Digital video steganalysis exploiting collusion sensitivity

Udit Budhia[a] and Deepa Kundur[a]

[a] Department of Electrical Engineering, Texas A&M University , College Station, U.S.A.

ABSTRACT

In this paper we present an effective steganalyis technique for digital video sequences based on the collusion attack. Steganalysis is the process of detecting with a high probability and low complexity 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 paper, we present a method that overcomes this limitation by using redundant information present in the temporal domain to detect covert messages in the form of Gaussian watermarks. 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. Applications of our scheme include cybersecurity and cyberforensics.

Keywords: Video steganalysis, video steganography, collusion attack, pattern recognition, cybersecurity, computer forensics

1. INTRODUCTION

Steganography is the art of hiding data in innocuous-looking mediums such as text, audio files, still images and video sequences. Unlike cryptography, in which the goal is to scramble the information using a secret transformation to deny access to the original content, steganography tries to hide the very presence of a message by "embedding" it in another *cover*-message such that it is not detected. The modern formulation is given in terms of the prisoner's problem[1] in which Alice and Bob are two prison inmates who are trying to hatch an escape plan. They are allowed to communicate so as long as the prison warden, Wendy, can scrutinize their exchanges. Wendy will put the inmates in solitary confinement if she is suspicious of their plans. Therefore, Alice and Bob must communicate their escape details in such a way that it does not arouse suspicion to third parties. Alice embeds a *secret message* 'm' into a *cover-object* 'c' to obtain a *stego-object* 's'[2] and sends the stego-object through the warden to Bob. Bob is able to remove the secret message 'm' from the stego-object 's' using an appropriate decoding technique. To be successful, the stego-object must exhibit similar characteristics to the cover object and hence appear "innocent" to the warden and Bob must successfully decode 'm'. Thus, the overall goal of steganography is to make it difficult for third parties to distinguish between a cover-object and a stego-object while guaranteeing accurate covert communication.

The process of detecting with high probability and low complexity the presence of covert communication through innocuous multimedia distribution is called steganalysis. Steganalysis is a way of distinguishing between a stego-object and a cover-object. With recent attacks on information systems, cybersecurity and cyberforensics 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 or to produce Trojan horse content. Identifying such communication must be automated in order to be able to effectively and practically monitor or trace such behavior.[3] Thus there is a need for efficient and reliable methods that detect the presence of covert data in innocuous looking mediums. A number of effective steganalysis techniques have been proposed for still images, text files and audio, but to the best of the authors' knowledge there have been no steganalysis methods targeting the unique characteristics of digital video.

In raw format, video sequences can be considered as a series of still images. The presence of the temporal dimension increases the volume of the covert data payload that can be embedded in the medium. Thus, from an

Further author information:
Udit Budhia: E-mail: ubudhia@tamu.edu, Telephone: 1-979-862-1190
Deepa Kundur: E-mail:deepa@ee.tamu.edu, Telephone: 1-979-862-8684

Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III, E. M. Carapezza, ed., Proc. of SPIE Vol. 5403 (SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.540814

embedder's point of view, using video sequences as a cover-object is attractive since the capacity or the amount of covert data that can be carried is very high in comparison to other mediums such as text and digital audio.

In this paper, we assert that there is also an advantage from a steganalysist's point of view; there is a greater chance of detection due to presence of statistical redundancy along the temporal dimension of the cover-video sequence. We concentrate on developing efficient steganlysis techniques for video sequences that take advantage of the inherent temporal redundancy unique to this media. The method is devised to work in compliment with known still image steganalysis techniques that can also be applied to video sequences.

There are many steganalysis techniques[2, 4–8] for still images that can be applied to video sequences on a frame by frame basis. Farid *et al.*[5, 6] shows that the embedding of a message disrupts the higher order statistical regularity within an image. This has been exploited to build a pattern classifier to distinguish between a watermarked and non-watermarked image. Memon *et al.*[2] use image quality metrics to build a pattern classifier based on multivariate regression analysis to detect presence of covert data. Since embedding a secret message in an image can be modeled as addition of noise, Pearlman *et al.*[4] hypothesize that the histogram characteristic function of a stego-image will change through embedding and can therefore be used for steganalysis by using a Bayesian classifier. Fridrich *et al.*[7, 8] have proposed a number of methods to detect LSB encoded hidden messages. In general, successful steganalysis requires that an implicit or explicit model of the cover-object, stego-object and/or steganographic method be incorporated in the detection problem. The effectiveness of the approach is determined by the accuracy and completeness of these models in practical situations. Most proposed steganalysis methods assume that the class of algorithms for steganography (i.e., the embedding technique) is known a priori. We also assume that the embedding approach is known and subsequently devise a method that exploits temporal video frame correlation to detect its use on a given digital video sequence.

The next section introduces the nomenclature used for this paper. Section 3 formulates our specific problem. Section 4 presents our novel method followed by results and final remarks in Sections 5 and 6.

2. 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 the how strongly and where in the cover-object the secret information is embedded, and seed information for pseudo-random number generation. In this formulation, the medium used for covert communication is digital video.

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 goal of the steganalysist, is to monitor the public channel and detect the presence of any covert communication in a given video sequence. The scenario is summarized in Figure 1.

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[9] denoted by $W_k(m,n)$. For compatibility, the watermark $W_k(m,n)$ is defined over the same domain as the host $U_k(m,n)$. The stego-video signal is represented by the commonly used equation[10]:

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

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:

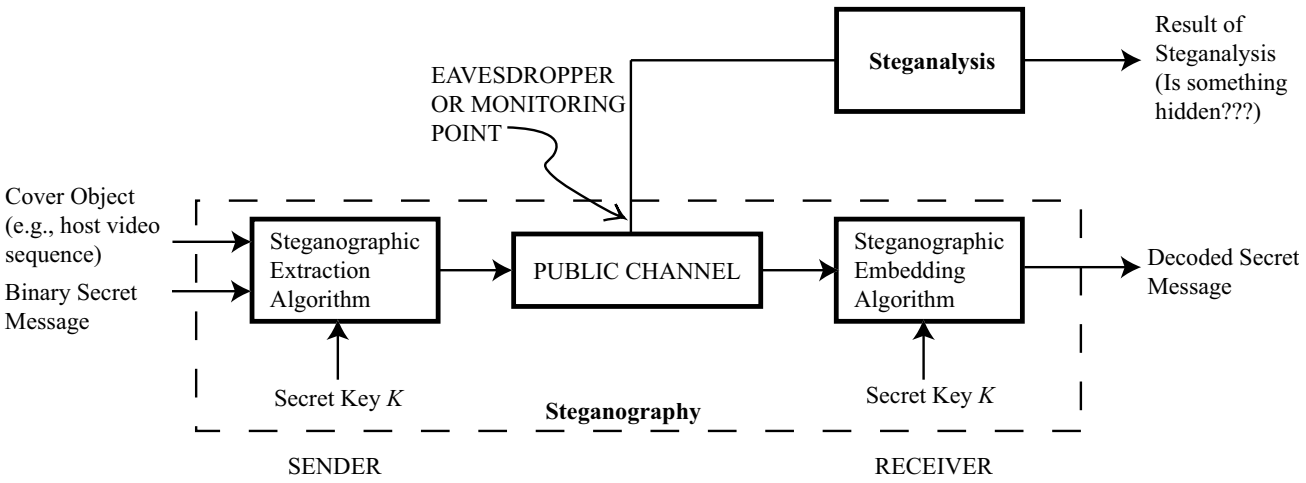


Figure 1. Steganography and steganalysis. Steganography consists of the process of embedding (by a sender) and extracting (by a receiver) covert information from innocuous messages. Steganalysis is the process of determining from a given message whether or not covert data has been embedded.

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

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 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.

3. PROBLEM FORMULATION

The overall goal of this paper is to design a steganalysis method for digital video sequences that is more optimal 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[9, 11] 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.[9, 11] 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 $\alpha = 5$.

Figure 2. 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 cybersecurity 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 3 summarizes the basic video steganalysis problem for spread spectrum embedding.

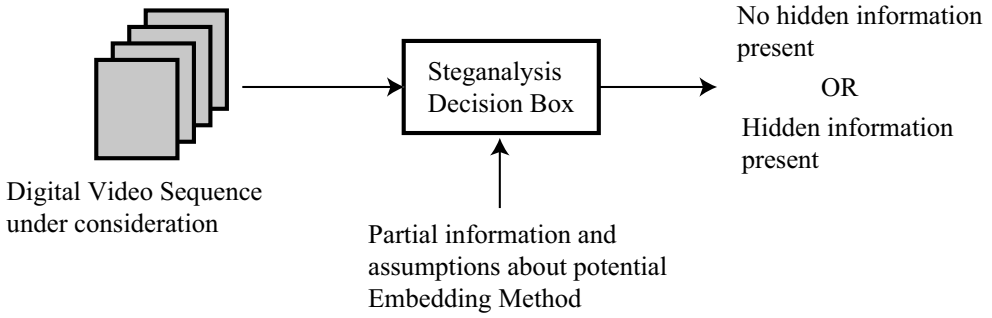


Figure 3. Video Steganalysis Problem. The objective is to design an decision box that takes a given video sequence and makes use of partial information about the potential embedding algorithm to decide whether or not hidden information is present in the given media.

4. COLLUSION AND CLASSIFICATION BASED STEGANALYSIS

The spirit of most steganalysis methods is to devise a function that differentiates between the general characteristics of a signal with and without embedding. This function is normally compared implicitly or explicitly to a threshold in order to decide whether or not a given signal $Y_k(m, n)$ contains hidden information. Thus,

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" $\hat{U}_k(m,n) = \mathcal{H}[Y_k(m,n)]$ signal from $Y_k(m,n)$. This "host" estimate $\hat{U}_k(m,n)$ is then compared in some way to $Y_k(m,n)$ in order to detect if something is hidden. The basic hypothesis is that the deviation of specific characteristics of $Y_k(m,n)$ and $\hat{U}_k(m,n)$ will differ if something is embedded in $Y_k(m,n)$ (i.e., $Y_k(m,n) = X_k(m,n) = U_k(m,n) + \alpha \cdot W_k(m,n)$) in comparison to when nothing is embedded in $Y_k(m,n)$ (i.e., $Y_k(m,n) = U_k(m,n)$). Pattern classification is often employed to characterize this deviation effectively.

In this work, 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.

Figure 4 presents our framework. The video sequence under consideration $Y_k(m,n)$ is passed through a digital watermarking attack block that attempts to estimate the host signal to produce $\hat{U}_k(m,n)$. This block may assume knowledge of the embedding algorithm (if any is used) to be effective. If something is in fact embedded in $Y_k(m,n)$, then $\hat{U}_k(m,n)$ should be a better estimate of $U_k(m,n)$ than $Y_k(m,n)$. If nothing is embedded in $Y_k(m,n)$, then $Y_k(m,n) = U_k(m,n)$ and $\hat{U}_k(m,n)$ may be some (possibly mildly) modified version of $U_k(m,n)$. To differentiate these two situations, the difference between $Y_k(m,n)$ and $\hat{U}_k(m,n)$ is taken and passed through an appropriate pattern classifier. If $Y_k(m,n)$ is a stego-video then the input to the pattern classifier is an estimate of the watermark. Otherwise, it is effectively independent of any assumed characteristics for the watermark $W_k(m,n)$ if any embedded. 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.

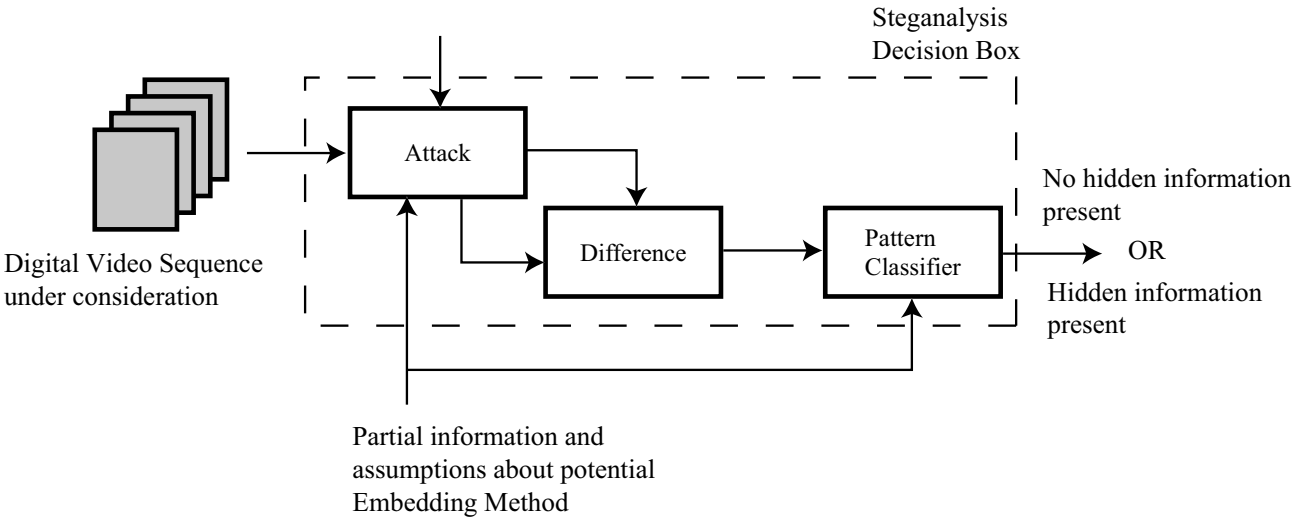


Figure 4. Proposed framework for steganalysis.

Since our goal is, in part, to develop a tool to enhance existing image steganalysis methods, we focus on algorithms for Figure 4 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, a more optimal 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.

An effective pattern classifier is also developed by incorporating knowledge that the watermark, if any present, is zero mean and Gaussian. In the next subsection we discuss the linear collusion attack and the classifier employed for our proposed video steganalysis.

4.1. Linear 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 to produce a resulting frame that has lower watermark content as follows:

$$Z_i(m,n) = \mathcal{C}_i[X_1(m,n), X_2(m,n), \ldots, X_N(m,n)] \tag{3}$$

where $Z_i(m,n)$ is called the colluded result and in this paper represents the estimate of the ith host frame $U_i(m,n)$ and $\mathcal{C}_i$ is the collusion operator that exploits the similarities and differences amongst all or a select subset of watermarked image frames $X_1(m,n), X_2(m,n), \ldots, X_N(m,n)$ to produce $Z_i(m,n)$. As we discuss, the colluded result $Z_i(m,n)$ in general contains significantly less contribution from $W_i(m,n)$ as compared to $X_i(m,n)$. Common forms of the collusion operator $\mathcal{C}_i$ 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 $\mathcal{C}_i$ represents a weighted average operation of select video frames; this attack has recently received much attention in the digital video watermarking community.[10,12] Intuitively, linear collusion on a sequence of video frames amplifies parts of the frames that are similar and attenuates components that are different. Thus, it has been shown analytically in Ref. 10 that if the linear correlation amongst host video frames $U_k(m,n)$ for some k differs from that of the watermark frames $W_k(m,n)$ over the same range of k then linear collusion will be successful in either attenuating or amplifying the presence of the watermark in the resultant frame $Z_i(m,n)$.

In this paper, we focus on the application of spread spectrum steganography on video sequences that in most applications requiring high covert data capacity implies that $W_k(m,n)$ 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 k centered at i, the watermarked video frames can be averaged in order to attenuate the presence of the watermark in the ith 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 $L+1$ frames (where L is assumed to be even) centered at frame i (except toward the beginning and end of the sequence since the window goes outside the range of k) to average the video sequence. The estimate of the ith host frame is given by:

$$Z_i(m,n) = \begin{cases} \frac{1}{L+1}\sum_{k=1}^{L+1} X_k(m,n) & 1 \le i \le L/2 \\ \frac{1}{L+1}\sum_{k=i-L/2}^{i+L/2} X_k(m,n) & L/2 < i < N - L/2 \\ \frac{1}{L+1}\sum_{k=N-L}^{N} X_k(m,n) & N - L/2 \le i \le N \end{cases} \tag{4}$$

where i is the frame under consideration to produce $Z_i(m,n)$, an estimate of $U_i(m,n)$. We next show why we assert that $Z_i(m,n) \approx U_i(m,n)$.

Substituting $X_k(m,n) = U_k(m,n) + \alpha.W_k(m,n)$ for all k from Equation 2 into Equation 4 we obtain:

$$Z_i(m,n) = \frac{1}{L+1}\sum_k U_k(m,n) + \frac{\alpha}{L+1}\sum_k W_k(m,n) \tag{5}$$

where the summations are over the appropriate domains for the various ranges of i shown in Equation 4. Since the watermarks $W_k(m,n)$ are independent and zero mean, the second term of the left hand side of Equation 5 approaches zero as L increases. Furthermore, because we assume $U_k(m,n) \approx U_i(m,n)$ for all k in the neighborhood of the sliding window centered at i, the first term will dominate resulting in the following approximation:

$$Z_i(m,n) \approx \frac{1}{L+1}\sum_k U_k(m,n) \quad (6)$$

$$\approx \frac{1}{L+1}\sum_k U_i(m,n) \quad (7)$$

$$\approx U_i(m,n) \quad (8)$$

The effectiveness of $Z_i(m,n)$ as an approximation of $U_i(m,n)$ depends on the value of L in relation to the rate of motion in the video sequence. This design parameter was found in our work by running simulations for various window lengths on common slowly moving test video sequences. A value of $L+1 = 11$ was found to be reasonable.

If collusion is applied to a given video sequence $Y_i(m,n)$ that may or may 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 $U_i(m,n)$. Thus if a watermark is embedded in the video, subtracting $Z_i(m,n)$ from $Y_i(m,n)$ gives $Y_i(m,n) - Z_i(m,n) \approx Y_i(m,n) - U_i(m,n) = \alpha W_i(m,n)$ an estimate of the scaled zero mean Gaussian watermark. If no watermark is present in $Y_i(m,n)$ 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 next section for steganalysis.

The reader should note that in the case of fast moving video sequences, the collusion attack applied to dissimilar frames may not result in a reasonable approximation for $U_k(m,n)$. We are currently working towards analysis that allows us to determine a lower threshold for the pairwise correlation between successive frames that guarantees successful steganalysis. However, in Section 6 we provide a practical alternative to improve linear collusion performance for steganalysis that involves block matching and reorganization in each frame for fast moving sequences whose correlation is below this as-of-yet undetermined threshold.

4.2. Classification

Our objective is to build a classifier that discriminates between an estimate of the scaled watermark and no watermark. The two main components of a typical classifier are feature extraction and the discriminator.[13] Feature extraction derives characteristics from the signal under consideration to provide relevant information to the discriminator for classification.

Figure 5 gives an example of the distribution (i.e., scaled histogram) of the difference between the $Y_k(m,n) - Z_k(m,n)$ when a Gaussian watermark is present and when no watermark is present. 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.

Kurtosis[14] is a value that partially measures the "shape" of a distribution. Kurtosis for a Gaussian distribution is zero and for most of the other distributions it is non-zero. It is defined as

$$Kurtosis = \frac{1}{\sigma^2 N}\sum (x-\mu)^4 - 3, \quad (9)$$

where σ and μ represent the variance and mean of the distribution.

Entropy[14] helps to determine the degree of "randomness" in a given distribution. For a fixed variance the Gaussian distribution has 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

$$Entropy = -\sum_{i=1}^{N} (p_X(i) log(p_X(i))), \tag{10}$$

where $p_X(i)$ is an estimate of the distribution of $Y_k(m,n) - Z_k(m,n)$ shown in Figure 5 for a specific test case.

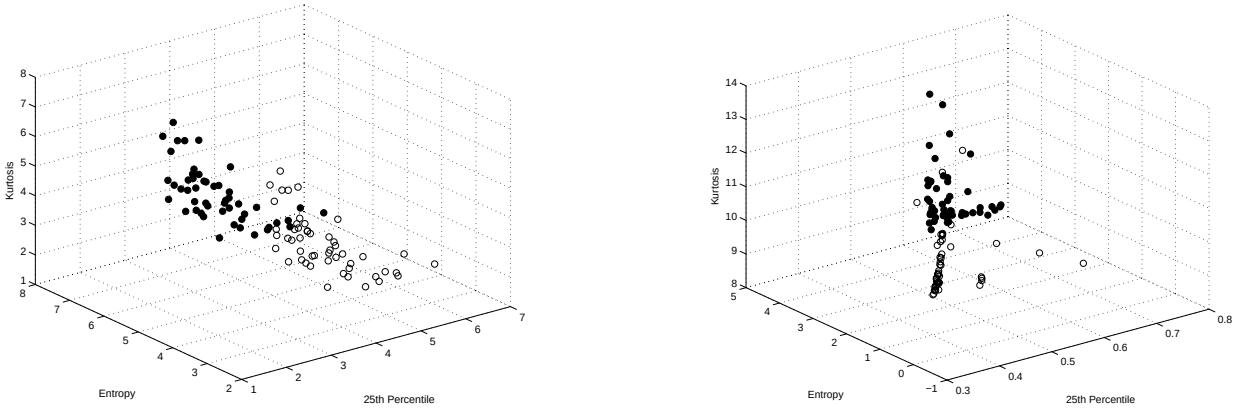
The last feature that we consider is the 25^{th} percentile of a given distribution defined as the value above which 25% of the points in the histogram reside. From Figure 5 it is clear that the distribution when a watermark is present is more spread than when no watermark is present resulting in a difference in this percentile value.

(a) Watermarked sequence

(b) Non-watermarked sequence

Figure 5. Distribution of the watermark estimates for a video sequence (a) with and (b) without steganographic data embedded.

Figure 6 represents a *scatter plot* of specific statistical features of $Y_k(m,n) - Z_k(m,n)$ 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 (defined later in this section) of $Y_k(m,n) - Z_k(m,n)$ 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.

Figure 6. 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.

Once the features are extracted, we build a kNN classifier.[13, 15] More sophisticated classifiers using support vector machines and neural networks[15] could have been employed for discrimination, but are higher in complexity without providing significantly improved performance. The kNN classifier must be trained to be able to operate for steganalysis. We use cross validation[13, 15] to determine the video set which would yield the lowest probability of false positive and false negative.

Table 1 summarizes the overall steganalysis method that incorporates linear collusion and classification.

Table 1. Collusion and Classification Based Steganalysis.

- Variable Definitions:

N	Number of Frames
$X_i(,)$	i^{th} frame of the video sequence
$Y_i(,)$	colluded version of the i^{th} frame
$E_i(,)$	estimate of the watermark in the i^{th} frame
O_i	Output from the pattern classifier i^{th} frame
$Coll()$	Collusion attack on L+1 frames as described in Section 4.1
$Patt()$	Pattern Classification on as described in Section 4.2

- Algorithm:

for i={1,2,. . . ,N}
$Y_i(,) := Coll(X_i(,))$
$E_i(,) := X_i(,) - Y_i(,)$
$O_i(,) := Patt(E_i(,))$
end

5. RESULTS

We applied our algorithm to 27 different grayscale video sequences in raw format consisting of 40 frames per set. The resolution of the still frames varied for different video sequences. Most of the video sequences used for simulation were "slow-moving" video sequences.

As discussed in Sections 2 and 3, 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 $W_k(m,n)$ with a zero-mean unit variance Gaussian distribution as presented in Equation 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 4 we use a sliding window to perform the collusion attack. The optimal window length of $L + 1 = 11$ is used for all simulations and was found through preliminary simulations. Different window lengths were employed for a collusion attack on test video sequences containing watermarks $X_k(m,n)$ to produce $Z_k(m,n)$. The difference $X_k(m,n) - Z_k(m,n)$ was then obtained to provide an estimate of $\alpha W_k(m,n)$. To determine the success of the window length for steganalysis, the pairwise correlation coefficient $\rho(W_k(m,n), X_k(m,n) - Z_k(m,n))$ was computed, where

$$\rho(A,B) = \frac{\mathrm{cov}(A,B)}{\sqrt{\mathrm{var}(A) \cdot \mathrm{var}(B)}}, \tag{11}$$

$\mathrm{cov}(\cdot,\cdot)$ denotes the covariance and $\mathrm{var}(\cdot)$ denotes the variance of the argument random variable(s). On average over all test video sequences, the window length of $L+1 = 11$ (i.e., $L = 10$) gave the highest pairwise value of ρ.

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 classifier[13, 15] 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.

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}.

Table 2 shows the probability of false negative P_{FN} and the probability of false positive P_{FP} for different values of embedding strength α. As we can see from Table 2, P_{FN} is reasonably low for most test video sequences except Sequence number 4. The failure of our method for this case is due to the rapid scene changes in this particular video sequence. We have proposed a slight modification to the collusion attack for fast sequences in Section 6 that should help overcome this limitation. We also note that P_{FP} is higher than P_{FN}. 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 2 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.

Table 2. False negative (P_{FN}) and False positive (P_{FP}) probabilities (in units of percent).

α	1		3		5	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq no :1	0	0	0	0	0	0
Seq no :2	0	0	0	0	0	0
Seq no :3	0	2.5	0	2.5	0	2.5
Seq no :4	27.5	40	27.5	42.5	27.5	42.5
Seq no :5	0	0	0	0	0	0
Seq no :6	0	0	0	0	0	0
Seq no :7	2.5	2.5	0	2.5	0	2.5
Seq no :8	0	0	0	0	0	0
Seq no :9	0	0	0	0	0	0
Seq no :10	0	0	0	0	0	0
Seq no :11	0	0	0	0	0	0
Seq no :12	0	0	0	0	0	0
Seq no :13	0	0	0	0	0	0
Seq no :14	0	5	0	5	0	5

continued on the next page

continued from the previous page						
α	1		3		5	
Sequence	P_{FN}	P_{FP}	P_{FN}	P_{FP}	P_{FN}	P_{FP}
Seq no :15	0	0	0	0	0	0
Seq no :16	0	0	0	0	0	0
Seq no :17	0	0	0	0	0	0
Seq no :18	0	0	0	0	0	0
Seq no :19	0	5	0	0	0	0
Seq no :20	0	0	0	0	0	0
Seq no :21	0	0	0	0	0	0
Seq no :22	0	0	0	0	0	0
Seq no :23	0	0	0	0	0	0
Seq no :24	0	0	0	0	0	0
Seq no :25	0	0	0	0	0	0
Seq no :26	0	2.5	0	0	0	0
Seq no :27	0	0	0	0	0	0

6. DISCUSSION AND FUTURE DIRECTIONS

The work presented in this paper demonstrates the potential of our framework and the use of temporal processing for effective steganalysis. In this section, we discuss some limitations of our algorithm and highlight areas of further research.

In comparison to spatial methods of image steganalysis, our temporal method gives slightly poorer performance for lower embedding strengths. Thus, integration of spatial and the proposed temporal approach for steganalysis must taken into account the varying degrees of accuracy for different α. This is a topic 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.

The collusion attack fails if the pairwise correlation between subsequent frames falls below a certain threshold.[10] We propose a strategy to improve collusion performance in such cases. For a given frame i, all other frames in the $L+1$ window can be block-reordered to form frames that are visually similar (and more correlated) to the ith one. A block matching strategy similar to that used for MPEG is feasible. We believe this will improve collusion performance for fast moving video sequences.

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[2, 5, 16] 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.

Current research efforts focuses on developing mathematical analysis to formally determine the strengths and limitations of our steganalysis approach.

7. ACKNOWLEDGMENTS

The picture, Elaine, used in Figure 3 was downloaded from USC-SIPI Image Database at http://sipi.usc.edu/services/database/Database.html. The 27 sets of video sequences used for simulations were downloaded from http://ise.stanford.edu/video.html and http://www.acticom.info/1489.html.

REFERENCES

1. G. J. Simmons, "The prisoner's problem and the subliminal channel," in *Advances in Cryptology, Proc. CRYPTO '83*, pp. 55–67, 1983.
2. I. Avcibas, B. Sankur, and N. Memon, "Steganalysis based on image quality metrics - differentiating between techniques," in *Proc. IEEE Workshop on Multimedia*, (Cannes, France), October 2001.
3. G. Mohay, A. Anderson, B. Collie, O. de Vel, and R. McKemmish, *Computer and Intrusion Forensics*, Artech House, 2003.
4. J. J. Harmsen and W. A. Pearlman, "Steganalysis of additive noise modelable information hiding," in *Proc. SPIE Security and Watermarking of Multimedia Contents V*, **5020**, January 2003.
5. H. Farid, "Detecting hidden messages using higher-order statistical models," in *Proc. IEEE International Conference on Image Processing*, (Rochester,New York), September 2002.
6. H. Farid and S. Lyu, "Detecting hidden messages using higher-order statistics and support vector machines," in *Proc. 5th International Workshop on Information Hiding*, (Noordwijkerhout, The Netherlands), 2002.
7. J. Fridrich, R. Du, and L. Meng, "Steganalysis of lsb encoding in color images," in *Proc. IEEE Conference on Multimedia and Expo*, (New York City, New York), July-August 2000.
8. J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in grayscale and color images," in *Proc. ACM Workshop on Multimedia and Security*, (Ottawa, Canada), October 2001.
9. I. J. Cox, J. Killian, F. T. Leighton, and T. Shamoo, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Proceedings* **6**, pp. 1673–1687, December 1997.
10. K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility in collusion-resistant digital video watermarking," *IEEE Transactions on Multimedia* , to appear.
11. L. M. Marvel, J. C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Transactions on Image Processing* **8**, pp. 1075–1083, August 1999.
12. 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, July 1998.
13. R. O. Duda, P. E. Hart, and D.G.Stork, *Pattern Classification*, Wiley, 2nd ed., 2001.
14. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw Hill, 4th ed., 2002.
15. R. Gutierrez-Osuna, "Cpsc 689-604: Special topics in pattern analysis." Lecture Notes, September 2003. http://faculty.cs.tamu.edu/rgutier/courses/cpsc689_f03/index.html.
16. I. Avcibas, B. Sankur, and N. Memon, "Image steganalysis with binary similarity measures," in *Proc. IEEE International Conference on Image Processing*, (Rochester, New York), September 2002.