pixel of each frame. The watermark strength parameter α is varied to test the affects on secrecy. The values used in our simulations are $\alpha = 1, 3, 5$. The smaller the value of α the less perceptible the mark both visually and through steganalysis, but the lower the capacity or robustness of the covert data embedding.

As mentioned in Section B of Chapter III we use a sliding window to perform the collusion attack. Different window lengths were employed for a simple linear collusion attack on test video sequences containing watermarks X_k to produce $\hat{X}_k$. The difference

[1]The sequences were downloaded from http://ise.stanford.edu/video.html and http://www.articom.info/1489.html

$Y_k - \hat{X}_k$ was then obtained to provide an estimate of αW_k. To determine the success of the window length for steganalysis, the pairwise correlation coefficient $\rho(\alpha W_k, \hat{W}_k)$ was computed, where

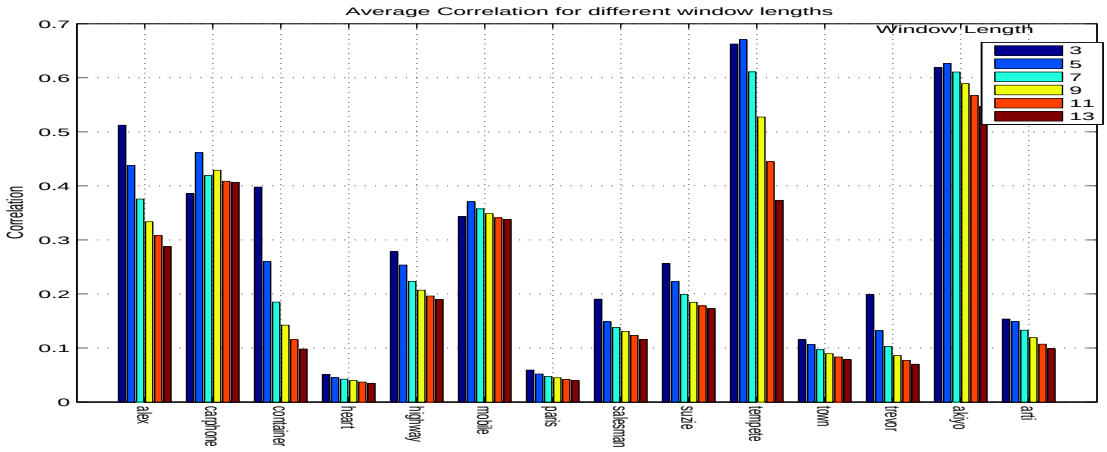
$$\rho(A, B) = \frac{\text{cov}(A, B)}{\sqrt{\text{var}(A) \cdot \text{var}(B)}}, \tag{4.1}$$

$\text{cov}(\cdot, \cdot)$ denotes the covariance and $\text{var}(\cdot)$ denotes the variance of the argument random variable(s).

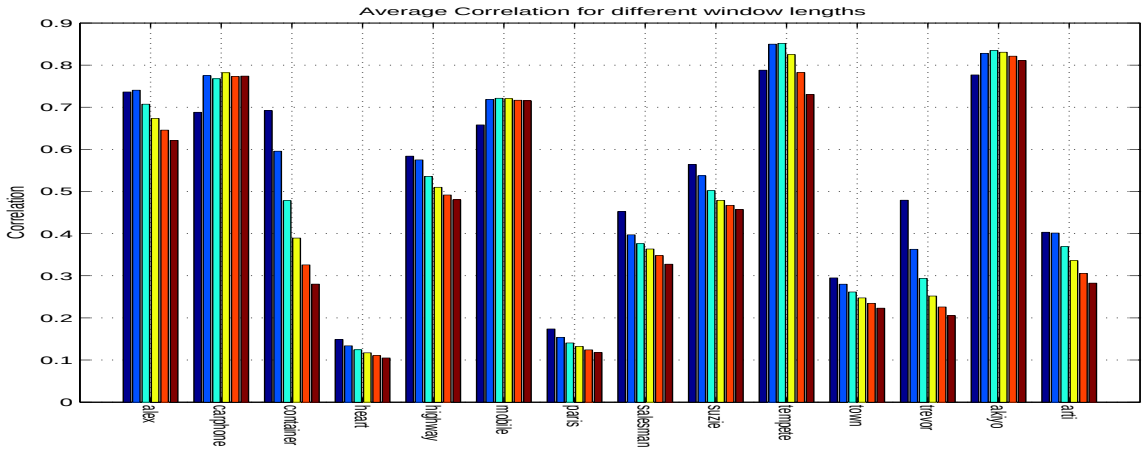
Figures 11(a),11(b) and 11(c) show the average correlation between the embedded watermarks and the estimated watermarks over 40 frames using simple linear collusion for different values of embedding strength and window lengths for various sequences. We see that in Figure 11(a) the average correlation is highest for majority of the sequences for a window length of 3. The optimum collusion length increases for higher embedding strengths which can be seen from Figures 11(b) and 11(c). This is in accordance with our earlier assertion that for with a fixed value of correlation coefficient and average variance of the host frames an increase in SNR will lead to a increase in the value of the optimum collusion length. e.g. From Figures 11(a),11(b) and 11(c) we see that the optimum collusion length of sequence "alex" is 3,5,13 for embedding strengths of 1,3,5 respectively.

We assume that the embedder uses a low embedding strength for watermark insertion since a higher embedding strength would leave significant statistical imprints. With this assumption we chose the optimum collusion length to be $2L + 1 = 5$ for the proposed steganalysis method.

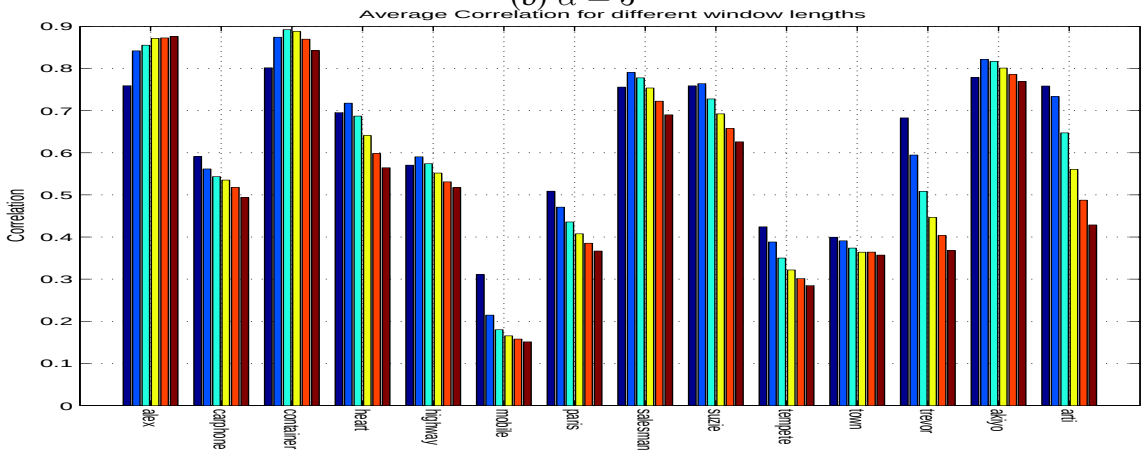
Other issues that require optimization are the training and parameter selection of the kNN classifier. The number of video sequences required for training for effective classification is application-dependent. In our work, we employed cross validation to minimize the probability of false negative with different numbers of training video sequence sets. It was found that two video sequences are effective for training. The parameter k in the kNN



(a) $\alpha = 1$



(b) $\alpha = 3$



(c) $\alpha = 5$

Fig. 11. Average correlation between W_k and $\hat{W}_k$ for different sequences.

classifier [39, 40] that determines the number of "nearest neighbors" searched to reach a classification decision also needs to be set. Increasing k increases computational complexity, so the optimal value must provide good performance without cost. Our tests showed that $k = 1$ gave a low probability of false negative and false positive and higher values of k did not improve performance.

In Section E of Chapter III we described an adaptive scheme of finding a set of weights for weighted collusion scheme to detect the presence of a watermark. Using simulations we depict that the weighted collusion scheme is superior than the simple linear collusion scheme in estimating the watermark in each frame. The success of the method is noted by measuring the correlation between the estimated watermark $\hat{W}_k$ and the embedded watermark αW_k in each frame. Table VI in Appendix B shows the comparison between the averaging based scheme and the weighted scheme. We can clearly see an improvement of about 0.05 in the correlation values for the weighted scheme over the averaging scheme for an embedding strength of 1. The improvement is not significant for higher embedding strengths. For the higher embedding strengths we might use the simple linear collusion scheme instead of the weighted scheme to get rid of the inherent complexity involved in computing the weights in the weighted collusion scheme.

The probabilities of false negative P_{FN} and false positive P_{FP} were computed for a given test video sequence by counting the number of misdetections over each of the 40 frames in the sequence; thus if one video frame out of the 40 results in a false detection the error probability is 2.5%. We estimated P_{FN} by embedding a Gaussian watermark into a given video sequence and then applying a collusion attack to estimate the watermark present. The result was then passed to the pattern classification algorithm to determine the detection result. The fraction of failed detections was counted to estimate P_{FN}. Similarly, the same approach was applied to unmarked video sequences to estimate P_{FP}.

Our aim is to detect the presence of covert data in a video sequence on the whole rather than estimating the presence of watermarks in individual frames. So if P_{FN} and P_{PN} is less than 50 we still have a successful steganalysis attack. Let us assume that the P_{FN} for a watermarked sequence is 30. It means that 30 percent of the total number of frames from a watermarked sequence were classified as non-watermarked and the rest as watermarked. Adopting a majority takes all scheme suggests that the video sequence is watermarked since the number of frames that were classified as watermarked were more than the other. The steganalysis method was tested on 3 different variations of the spread spectrum steganography. The first method does embedding in the spatial domain as described in Section E of Chapter I. The other two methods require adding Gaussian watermarks in the DCT domain which are explained in Appendix C.2.

Tables VIII, IX and X shows the probability of false negative P_{FN} and the probability of false positive P_{FP} for the proposed steganalysis method for embedding in spatial domain. The tables show the error encountered in detection of watermarked and non-watermarked sequences for different values of alpha using different steganalysis methods. The comparison between the spatial based steganalysis method based on weiner filtering to estimate the hidden watermark and the temporal methods such as simple linear collusion scheme, weighted collusion scheme and the block based scheme has been provided. As we can see from Table VIII, for an embedding strength of $\alpha = 1$ the P_{FN} is reasonably low for most test video sequences. We also note that P_{FP} is higher than P_{FN} and the method classifies most of the frames of an unwatermarked sequence as watermarked. This is not of great concern because the overall goal of steganalysis in most applications is to avoid a false negative detection. Any sequences that is (rightly or wrongly) flagged as potentially containing hidden information can go under more thorough processing for better detection results.

Table IX and X also shows how the performance of the steganalysis technique improves as the magnitude of the embedding strength α increases. It follows that a steganalysis technique that works well for a lower value of α will work at least as well for higher values. Thus, our analysis of small values of α provides a minimum performance limit on the algorithm.

The performance of the temporal based simple linear collusion detection is almost the same to that of the spatial based weiner filtering method to detect the messages. This suggests no added advantage of using the temporal based schemes over the spatial based detection schemes that work on individual frames. However we note the improvement in the error detection probabilities in case of the block-based temporal detection method over the spatial and simple linear collusion detection schemes.

The steganalysis technique does not quite work well for Sequence No:15 which is a sequence having a lot of motion. We see that inspite of using block based techniques the P_{FP} is significantly high and the sequence is always classified as watermarked. This suggests that the proposed steganalysis method fails if the correlation between the frames is very low. We can see from Figure 6 that a very high value of SNR is required for video sequences where the correlation between frames is pretty low in order to have a successful collusion attack. This is further verified by the simulation results.

The proposed steganalysis method was also tested on video sequences embedded with watermarks in the DCT domain. The results for false positives and false negatives are shown in Appendix C.2 for the two different DCT based embedding schemes.

The P_{FN} and P_{FP} for DCT based embedding scheme using Method A shows a similar trend to that of the spatial based embedding scheme. This is attributed to the fact that the addition of Gaussian watermarks in the DCT domain can be modeled as addition of Gaussian watermarks in the spatial domain using Equation 1.2. The DCT transform is linear and hence any Gaussian watermark added in the DCT domain remains Gaussian in

the spatial domain. We would like to point out that the proposed steganalysis method has to undergo no change apart from the training set in order to foil the DCT based spread spectrum steganography for video sequences.

CHAPTER V

CONCLUSION

A. Discussion

The work presented in this thesis demonstrates the potential of our framework and the use of temporal processing for effective steganalysis. In this chapter we discuss the salient features of our algorithm and to what extent we were able to achieve our objectives.

To the best of our knowledge we developed the first video steganalysis algorithm that takes advantage of the temporal redundancy present in the video. We see the improvement in the performance of our method over the spatial methods that work on frame-by-frame basis. This clearly shows that the essence of future steganalysis methods for video lies in the utilization of the temporal information.

From an embedders point of view the presence of temporal redundancy makes video sequences an attractive choice for cover objects. But the statistical redundancy in the video aids the steganalyst too in detecting the hidden Gaussian watermarks. This poses serious challenges to an embedder since the effective data rate is reduced substantially in order to prevent the insertion of statistical imprints. We earlier asserted that the success of the proposed method is a factor of the correlation between the host frames. The chances of detection increase exponentially with the increase in correlation for a fixed SNR. The block-based scheme further tries to maximize the correlation by finding the best match in the previous frames corresponding to the reference frame. We conclude that slow moving video sequences is not an ideal choice for steganography. The notion is also supported by Chandramouli in [6]. The embedder should therefore hide messages in moving parts of a video or choose a video that has a lot of motion as a cover object.

One of our objectives was to study the trade off between robust embedding of messages and detection capability of our steganalysis method. We see that the detection rate increases with an increase in the embedding strength of the watermark suggesting robustness increases the chances of detection. The theoretical bounds plotted earlier suggest using a range of 1 to 3 for α to foil the collusion attack. We note from simulations that for an embedding strength of 1 the probability of false positive and false negative are relatively high as compared to higher embedding strengths. The P_{FN} and P_{FP} are low enough for a successful detection of a covert data in a video sequence for an embedding strength of 3. Thus based on simulations and the theoretical analysis one should embed the Gaussian watermarks with an embedding strength of 1-3.

We would like to reiterate the fact that the field of watermarking and steganography complement each other. The method of estimating the watermark in each frame using collusion attack as proposed in our steganalysis method can be applied to the field of watermarking. It can be used for watermark detection in watermarking applications such as content authorization, fingerprinting etc. Collusion attack used in our steganalysis method helps in getting a mark free copy from watermarked sequences. Thus we have shown that spread spectrum may not be the most robust data hiding scheme for video as proposed in the literature. The same conception is supported by authors in [6, 31] who propose different methods to break spread spectrum steganography.

The complexity of the simple linear collusion scheme is very low and can be applied for real time applications. For every frame that is under a steganalysis test we need to wait for 2-4 future frames to arrive before one can perform the collusion attack. At a display rate of 30 frames per sec this corresponds to time lag of 1/10 of a second. The processing time is bare minimum and thus the total time it takes to predict the presence or absence of a message after the arrival of a frame is small enough to be applied for real time applications. The weighted linear collusion scheme or the block based schemes have

a larger time complexity and hence is not feasible to be used for real time applications. The increase in performance of these methods over the simple linear collusion schemes makes it an ideal choice for all situations other than real-time applications.

B. Limitations and Future Directions

In this section we discuss limitations of our algorithm and highlight the areas of future research.

Apart from the assumption that the watermark is additive white and Gaussian, our scheme also presumes that the sender embeds the watermark in each pixel of every frame. To maximize covert communication capacity, this may be reasonable. However, future investigation must consider how the affects of interleaving the watermark in select pixels and frames affects the detection accuracy of steganalysis. Such interleaving will provide the sender with greater secrecy at the expense of capacity or robustness. We expect that there is a threshold for interleaving below which steganalysis detection will become inaccurate. Thus, this value determines the effective covert communication capacity that cannot be detected.

In order to develop a strategy that works for all embedding schemes (not just the spread-spectrum based Gaussian watermarks discussed in this paper), we need to target the statistics of the video sequence [1, 17, 19, 15] rather than solely consider the statistics of a possibly hidden message. The proposed steganalysis schemes uses a model of the distribution of the embedded message as reference information. A steganalysis technique that also accounts for the statistics of a natural video sequence may be more general. However this method may have some disadvantages and may not target all classes of video sequences.

This method will fail for those situations where the characteristic of the sequence is different from natural sequences. The proposed method in this thesis on the other hand will be more robust to the outliers.

Another possible change that can be made to the block based collusion scheme is to detect the presence of a watermark at block level rather than a frame level. A collective decision such as majority wins can be made on each frame using the individual detection results on the blocks. The detection results for each frame can be used to detect the presence and absence of a message in the entire video sequence. It is shown in [41] that a distributed framework can help lower the probability of false negative and false positive.

REFERENCES

[1] I. Avcibas, B. Sankur, and N.D. Memon, "Steganalysis based on image quality metrics - differentiating between techniques," in *Proc. IEEE Workshop on Multimedia*, Cannes, France, October 2001.

[2] A. Kerckhoffs, "La cyprotgraphie militaire," *Journal des Sciences Militaires*, vol. 9, pp. 5–38, January 1883.

[3] J. Watkins, "Steganography–messages hidden in bits," Multimedia systems coursework, Department of Electronics and Computer Science,University of Southampton, UK, December 2001.

[4] G.J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology: Proceedings of Crypto'83*, pp. 51–67, Plenum Press, London, 1984.

[5] D. Kahn, *The Code-Breakers: The Story of Secret Writing*, New York: Macmillan and Co., 1st edition, 1967.

[6] R. Chandramouli, "A mathematical framework for active steganalysis," *ACM Multimedia Systems Journal, Special Issue on Multimedia Watermarking*, 2003.

[7] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Proceedings*, vol. 6, no. 12, pp. 1673–1687, December 1997.

[8] L.M. Marvel, C.G. Boncelet Jr., and C.T. Retter, "Spread spectrum image steganography," *IEEE Transactions on Image Proceedings*, vol. 8, pp. 1075–1083, August 1999.

[9] G. Mohay, A. Anderson, B. Collie, O. de Vel, and R. McKemmish, *Computer and Intrusion Forensics*, Boston: Artech House, 2003.

[10] J. Kelly, "Terror groups hide behind web encryption," May 2001, http://www.usatoday.com/tech/news/2001-02-05-binladen.htm.

[11] D. Lewis, "Terrorists and steganography," Septmeber 2001, http://www.linuxsecurity.com/content/view/110558/151/.

[12] T. Reed, *At the Abyss: An Insiders History of the Cold War*, New York: Random House, 2004.

[13] G.C. Kessler, "Steganography: Implications for the prosecutor and computer forensics examiner," April 2004, http://www.garykessler.net/library/ndaa_stego.html.

[14] B.H. Astrowsky, "Steganography: Hidden images, a new challenge in the fight against child porn," *UPDATE*, vol. 13, no. 2, 2000.

[15] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, 2003.

[16] B. Krebs, "Danger of image-borne viruses looms," September 2004, http://www.washingtonpost.com/wp-dyn/articles/A45126-2004Sep23.html.

[17] H. Farid, "Detecting hidden messages using higher-order statistical models," in *Proc. IEEE International Conference on Image Processing*, Rochester, New York, September 2002.

[18] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *Proc. 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[19] I. Avcibas, B. Sankur, and K. Sayood, "Image steganalysis with binary similarity measures," in *IEEE International Conference on Image Processing*, Rochester, New York, June 2002, vol. 3, pp. 645–648.

[20] J.J. Harmsen and W.A. Pearlman, "Steganalysis of additive noise modelable information hiding," in *Proc. SPIE Security and Watermarking of Multimedia Contents V*, Santa Clara, California, January 2003, vol. 5022.

[21] U. Budhia and D. Kundur, "Video steganalysis using collusion sensitivity," in *Proc. SPIE: Sensors, Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense*, Edward M. Carapezza, Ed., Orlando, Florida, April 2004, vol. 5403.

[22] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 43–51, February 2005.

[23] J. Fridrich, R. Du, and L. Meng, "Steganalysis of LSB encoding in color images," in *Proc. IEEE Conference on Multimedia and Expo*, New York, July-August 2000.

[24] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compability," in *Proc. SPIE Multimedia Systems and Applications IV*, Denver, Colorado, August 2001.

[25] A. Westfeld and A. Phitzmann, "Attacks on steganographic systems," in *Proc. 3rd Information Hiding Workshop*, Dresden, Germany, September-October 1999.

[26] N. Provos, "Defending against statistical steganalysis," in *Proc. of the 10th USENIX Security Symposium*, Dresden, Germany, August 2001.

[27] S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machines," in *SPIE Symposium on Electronic Imaging*, San Jose, California, 2004.

[28] I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality measures," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 206–223, April 2002.

[29] S. Liu, H. Yao, and W. Gao, "Steganalysis of data hiding techniques in wavelet domain," in *International Conference on Information Technology: Coding and Computing*, Las Vegas, Nevada, April 2004, vol. 1, pp. 751–754.

[30] S. Liu, H. Yao, and W. Gao, "Steganalysis based on wavelet texture analysis and neural network," in *9th China Conference on Machine Learning*, Shanghai, China, October 2004.

[31] S. Trivedi and R. Chandramouli, "Secret key estimation in sequential steganography," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 53, no. 2, pp. 746–757, February 2005.

[32] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis of digital images: estimating the secret message length," *ACM Multimedia Systems Journal, Special Issue on Multimedia Watermarking*, vol. 9, no. 3, pp. 288–302, 2003.

[33] J. Kilian, F.T. Leighton, L.R. Matheson, T.G. Shamoon, R.E. Tarjan, and F. Zane, "Resistance of digital watermarks to collusive attacks," Technical Report TR-585-98, Computer Science Department, Princeton University, Princeton, New Jersey, July 1998.

[34] H. Zhao, M. Wu, J. Wang, and K.J.R. Liu, "Nonlinear collusion attacks on independent multimedia fingerprints," in *IEEE International Conference on Multimedia and ExpoSignal Processing*, Baltimore, Maryland, July 2003.

[35] H. Zhao, M. Wu, J. Wang, and K.J.R. Liu, "Nonlinear collusion attacks on independent multimedia fingerprints," *IEEE Trasactions on Image Processing*, to appear.

[36] G. Doërr and J.L. Dugelay, "New intra-video collusion attack using mosaicing," in *IEEE International Conference on Multimedia and Expo*, Baltimore, Maryland, July 2003.

[37] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, New York: McGraw Hill, 4th edition, 2002.

[38] R.J. Anderson and F.A.P. Petitcolas, "On the limits of steganography," *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 2, pp. 474–481, May 1998, (Special Issue on Copyright & Privacy Protection).

[39] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification and Scene Analysis*, New York: Wiley, 2nd edition, 2001.

[40] R. Gutierrez-Osuna, "Cpsc 689-604:special topics in pattern analysis," Lecture Notes, September 2003, http://research.cs.tamu.edu/prism/lectures.htm.

[41] R. Chandramouli and N.D. Memon, "A distributed detection framework for steganalysis," in *ACM Workshop on Multimedia Security*, Los Angeles, California, November 2000, pp. 123–126.

APPENDIX A

PROOFS

1. Proof of Proposition 1

The expeted MSE between the estimated watermark and the orginal watermark as defined in equation 3.14 is given by

$$
\begin{aligned}
\mathbf{E}[(\hat{W}_k - \alpha W_k)^2] &= \mathbf{E}[(U_k - \hat{X}_k)^2] \\
&= \mathbf{E}[(U_k - \frac{1}{2L+1}\sum_{i=k-L}^{k+L} X_i)^2] \\
&= \mathbf{E}[(U_k - \frac{1}{2L+1}\sum_{i=k-L}^{k+L} (U_i + \alpha \cdot W_i))^2] \\
&= \mathbf{E}[U_k^2 + \frac{1}{(2L+1)^2}(\sum_{i=k-L}^{k+L} (U_i + \alpha \cdot W_i))^2 \\
&\qquad -\frac{2}{2L+1}U_k(\sum_{i=k-L}^{k+L} (U_i + \alpha \cdot W_i))] \\
&= \mathbf{E}[U_k^2 + \frac{1}{(2L+1)^2}((\sum_{i=k-L}^{k+L} U_i)^2 + \alpha^2(\sum_{i=k-L}^{k+L} W_i)^2 \\
&\qquad +2\alpha(\sum_{i=k-L}^{k+L} U_i \cdot \sum_{i=k-L}^{k+L} W_i)) - \frac{2}{2L+1}(\sum_{i=k-L}^{k+L} (U_k \cdot U_i + \alpha \cdot U_k \cdot W_i))] \\
&= \mathbf{E}U_k^2 + \frac{1}{(2L+1)^2}(\mathbf{E}[(\sum_{i=k-L}^{k+L} U_i)^2] + \mathbf{E}[\alpha^2(\sum_{i=k-L}^{k+L} W_i)^2] \\
&\qquad +2\alpha\mathbf{E}[\sum_{i=k-L}^{k+L} U_i \cdot \sum_{i=k-L}^{k+L} W_i] - \frac{2}{2L+1}(\sum_{i=k-L}^{k+L} \mathbf{E}[U_k \cdot U_i + \alpha \cdot U_k \cdot W_i])] \\
&= \sigma_u^2 + \frac{\alpha^2\sigma_w^2}{2L+1} + \frac{1}{(2L+1)^2}\mathbf{E}[(\sum_{i=k-L}^{k+L} U_i)^2] - \frac{2}{2L+1}\sum_{i=k-L}^{k+L} \mathbf{E}[U_i \cdot U_k] \\
&\qquad \text{(Using Assumption 2, i.e. } \mathbf{E}W_k = 0, \mathbf{E}W_i.W_j = 0 \text{ for } i \neq j \text{ and}
\end{aligned}
$$

$\mathbf{E}W_i.U_j = 0$ for all i,j)

$$= \sigma_u^2 + \frac{\alpha^2\sigma_w^2}{2L+1} + A - B \tag{A.1}$$

where,

$$A = \frac{1}{(2L+1)^2}\mathbf{E}[(\sum_{i=k-L}^{k+L} U_i)^2]$$

$$B = \frac{2}{2L+1}\sum_{i=k-L}^{k+L}\mathbf{E}[U_i \cdot U_k]$$

$Now,$

$$\begin{aligned} B &= \frac{2}{2L+1}\sum_{i=k-L}^{k+L}\mathbf{E}[U_i \cdot U_k] \\ &= \frac{2}{2L+1}\mathbf{E}[U_k \cdot U_{k-L} + U_k \cdot U_{k-L+1} + \ldots + U_k \cdot U_{k-1} \\ &\quad + U_k \cdot U_k + U_k \cdot U_{k+1} + \ldots + U_k \cdot U_{k+L-1} + U_k \cdot U_{k+L}] \\ &= \frac{2\sigma_u^2}{2L+1}(\rho^L + \rho^{L-1} + \ldots + \rho + 1 + \rho + \ldots + \rho^{L-1} + \rho^L) \end{aligned}$$

(Using Markov Model defined in Assumption 2)

$$\begin{aligned} &= \frac{2\sigma_u^2}{2L+1}(1 + 2(\rho + \rho^2 \ldots + \rho^{L-1} + \rho^L)) \\ &= \frac{2\sigma_u^2}{2L+1}(1 + \frac{2\rho(1-\rho^L)}{1-\rho}) \end{aligned}$$

(Assuming $|\rho| < 1$, it dosen't work if $\rho = 1$)

$Now,$

$$\begin{aligned} A &= \frac{1}{(2L+1)^2}\mathbf{E}[(\sum_{i=k-L}^{k+L} U_i)^2] \\ &= \frac{1}{(2L+1)^2}\mathbf{E}[\sum_{i=k-L}^{k+L} U_i \sum_{j=k-L}^{k+L} U_j)] \\ &= \frac{1}{(2L+1)^2}\mathbf{E}[(U_{k-L}\sum_{j=k-L}^{k+L} U_j) + (U_{k-L+1}\sum_{j=k-L}^{k+L} U_j) + \ldots \\ &\quad + (U_{k+L-1}\sum_{j=k-L}^{k+L} U_j) + (U_{k+L}\sum_{j=k-L}^{k+L} U_j)] \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{(2L+1)^2}\mathbf{E}[\sum_{j=k-L}^{k+L} U_{k-L}\cdot U_j + \sum_{j=k-L}^{k+L} U_{k-L+1}\cdot U_j + \ldots \\
&\qquad + \sum_{j=k-L}^{k+L} U_{k+L-1}\cdot U_j + \sum_{j=k-L}^{k+L} U_{k+L}\cdot U_j] \\
The\quad 1^{st}\quad term\quad is\quad &= \frac{\sigma_u^2}{(2L+1)^2}(1+\rho+\rho^2+\ldots+\rho^{2L-1}+\rho^{2L}) \\
2^{nd}\quad term\quad is\quad &= \frac{\sigma_u^2}{(2L+1)^2}(\rho+1+\rho+\ldots+\rho^{2L-2}+\rho^{2L-1}) \\
&\vdots \qquad\qquad\qquad \vdots \\
&\vdots \qquad\qquad\qquad \vdots \\
2L+1^{th}\quad term\quad is\quad &= \frac{\sigma_u^2}{(2L+1)^2}(\rho^{2L}+\rho^{2L-1}+\rho^{2L-2}+\ldots+\rho+1)
\end{aligned}
$$

The terms can be put together as rows of a Toeplitz matrix and the sum of all the terms is given by the sum of all the elements in the matrix.

$$
\begin{aligned}
A &= \frac{\sigma_u^2}{(2L+1)^2}\sum_{r=1}^{2L+1}\sum_{s=1}^{2L+1} T_{r,s} \\
&= \frac{\sigma_u^2}{(2L+1)^2}\sum_{r=-2L}^{2L}\sum_{s=1}^{2L+1-|r|} T_{1,|r|+1} \qquad \text{(A.2)}
\end{aligned}
$$

where

$$
T = \begin{bmatrix}
1 & \rho & \rho^2 & \ldots & \rho^{2L} \\
\rho & 1 & \rho & \ldots & \rho^{2L-1} \\
\rho^2 & \rho & 1 & \ldots & \rho^{2L-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\rho^{2L} & \rho^{2L-1} & \ldots & \rho & 1
\end{bmatrix}
$$

Evaluating Equation A.2 and assuming $z = 2L + 1$ we have,

$$\begin{aligned}
A &= \frac{\sigma_u^2}{z^2}[z + 2(z-1)\rho + 2(z-2)\rho^2 + \ldots + 2(z-(z-1))\rho^{z-1}] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z(\rho + \rho^2 + \rho^3 + \ldots + \rho^{z-1}) - 2(\rho + 2\rho^2 + 3\rho^3 + \ldots + (z-1)\rho^{z-1})] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho\sum_{j=1}^{z-1} j\rho^{j-1}] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho\sum_{j=1}^{z-1} \frac{d}{d\rho}\rho^j] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho\frac{d}{d\rho}\sum_{j=1}^{z-1} \rho^j] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho\frac{d}{d\rho}(\rho\frac{(1-\rho^{z-1})}{1-\rho})] \\
&= \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho(\frac{(1-\rho^z - z\rho^{z-1}(1-\rho))}{(1-\rho)^2})]
\end{aligned}$$

Substituting the values of A, B and $z = 2L + 1$ in equation A.1 we have

$$\begin{aligned}
\mathbf{E}[(\hat{W}_k - W_k)^2] &= \sigma_u^2 + \frac{\alpha^2\sigma_w^2}{z} + \frac{\sigma_u^2}{z^2}[z + 2z\rho\frac{(1-\rho^{z-1})}{1-\rho} - 2\rho(\frac{(1-\rho^z - z\rho^{z-1}(1-\rho))}{(1-\rho)^2})] \\
&\quad -\frac{2\sigma_u^2}{z}(1 + \frac{2\rho(1-\rho^{\frac{z-1}{2}})}{1-\rho})
\end{aligned}$$

This simplifies to

$$\mathbf{E}[(\hat{W}_k - W_k)^2] = \sigma_u^2[\frac{z-1}{z} - \frac{2\rho}{z(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{z(1-p)} - \frac{2\rho(1-\rho^z)}{z^2(1-\rho)^2}] + \frac{\alpha^2\sigma_w^2}{z}$$

The mean μ of the host frames in our proof has been ignored and is assumed to be zero. However the final term will be independent of the mean even if we take it into account.

2. Proof of Proposition 2

The expected MSE between the watermarks in case there is no collusion attack used is given by substituting $L = 0$ or $z = 1$ in equation 3.16.

$$\begin{aligned} \mathbf{E}[(\hat{W}_k - \alpha W_k)^2] &= \sigma_u^2[-\frac{2\rho}{(1-\rho)} + \frac{4\rho}{(1-\rho)} - \frac{2\rho}{(1-\rho)}] + \alpha^2\sigma_w^2 \\ &= \alpha^2\sigma_w^2 \end{aligned}$$

3. Proof of Proposition 3

From Proposition 1 we know the expected MSE between the watermarks when we apply collusion attack. To determine the conditions for which the collusion attack is successful in estimating the watermark, we consider the case in which the estimated MSE obtained from collusion attack is smaller than the estimated MSE obtained without the collusion attack.

$$\begin{aligned} & \sigma_u^2[\frac{z-1}{z} - \frac{2\rho}{z(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{z(1-\rho)} - \frac{2\rho(1-\rho^z)}{z^2(1-\rho)^2}] + \frac{\alpha^2\sigma_w^2}{z} < \alpha^2\sigma_w^2 \\ \Rightarrow \quad & \sigma_u^2[\frac{z-1}{z} - \frac{2\rho}{z(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{z(1-\rho)} - \frac{2\rho(1-\rho^z)}{z^2(1-\rho)^2}] < \frac{(z-1)}{z}\alpha^2\sigma_w^2 \\ \Rightarrow \quad & \sigma_u^2[1 - \frac{2\rho}{(z-1)(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{(z-1)(1-\rho)} - \frac{2\rho(1-\rho^z)}{z(z-1)(1-\rho)^2}] < \alpha^2\sigma_w^2 \\ \Rightarrow \quad & \frac{\alpha^2\sigma_w^2}{\sigma_u^2} > 1 - \frac{2\rho}{(z-1)(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{(z-1)(1-\rho)} - \frac{2\rho(1-\rho^z)}{z(z-1)(1-\rho)^2} \\ \Rightarrow \quad & \frac{\sigma_u^2}{\alpha^2\sigma_w^2} < \frac{1}{1 - \frac{2\rho}{(z-1)(1-\rho)} + \frac{4\rho^{\frac{z+1}{2}}}{(z-1)(1-\rho)} - \frac{2\rho(1-\rho^z)}{z(z-1)(1-\rho)^2}} \end{aligned}$$

4. Proof of Proposition 5

The entropy of the estimate of the watermark from a watermarked frame is given by

$$\begin{aligned} E_1 &= H(n_k + w_k^{'}) \quad \text{(where } H \text{ is the entropy operator)} \\ E_1 &= H(n_k) + H(W_k^{'}) \quad \text{(Since } W_k^{'} \text{ is independent of } n_k) \end{aligned}$$

$$
\begin{aligned}
E_1 &= E_0 + H(W_k^{'}) \\
Now, & \\
W_k^{'} &= \alpha(W_k - C(W_k)) \quad \text{(From Equation 3.13)} \\
W_k^{'} &= \alpha(W_k - \frac{1}{2L+1}\sum_{i=k-L}^{k+L} W_k)
\end{aligned}
$$

Now since $W_k^{'}$ is a linear combination of independent gaussian variables, $W_k^{'}$ also belongs to a gaussian distribution.

The mean and the variance of this gaussian distribution is given by:

$$
\begin{aligned}
\mu &= \mathbf{E}W_k^{'} \\
&= \mathbf{E}[\alpha(W_k - \frac{1}{2L+1}\sum_{i=k-L}^{k+L} W_k)] \\
&= 0 \quad \text{(Since } W_k = N(0, \sigma_w^2)) \\
\sigma &= \mathbf{E}W_k^{'2} - \mu^2 \\
&= \alpha^2\mathbf{E}[(W_k - \frac{1}{2L+1}\sum_{i=k-L}^{k+L} W_k)^2] \\
&= \alpha^2(\sigma_w^2 + \frac{\sigma_w^2}{2L+1} - \frac{2\sigma_w^2}{2L+1}) \\
&= \frac{2L}{2L+1}\alpha^2\sigma_w^2
\end{aligned}
$$

We know that the entropy a Gaussian distrbuted random variable is given by $\frac{1}{2}log(2e\pi\sigma^2)$. Since $W_k^{'}$ is Gaussian distributed with a variance of $\sigma_w^{'2} = \frac{2L}{2L+1}\alpha^2\sigma_w^2$, the entropy of it is given by $\frac{1}{2}log(2e\pi\sigma_w^{'2})$.

Therefore,

$$E_1 = E_0 + \frac{1}{2}log(2e\pi\sigma_w^{'2}) \tag{A.3}$$

5. Proof of Proposition 4

From equation 3.14 and 3.19 the cost function that needs to minimized can be written as $\mathbf{E}[(U_k - \sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i))^2]$ subject to a linear constraint $\sum_{i=k-L}^{k+L} \beta_i = 1$. We can solve the above linear constraint problem using Lagrange Mulitpliers. The cost function can be written as

$$
\begin{aligned}
f(\beta,\lambda) &= \mathbf{E}[(U_k - \sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i))^2] + 2\lambda(\sum_{i=k-L}^{k+L} \beta_i - 1) \\
\frac{\partial(f(\beta,\lambda))}{\partial\beta_j} &= 0 \quad for \;\; j = k-L, k-L+1, \ldots, k+L \\
\frac{\partial(f(\beta,\lambda))}{\partial\beta_j} &= \mathbf{E}[2(U_k - \sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i))(-1)(U_j + \alpha \cdot W_j)] + 2\lambda \\
0 &= \mathbf{E}[-2(U_k(U_j + \alpha \cdot W_j) - \sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i)(U_j + \alpha \cdot W_j))] + 2\lambda \\
0 &= \mathbf{E}[(U_k U_j - \sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i)(U_j + \alpha \cdot W_j))] - \lambda \\
\mathbf{E}(U_k U_j) &= \mathbf{E}[\sum_{i=k-L}^{k+L} \beta_i(U_i + \alpha \cdot W_i)(U_j + \alpha \cdot W_j)] + \lambda \\
\mathbf{E}(U_k U_j) &= \beta_j \mathbf{E}(\alpha^2 \cdot W_j^2) + \sum_{i=k-L}^{k+L} \beta_i \mathbf{E}(U_j U_i) + \lambda \\
\sigma_u^2 \rho^{|k-j|} + \mu^2 &= \beta_j \alpha^2 \sigma_w^2 + \sum_{i=k-L}^{k+L} \beta_i(\rho^{|i-j|}\sigma_u^2 + \mu^2) + \lambda \\
\sigma_u^2 \rho^{|k-j|} + \mu^2 &= \beta_j \alpha^2 \sigma_w^2 + \sum_{i=k-L}^{k+L} \beta_i \rho^{|i-j|}\sigma_u^2 + \sum_{i=k-L}^{k+L} \beta_i \mu^2 + \lambda \\
\sigma_u^2 \rho^{|k-j|} &= \beta_j \alpha^2 \sigma_w^2 + \sum_{i=k-L}^{k+L} \beta_i \rho^{|i-j|}\sigma_u^2 + \lambda
\end{aligned}
$$

The $2L+1$ linear set of equations alongwith the constraint equation can be represented in a matrix form in the following way.

$$\begin{bmatrix} 1 \\ \sigma_u^2\rho^L \\ \sigma_u^2\rho^{L-1} \\ \vdots \\ \sigma_u^2\rho^L \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ \sigma_u^2+\alpha^2\sigma_w^2 & \sigma_u^2\rho & \dots & \sigma_u^2\rho^{2L} & 1 \\ \sigma_u^2\rho & \sigma_u^2+\alpha^2\sigma_w^2 & \dots & \sigma_u^2\rho^{2L-1} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_u^2\rho^{2L} & \sigma_u^2\rho^{2L-1} & \dots & \sigma_u^2+\alpha^2\sigma_w^2 & 1 \end{bmatrix} \begin{bmatrix} \beta_{k-L} \\ \beta_{k-L+1} \\ \vdots \\ \beta_{k+L} \\ \lambda \end{bmatrix}$$

$$\begin{aligned} AB &= P \\ B &= A^{-1}P \end{aligned}$$

where,

$$P = [1 \quad \sigma_u^2\rho^L \quad \sigma_u^2\rho^{L-1} \dots \sigma_u^2\rho^L]^T$$

$$B = [\beta_{k-L} \quad \beta_{k-L+1} \dots \beta_{k+L} \quad \lambda]^T$$

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ \sigma_u^2+\alpha^2\sigma_w^2 & \sigma_u^2\rho & \dots & \sigma_u^2\rho^{2L} & 1 \\ \sigma_u^2\rho & \sigma_u^2+\alpha^2\sigma_w^2 & \dots & \sigma_u^2\rho^{2L-1} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_u^2\rho^{2L} & \sigma_u^2\rho^{2L-1} & \dots & \sigma_u^2+\alpha^2\sigma_w^2 & 1 \end{bmatrix}$$

APPENDIX B

RESULTS

1. Description of Sequences

The 27 gray scale video sequences in raw format having 40 frames each that are used for simulations is shown in Table II. Most of the sequences are slow moving sequences containg scences having little eye, lip or hand movement of the subject. The sequences 1 to 14 will be used for training our classifier. The remaining sequences from 15 to 27 will be used to test the performance of the proposed steganalysis methods. The training set has sequences representing from minimal to little motion. The test set also contains sequences having minimal to low motion. Sequence ”backyard” in the test set is a sequence that shows some motion due to the movement of the camera.

Table II. Sequence description.

Sequence Number	Sequence Name
1	alex
2	carphone
3	container
4	heart
5	highway
6	mobile
7	paris

Table II. continued.

Sequence Number	Sequence Name
8	salesman
9	suzie
10	tempete
11	town
12	trevor
13	akiyo
14	arti
15	backyard
16	bridge-close
17	bridge-far
18	building
19	claire
20	diskus
21	foreman
22	grandma
23	hand
24	house
25	miss
26	mother
27	silent

Table III. Average correlation between W_k and $\hat{W}_k$ for $\alpha = 1$.

Window Length(2L+1)	3	5	7	9	11	13
Seq No:1	0.5124	0.4379	0.3757	0.3336	0.3082	0.2877
Seq No:2	0.3856	0.4616	0.4193	0.4290	0.4086	0.4067
Seq No:3	0.3979	0.2600	0.1851	0.1422	0.1154	0.0981
Seq No:4	0.0512	0.0449	0.0424	0.0397	0.0371	0.0348
Seq No:5	0.2783	0.2535	0.2232	0.2070	0.1963	0.1897
Seq No:6	0.3430	0.3706	0.3575	0.3485	0.3408	0.3378
Seq No:7	0.0593	0.0520	0.0474	0.0448	0.0418	0.0397
Seq No:8	0.1905	0.1484	0.1380	0.1308	0.1233	0.1160
Seq No:9	0.2565	0.2228	0.1991	0.1847	0.1778	0.1729
Seq No:10	0.6626	0.6708	0.6110	0.5274	0.4447	0.3734
Seq No:11	0.1153	0.1062	0.0969	0.0894	0.0835	0.0786
Seq No:12	0.1993	0.1325	0.1030	0.0862	0.0772	0.0700
Seq No:13	0.6190	0.6267	0.6107	0.5892	0.5676	0.5462
Seq No:14	0.1537	0.1488	0.1330	0.1193	0.1071	0.0985

Table IV. Average correlation between W_k and $\hat{W}_k$ for $\alpha = 3$.

Window Length(2L+1)	3	5	7	9	11	13
Seq No:1	0.7363	0.7406	0.7074	0.6734	0.6457	0.6214
Seq No:2	0.6882	0.7750	0.7675	0.7819	0.7731	0.7739
Seq No:3	0.6922	0.5957	0.4784	0.3896	0.3258	0.2802
Seq No:4	0.1489	0.1337	0.1243	0.1171	0.1106	0.1048
Seq No:5	0.5839	0.5751	0.5360	0.5100	0.4919	0.4805
Seq No:6	0.6582	0.7186	0.7213	0.7204	0.7167	0.7155
Seq No:7	0.1736	0.1539	0.1404	0.1321	0.1241	0.1179
Seq No:8	0.4526	0.3972	0.3762	0.3636	0.3479	0.3275
Seq No:9	0.5643	0.5374	0.5019	0.4789	0.4673	0.4574
Seq No:10	0.7881	0.8499	0.8515	0.8251	0.7824	0.7305
Seq No:11	0.2950	0.2801	0.2616	0.2474	0.2345	0.2232
Seq No:12	0.4794	0.3629	0.2936	0.2518	0.2259	0.2054
Seq No:13	0.7768	0.8280	0.8348	0.8306	0.8213	0.8111
Seq No:14	0.4033	0.4014	0.3691	0.3359	0.3059	0.2827
Seq No:15	0.6168	0.6029	0.5505	0.4980	0.4566	0.4247

Table V. Average correlation between W_k and $\hat{W}_k$ for $\alpha = 5$.

Window Length(2L+1)	3	5	7	9	11	13
Seq No:1	0.7590	0.8419	0.8554	0.8715	0.8726	0.8762
Seq No:2	0.5912	0.5617	0.5433	0.5351	0.5177	0.4943
Seq No:3	0.8017	0.8744	0.8922	0.8879	0.8697	0.8428
Seq No:4	0.6954	0.7178	0.6870	0.6408	0.5983	0.5644
Seq No:5	0.5705	0.5904	0.5740	0.5518	0.5309	0.5174
Seq No:6	0.3112	0.2145	0.1798	0.1655	0.1576	0.1510
Seq No:7	0.5087	0.4709	0.4357	0.4076	0.3852	0.3667
Seq No:8	0.7560	0.7909	0.7778	0.7537	0.7224	0.6901
Seq No:9	0.7590	0.7640	0.7278	0.6925	0.6578	0.6260
Seq No:10	0.4241	0.3882	0.3501	0.3217	0.3013	0.2848
Seq No:11	0.3996	0.3910	0.3738	0.3639	0.3642	0.3572
Seq No:12	0.6828	0.5947	0.5084	0.4466	0.4036	0.3683
Seq No:13	0.7791	0.8216	0.8165	0.8010	0.7860	0.7694
Seq No:14	0.7585	0.7336	0.6472	0.5605	0.4875	0.4288

Table VI. Correlation between αW_k and $\hat{W}_k$ for different values of alpha for sequence "alex" using averaging and weighted collusion attack.

α	1		3		5	
Frame No.	Averaging	Weighted	Averaging	Weighted	Averaging	Weighted
1	0.3720	0.4524	0.7383	0.7672	0.8435	0.8504
2	0.3679	0.4446	0.7353	0.7647	0.8419	0.8487
3	0.4391	0.4874	0.7872	0.7994	0.8679	0.8699
4	0.4364	0.4926	0.7859	0.8000	0.8673	0.8702
5	0.4764	0.5515	0.8095	0.8254	0.8787	0.8818
6	0.4970	0.5499	0.8186	0.8291	0.8828	0.8845
7	0.5167	0.6140	0.8289	0.8444	0.8868	0.8895
8	0.4853	0.5484	0.8146	0.8274	0.8808	0.8829
9	0.4912	0.5827	0.8158	0.8329	0.8809	0.8838
10	0.4636	0.5342	0.8013	0.8182	0.8747	0.8783
11	0.4167	0.4825	0.7722	0.7930	0.8605	0.8646
12	0.4337	0.4989	0.7840	0.8023	0.8664	0.8701
13	0.4463	0.5077	0.7919	0.8071	0.8704	0.8724
14	0.3774	0.4306	0.7425	0.7653	0.8457	0.8506
15	0.4350	0.5104	0.7845	0.8076	0.8673	0.8721
16	0.4325	0.4707	0.7823	0.7919	0.8655	0.8665
17	0.4711	0.5424	0.8058	0.8205	0.8764	0.8789
18	0.4216	0.4806	0.7749	0.7926	0.8616	0.8640
19	0.4163	0.5055	0.7714	0.7996	0.8600	0.8667
20	0.4335	0.4722	0.7859	0.7969	0.8674	0.8700

Table VI. continued.

α	1		3		5	
Frame No.	Averaging	Weighted	Averaging	Weighted	Averaging	Weighted
21	0.3777	0.4507	0.7458	0.7695	0.8470	0.8516
22	0.4087	0.4646	0.7657	0.7841	0.8571	0.8609
23	0.4301	0.4722	0.7818	0.7946	0.8648	0.8673
24	0.4021	0.4314	0.7601	0.7691	0.8542	0.8551
25	0.4041	0.4492	0.7633	0.7778	0.8559	0.8583
26	0.4221	0.4811	0.7755	0.7926	0.8623	0.8659
27	0.4528	0.5172	0.7968	0.8123	0.8731	0.8760
28	0.4572	0.5110	0.7982	0.8098	0.8728	0.8749
29	0.4791	0.5470	0.8104	0.8227	0.8793	0.8812
30	0.3843	0.4207	0.7459	0.7571	0.8470	0.8488
31	0.4266	0.4679	0.7770	0.7878	0.8630	0.8651
32	0.4061	0.4434	0.7657	0.7788	0.8574	0.8599
33	0.4191	0.4792	0.7736	0.7915	0.8611	0.8647
34	0.4703	0.5323	0.8069	0.8202	0.8774	0.8794
35	0.3803	0.4545	0.7455	0.7764	0.8468	0.8543
36	0.3764	0.4825	0.7402	0.7804	0.8436	0.8538
37	0.3867	0.4476	0.7495	0.7804	0.8485	0.8567
38	0.3492	0.4070	0.7149	0.7408	0.8301	0.8363
39	0.3143	0.3729	0.6808	0.7146	0.8104	0.8194
40	0.1926	0.2030	0.4902	0.5008	0.6660	0.6706

Table VII. Average kurtosis values of $\hat{W}_k$ in case of watermarked and non-watermarked video sequences.

α	1		3		5	
Sequence	Watermarked	Non-Watermarked	Watermarked	Non-Watermarked	Watermarked	Non-Watermarked
1	11.670	53.3703	3.5075	53.3703	2.938	53.3703
2	8.1279	21.3997	3.2317	21.3997	2.9076	21.3997
3	3.1629	102.850	2.8786	102.851	2.8559	102.850
4	5.2722	10.3778	3.3362	10.3778	3.1906	10.3778
5	3.0644	4.1163	2.6771	4.1163	2.6119	4.1163
6	7.0148	7.6084	4.7217	7.6084	3.5211	7.6084
7	21.125	40.0643	5.3969	40.0643	3.4368	40.0643
8	10.561	76.399	3.3026	76.399	3.0214	76.399
9	8.4182	28.2341	3.3834	28.2341	3.1313	28.2341
10	20.625	42.7215	5.0787	42.7215	3.4108	42.7215
11	54.115	60.0454	30.289	60.0454	15.370	60.0454
12	16.475	33.8751	4.2042	33.8751	3.3169	33.8751
13	29.396	188.678	4.6948	188.678	3.2693	188.678
14	3.9711	21.2551	2.724	21.2551	2.7965	21.2551

APPENDIX C

RESULTS

1. Embedding in spatial domain

Table VIII. False negative (P_{FN}) and False positive (P_{FP}) probabilities for steganography in spatial domain using $\alpha = 1$.

$\alpha = 1$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	30	75	37.5	62.5	0	97.5
Seq No:16	10	0	0	0	12.5	0	57.5	12.5
Seq No:17	0	100	2.5	15	0	92.5	10	100
Seq No:18	37.5	100	35	77.5	30	55	7.5	57.5
Seq No:19	100	0	15	0	17.5	15	2.5	0
Seq No:20	75	35	10	90	17.5	97.5	5	77.5
Seq No:21	37.5	40	2.5	42.5	12.5	70	15	55
Seq No:22	0	92.5	20	22.5	10	10	2.5	2.5
Seq No:23	0	100	22.5	87.5	35	72.5	2.5	80
Seq No:24	80	0	47.5	0	25	0	97.5	0
Seq No:25	82.5	0	2.5	12.5	7.5	17.5	0	0
Seq No:26	75	30	22.5	2.5	20	20	5	5
Seq No:27	82.5	100	5	0	15	0	10	0

Table IX. False negative (P_{FN}) and false positive (P_{FP}) probabilities for steganography in spatial domain using $\alpha = 3$.

$\alpha = 3$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	100	0	100
Seq No:16	0	0	0	0	0	0	0	0
Seq No:17	0	0	0	0	0	60	0	100
Seq No:18	0	0	0	100	0	100	2.5	0
Seq No:19	0	0	0	0	0	0	0	0
Seq No:20	0	0	0	0	0	27.5	0	0
Seq No:21	0	0	0	5	0	65	0	7.5
Seq No:22	0	0	0	0	0	0	0	0
Seq No:23	0	100	0	100	0	100	0	100
Seq No:24	0	0	0	0	0	0	65	0
Seq No:25	0	0	0	0	0	0	0	0
Seq No:26	0	0	0	0	0	0	0	0
Seq No:27	0	100	0	0	0	0	0	0

Table X. False negative (P_{FN}) and false positive (P_{FP}) probabilities for steganography in spatial domain using $\alpha = 5$.

$\alpha = 5$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	97.5	0	100
Seq No:16	0	0	0	0	0	0	0	0
Seq No:17	0	0	0	0	0	0	0	0
Seq No:18	0	0	0	100	0	100	0	0
Seq No:19	0	0	0	0	0	0	0	0
Seq No:20	0	0	0	0	0	0	0	0
Seq No:21	0	0	0	0	0	2.5	0	0
Seq No:22	0	0	0	0	0	0	0	0
Seq No:23	0	100	0	95	0	100	0	52.5
Seq No:24	0	0	0	0	0	0	0	0
Seq No:25	0	0	0	0	0	0	0	0
Seq No:26	0	0	0	0	0	0	0	0
Seq No:27	0	0	0	0	0	0	0	0

2. Embedding in DCT Domain

Messages can be hidden in a video sequence by embedding Gaussian watermarks in the frequency domain too. We discuss two different ways of hiding Gaussian watermarks in the DCT domain in the next subsections and present the results of our steganalysis method.

1. Method A

The embedding in the DCT domain is done by first taking a 2D DCT transform of the host frame. A Gaussian watermark is then added to all the DCT coefficients except the DC coefficient. The DC coefficient is left unwatermarked to prevent any significant changes in the visual quality of the watermarked frame from the host frame. An inverse DCT transform of the watermarked image is taken to arrive back to the spatial domain. The whole process of watermarking a frame in the DCT domain can be represented by the following equations.

$$X_k^D(m,n) = U_k^D(m,n) + \alpha \cdot W_k(m,n) \qquad k = 1, 2, 3 \ldots N\,. \tag{C.1}$$

where $U_k^D(m,n)$ represents the 2D DCT transform of the host frame $U_k(m,n)$ and $X_k^D(m,n)$ represents the watermarked frame in the DCT domain. $W_k(m,n)$ like before represents a Gaussian watermark and α represents the embedding strength which is constant for the entire video sequence. An inverse DCT transform of the watermarked frame in DCT domain $X_k^D(m,n)$ is taken to arrive back in the spatial domain. The watermarked signal in the spatial domain is given by $X_k(m,n) = IDCT^{-1}[X_k^D(m,n)]$.

Table XI. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method A) using $\alpha = 1$.

$\alpha = 1$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	100	0	0
Seq No:16	0	100	0	2.5	0	20	0	100
Seq No:17	0	100	0	100	0	100	0	0
Seq No:18	0	100	0	100	0	100	0	5
Seq No:19	92.5	0	0	7.5	0	92.5	0	100
Seq No:20	0	100	0	100	0	100	0	5
Seq No:21	0	100	0	92.5	0	100	0	5
Seq No:22	0	100	0	2.5	0	87.5	0	0
Seq No:23	0	100	0	100	0	97.5	0	0
Seq No:24	100	0	0	0	5	0	85	5
Seq No:25	2.5	0	0	10	0	90	0	100
Seq No:26	0	100	0	22.5	0	95	0	5
Seq No:27	0	100	0	2.5	0	47.5	0	0

Table XII. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method A) using $\alpha = 3$.

$\alpha = 3$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	100	0	100
Seq No:16	0	0	0	0	0	0	0	0
Seq No:17	0	0	0	87.5	0	100	0	100
Seq No:18	0	0	0	100	0	100	0	0
Seq No:19	0	0	0	0	0	67.5	2.5	0
Seq No:20	0	0	0	32.5	0	90	0	0
Seq No:21	0	0	0	5	0	100	0	7.5
Seq No:22	0	0	0	0	0	82.5	0	2.5
Seq No:23	0	100	0	100	0	100	0	100
Seq No:24	0	0	0	0	0	0	72.5	0
Seq No:25	0	0	0	0	0	75	0	0
Seq No:26	0	0	0	0	0	70	0	2.5
Seq No:27	0	100	0	0	0	12.5	0	0

Table XIII. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method A) using $\alpha = 5$.

$\alpha = 5$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	97.5	0	100
Seq No:16	0	0	0	0	0	0	0	0
Seq No:17	0	0	0	0	0	67.5	0	0
Seq No:18	0	0	0	100	0	100	0	0
Seq No:19	0	0	0	0	0	0	0	0
Seq No:20	0	0	0	17.5	0	65	0	0
Seq No:21	0	0	0	0	0	82.5	0	0
Seq No:22	0	0	0	0	0	15	0	0
Seq No:23	0	100	0	95	0	97.5	0	52.5
Seq No:24	0	0	0	0	0	0	0	0
Seq No:25	0	0	0	0	0	5	0	0
Seq No:26	0	0	0	0	0	15	0	0
Seq No:27	0	0	0	0	0	0	0	0

2. Method B

The second method of embedding messages in DCT domain is similar to the first one. This method is popularly known as the spread spectrum watermarking method designed by Cox *et al.* [7]. The only change that is made is the way the Gaussian watermark is added to the DCT coefficients. The embedding of the watermark is done in the following way.

$$X_k^D(m,n) = U_k^D(m,n)(1 + \alpha \cdot W_k(m,n)) \qquad k = 1, 2, 3 \ldots N\,. \tag{C.2}$$

The typical value of α that is used for spread spectrum watermarking is 0.1.

Table XIV. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method B) using $\alpha = 0.1$.

$\alpha = 0.1$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	100	0	0
Seq No:16	0	100	0	0	7.5	0	0	100
Seq No:17	0	100	0	97.5	0	95	0	0
Seq No:18	0	100	0	100	0	100	0	5
Seq No:19	77.5	20	0	0	0	52.5	0	100
Seq No:20	0	100	0	62.5	0	97.5	0	5
Seq No:21	0	100	0	90	0	100	0	5
Seq No:22	0	100	0	0	0	92.5	0	0
Seq No:23	0	100	0	100	2.5	97.5	0	0
Seq No:24	100	0	0	0	17.5	0	0	5
Seq No:25	52.5	75	0	0	0	65	0	100
Seq No:26	0	100	0	15	0	55	0	5
Seq No:27	0	100	0	0	0	10	0	0

Table XV. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method B) using $\alpha = 0.3$.

$\alpha = 0.3$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	100	0	100	0	90
Seq No:16	0	100	0	0	7.5	7.5	0	0
Seq No:17	12.5	0	0	0	0	92.5	0	87.5
Seq No:18	0	32.5	0	100	0	100	0	0
Seq No:19	7.5	0	0	0	0	0	0	0
Seq No:20	0	95	0	30	2.5	12.5	0	0
Seq No:21	0	65	0	0	0	17.5	0	0
Seq No:22	0	100	0	0	0	55	0	0
Seq No:23	0	100	0	67.5	0	100	0	95
Seq No:24	20	0	0	0	0	15	0	0
Seq No:25	0	0	0	0	0	7.5	0	0
Seq No:26	0	20	0	0	0	2.5	0	0
Seq No:27	0	100	0	0	0	0	0	0

Table XVI. False negative (P_{FN}) and false positive (P_{FP}) probabilities for DCT based steganography(Method B) using $\alpha = 0.5$.

$\alpha = 0.5$								
Method	Weiner		Averaging		Weighted		Block based	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq No:15	0	100	0	90	0	95	0	70
Seq No:16	0	0	0	0	7.5	2.5	0	0
Seq No:17	0	50	0	0	0	72.5	0	0
Seq No:18	0	57.5	0	10	0	100	0	0
Seq No:19	0	95	0	0	0	0	0	0
Seq No:20	0	57.5	0	0	0	7.5	0	0
Seq No:21	0	0	0	0	0	32.5	0	0
Seq No:22	0	100	0	0	0	17.5	0	0
Seq No:23	0	100	0	0	5	77.5	0	42.5
Seq No:24	0	100	0	0	10	0	0	0
Seq No:25	0	97.5	0	0	0	0	0	0
Seq No:26	0	75	0	0	0	0	0	0
Seq No:27	0	100	0	0	0	0	0	0

VITA

Udit Budhia

AR Tyres and Cold Retreaders

Shanti Niwas, 1st Floor

Main Road, Ranchi-834001

India.

Education:

B.E. Birla Institute of Technology, Ranchi, India.

Publications:

U. Budhia and D. Kundur,"Digital Video Steganalysis Exploiting Collusion Sensitivity," in *Proc. SPIE: Sensors, Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense*, vol. 5403, Orlando, Florida, April 2004.