DEFENSES AGAINST COVERT-COMMUNICATIONS IN MULTIMEDIA AND SENSOR NETWORKS

A Dissertation

by

JULIEN SEBASTIEN JAINSKY

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

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Deepa Kundur
Committee Members,	Jim Ji
	Michael Longnecker
	Erchin Serpedin
Head of Department,	Chanan Singh

December 2012

Major Subject: Electrical Engineering

ABSTRACT

Steganography and covert-communications represent a great and real threat today more than ever due to the evolution of modern communications. This doctoral work proposes defenses against such covert-communication techniques in two threatening but underdeveloped domains. Indeed, this work focuses on the novel problem of visual sensor network steganalysis but also proposes one of the first solutions against video steganography.

The first part of the dissertation looks at covert-communications in videos. The contribution of this study resides in the combination of image processing using motion vector interpolation and non-traditional detection theory to obtain better results in identifying the presence of embedded messages in videos compared to what existing still-image steganalytic solutions would offer. The proposed algorithm called MoViSteg utilizes the specifics of video, as a whole and not as a series of images, to decide on the occurrence of steganography. Contrary to other solutions, MoViSteg is a video-specific algorithm, and not a repetitive still-image steganalysis, and allows for detection of embedding in partially corrupted sequences.

This dissertation also lays the foundation for the novel study of visual sensor network steganalysis. We develop three different steganalytic solutions to the problem of covert-communications in visual sensor networks. Because of the inadequacy of the existing steganalytic solutions present in the current research literature, we introduce the novel concept of preventative steganalysis, which aims at discouraging potential steganographic attacks. We propose a set of solutions with active and passive warden scenarii using the material made available by the network. To quantify the efficiency of the preventative steganalysis, a new measure for evaluating the risk of steganography is proposed: the embedding potential which relies on the uncertainty of the image's pixel values prone to corruption.

DEDICATION

To my Parents

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. Background

The fact is that we now live in a world where everything is dominated by computers and numerous electronics systems. By using these new age tools, people seek to make their life easier and somewhat more civilized and secure but it is clearly apparent from the current state of things that these are false assumptions. Never before have our systems been under so many threats and attacks [1][2][3]. Hackers and e-pirates form a large group of clever minds who know perfectly well how to manipulate state-of-the-art technology for their own devious purposes. It does not matter which field of studies, for every new software created, knowledgeable hackers can and do develop countless attacks for any reason that suit them and sometimes for no reason at all except the challenge it represents. As a consequence, privacy issues and security problems are constantly multiplied and can threaten the life and identity of individuals every day. Such is the reality. And it requires defenses.

The challenges for a security specialist are renewed every day due to his, or her, eternal struggle with the ever changing attacker's schemes. In parallel with the evolution of technology, the role of the security consultant becomes more demanding and difficult but it is also most important today and tomorrow than it ever was [4][5]. Terrorists' cells nowadays are not solely composed of amateurish home bomb makers but skilled scientists and engineers capable of sophisticated attacks on sensible networks or governmental databases. It is therefore of the utmost importance to create original and efficient means to protect ourselves and, to the best of our abilities, prevent further attacks.

One of the fields that have been used on several occasions by terrorist is steganography [6]. It was reported in 2001 that Al Qaeda hid specific plans for the attacks of 09/11 in innocuous-looking images and that members of this terrorists' cell received training in covert communications [7][8]. Proof that Al Qaeda is still using steganography has been

found in May 2012 when hidden messages concerning an attack plot were retrieved from a pornographic video carried by a man identified as a terrorist [9]. In 2006, the National Science and Technology Council pointed out the seriousness of the threat associated with steganography and encouraged further research in defenses against covert-communications [10].The threat is therefore real and has renewed interest in the study of steganalysis which is used to detect the presence of hidden messages.

This is the reason why, in this dissertation, we focus our attention to the problem of covert communications and the potential solutions to identify their occurrences. Given the high degree of collaboration and cooperation in modern information systems such as emerging multimedia sensor networks, covert communications is a greater threat than ever. Network-level approaches to covert transmission have classically involved passing information innocuously via shared resources by having one communicating entity modulate network characteristics (such as transmission times or storage elements) such that a second party (who can monitor the resources) deduces the secret message. More modern approaches to covert transmission in networks have included the use of steganographic mechanisms in network protocol packets [11]; however, recent research [12] has suggested that communicating covertly at such a structured level is easily detectible making it more attractive for attackers to employ subversive communications at the multimedia content level.

Studies in enabling and preventing covert transmissions have repeatedly demonstrated a fundamental trade-off between the reduction of covert communications capacity and the performance of overt communications. As communications, computation and sensing converge to create advanced multimedia sensor networking, we assert that it will become practically impossible to design high performance networks that prevent covert communications. The high levels of redundancy of such networks provide a rich environment for data hiding without significantly affecting network performance. In addition, the scalable network design often requires collaboration on the part of network entities enabling subversive communications to a much greater extent.

Furthermore, the acquisition of highly correlated multimedia provides fertile ground for advanced steganographic approaches. Covert communications in multimedia networks is of special concern for several reasons. Given recent interest in employing multimedia sensor systems for tactical military and healthcare applications, such networks are natural targets for attack. A distinguishing assumption in threat models of these systems is the high likelihood of insider attack via corrupt network entities that facilitate subversive behavior. In addition, sensor network security strategies often entail intrusion detection mechanisms that only exploit deviations in overt communication statistics to assign a trust-level to each network entity encouraging stealthy behavior [13]. Moreover, covert communications among select network participants allows for strategic cooperation amongst corrupt nodes resulting in highly effective denial-of-service attacks [14].

1.2. Concepts: Covert Communications, Steganography and Steganalysis

1.2.1. Covert Communications and Steganography

Covert communication is an illegitimate mean of exchanging information. Covert channels have traditionally been employed for stealthy communication of sensitive information from high security areas to low security areas [15-16]. This often undetectable information leakage has the potential to breach both security and privacy of a given system. On the other hand, overt channels are used for authorized transmission of information over a network. Attacks on overt channels often try to take advantage of the existing communication path to convey additional data, usually restricted, and doing so secretly. Such attacks include the embedding of information within the original, authorized and legitimate media travelling through the overt channel: this is steganography.

A steganographic system involves two parties: the sender who embeds the secret message in the cover-media to produce the stego-media and the receiver who extracts it. Security comes, in part, from the presence of a symmetric 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 particular to the steganography algorithm and may impose specifics such as how strongly and where in the cover-object the secret information is embedded, and/or seed information for pseudo-random number generation.

Figure 1 shows the steganographic embedding process where the sender takes the host media called the cover-media, which can be an image, a sound or a video, and embeds a secret binary message vector using K to produce a stego-media that is perceptually identical (and possibly similar in some statistical sense) to the cover- media [17], more specifically, a steganography system has been defined as secured in [18] if the *Kullback-Leibler* distance between cover- and stego-media is *0*. A more general and flexible concept considers the ε-security of a steganographic system if the same distance is less than ε. The stego-media is then communicated along a public channel to the receiver.

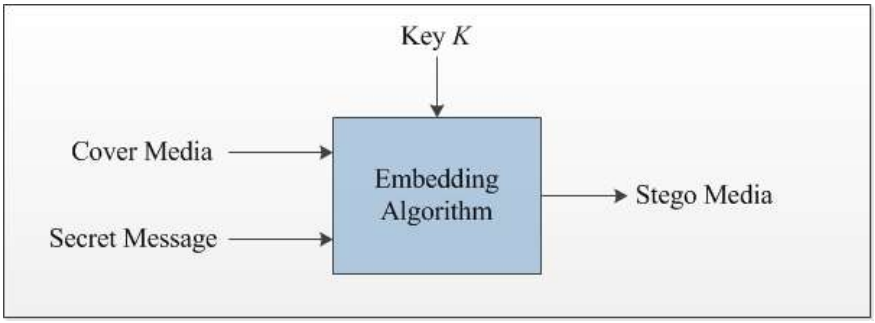

Figure 1. Steganography - sender process.

At the receiver the stego-object and secret key K are used to extract the secret binary message as illustrated in Figure 2. The public channel may be monitored by an active or a passive steganalyst whose goal is to detect the presence of covert communication.

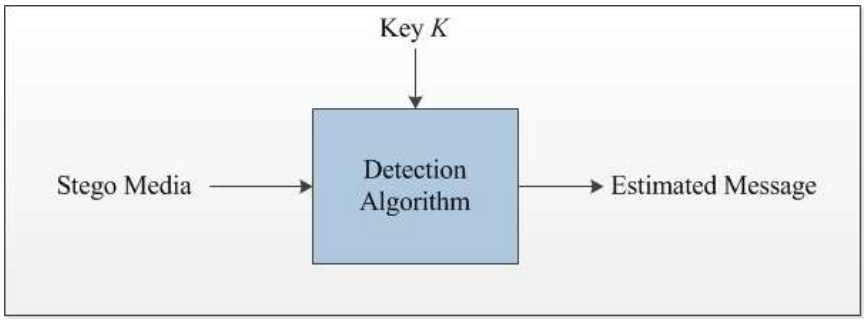

Figure 2. Steganography - receiver process.

The security of a steganographic system can be measured in terms of how robust it is to steganalysis, which measures how easily the presence of hidden data can be detected by an external party without access to *K*. Thus, when data is hidden in a media, the associated "mark" (often known as a watermark) that is embedded should not reveal with high probability that hidden data exists or any characteristics of the covert message such as its length or embedding location. However, the cover-media and stego-media cannot necessarily be identical if some non-zero capacity of hidden data is embedded.

1.3. Steganalysis

The general process of "attacking" a steganographic system is known as steganalysis and is used to detect, and potentially annihilate the presence of steganography [19]. Specifically, the steganalyst will attempt to effectively exploit information about the differences between the cover- media and stego- media (using information about the embedding algorithm and/or characteristics of the cover-media) in order to detect the presence of steganography.

Several classifications of steganalysis exist; the most commonly used are illustrated in Figure 3.

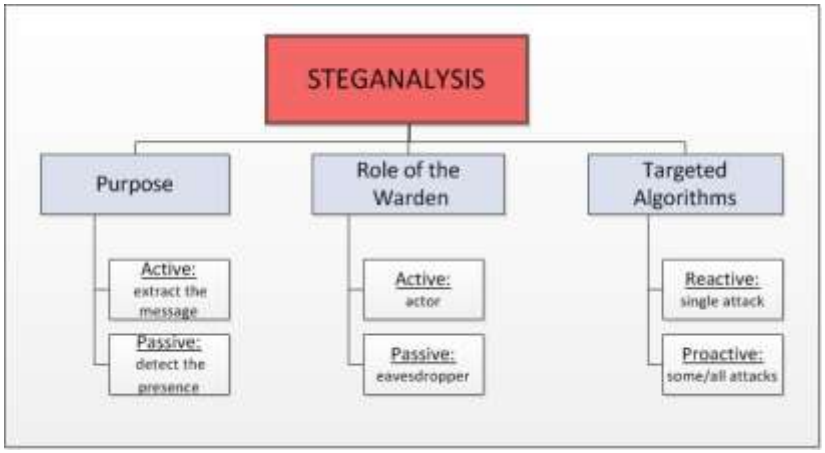

Figure 3. Classification of steganalysis.

Although most steganalysis share a passive purpose where the steganalyst only tries to detect the presence of the message, some rare proposed algorithms aim at discovering more information about the embedding such as its length, position in the media and, very rarely, its content, making these algorithms active in their purpose.

In most publications, active or passive are two adjectives that define the action of the steganalyst, often referred as the warden, and are also illustrated in Figure 4. The steganalyst himself, or herself, can be either active or passive [20]. An active steganalyst destroys the presence of any hidden information in a given image via signal processing such as lossy compression techniques or by introducing imperceptible distortions to an image. Active steganalysis processes all images (whether or not they contain hidden data) and thus acts as an insurance policy against steganographic activity. In passive steganalysis, one wants to detect the possible presence of a hidden message by eavesdropping on the communication channel. This task becomes more difficult to achieve as steganographic technology improves to provide smaller deviations between the cover- and the stego-images. Still, using various statistical tools, one can find some anomalies for steganalysis. As for the last classification illustrated in Figure 3, a reactive

steganalysis targets one specific type of steganography whereas proactive steganalysis targets a class of steganographic techniques.

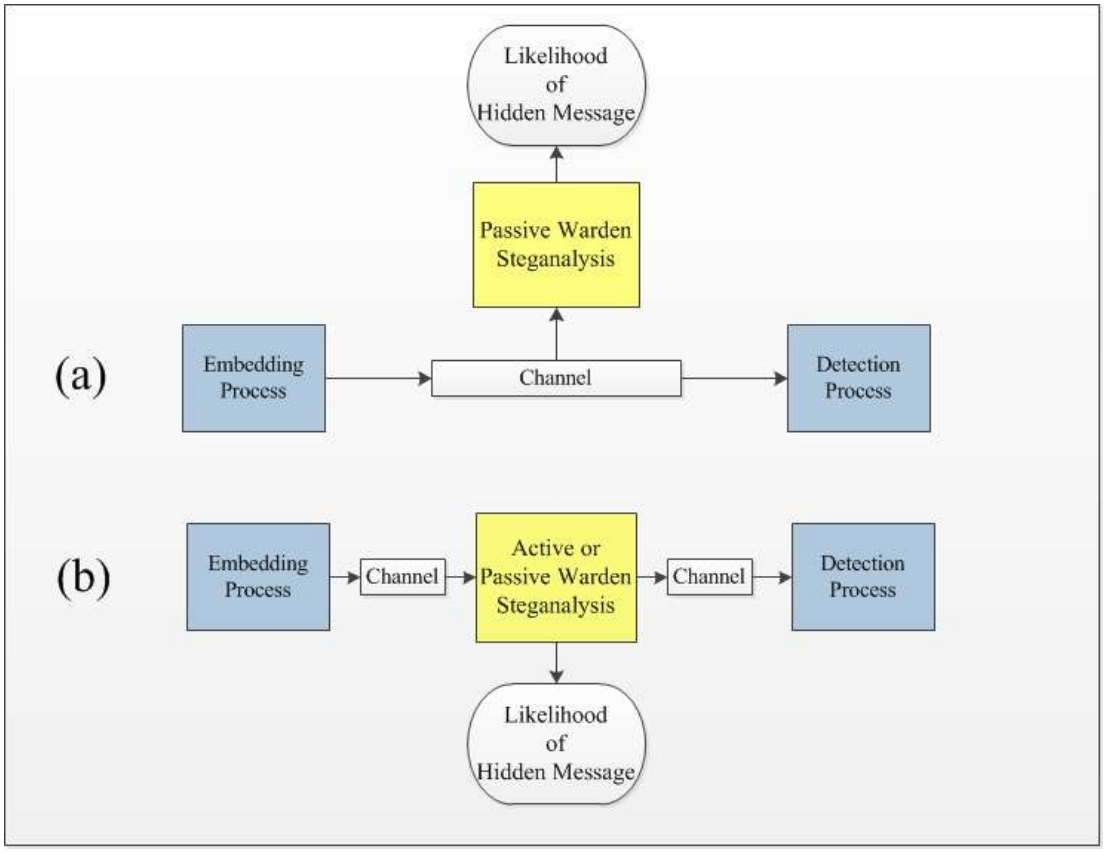

Figure 4. Steganalytic scenario: (a) the steganalyst is an eavesdropper on the communication channel, (b) the communication channel goes through the steganalyst who acts as a middleman.

1.4. Literature Review

Steganography has been used since the Antiquity and the first examples were related by Herodotus, considered to be the first Greek Historian, in 500 B.C.E [21]. In his book entitled 'Histories', he wrote about two incidents dealing with covert-communications. The first one mentions the use of a tablet where a message was inscribed warning the Lacedemonians about an upcoming attack. The tablet, once the message inscribed, was covered with wax. The Lacedemonians only had to scrap the wax off to be able to read

the message. The second example concerns a message sent by Histiaeus, ruler of Miletus to his nephew Aristagoras, asking him to rebel against the Persians. In order to secretly pass this message, Histiaeus shaved the head of a trusted slave and imprinted the message like a tattoo. Once the slave's hair had grown back, he was sent to Aristagoras who only had to shave the slave once more to read the intended message. Of course, since then, progress in technology and science has made covert-communications more elaborate and efficient [22-25].

Earlier steganographic methods made use of the least significant bit (LSB) plan of an image to hide information while introducing the least amount of distortion. EzStego was proposed by Romana Machado [26] and essentially embeds the hidden message in the LSBs of select pixels of paletted images. Given that each pixel of an image is coded with n bits, EzStego orders the colors of the cover-image into pairs of values (PoVs) that are in close color proximity (ideally, perceptually identical when swapped for one another in the image). Once the palette is sorted, the message is embedded in select pixels by possibly swapping the color of each pixel with its associated pair member to reflect the the data bit to be hidden. Another widely used algorithm is F5 [27-28]. First proposed by Pfitzmann and Westfeld in 2001, the method was considered to be both secure and have high-embedding capacity; that is, a high volume of hidden message bits can be embedded without detection using existing steganalysis methods. The F5 algorithm was designed, in particular, to counter a well-known first-order statistical steganalysis attack (also known as the $\chi 2$ -attack) also developed by the authors of [27-28]. OutGuess was proposed by Neils Provos in 2001 [29]. Analogous to the F5 algorithm, it was developed to counter attacks on the first-order statistics and is compatible with the JPEG image format. OutGuess essentially embeds the hidden message in the LSBs of a selected group of DCT coefficients, and then compensates for any changes in the first-order statistics by judiciously modifying the remaining DCT coefficient LSBs. As of today, these three algorithms have been defeated [30-32].

More recent publications in still image steganography have focused on reaching higher embedding capacity while introducing less distortion in the cover-image, especially since new picture formats aim for high level of compression in order to reduce the size of images. For example, Zhang et al. developed a model to increase embedding capacity of JPEG2000 images, a highly compressed image format [33]. Others chose to study the embedding of information in the frequency domain of the video where steganography can gain in capacity. Although DCT has been the primary choice in steganography before, studies have used the discrete wavelet transforms (DWT) [34] and curvelet transform [35] to achieve high embedding capacity.

An interesting novel steganographic scheme has involved the use of Sudoku puzzles as the key in the embedding and data extraction algorithms [36-38]. In 2008, Chang et al. introduced the concept of Sudoku based steganography in greyscale images [36]. In this publication, the Sudoku puzzle is used as a key to embed the secret message. Since there are 6.67×10^21 possible Sudoku grids for a 9x9 system, the key is relatively secure, even more so than the Data Encryption Standard (DES) keys. Later publications, respectively in 2009 and 2012, have developed further the concept of Sudoku steganography by making it available for color-images. In [37], Roshan Shetty et al. only consider the embedding to occur in two of the three color plans, namely red and green, and in [38] Sanmitra et al. extend the embedding scheme to the three RGB plan.

Video steganography is a very recent research subject. Until now, the assumption was that each frame of the video would be corrupted by an existing steganographic algorithm, i.e. a still image embedding technique. However, recently researchers have embedded message on features that are specific to videos such as motion vectors. Researchers chose to embed information in video by modifying the characteristics of motion vectors of coded sequences, preferably the largest motion vectors in order to remain as stealthy as possible. These characteristics include the horizontal and vertical conponents of the vectors [39] and the angle phase [40].

The majority of the steganalysis methods developed has focused on still images steganalysis[41-47]. Early steganalysis methods were reactive, meaning that they targeted only a specific type of embedding algorithm. For instance, Westfeld and Pfitzmann, in 1999, developed both visual and statistical attacks that observe the distribution of pairs of values against the software EzStego which used a LSB replacement method for embedding data [41]. Later methods were applicable to a broader range of embedding methods. For example, Fridrich et al. developed RS-steganalysis [42] which can identify the application of a class of LSB embedding methods. Subsequently, in [43], Fridrich et al. used the characteristics of JPEG compressed images as ``signatures" in order to detect data hiding.

More proactive steganalysis approaches that apply to a broader class of embedding methods have also been proposed. These methods are often called blind or universal steganalysis techniques since no (or very few) assumptions on the embedding process are made. Within this class, Farid et al. employed the higher order statistics of image features [44]. In this paper, the image is decomposed via a wavelet transform and the mean, variance, skewness and kurtosis of associated coefficients and their features are used to differentiate cover-images from stego-images. Since few assumptions on embedding are made, training to estimate thresholds and soft computing are employed. In [45], Avcibas et al. introduced the use of image quality metrics (IQMs) in order to discern between cover and stego-images by taking into account the more global characteristics of natural images. Their approach uses discriminative image statistics such as the Minkowsky metric, the spectral magnitude and the normalized mean square error as well as regression analysis to build a composite measure to identify the presence of hidden data.

More recently methods founded on detection theory have been presented with the hope of leading to a more universal and high-performance steganalysis solution. In [46][47],

Sullivan et al. developed a detection theoretic approach that employs a Markov chain (MC) model for spatial correlation in the cover-image order to identify the presence of hidden data. The authors argue that their approach provides a fundamental benchmark for evaluating the security of data hiding algorithms. Their detection algorithm is based on the observation that the divergence of the transition matrix, comprised of conditional probability values governing the MC source model, behaves somewhat predictably when data is embedded in an image; in particular, significant matrix values ``spread out'' from the main diagonal. To quantify this characteristic, features are extracted from an empirically generated transition matrix of the suspect-image and classification based on supervised learning strategies is employed to deduce whether the suspect is a cover-image or stego-image. Their approach illustrates the necessary interaction between detection theory and signal processing to develop a practical method governed in well-developed theory.

Against steganography in the frequency domain, which receives more attention than spread-spectrum steganography recently, recent publications have looked at the correlation between components in the transformed domain. For example, in [48], Chamorro et al. look at the probability density function of DCT blocks, called intra-block density) and the joint pdf of neighboring 8x8 DCT blocks (called inter-block density) to find statistics to separate cover-images from stego-images. Similarly, in [49], Zang et al. compare the pdf and joint pdf of the wavelet component of the image after a two-level DWT decomposition. Both algorithms target potential steganography in JPEG images with high performance.

Recently, researchers have been interested in studying the potential of video steganalysis [50-51]. The proposed technique leverages the differences in correlation from frame-to-frame between the watermark and the cover-video in order to statistically separate the components. In particular, the method assumes that each frame of a stego-video contains a hidden spread spectrum watermark and employs the collusion attack on adjacent video

frames in order to estimate the cover-video. Features of the difference between the suspect-video and cover-video are then classified using a kNN classifier that is trained a priori. The introduction paper on MoViSteg that is the part of this study and will be further discussed in this dissertation has also been published [52-53].

Since then, the field of video steganalysis has rather flourished, especially in the last couple of years. For example, in 2012, Htet and Mya proposed to use images higher order statistics in order to observe a particular behavior that would help separate corrupted videos from uncorrupted ones via the use of a Bayes classifier. These statistics include the entropy, the contrast, the angular second moment and the inverse difference moment [54]. Other attempts have been made to derive reactive video steganalysis [55-56]. Su et al. chose to work on defeating the Moscow State University steganographic algorithm for video, one of the only available video embedding technique whereas Pankajakshan created a steganalytic solution against steganography in MPEG videos.

Sensor networks receive also the attention of many researchers. It has been said that sensor network would represent a market of 8 billion dollars in 2010 [57]. A sensor network is commonly described as a group of nodes with sensing capabilities that are deployed in a field of interest for various applications. For example, a typical military application is the monitoring of an area for intrusion detection [58-60]. A more environmental application would be the monitoring of birds and their behavior as done by the University of California, Berkeley in the Great Duck Island [61].
Sensor networks are also targets of a wide range of attacks such as denial of service attacks and attacks on routing [62-64]. In the case of environmental applications such as bird monitoring, the security threats can be ignored, except in the case of physical tampering of nodes by animals but the probability of such attacks occurring is rather small. However in military applications, security is of main concern especially because nodes are usually deployed in a hostile environment therefore the potential attacker can have direct physical access to the network and can therefore corrupt several nodes.

Concerning sensor networks, most well-known measures to protect WVSNs, to date, have focused on the problem of providing privacy in vision-rich systems. Lo *et al.* [65] introduce an automated homecare monitoring system for the elderly named *UbiSense* where image processing is conducted directly at the camera to convert visual data directly into abstractions that reveal no personal information and hence protect the privacy of the monitored individuals. Fidaleo *et al.* [66] introduce the *Networked Sensor Tapestry (NeST)* architecture designed for the secure sharing, capture, and distributed processing and archiving of multimedia data. They introduce the notion of "subjective privacy" in which processing of raw sensor data is conducted to remove personally identifiable information; thus the behavior, but not the identity of an individual under surveillance is conveyed. The resulting data, approved for public viewing, is communicated in a network that employs the secure socket layer protocol and client authorization for network-level protection. Wickramasuriya *et al.* [67] present a privacy preserving video surveillance system that monitors subjects in an observation region using video cameras along with motion sensors and RFID tags. The motion detectors are used to trigger the video cameras on or off, and the RFIDs of the subjects provide authorization information in order to specify which individuals are entitled to privacy and hence have their visual information masked through image processing. Kundur *et al.*[68] present the HoLiSTiC (Heterogeneous Lightweight Sensornet for Trusted Visual Computing) framework for WVSN security that exploits secure protocols in a hierarchical directional link communication network to achieve broadband low power communications. A decentralized visual secret sharing approach is used to preserve privacy.

More recently, research has also emerged with the goal of assuring the authenticity of the data collected by sensor networks. When nodes are corrupted and provide false information, the entire network's legitimacy is compromised. The authentication of each node allows for the network to remain trusted. Several proposed solutions utilize common cryptographic concepts to provide such security [69-74]. For example, Feng et

al. [69] introduce a paradigm to cryptologically embed signatures into the collected data via watermarking techniques. Their objective is to efficiently watermark the data while introducing as little distortion as possible. Zheng et al. [70] propose to offer authenticity assurance using a public key cryptographic scheme. A derivable public key scheme is used which has the effect of simplifying the cryptography and reducing the need for key storage, therefore making it more suitable for large scale sensor networks. Because these methods still increase the workload of the WSN, Martinovic et al. [75] propose a novel paradigm that relies on the properties of wireless communications to provide authentication capabilities. They focus their study on taking advantage of frequency jamming to detect attacks and strengthen the WSN's security. Given the need for energy conservation in distributed wireless networks, Blaß et al. [76] develop Extended Secure Aggregation for Wireless Sensor Networks (ESAWN). ESAWN finds a trade-off between decreasing the energy consumption of the network via data aggregation and providing authentication mechanisms that are fundamentally weaker compared to techniques that are not driven by energy preservation. Various leads for research on security for wireless sensor networks have also been proposed in [14] and in particular data correlation in dense networks has been described as an important parameter that could help for detecting node corruption.

These existing approaches for WVSN protection all focus on protecting the *overt* data acquisition and communications systems. Fundamental questions however arise regarding the possibility of *covert* approaches for networking leading to breaches in both security and privacy. In this dissertation, we propose to study the possibility of, implications to and mitigation approaches for covert networking in the context of WVSNs.

1.5. Contributions

In this dissertation, we aim at identifying the existence of steganographic activity in two research fields that are at an early stage of their development. Due to the high demand

for video security and the projected extensive use of sensor networks in the future, we focus our attention to the study of steganalysis in both videos and sensor networks.

To the field of video steganalysis, our contribution will be twofold:

1- We derive a simple method to estimate the cover-media by using features characteristic of video and the high redundancy between consecutive frames,
2- We use a detector that favors the temporal distribution of the video frames instead of borrowing a detector from still-image steganalytic protocols therefore taking full advantage of the video.

Our next contributions are to the field of Video Sensor Networks. We lay the foundation for the study of steganalysis in the field by proposing the first solution against steganography in sensor networks. Because of the shortcomings of current steganalytics solutions, unsuitable for the challenges met in sensor networks, we introduce the concept of preventative steganalysis which aims at discouraging potential attackers from embedding messages in the images captured by the network's cameras.

We study different approaches to preventative steganalysis in sensor networks but eventually decide to create a new measure to quantify the risk for steganography to occur in images, called the embedding potential. We show how to reduce that embedding potential by using specifics of visual sensor networks, such as spatial and temporal redundancies expected between captured samples, and how it facilitates the steganalysis.

Our contribution to the field of Visual Sensor Networks Steganalysis can be separated as follows:

1- Create awareness of the dangers of steganography in sensor networks,
2- Provide a new classification for steganalysis illustrated by a set of non-invasive steganalytic solutions from an active-warden to a passive-warden scenario,
3- Offer a new measure on the potential for steganography in images.

2. STEGANALYSIS IN VIDEO

The existing literature focuses mainly on the steganalysis of still images. The results for still image steganalysis are typically obtained by first determining an estimate as close as possible to the original image. The estimated image is then subtracted to the input image and the difference is compared to a predefined threshold. However, with the fast growing use of videos on the Internet and the great hiding capacity of videos, it seems legitimate to insist on the importance of developing new video steganalytic methods.

2.1. Objectives of the Study

The proposed research will be directed at determining the presence of steganography in video for passive steganalysis using motion vectors and advanced decision theory. A statistical approach is adopted to decide whether hidden content is present. Some advantages of using this approach include:

1) Cover-video with only few contaminated frames should be detected easily
2) Both the watermark and host video are considered as random variables allowing more flexibility
3) The resulting correlation model for the video fits the reality

2.2. Methodology and Notions

Videos can be seen as sequences of images, allowing the existing steganalyses for still images to be adapted to detect the presence of hidden messages. However, since images in a video are usually highly correlated, by using the video redundancy we can take advantage of video features to detect steganography.

The steganalysis will incorporate two different stages. First stage is the estimation of the original media using motion vectors. The second stage is the detection which will result in the decision whether hidden content has been embedded in the video.

In this section, general notions about motion vectors and detection theory are detailed.

2.2.1. Motion Estimation

Typically motion estimation is done by prediction of the current frame from a previous or future reference frame.

In the case of a video coded with motion vectors, it would prove to be more time efficient to use directly the motion vectors from the coding. However, in this study the videos are considered as non-coded in order to generalize the process to any type of video; hence a motion estimation algorithm is needed.

When examining a video sequence, it should be realized that each frame of the sequence can be corrupted by steganography. Keeping that in mind, we want to evaluate each n^{th} frame of the video sequence without using the original frame n. To ensure this, it has been decided to consider the original n^{th} frame as being blank. Hence our estimation problem becomes an error-concealment problem: suppose that frame n is missing in the video sequence, an error concealment method to the whole frame n will be used to retrieve it. Instead of using only one reference frame, the steganalytic scheme developed here to evaluate frame n will use both frames $(n-1)$ and $(n+1)$ as references frames.

In an image sequence, the transformation from one frame to another can be determined by motion vectors. In the present case, the motion vectors will define the displacement of a block of pixel, or macro block, from one frame to another. This kind of approach typically uses motion vectors to evaluate the current frame. Frame n can be evaluated using the following method:

1) Estimate the motion vectors between frames $(n-1)$ and $(n+1)$
2) Take the average resulting motion vectors to estimate frame n.

Figure 5 illustrates this motion vectors method. The two frames contain a gray circle which had moved from one frame to another and the motion vector representing the

circle's displacement is generated. Assuming uniform motion, the inter-frame (frame *n*) would have its gray circle displaced by half of the motion vector.

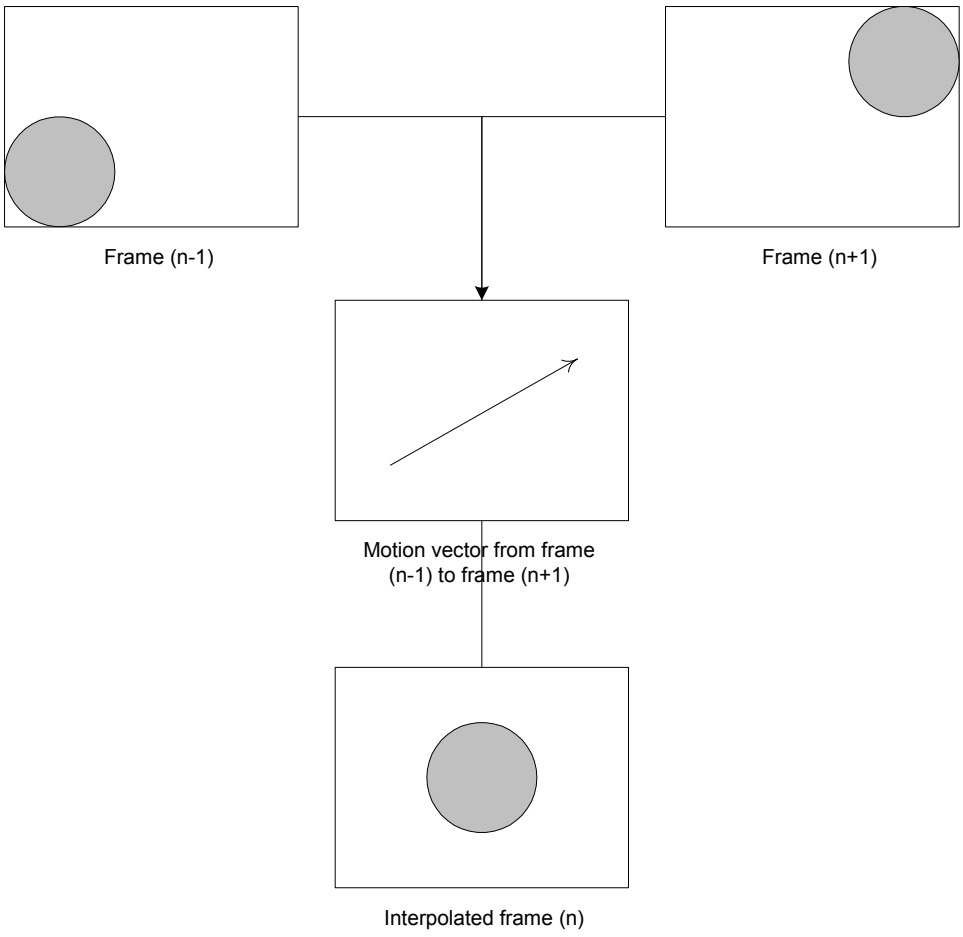

Figure 5. Illustration of the motion vectors method.

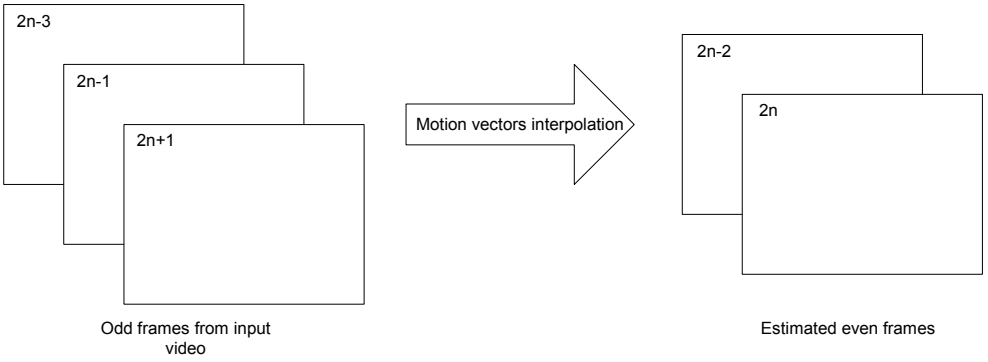

Figure 6. Estimation of even frames using odd frames.

For faster computation, it was decided to separate the odd frames from the even frames. Figure 6 shows one part of the algorithm, only the odd frames are retained. The motion vectors to go from one frame to another are generated, and, using the same scheme as illustrated in Figure 5, the inter-frames are interpolated. These interpolated frames correspond to the estimated even frames. In the same way, odd frames can be estimated by using even frames.

Once all the frames have been processed, the estimated sequence obtained can be subtracted from the input video and the resulting sequence is then forwarded to the detector for the second stage of the steganalysis process.

2.2.2. Detection Theory

Most papers on steganalysis in the open literature focus on estimating the original media. Once the estimated sequence is obtained, the difference between the cover and the estimated medias is numerically compared to a threshold. All the performance of the algorithm is focused on the signal processing part of the steganalysis. As a consequence, the detection algorithms have been undermined so far. It is believed that developing methods with a much complex detection scheme might improve the overall performance of steganalysis.

Before going on the subject, it is important to first define several notions that will be used later in the text.

- *Neyman-Pearson criterion* [77]:

  The Neyman-Pearson criterion is a notion of detector's optimality. It states that to optimize a detector, one should constrain the probability of false alarm (P_{FA}) and maximize the probability of correct detection (P_D), or minimize the probability of misdetection (P_M).

- *Asymptotic relative efficiency (ARE)* [78]:

  The ARE is also a performance criterion. It is usually used to compare the asymptotical optimality of two different statistical tests. The ARE is independent of P_{FA} and P_D and is adapted for weak-signals and large samples. It is also compatible with the Neyman-Pearson criterion.

- *Efficacy* [78]:

  The efficacy is a measure used in the ARE tests: the larger the efficacy, the more efficient the test.

Detection theory is used for decision making under uncertainty. A steganalyst is more interested in signal detection theory (SDT) whose goal is to differentiate between signal and noise. In video steganalysis the signal is going to be the watermark and the noise the original video. The uncertainty translates the unknown presence of any watermark. Typically in a signal detection problem, the noise is considered as being a random variable which changes over time. The signal however is usually considered constant. However, in steganography, the watermark is rarely the same from one image to another; hence it also needs to be considered as a random variable.

Both the video host and the hidden data have their own distribution and are independent from each other. Using a simple detector where the signal (hidden data) is supposed constant is not sufficient, which is why both the cover-video (noise) and the hidden data (signal) are considered to be independent random variables in the current scheme. Two hypotheses, H_0 and H_1, are derived to formulate the problem:

- $H_0: \quad Y_i = N_i$
- $H_1: \quad Y_i = N_i + \theta \cdot S_i$

Where θ is a parameter allowed to approach zero at a defined rate. It is also used to vary the intensity of the hidden watermark in the video. In the present case, N represents the host video whereas S represents the watermark. i is the index indicating the *ith* frame.

A detector is needed to take the decision on whether the surveyed video contains steganography or not. We chose a detector based on the asymptotic relative efficiency (ARE) criterion because it performs well in the case of weak signals and large samples.

To assess the problem, the general form of detector is used. The detector can be reduced into the formula:

$$\sum_{i=1}^{n} g(y_i) \underset{H_0}{\overset{H_1}{\gtrless}} T \qquad (1)$$

where:

- T is the threshold,
- H_1 hypotheses hidden content is embedded,
- H_0 hypotheses no hidden content is present,
- y_i is a sequence of input variables,

- g is a nonlinearity approximated by a polynomial: $g(y_i) = \sum_{j=0}^{M} a_j y_i^j$

The structure of the detector is described in Figure 7.

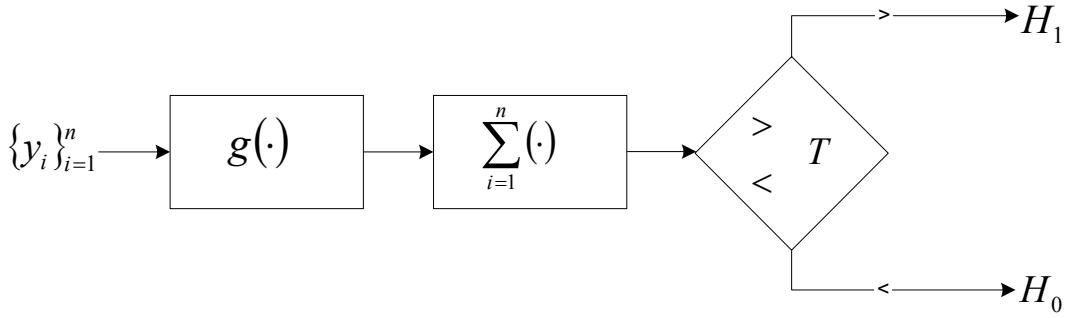

Figure 7. Structure of the detector.

The input to the detector is a sequence of variables $\{y_i\}_{i=1}^{n}$ which will later be chosen to be the sum of absolute difference between the input sequence and the estimated one. The input sequence is first transformed by the nonlinearity and then into a polynomial. Whether the output is greater or smaller than the chosen threshold, the detector takes a decision in favor of the hypodissertation H_0 or the hypodissertation H_1.

The detection stage uses several legitimate assumptions:

1) The hidden message in frame i is independent from the message in frame j when $j \neq i$
2) The hidden message is independent from the cover-video
3) The video has a defined correlation model

The steganalysis algorithm is a trained algorithm. Therefore it will be tested on two sets of video. One will contain only untouched videos; the other one will contain only videos with hidden content. This stage will allow us to determine the value of the threshold which differentiates the videos with the lowest false negative.

2.3. Algorithm

In this section, the algorithms for the motion vector computation and detection scheme used for the current study are explained in detail. The notions presented previously are directly applied to the steganalysis algorithm.

2.3.1. Motion Vectors

There exist several methods to compute the motion vectors between two frames. In the current study, a block-matching method was chosen, mostly for its simplicity of computation: first, select a block of $m \times n$ pixels in frame A centered at the position (x, y), then search in frame B the best candidate in the neighborhood of the position (x, y). The best candidate is the block in B whose sum of absolute difference (SAD) with the original block of frame A is the smallest. Because of the high number of frames per second in videos, it can be assumed that the motion from one frame to the next is very small. Hence the chosen searching neighborhood can be restricted to the adjacent pixels of the position (x, y).

The block matching method between two frames is straightforward: a block from the first frame is chosen, and then the second frame is searched until a matching block is found. In order to compare one block from the first frame to another block from the second frame, the sum of absolute differences (SAD) is computed. For a $N \times N$ block, it can be expressed as:

$$SAD_{N\times N} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left|B_{ij} - B'_{ij}\right| \tag{2}$$

Where B_{ij} is the pixel of coordinates (i, j) in the block B of the first frame, and B'_{ij} is the pixel of coordinates (i, j) in the block B' of the second frame.

Obviously, the block with the smallest SAD in the searching area of the second frame is selected as the best match for the original block. The motion vector will define the displacement from the original block to the best-matching block.

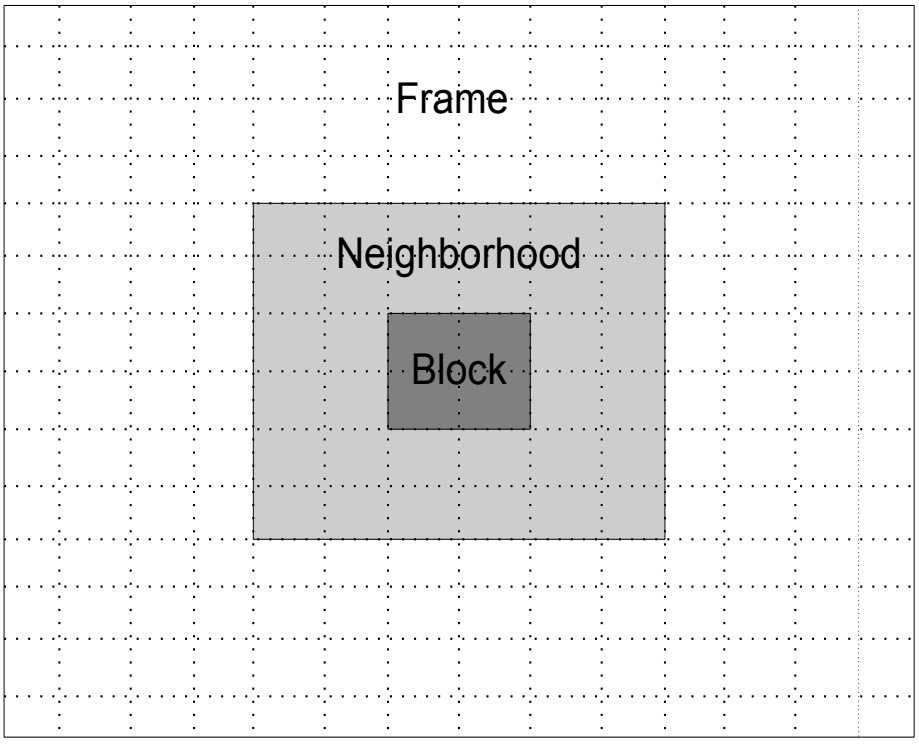

Figure 8. Searching area for best matching block.

Figure 8 shows the searching area for the best matching block for a 2-by-2 block. There are nine potential matching blocks in the neighborhood of search: the one in the exact same position as the reference block and the one surrounding it (light gray area on Figure 8). The searching area is small as the motion between successive frames is assumed to be small. The distance between the potential matching blocks and the reference block needs to be even, hence the dimensions *Vx* and *Vy* of the motion vectors need to be even. This is simply due to the fact that during the interpolation step of the algorithm the motion vectors are cut in half; as Matlab does not support half-pixel motion, an error would occur if the displacement of the block was odd. Figures 9 and 10 illustrate further the motion vector scheme used in this study by showing its application to the real sequence 'akiyo' available on Matlab.

At the top of Figure 9, two frames are shown. The difference between these frames is focused in the area of the eyes, opened in frame 36 and closed in frame 38. The background remains still, only the body of the show host is capable of moving. The next image in figure 9 is the motion vector field computed using the developed algorithm. As it can be seen, the motion vectors are non-zero only in the facial area of the person. Using interpolation, the estimated inter-frame is generated. For more details, Figure 10 points out the difference between the estimated inter-frame and the original one.

The difference is the sum of absolute differences. As expected, the difference between both images is concentrated around the eyes and face of the host.

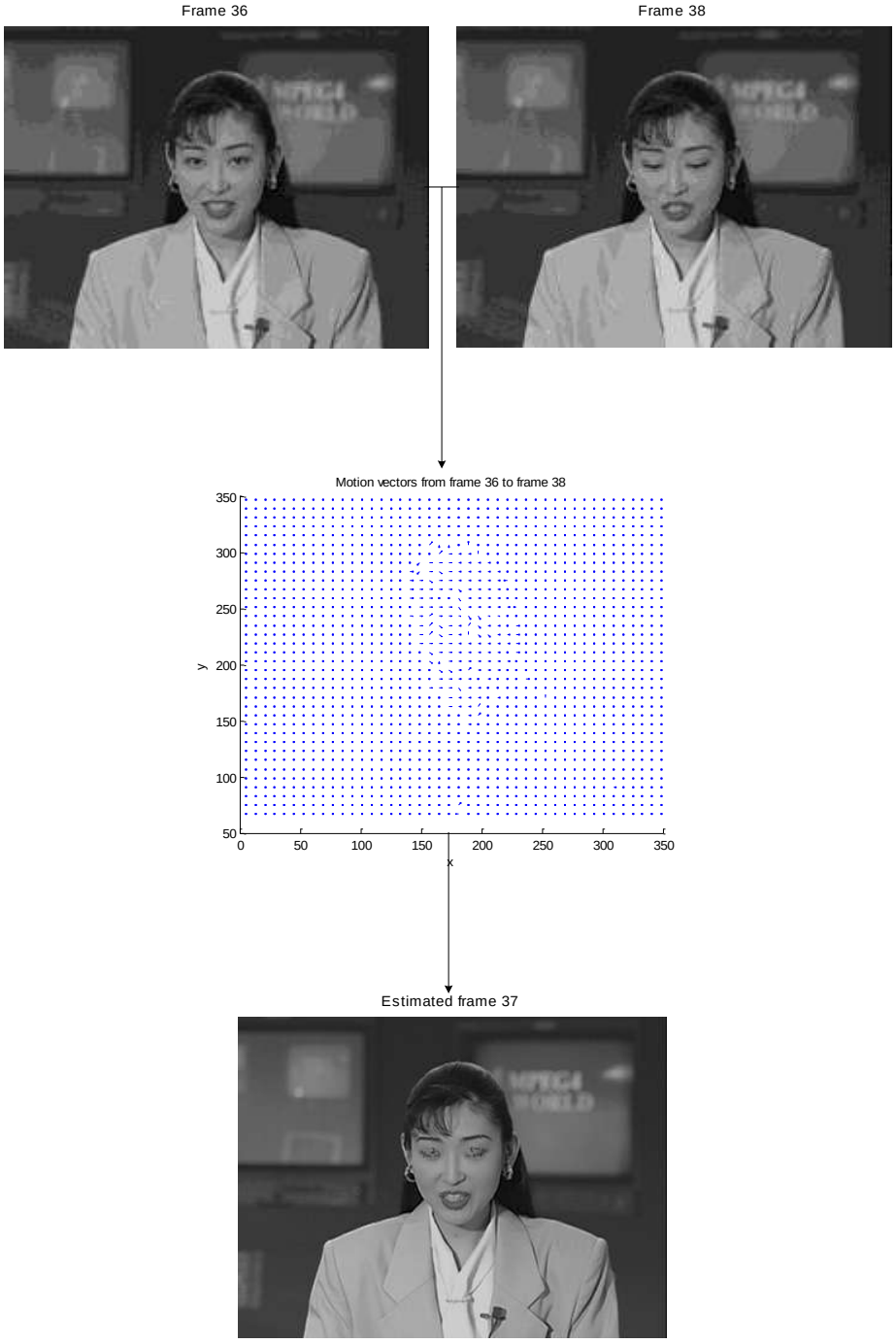

Figure 9. Application of frame estimation via motion vectors.

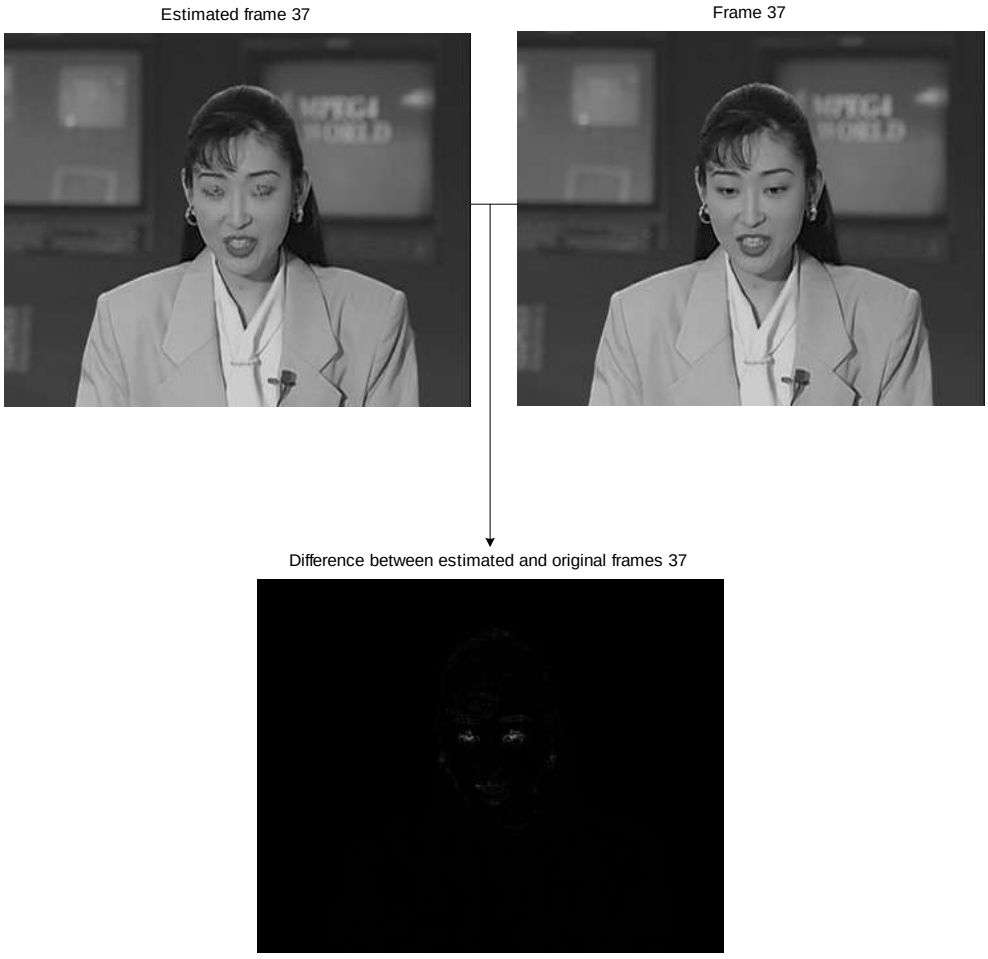

Figure 10. Difference between estimated and actual frames.

2.3.2. Detector

As mentioned earlier, the detection scheme is based on the asymptotic relative efficiency (ARE) criterion. This choice is motivated by its simplicity and its satisfying results in the case of weak signals.

2.3.2.1. Efficacy

Remember that the detector used has the form $\sum_{i=1}^{n} g(y_i) \underset{H_0}{\overset{H_1}{\gtrless}} T$ and g is the nonlinearity defined as $g(y_i) = \sum_{j=0}^{M} a_j y_i^j$. This detector becomes optimal when the efficacy $\eta(g)$ is maximized. As a consequence, when maximizing the efficacy, it is possible to determine the optimal coefficients a_j of the nonlinearity g . The efficacy $\eta(g)$ is defined as [79-80]:

$$\eta(g) = \frac{\left[\frac{\partial^2}{\partial \theta^2} \mathrm{E}_\theta \{ g(Y_i) \}|_{\theta=0} \right]^2}{\sigma_0^2(g)} \tag{3}$$

First, $\eta(g)$ needs to be expressed in terms of the coefficients a_j . After derivations, the following formula is obtained:

$$\eta(g) = \frac{\left[\sum_{i=2}^{M} a_i \cdot i \cdot (i-1) \cdot E\{ N_1^{i-2} S^2 \} \right]^2}{\sigma_0^2(g)} \tag{4}$$

With

$$\sigma_0^2(g) = \sum_{i=0}^{M}\sum_{j=0}^{M} a_i a_j E\left\{N_1^{i+j}\right\} + 2\sum_{k=1}^{\infty}\sum_{i=0}^{M}\sum_{j=0}^{M} a_i a_j E\left\{N_1^i N_{k+1}^j\right\} \tag{5}$$

In the next step, Lagrange multipliers technique is used in order to find the coefficients a_j that maximize $\eta(g)$.

2.3.2.2. Lagrange multipliers

The Lagrange multipliers have a critical role in every optimization problem [81]. They help finding the local extrema of multi-variable functions subject to a certain constraint. There exists an extrema where the derivative of the function of interest equals zero.

In the present case, it is desired to find the extrema of the efficacy. This can be done by holding the denominator constant and maximizing the numerator in the expression of the efficacy. Using the constant α, the constraint (i.e. denominator constant) can be expressed as:

$$\sigma_0^2(g) - \alpha^2 = 0 \tag{6}$$

The multi-variable function of the current problem is the numerator associated to the constraint:

$$\begin{aligned} f(a_0, a_1, ..., a_M, \lambda) &= \sum_{i=2}^{M} a_i \cdot i \cdot (i-1) \cdot E\left\{N_1^{i-2} S^2\right\} - \lambda \cdot \left(\sigma_0^2(g) - \alpha^2\right) \\ &= \sum_{i=2}^{M} \left(a_i \cdot i \cdot (i-1) \cdot E\left\{N_1^{i-2} S^2\right\}\right) - \ldots \\ &\quad \cdots \lambda \cdot \left(\sum_{i=0}^{M}\sum_{j=0}^{M} a_i a_j E\left\{N_1^{i+j}\right\} + 2\sum_{k=1}^{\infty}\sum_{i=0}^{M}\sum_{j=0}^{M} a_i a_j E\left\{N_1^i N_{k+1}^j\right\} - \alpha^2 \right) \end{aligned} \tag{7}$$

When deriving $f(\ )$ with respect to the a_i's, $0 \le i \le M$ and λ, we get:

$$n=0:\ -\lambda\cdot\left(2\cdot a_0\mathrm{E}\{1\}+2\cdot\sum_{k=1}^{M}a_k\mathrm{E}\{N_1^k\}+2\cdot\sum_{j=1}^{\infty}\left[2\cdot a_0\mathrm{E}\{1\}+\sum_{k=1}^{M}\mathrm{E}\{N_{j+1}^k\}+\sum_{k=1}^{M}\mathrm{E}\{N_1^k\}\right]\right)=0$$

$$n=1:\ -\lambda\cdot\left[2\cdot a_1\mathrm{E}\{N_1^2\}+2\cdot\sum_{\substack{k=0\\k\neq 1}}^{M}a_k\mathrm{E}\{N_1^{k+1}\}+\cdots\right.$$

$$\left.\cdots 2\cdot\sum_{j=1}^{\infty}\left(2\cdot\sum_{j=1}^{\infty}N_1N_{j+1}+\sum_{\substack{k=0\\k\neq 1}}^{M}\mathrm{E}\{N_1N_{j+1}^k\}+\sum_{\substack{k=0\\k\neq 1}}^{M}\mathrm{E}\{N_1^kN_{j+1}\}\right)\right]=0 \qquad (8)$$

$$n\ge 2:\ n(n-1)\mathrm{E}\{N_1^{n-2}S^2\}-\lambda\cdot\left[2\cdot a_n\mathrm{E}\{N_1^{2n}\}+2\cdot\sum_{\substack{k=0\\k\neq n}}^{M}a_k\mathrm{E}\{N_1^{k+n}\}+\cdots\right.$$

$$\left.\cdots 2\cdot\sum_{j=1}^{\infty}\left(2\cdot\sum_{j=1}^{\infty}N_1^nN_{j+1}^n+\sum_{\substack{k=0\\k\neq n}}^{M}\mathrm{E}\{N_1^nN_{j+1}^k\}+\sum_{\substack{k=0\\k\neq n}}^{M}\mathrm{E}\{N_1^kN_{j+1}^n\}\right)\right]=0$$

This system of $n+1$ equations and $n+1$ unknowns can be represented in a matrix form (the reduced system of n equations and n unknowns is considered):

$$X\cdot A=\frac{1}{\lambda}\cdot Y \qquad (9)$$

Where X is an $n\times n$ matrix with coefficients:

$$X(i,j)=2\cdot\mathrm{E}\{N_1^{i+j}\}+2\cdot\sum_{k=1}^{\infty}\left[\mathrm{E}\{N_1^iN_{k+1}^j\}+\mathrm{E}\{N_1^jN_{k+1}^i\}\right] \qquad (10)$$

and Y is an $n\times 1$ matrix with coefficients:

$$Y(i,j) = i \cdot (i-1) \cdot \mathrm{E}\{N_1^{i-2} S^2\} \tag{11}$$

Consider for the moment that the steganographic content is independent from the cover-video. In this case, the coefficients of the matrix Y can be reformulated as:

$$Y(i,j) = i \cdot (i-1) \cdot \mathrm{E}\{N_1^{i-2}\} \cdot \mathrm{E}\{S^2\} \tag{12}$$

The detector needs to be as proactive as possible in order to work for a large range of steganographic methods. It means that no assumption about the distribution of the signal, i.e. the steganographic content, should be made; that is why the quantity $\mathrm{E}\{S^2\}$ is injected into the threshold T of the detector. Finally, the coefficients of the matrix Y can be simplified to:

$$Y(i,j) = i \cdot (i-1) \cdot \mathrm{E}\{N_1^{i-2}\} \tag{13}$$

The difficulty of the calculations resides in the computation of the moments elevated at different powers $\mathrm{E}\{N_k^n N_l^m\}$. Without prior knowledge of the distribution of the noise, or host-video, this computation will prove to be quite unfeasible. Therefore a certain distribution model is assumed before continuing the derivation. The assumed distribution needs to have a correlation that fits reality. It means that the correlation model associated with the selected distribution should have the curve shown in Figure 11.

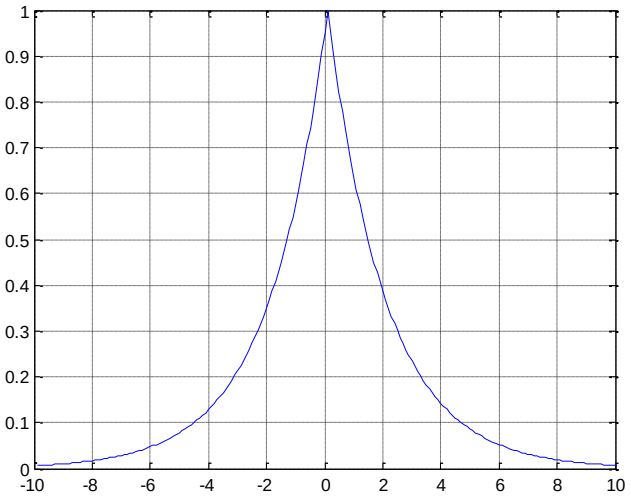

Figure 11. Typical correlation model wanted.

The distribution shown is Figure 11 indicates that the farther from the analyzed frame one gets, the less influence it will have on that frame. This obviously makes sense, which is why that is the correlation model wanted.

Two different distributions are considered in this study: the Gaussian, or normal, distribution and also the Gauss-Markov distribution.

2.3.2.3. Gaussian distribution

The Gaussian distribution is probably one of the most commonly used distributions. Its correlation model fits the reality as can be seen in Figure 12. The shape of the correlation can change depending on the value of the correlation coefficient ρ.

The correlation matrix chosen has the following form:

$$C = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^n \\ \rho & 1 & \ddots & \ddots & \vdots \\ \rho^2 & \ddots & \ddots & \ddots & \rho^2 \\ \vdots & \ddots & \ddots & 1 & \rho \\ \rho^n & \cdots & \rho^2 & \rho & 1 \end{bmatrix} \tag{14}$$

ρ being less than 1, this correlation matrix reflects the fact that the farther one gets from the reference frame, the less correlated it is. Using this matrix, the correlation model for the Gaussian case is plotted in order to verify that it matches with the theoretical correlation model desired shown in Figure 13 as well as with the correlation model of real sequences of images. As an example, the correlation model of the sequence 'paris' in Matlab is derived; Figure 12 shows only the first frame of the sequence. The correlation between all the frames is plotted in Figure 13.

Figure 12. Frame from sequence 'Paris'.

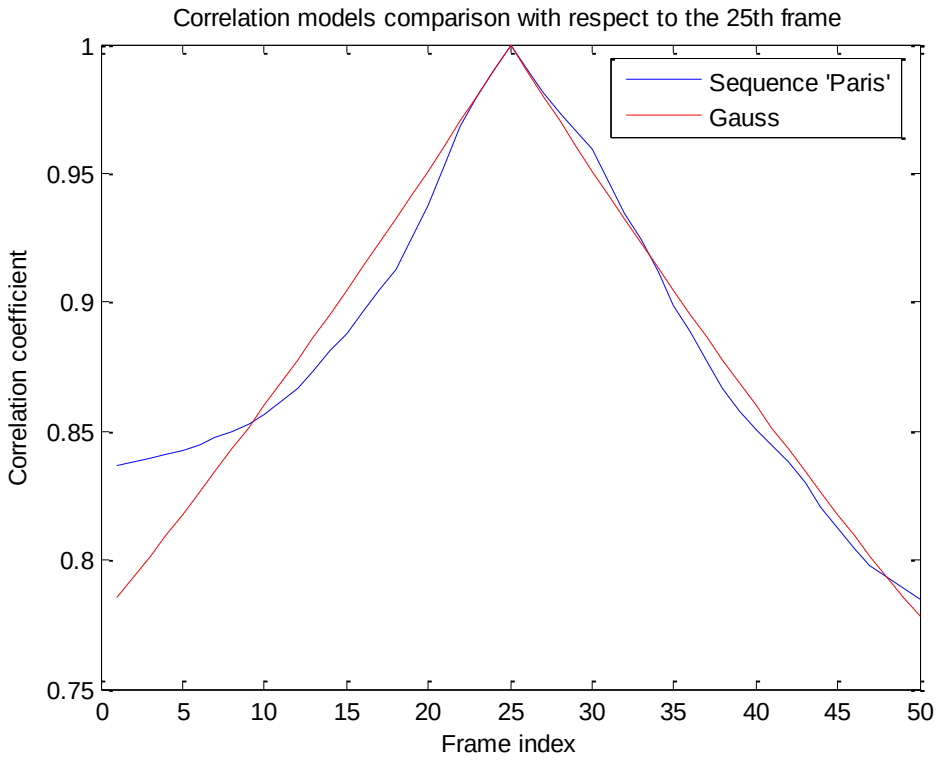

Figure 13. Gaussian correlation model.

The correlation models in Figure 13 have similar shapes; hence the choice of the Gaussian distribution to model this particular video can be validated. The correlation models of other sequences were tested in the same way and the Gaussian model was concluded to be valid for videos in general.

The computation of the moments $\mathrm{E}\{X^n Y^m\}$ can be done by using the joint characteristic function of two jointly Gaussian random variables.

$$\Phi_{XY}(u,v) = \exp\left(j(\mu_X \cdot u + \mu_Y \cdot v) - \frac{1}{2}\left(u^2 + 2\rho \cdot u \cdot v + v^2\right)\right) \qquad (15)$$

Or, when assuming zero mean and variance 1:

$$\Phi_{XY}(u,v) = \exp\left(-\frac{1}{2}\left(u^2 + 2\rho \cdot u \cdot v + v^2\right)\right) \tag{16}$$

The moments $\mathrm{E}\{X^n Y^m\}$ are expressed in the following equation:

$$\mathrm{E}\{X^n Y^m\} = -j^{n+m} \cdot \frac{\partial^{n+m} \Phi_{XY}(u,v)}{\partial u^n \partial v^m} \tag{17}$$

2.3.2.4. Gauss-Markov distribution

The Gauss-Markov noise process is solution of the equation

$$N_n = e^{-a} \cdot N_{n-1} + G_n \tag{18}$$

where $a \in \Re_+$ and $\{G_n\}_{n=1}^{\infty}$ is the sequence of iid Gaussian random variables $\sim \aleph\left(0,1-e^{-2a}\right)$. It is assumed to be zero-mean.

Same as in the simply Gaussian case, both correlation models shown in Figure 14 have the similar shape which validates the choice of the Gauss-Markov model.

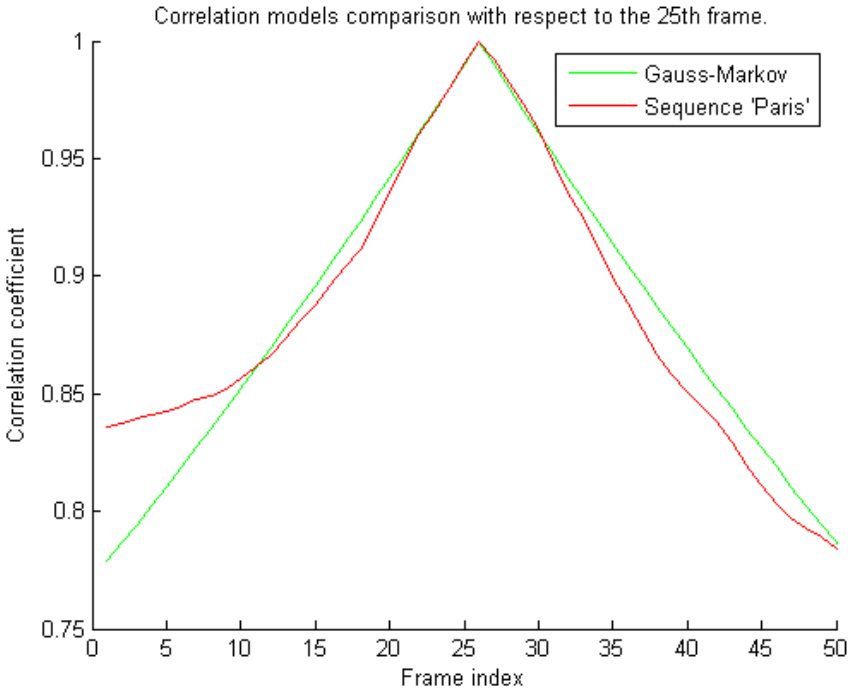

Figure 14. Gauss-Markov correlation model.

This choice of distribution is motivated principally by the practical resulting correlation model. It also gives a more workable and more general distribution compared with the commonly used simply Gaussian or Markov models, and with simpler resulting computations as will be proved later on. Of course, the legitimacy of this choice of distribution needs to be checked when the final test results are available.

With the Gauss-Markov model and supposing that $N_0 \sim 0$, the previous equation can be rewritten as:

$$N_i = \sum_{j=1}^{i} e^{-(i-j)a} \cdot G_j \ , \ i = 1,2,\ldots \tag{19}$$

Then, the computation of $N_k^n N_l^m$ simplifies to:

$$N_k^n N_l^m = \left(\sum_{j=1}^{k} e^{-(k-j)a} \cdot G_j \right)^n \cdot \left(\sum_{i=1}^{l} e^{-(l-i)a} \cdot G_i \right)^m \tag{20}$$

After calculations and using the multinomial theorem, the previous equation becomes:

$$N_k^n N_l^m = C \cdot \left\{ \sum_{\sum_{j=1}^{k} n_j = n} \left(\prod_{j=1}^{k} \frac{\left(e^{ja} \cdot G_j\right)^{n_j}}{n_j!} \right) \right\} \cdot \left\{ \sum_{\sum_{i=1}^{l} m_i = m} \left(\prod_{i=1}^{l} \frac{\left(e^{ia} \cdot G_i\right)^{m_i}}{m_i!} \right) \right\} \tag{21}$$

Where C is a constant equal to $e^{-a(kn+lm)} \cdot n!m!$.

The previous equation involves the finding of the sets of k and l coefficients that add up to n and m respectively. These sets are computed using Programs 4 and 5 in the Appendices.

$\{G_n\}_{n=1}^{\infty}$ being *iid* Gaussian, the computation of the expectation of $N_k^n N_l^m$ can be simplified since only the sums and products of the i^{th} moments of Gaussian processes will be involved. For example, for $(k,l,m,n) = (1,2,2,3)$, we get the moment:

$$\mathrm{E}\{N_1^3 N_2^2\} = e^{-2a}\mathrm{E}\{G_1^5\} + \mathrm{E}\{G_1^3\}\mathrm{E}\{G_2^2\} + 2e^{-a}\mathrm{E}\{G_1^4\}\mathrm{E}\{G_2\} \tag{22}$$

These moments can be directly expressed using the variance of the Gaussian sequence defined earlier as $1 - e^{-2a}$.

Recall that the moment-generating function of a Gaussian distribution with mean μ and variance σ^2 is [82]:

$$M(t) = \left\langle e^{t \cdot x} \right\rangle = \int_{-\infty}^{+\infty} \frac{e^{t \cdot x}}{\sigma\sqrt{2\pi}} e^{-(x-\mu)/2\sigma^2} dx$$
$$= e^{\mu \cdot t + \sigma^2 \cdot t^2 / 2} \tag{23}$$

However in our scenario, the mean of the Gaussian distribution is zero. Hence, the previous relation simplifies to:

$$M(t) = e^{\sigma^2 \cdot t^2 / 2} \tag{24}$$

When this function is derived, each n^{th} derivative corresponds to the n^{th} raw-moment. Therefore, taking these derivatives at $t = 0$ give the wanted moments.

$$M^{(n)}(0) = \left\langle x^n \right\rangle \tag{25}$$

When computing the derivatives of the moment-generating function, one can extract a recurring formula for zero-mean Gaussian distributions:

$$M^{(n)}(0) = (n-1) \cdot \sigma \cdot M^{(n-2)}(0) \tag{26}$$

This implies that each odd derivative will be equal to 0 (because the mean is zero).

These results can now be injected in the main expression which will be solved to get the coefficients a_i's. When computed, the odd elements of the matrix A equal 0; thus the nonlinearity $g(y_i) = \sum_{j=0}^{M} a_j y_i^j$ is expected to be even symmetric. It is also expected that the nonlinearity will be similar to a quadratic. As for the Gaussian case, plotting the nonlinearity will help verify these expectations and also determine the range of trust of the detector. The range of trust includes the nonlinearity around the origin. The nonlinearity is known to be optimal in the range of trust. On the tails, the dominant

factor is too strong and overcome the other coefficients and as a result the outcome cannot be trusted.

There are three parameters involved in the computation of the coefficients a_i . These parameters are:

- a , in the exponential of the Gauss-Markov equation,
- i , defining the length of the sequence to be analyzed,
- k , being the number of samples or frames included in the computation of the moments.

These three parameters can all be triggered to change the range of trust of the nonlinearity g .

Figure 15 shows graphically the influence of the three parameters mentioned above. Only one parameter varies in each case, the others being held constant. The upper-left corner graph represents the reference case with $a = 0.05,\ i = 5$ and $k = 5$. The other graphs show what happens when one parameter varies from its reference value. In the upper-right corner a goes from 0.05 to 0.1. In the lower-left corner, i goes from 5 to 10. And in the lower-right corner, k goes from 5 to 9.

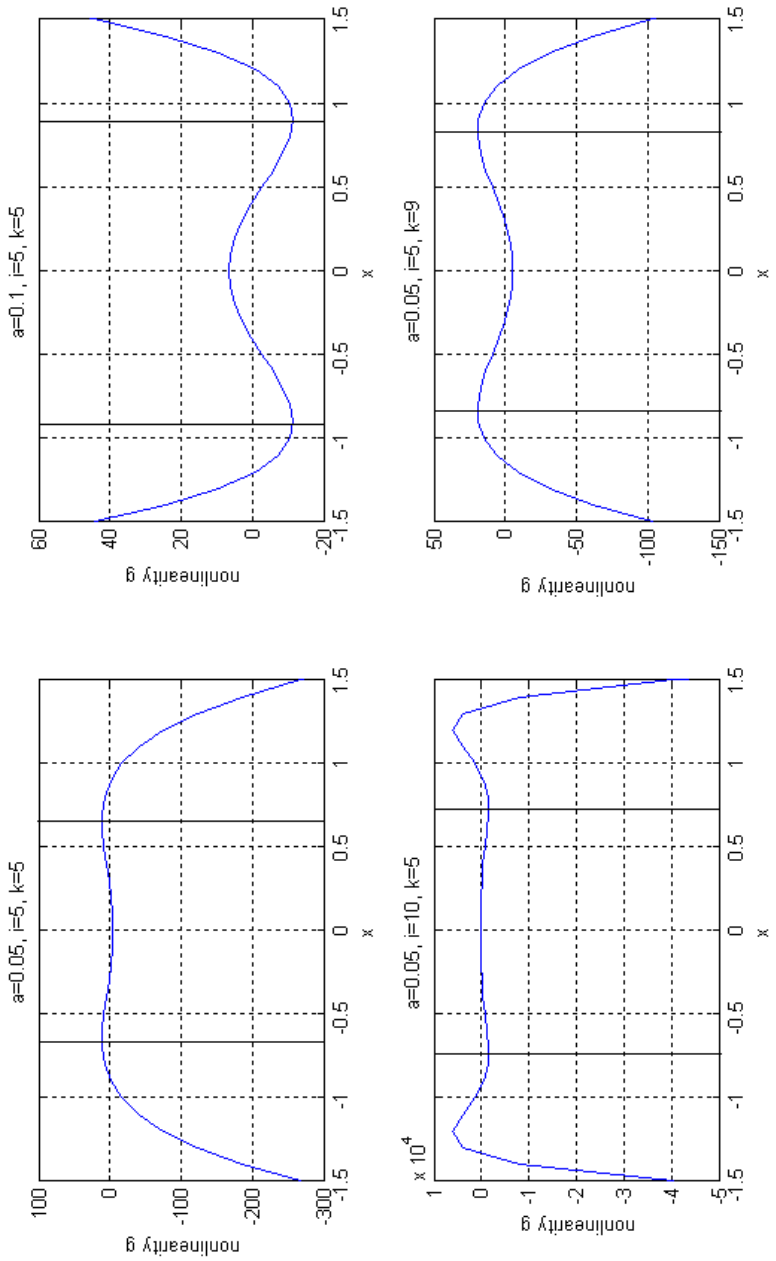

Figure 15. Parameters' influence on nonlinearity.

The results show the following tendencies:

- as a grows bigger, the range of trust becomes larger,

- as i grows bigger, the range of trust becomes slightly larger but odd shape appears when k is less than i,
- as k grows bigger, the range of trust becomes larger.

However, since a is chosen so that the correlation model issued from the Gauss-Markov distribution is practical, it cannot vary dramatically. As for i, it depends on the length of the sequence to be analyzed and the nonlinearity's shape is not satisfying when i gets bigger, it does not make much sense to make it vary. k, on the other hand, can be seen as an optional parameter and does not harm the detector when it varies. The added performance when k gets bigger also confirms that the ARE detector performs well for large samples. There are only two possibilities left to make the detector perform better, varying k or scaling the output of the SAD so that it fits in the range of trust of the nonlinearity. As making k vary induces more complexity in term of algorithm computation, scaling the output of the SAD seems to be the best solution to choose. The values chosen for a, i, and k are respectively 0.1, 5 and 9. The coefficients for a 5-frame-long sequence, with $a = 0.1$ and $k = 9$ are:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \end{pmatrix} = \begin{pmatrix} 8.2643 & 0 & -51.729 & 0 & 23.27 \end{pmatrix} \qquad (27)$$

2.3. Tests

In order to verify the efficiency of the proposed scheme, a series of tests has been performed. Three distinctive tests will be presented here.

2.3.1. Test 1 - Time Complexity

The first test consists in time-wise complexity tests. The length of the $a_i's$ vector has direct impact on the number of frames to be analyzed at each passing into the detector. Even though analyzing five or ten frames at a time eventually results in the same output, one might prefer to process more frames at each passing into the detector. The results have been obtained with a Pentium 4 computer running at 2.80GHz with 1Go of DDR.

2.3.1.1. Gaussian distribution

For the simply Gaussian distribution, the results of the time-complexity test are regrouped in Table 1. To have a better insight of these numbers, a graph is plotted representing how the size of the vector of $a_i's$ influences the time of computation.

Table 1. Time complexity table for Gaussian case.

Size of a_i 's vector	5	8	10	12	15
Computation time (s)	75	267.687	424.187	639.891	1127.813

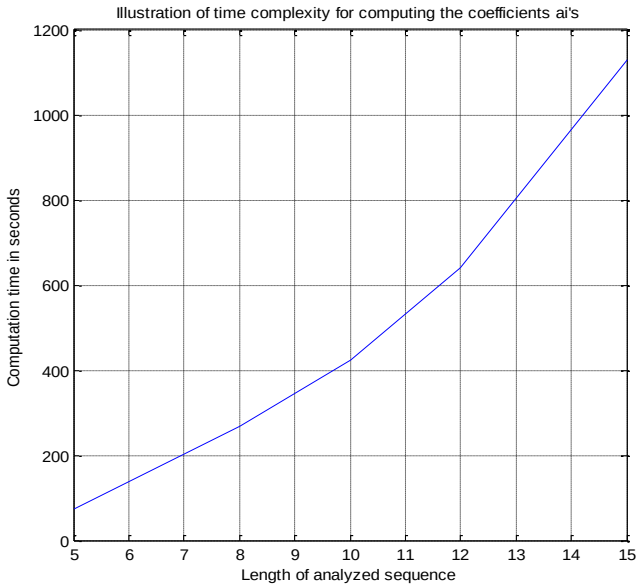

Figure 16. Time complexity for Gaussian case.

Figure 16 shows the strong increase of the time complexity when the number of coefficients $a_i's$, i.e. the number of analyzed frames during each passing into the detector, increases.

2.3.1.2. Gauss-Markov distribution

The time needed to compute the $a_i's$ vector in the Gauss-Markov case is generated and results are shown in Table 2 and Figure 17:

Table 2. Time complexity table for Gauss-Markov case.

Size of a_i 's vector	5	8	10	12
Computation time (s)	1716.078	13383.313	67951.453	319105.65

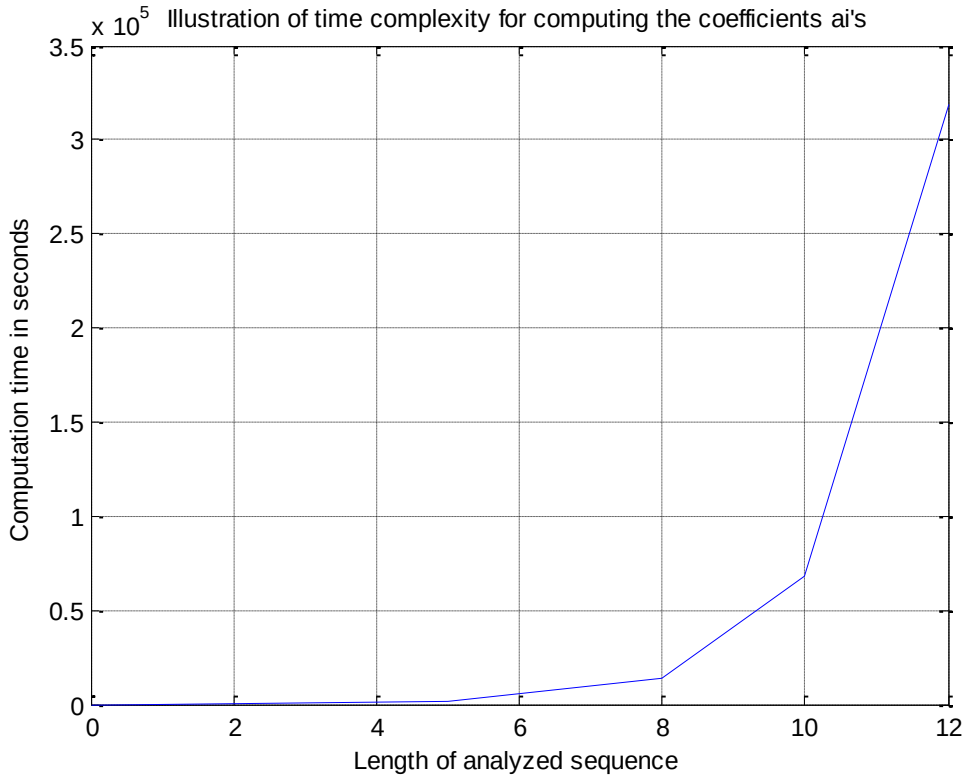

Figure 17. Time complexity for Gaussian-Markov case.

Both show the very strong increase in the computation time for Gauss-Markov distributions. Although faster computations could be obtained with faster computers, the ratio between the size of the analyzed data is likely to remain the same, i.e. it will take 40 times longer to compute the coefficients for a 10-frame-long analysis that for a 5-frame-long one.

Knowing that the nonlinearity g looks actually better for smaller vector size, it seems unreasonable to increase the number of coefficients too high. However, the coefficients a_i 's are computed once and for all, so it is up to the user to increase the number of a_i 's.

It should also be noticed that a smaller number of coefficients a_i 's can be used to identify the corrupted frames. For example, if the presence of steganography simply needs to be detected, then analyzing the input video five frames at a time can induce a gain of time. As soon as hidden content is detected, the steganalysis can be stopped and the input video is flagged. The rest of the video does not need to be processed as steganography has already been found. In another example, if the user is trying to detect steganography and locate it, then analyzing the input video five frames at a time can also prove to be a legitimate choice. If steganography is detected on one passing of the detector, then the user knows that there is steganographic content in the five frames that have just been analyzed. Taking these advantages into account, the next tests are conducted five frames at a time.

2.3.2. Test 2 - General Performance - Threshold

These tests consist in applying the steganalysis scheme to various videos randomly possessing hidden content. The watermarked sequences are created by adding some noise to a non-watermarked sequence. The results are then differentiated into four categories depending on whether the secret content was actually hidden in the video and whether the video was flagged as hiding steganography. These categories are represented in Table 3. Because of the lower performance of the detector when the number of frames analyzed gets bigger (other parameters held constant), it is decided to perform a 5 frame-long analysis and work step by step on the whole video. This also allows identifying the presence or absence of steganography at every stage, meaning that as soon as steganography is detected, the suspicious video can be flagged as being corrupted and the rest of the video does not need to pass through the detector, hence generating a considerable gain of time. Also the computation of the coefficients a_i for a 5 frame-long analysis is computationally less complex. The goal for these tests is to get a low rate of false positives and also a low rate of false negatives.

Table 3. Test outputs categories.

	Steganography detected	Steganography non-detected
Steganography present	HIT	MISS
Steganography absent	FALSE DETECTION	CORRECT REJECTION

First, to give an idea on the performance of the detector itself, some figures are plotted to illustrate the differences of the steganalysis scheme with or without detector.

Figures 18 and 19 show some outputs for the Gaussian case and Gauss-Markov case respectively. The first graph on each figure represents the output of the steganalysis if the detector was not used. This proves that the signal processing part of the steganalytic scheme performs well. The second graph shows the output of the steganalysis with the detector.

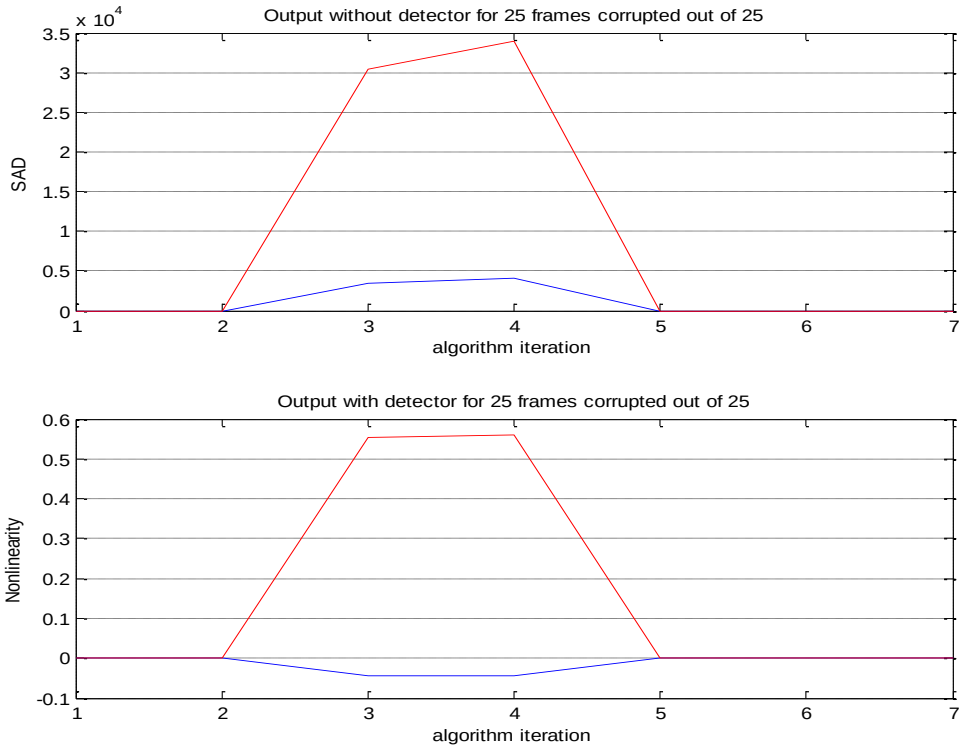

Figure 18. Examples of outputs without or with detector for Gaussian.

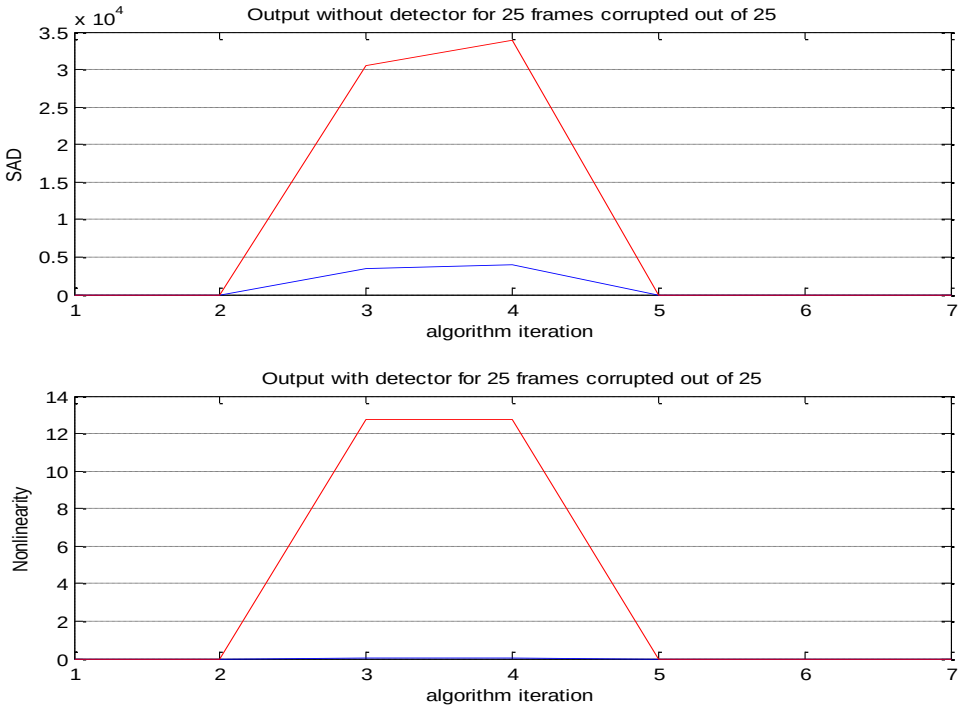

Figure 19. Examples of outputs without or with detector for Gauss-Markov.

The distinction between watermarked and non-watermarked videos without detector might not be easy to make, especially as the number of corrupted frames decreases. However, the output of the detector shows a better differentiation between both types of videos as it pushes the plot towards zero in the case of no steganography. Also in the Gaussian case, the difference in absolute value between the output of the steganalysis with or without detector appears to be very small. Therefore, the decision is taken not to pursue tests with the Gaussian case.

Before conducting the final tests, the algorithm needs to be trained in order to find the threshold which could best differentiate between video with and without steganography. The threshold is also chosen so that the rate of "miss" is minimized. For this, several videos were utilized, some with steganographic content, others without. The output of the detector is plotted for these different videos and a decision on the best threshold is

made. The difference between the two cases (with or without steganography) is quite obvious when looking at the graphs in the previous figures.

The theory in the field of Detection states that the threshold should be a constant fixed for every videos which can be seen as different realizations of the same signal. However, after observing the behavior of the output of the detector for various videos, it is to be noticed that depending on the video, the scale of the output varies between values less than 1 to values more than 1000. Therefore it seems improbable that a constant threshold could give satisfying results. This can only mean that the threshold depends on the video characteristics. The correlation between the frames of the video appears to be one of these characteristics: the less correlated two successive frames, the more different two successive estimated frames. In addition, the standard deviation of the video is to be considered: the greater the standard deviation, the noisier the frames, in which case the frames have a greater probability to be corrupted by a watermark. One other parameter that will influence the threshold is the size of the $a_i's$ vector, i.e. number of frames analyzed by each passing into the detector, because the scale of the outputs from the detector is directly linked to this parameter.

All considerations made, a formula has been developed taking into account the correlation between the frames of the video, the size and the standard deviation of the frames, and finally the number of frames analyzed at the same time by the detector. In the present study, the formula works the best is found to be the following:

$$T = \left[C \cdot \left(StD(F_1) \right) \cdot \left(Corr(F_1, F_2) \right) \cdot \frac{(x \cdot y)}{(288 \cdot 352)} \right]^2 \cdot \left(\frac{N}{5} \right) \qquad (28)$$

With:

- C a constant that can vary depending on the application requirements concerning the rates of hit, miss, false alarm and correct rejection,
- $StD(F_1)$ the standard deviation of the frame F_1 in the video,

- $Corr(F_1, F_2)$ the correlation between the successive frames F_1 and F_2 in the video,
- $\frac{(x \cdot y)}{(288 \cdot 352)}$ the size of the frames normalized by the size of the frames (x, y) used during the setting of the threshold, i.e., $(x_{setting}, y_{setting}) = (288,352)$,
- $\left(\frac{N}{5}\right)$ the number of frames analyzed simultaneously by the detector N normalized by the number used during the setting of the threshold, i.e., $N_{setting} = 5$.

Using this threshold, the final tests are executed. Several videos, with or without watermark, are imported into the steganalytic algorithm. The motion vectors are computed and the estimated sequence is examined by the detector. This latter will decide whether the corruption of the media occurs based on the value of the threshold.

These tests consist in 28 different sequences of 25 frames and are conducted with a 5 frame-analysis during each passing into the detector $(N_{setting} = 5)$. From each of these 28 uncorrupted sequences, 26 new sequences are created, each possessing respectively from 0 to 25 corrupted frames. Therefore 728 sequences have been used for these tests. The varying parameter is the constant C from the threshold. The detector ROC curve is plotted in Figure 20.

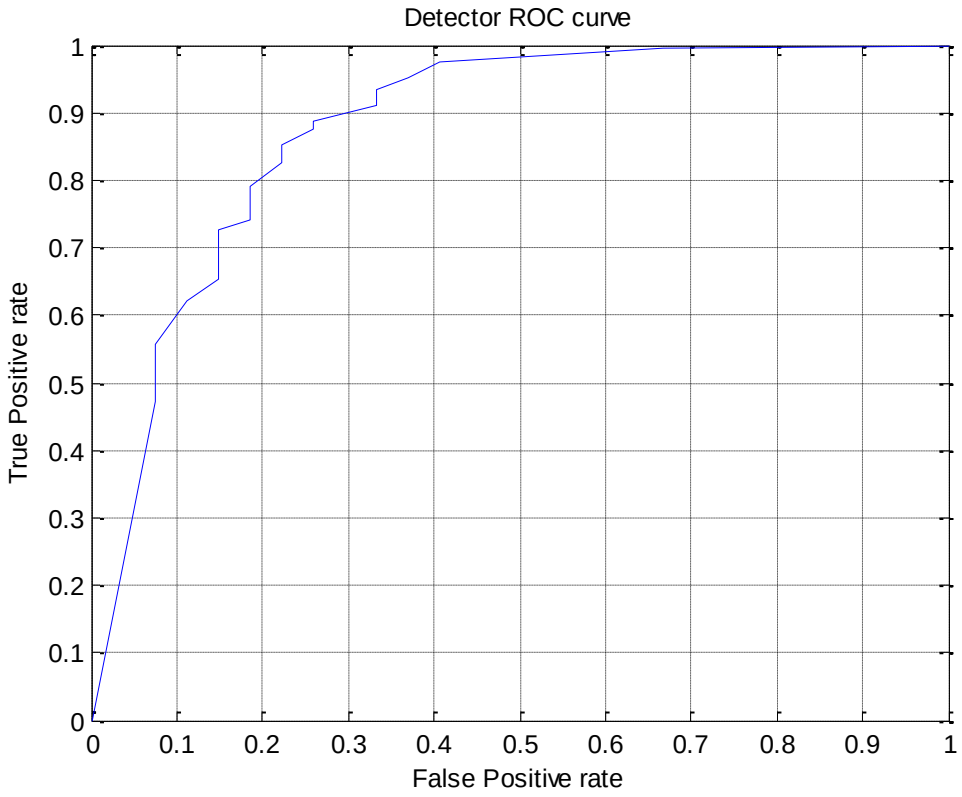

Figure 20. Detector ROC curve.

The ROC curve in Figure 20 is a witness of the performance of the steganalysis. The detector would be useless if the ROC curve would have been close to a straight line going from the coordinates $(0,0)$ to $(1,1)$. From this ROC curve, it can be concluded that the steganalysis developed offers satisfying results. Depending on the needs of the user, the algorithm can achieve a 80% of true positive and 18% of false positive, or a 94% of true positive and 33% of false positive for example.

The final results for *C=100* are grouped in Table 4 with the percentage of miss, hit, false detection and correct rejection.

Table 4. Test results, in percentage, for C=100.

	Steganography detected	Steganography non-detected
Steganography present	93.48	6.52
Steganography absent	33.33	66.67

2.3.3. Test 3 - Performance vs. Number of Frames Corrupted

Furthermore, additional tests are executed to compare the efficiency of the detection depending on the number of frames that are corrupted by steganography.

As an example, Figure 21 shows the difference in the output of the detector when, out of a 25 frame sequence, only 4 frames are corrupted (first graph), and when every frame is corrupted (second graph). As the number of corrupted frame decreases, the plots representing the outputs of the detector with and without watermark get closer. More tests need to be processed in order to judge the efficiency of the detector with more precision.

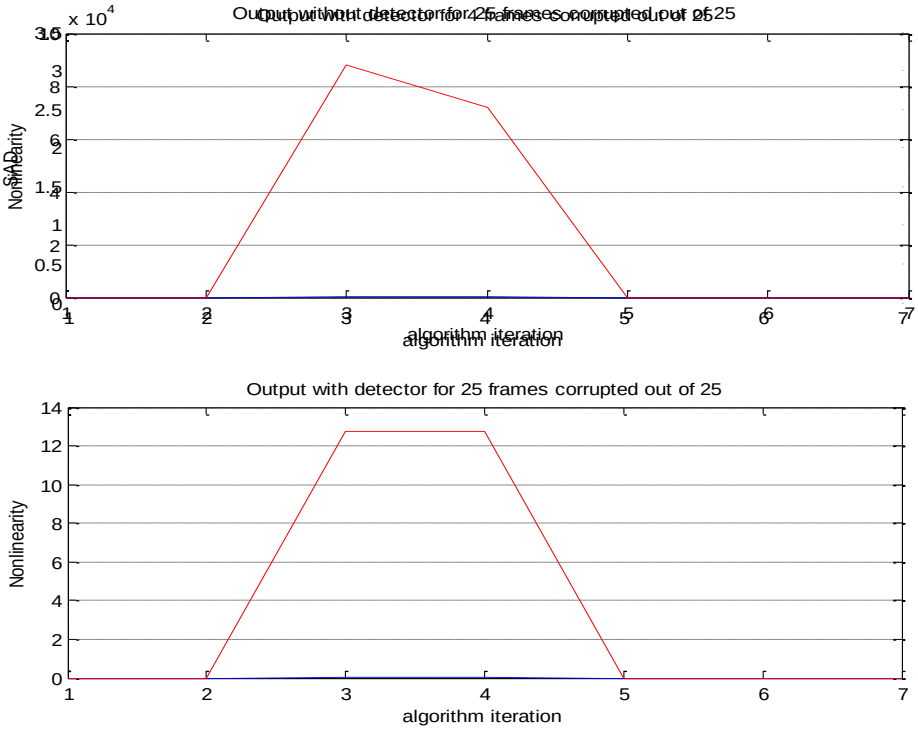

Figure 21. Examples of outputs with detector for Gauss-Markov.

In the previous section, the ROC curve (Figure 20) illustrates the results of the steganalysis regardless the number of frames corrupted by watermarks. Here, Table 5 shows more precisely the efficiency of the detector depending on the number of frames corrupted. The constant C is chosen to be 100 for all cases. Again these tests are realized over 28 sequences for every number of corrupted frames between 1 and 25.

Table 5. Test results: performance vs. number of corrupted frames for C=100.

Corrupted frames	Probability of HIT	Probability of MISS	Corrupted frames	Probability of HIT	Probability of MISS
25	1	0	12	1	0
24	1	0	11	1	0
23	1	0	10	0.963	0.037
22	1	0	9	1	0
21	1	0	8	0.926	0.074
20	1	0	7	0.8889	0.1111
19	1	0	6	0.963	0.037
18	1	0	5	0.7778	0.2222
17	1	0	4	0.7778	0.2222
16	1	0	3	0.8519	0.1481
15	1	0	2	0.6296	0.3704
14	0.963	0.037	1	0.6667	0.3333
13	0.963	0.037			

Table 5 shows that the greater the number of corrupted frames is, the better the steganalysis. These results from this table are obtained for $C = 100$ which, from Table 4, leads to a 33.33% rate of false positive. However the value of the parameter C can be increased in order to reach a smaller false positive rate. In case where at least 80% of

the sequences are corrupted, a smaller false alarm rate and a still very high true positive rate can be achieved. The ROC curve when at least 20 out of the 25 frames of the sequences are corrupted can be found in Figure 22.

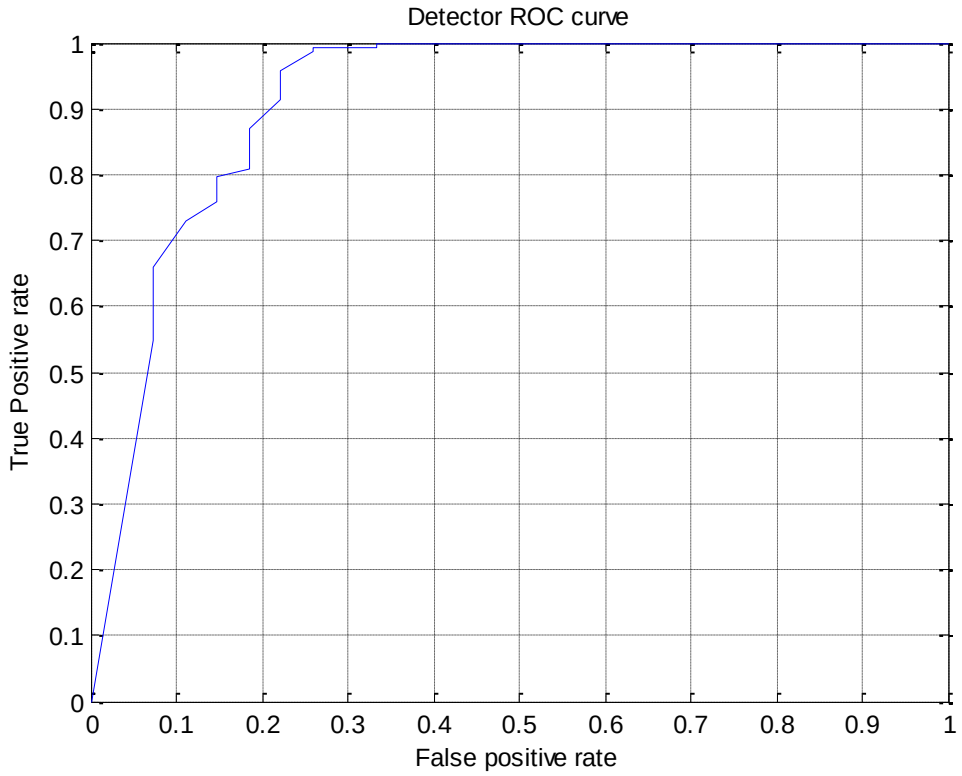

Figure 22.Detector ROC curve for at least 20 corrupted frames out of 25.

Clearly, although the algorithm performs rather well with a subset of the sequence being corrupted, the higher the number of watermarked frame, the better the steganalysis.

2.4. Comparison with Other Algorithms

For comparison, we chose two other works published in [83-84]. We decided to use these two studies because [83] was the first work on video steganalysis and [84], published in 2009, is similar to MoViSteg as it also uses motion vectors to create a

cover-video estimate. Both solutions, as well as MoViSteg, are used to defeat spread-spectrum steganography.

Budhia's algorithm uses a collusion technique and a kNN classifier in order to decide on the presence of steganography. The collusion technique consider a sliding a window over the video so that a total of L frames are processed at the same time (processing the whole video at once proves to be laborious a project without providing much performance gain). The watermarks are assumed to exist in each and every frame and follow a zero-mean Gaussian distribution. Over the length of the window, which was optimized to L=11, the frames are averaged so that effect of the zero-mean watermarks gets nullified and an estimate of the cover-media is derived. By subtracting this estimated cover-video to the one originally received, the author expects to retrieve frames containing mainly the watermark. A kNN classifier is then used to separate corrupted sequences from untainted ones using three image parameters: kurtosis, entropy and 25^{th} percentile.

Kancherla's algorithm, as most steganalytic solutions, also has two parties. The first one aims at estimating the cover-media whereas the second one makes the decision on the presence of steganography. In order to estimate the cover-video from a potentially corrupted sequence, the author chose the same solution as the one used for MoViSteg: in order to estimate frame n, the motion vectors from frame n-1 to frame n+1 are derived and the estimate of frame n is obtained via interpolation, using half the motion vectors derived in the process. The detector making the decision on steganography is however very different. Kancherla uses the merged Discrete Cosine Features (DCT) and Markov features of the watermarks' estimates and a SVM classifier to make a final decision. Using the DCT and Markov features is an idea developed by Fridrich et al. [85] towards still image steganalysis in JPEG files. Kancherla borrowed this study and applied it to the field of video steganalysis.

Both algorithms have been implemented in the Matlab environment and performance results have been obtained on the same sequences used to evaluate MoViSteg. We

consider sequences where all frames have been compromised and draw the ROC curves for these algorithms (Figure 23) and to compare them to MoViSteg's.

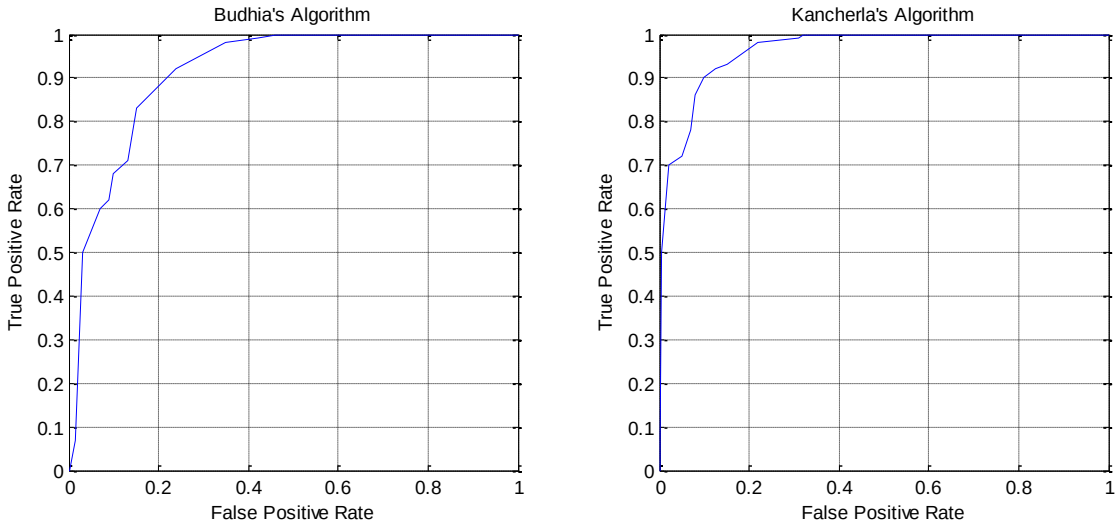

Figure 23. ROC curves for steganalytic algorithms in [83] (left) and [84] (right).

Budhia's work shows promising results and is quite comparable with MoViSteg in its performance. However, it assumes that all frames in the video have to contain a zero mean Gaussian watermark, present in every pixel, whereas MoViSteg is designed to detect steganography even when a portion of the video is corrupted as results have shown. Also, [83] is very dependent on how fast the content of the video changes since it only uses averaging. For example, if the video is that of a high speed train, the average over the sliding window could create an estimate of the cover-frame with very little similarities with the actual cover-frame due to the train 'shadows' expected from all contributing frames. Although MoviSteg is also dependent on the same factor to some extent, using motion vectors and interpolation provide more leniency.

Kancherla, although the performance derived in [84] could not be reiterated here, also shows great results. The derivation of the estimate cover-media is the same as the one

done in MoViSteg so the difference resides in the detection algorithm leading to the decision on steganography. While it is a proven efficient algorithm that Kancherla has borrowed, it was derived for purely still JPEG images. That means, although the author fails to point it out, that the algorithm should target M-JPEG videos only, making it a reactive algorithm. MoViSteg only assumes a distribution model for the video based on visual content only, no video format is ever assumed in the evaluation of MoViSteg's performance. We also believe the properties of videos are not well-utilized in this algorithm. Although motion vectors are used to estimate the cover-media, the steganalys (detection) is for still images and has to be applied frame-by-frame which makes it a time consuming and under-optimized solution to the problem of video steganalysis.

2.5. Conclusion

In the present study, we proposed a steganalysis scheme consisting in two distinct stages. The first stage is the use of a signal processing algorithm. It aims at emphasizing the presence of a watermark in the sequence using a motion estimation scheme. The second stage is the formulation of a decision theory algorithm which takes the final decision on whether steganography is hidden or not in the input sequence. This algorithm uses an advanced detection scheme where the noise (video) and signal (watermark) are both considered to be random variables.

The steganalysis scheme employes very few assumptions about the watermark except that it is zero-mean. This makes the proposed steganalysis proactive hence adapted to different kinds of additive watermarking.

For a steganographic scheme to be defeated, the steganalysis must have a true positive rate greater than 50% and a false positive rate less than 50%. The steganalysis proposed in this study accomplished both requirements. It can therefore be concluded that the steganalysis is successful.

3. STEGANALYSIS IN SENSOR NETWORKS

3.1. Current Issues

It is well known that wireless visual sensor networks (WVSNs) can be used in a wide range of applications from the surveillance of potentially dangerous areas, such as war zones, to habitat monitoring including the supervision of animal territories. Because of their relatively low cost, small component size and reasonable autonomy, WVSNs enable environmental monitoring in areas considered to be hostile for direct human interaction. Given their image capturing capabilities, WVSNs can record huge volumes of data which, depending on the application, can carry private and sensitive information. For this reason security and privacy is of fundamental concern for WVSN development and use.

Much research activity has been dedicated to the investigation of security issues in the overt communication infrastructure of a WVSN. In this section, we investigate WVSN security in the context of covert communications. The usual passive and active steganalytic concepts can be used in many applications but are currently unsuitable to defend against WVSN steganography because with a passive warden, whether the steganalysis is reactive or proactive, it only identifies the presence of steganography once the cover data has been sent through the communication channel. This means that even though that communication could be flagged as contaminated thanks to the steganalyst, the exchange of information between the malicious sender and the receiver has already occurred. This is where the passive steganalyst comes short: the detection of steganography can only happen after the attacker manages his or her malicious objective. In the case of the active warden, the lossy transformation usually used by the steganalyst compromises the integrity of the cover data which, in many cases, contains critical information. In WVSN, collecting important data, corrupting the information is not an option. If the WVSN works toward tracking intruders in a zone of interest, any lossy transformation might affect the network's primary function by raising the rates of false-positive or false-negative detections. From a steganalysis perspective, it is possible that a

certain visual processing approach could overcome these two weaknesses by discouraging steganographic activity while reaching a WVSN intended overall goal. Specifically, we introduce and present the problem of preventative steganalysis. We highlight the steganographic security concerns in visual sensor networks and describe the competing goals of the steganographer (a party involved in covert communications) and the steganalyst (a party attempting to discourage covert activity through network development and operation) in order to derive system design principles to either mitigate or limit steganalytic activity in WVSNs.

3.2. Sensor Networks

A sensor network is usually composed of many interconnected nodes that relay valuable information to a base station. The nodes are responsible for collecting any data important to the network's mission. For example, the nodes might capture the surrounding temperature, humidity level and pressure for environmental purposes. We separate the network in three main components (Figure 24): the node capturing the data, in this case a camera; the relay nodes that convey the data; and the base station, where all the data is forwarded and processed.

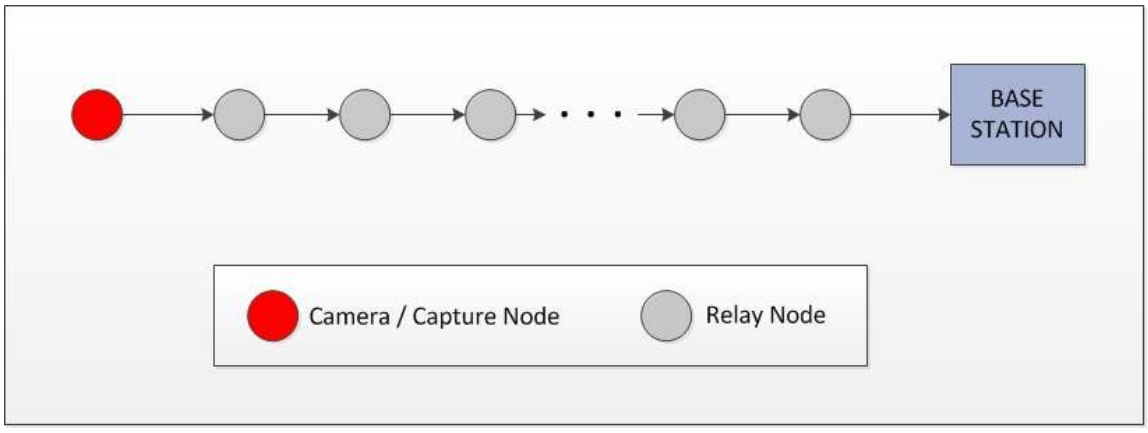

Figure 24. Components of a sensor network.

The nodes are preferably lightweight entities with very low production costs and rudimentary processing capabilities. The base station on the other hand is the brain of the network and is responsible for the majority of the processing that needs to be done. The main concern that has been raised about wireless sensor network is their apparent limited resources [86]. An overwhelming number of publications has been focused on finding deriving basic protocols such as collection of information, data forwarding, etc.. in the most energy efficient way [87-91]

In a visual sensor network the notes correspond to video cameras that are able to capture images. Such networks are useful for surveillance purposes e.g. the monitoring of a sensible area. In this type of scenario, the sensors are most likely deployed in the open, and even in hostile areas, meaning that any attacker can relatively easily access the cameras and corrupt them for his or her own malicious purposes.

3.3. Trials and Errors

3.3.1. MoViSteg

Our initial wish was to adapt the MoViSteg algorithm in order to provide a steganalytic protection to WVSN. However, the strength of MoViSteg lies on the high refreshing rate of videos so that the motion estimation based algorithm provides close estimates of interframes. Sensor networks are different. The capture of images is usually triggered by a specific event such as the presence of intruders in a monitored area. The capture usually consists of one or several photos of the monitored area when the intrusion occurs, and is by no mean a continuous video stream due to the specific and rather limited resources of the network. With such a low refreshing frame rate, the interframe estimation becomes flawed and the false positive rate, identifying the presence of watermark when there is not any, gets very high. Therefore it was concluded that MoViSteg was not an appropriate solution to the problem of steganography in sensor networks.

3.3.2. Aggressive Active Warden Steganalysis

Because the sensor network belongs to the same party as the steganalyst does, it is natural to assume that an active warden steganalysis where the steganalyst can perform various types of processing should be preferred. In that optic, we started deriving an aggressive solution that would scramble the transmitted data such as cryptography. In this case, cryptography would not be used as a solution to make the data private but as a way to introduce lossy transformation towards the annihilation of any potential watermark while keeping the original data retrievable via the decryption process. Although this method works for already compromised images (the scrambling as a lossy transformation makes the existing watermark unreadable by the targeted receiver), it does not provide any protection against future steganographic embedding. The reason why is that the watermarking does not rely on the actual data shown in the original image. As long as there is an available media, embedding can occur. And in fact, scrambling the original data has the effect of randomizing the pixel distributions and makes it easier for the attacker to hide a message without raising any flag.

One solution to this problem is to develop a Multi-Point steganalysis which consists in, not one, but a series of measures applied at every step of the transmission towards the base station, i.e. at each node. The goal here is to make any potential hidden content unreadable to the potential receiver and render the steganography obsolete.

Applying this method would require the node's capture to be scrambled at each node between the one capturing the image and the base station. Although efficient, this solution goes against the assumption that resources for sensor networks are limited. It becomes an even more difficult task to perform since it is of the utmost importance that the original data can be retrieved at the end of the transmission line since whatever decision the base station needs to make on the reading it receives depends on the actual data captured. This crucial point implies that the scrambling algorithms need to be invertible. And because several algorithms, preferably different for better steganalytic results, need to be applied for each frame captured by the network, a map of algorithms

associated with each node needs to be written and a different invertible function needs to be derived for each path leading from a capturing node to the base station. Such nomenclature adds a heavy load to the network and shortens its lifespan considerably. Therefore such a steganalytic solution needs to be avoided when possible.

3.4. Preventative Steganalysis

Traditional image steganalysis remains a difficult challenge because the steganalyst has little prior knowledge of the original cover-media. In videos however, the temporal redundancy between subsequent frames eases the job of the steganalyst who can compare close frames and decide with a low error-rate on the presence of covert communication. The sensor network environment offers video-like elements, such as the high temporal redundancy, that could be used for steganalytic purposes so that covert-communications could be "easily" identified.
It is also important to note that the network belongs to the trusted party and in that aspect helps the steganalyst. Indeed, in sensor networks, the attacker cannot choose the cover-media but is given one. It means that the attacker cannot choose an image with high embedding capacity but has to work with what is available.

Most steganalytic solutions are developed to be applied after steganography has occurred and in that sense are globally reactive to the attack. However, in large scale systems, it can be very difficult and resource-consuming to check at every step the presence of hidden messages. Therefore, the possibility for the attack to be successful is increased. For example, in a large scale network, there would be a need for each and every node to perform steganalysis which most of the time requires statistical analysis of the potentially corrupted media. Given the limited resources of networks' nodes, this solution would be highly impractical. This is the reason why a preventative solution, one that would discourage the attacker to use steganography, is needed. Our novel of a preventative steganalysis has been published on several occasions [92-94].

The goal of preventative steganalysis is to provide security by ensuring that the cover-media has statistical characteristics such that any previously hidden data has close to no chance of being undetected by a covert receiver and any possible future data to be hidden has limited opportunity to be communicated imperceptibly within that cover-media. Stated more specifically, the presence of steganography within the cover-media has a detection probability 1-ε where ε →0 thus discouraging any potential attacker from conducting data hiding.

3.5. Distributed Data Processing Schemes

One strategy to discourage steganalysis is to reduce the entropy of the cover-media. This has the effect of reducing the uncertainty of the cover information and hence detecting any statistical deviations due to data hiding. By definition, the entropy, or uncertainty, *H(X)* of a random variable *X* is a measure of its randomness and is given by [95]:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2(p(x_i)) \tag{28}$$

where $X = \{x_1, x_2, \dots, x_n\}$ and $p(x_i)$ is the probability of occurrence of x_i.

In the case of images, defining the entropy objectively is more complex because the purely statistical content of the image data does not correspond exactly to visual cues; thus entropy is considered to be an estimate of the visual uncertainty within an image. For this study, we chose a first-order estimate of the entropy which can be defined as:

$$H_{im} = N * \left[-\sum_{i=1}^{255} p(l_i) log_2(p(l_i)) \right] \tag{29}$$

where *N* is the total number of pixel in the image and $p(l_i)$ is the probability of gray level *i* to occur in the image examined. The image entropy is expressed in bits.

From the well-known *data processing inequality*, it is known that any type of processing on a data set cannot increase the entropy of the signal. The inequality states that for random variables X, Y, Z forming a Markov Chain X →Y→Z, the mutual information between X and Z is less than or equal to that of X and Y [1]:

$$I(X;Y) \geq I(X;Z) \tag{30}$$

We can rewrite this equation for the specific case of the X →X→f(X) Markov Chain where f(X) is a function of X representing the processing on X. Therefore we have:

$$\begin{aligned} & I(X;X) \geq I\big(X;f(X)\big) \\ & \Leftrightarrow H(X) - H(X|X) \geq H\big(f(X)\big) - H(f(X)|X) \\ & \Leftrightarrow H(X) \geq H(f(X)) \end{aligned} \tag{31}$$

Since $H(X|X)$ and $H(f(X)|X)$ equal zero (there is no uncertainty about X or f(X) when X is known). Therefore data processing can never increase the entropy.

Moreover if the process *f* is non-invertible (i.e. one-way) the processing strictly reduces the entropy. Most steganographic schemes embed covert data within noisy or irrelevant parts of a cover-media in order to maintain covertness both perceptually and statistically. For instance, “noisy” image characteristics often visually represent high texture areas that can be modified to embed stego-data through slight color variations; the changes can be applied to a greater extent than visually flat areas consisting of almost one color embedding significantly more covert information. It is therefore well know that by reducing the image entropy of the cover-media, the image irrelevancies or noise are reduced and the steganography becomes more difficult because the available covert communication bandwidth is reduced.

It is thus clear that one-way data processing inherent to WVSNs (for example for privacy preservation as highlighted above) will help prevent steganography. Consider

the example of a camera network for identifying the presence of an intruder in a safe area. The usual setup would be for the cameras in that safe area to forward the data to a specific base station which would, through various processes such as color conversion, contour extraction and pattern recognition, decide whether content resembling a human figure is present in the zone under surveillance. If the raw data is sent to the base station through the multihop network, there exists much room for an attacker to hide information within this network as the raw data is being communicated throughout. However if the cameras themselves directly extract the contours of the image and then directly communicate the extracted contours to the base station, it becomes more difficult for an attacker residing within the multihop network to hide information. Furthermore, if the cameras conduct all necessary processing only forwarding to the base station the decision of whether or not an intruder is present, there is close to no room for data hiding. Following the logic of this example, we assert that there is an opportunity to discourage and mitigate against steganography within a WVSN by distributing the signal processing effectively throughout the overall network taking into account practical network constraints.

Sensor networks are comprised of nodes often with limited computational and communication capabilities. Low computing power and short battery life are two of the most critical limitations of sensor nodes. In a WVSN, using cameras as sensors, these limitations become even more significant due the volume of data being acquired. Thus, we assert that when security design is concerned a simple yet efficient solution is required for WVSNs. This is especially true since information security is not the primary function of the network and is often considered a "tax" on the overall system. In that mindset, techniques to encourage security that employ existing system components and processing have the potential to provide protection without possible overhead. In this chapter, we look at how network topology and the distribution of integral processing affect steganographic security. In addition, because we do not want the steganalysis to add any unnecessary load to or interfere with the WVSN primary purpose, it is wise to first utilize the resources at hand and how they can be shaped in order to help the

steganalysis without disturbing the WVSN. In this state of mind, we choose to study how its function or objectives can be adapted for a steganalytic purpose.

Steganographic security is often not the primarily objective of a WVSN which means that in the quest for preventative steganalysis, priority must be given to essential network properties such as its application-based function and lifespan. An ultimate goal is to achieve a certain level of security without compromising the network's priorities. By distributing the processing related to the existing network's function, we aim to create a certain level of steganographic security without interfering with the WVSN goals. What a distributed approach is intended to do is break the overall WVSN objective into tasks that are then separately assigned amongst select nodes. For example, in the case of an intruder target-tracking surveillance WVSN, a goal of the network is to determine the presence of a person or an intruder in the monitored area. This goal could be divided into the following three possible steps as illustrated in Figure 25:

- Step 1. Convert the raw data into a black and white image;
- Step 2. Compute the contours of the black and white image;
- Step 3. Using one or more pattern recognition algorithms, determine the presence of an intruder.

Figure 25 also shows the respective image entropy after each processing step. It proves that processing the data greatly reduces its entropy and therefore its embedding capacity.

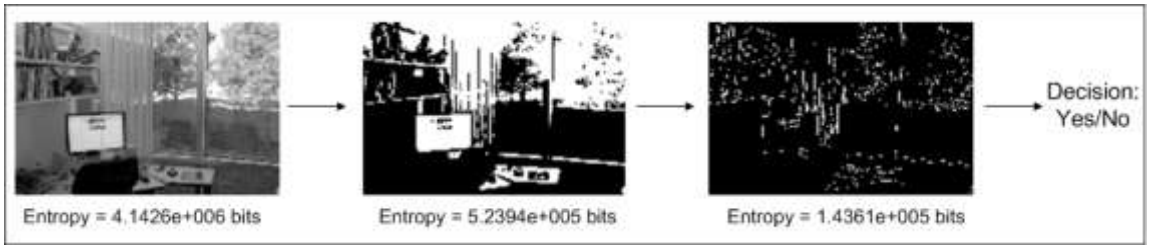

Figure 25. Illustration of image processing steps for surveillance application (source image 'office_4.jpg' from the Matlab library).

The reader should note that performing each one of these steps will eliminate part of the data that is irrelevant to the application such that the volume of cover-media that can mask the steganography is decreased. Therefore the entropy and the associated embedding capacity are also reduced. Once this embedding capacity further decreases, the data will become less attractive to the attacker for steganographic purposes. In Figure 25, the last step of the processing results in a decision on the presence of intruders that can be represented as binary data, i.e. '0' and '1' or 'yes' and 'no'. By achieving this level of data reduction, the bandwidth of effective steganography has reduced and preventative steganalysis has succeeded.

As discussed, preventative steganalysis can be achieved via breaking up network processing into multiple data processing steps and effectively distributing it. We next explore how this preventative steganalysis can be implemented in a WVSN and its associated costs. To do so, we consider three data processing architectures as applied to a variety of network configurations. The three data processing schemes are:
Central data processing: here, the base station is responsible for all of the processing; the nodes only transfer the raw data from the recording node to the base station;
Uniform distributed data processing: here, the nodes take some or all of the data processing; Each node, starting with the initial data sensing node, performs one and only one step of data processing if its battery level allows it;

Greedy distributed data processing: again, the nodes take some or all of the data processing. However, each node starting with the recording node tries to perform as many of the processing steps as its battery allows. We assume that the batteries are rechargeable after a period of time (for example, a day if solar recharging is employed) so that nodes that deplete power can be effectively revived within a period of time.
If preventative steganalysis can be achieved in theory, when it comes to WVSN, several challenges arise. Our preventative steganalytic solution requires the application of a distributed data processing approach, however, in a WVSN, by decentralizing the processing of the raw data, which takes place at the base station in most cases, the overall computational load on the network is intuitively increased and its lifespan could

shorten. Therefore, we also study the effect of providing steganographic security on the battery usage in a WVSN.

We consider a network as being comprised of a set of nodes connected through wireless communication links; two nodes are considered to be neighbors if they are one-hop (i.e., a direct communication link) apart. From the perspective of communication of sensed data from a source node to the base station we consider the information to be routed through neighboring nodes in a multi-hop fashion; along such a route a non-source node within this route usually has predecessor and successor neighbor(s) that aid in routing information from a source node to the base station. A successor is a node is defined as being one-hop further from the base station whereas a predecessor is said to be one-hop closer to the base station. Many types of node placements forming a given topology are possible. Whether the nodes are organized in a straight line or are all gathered around the base station, the network follows a communication topology. We define a termination node as one with no successors at the "edge" of the communication network. The reader should note that the term "node" does not include reference to the base station. Four common communication topologies are considered for the tests in this work:

- Horizontal topology: every node is only one-hop away from the base station; in other words, every node is a termination node.
- Fixed Tree topology: every node has only one predecessor and two successors;
- Random Tree topology: every node has only one predecessor but can have multiple successors;
- Vertical or straight line topology: every non-termination node has only one predecessor and only one successor.

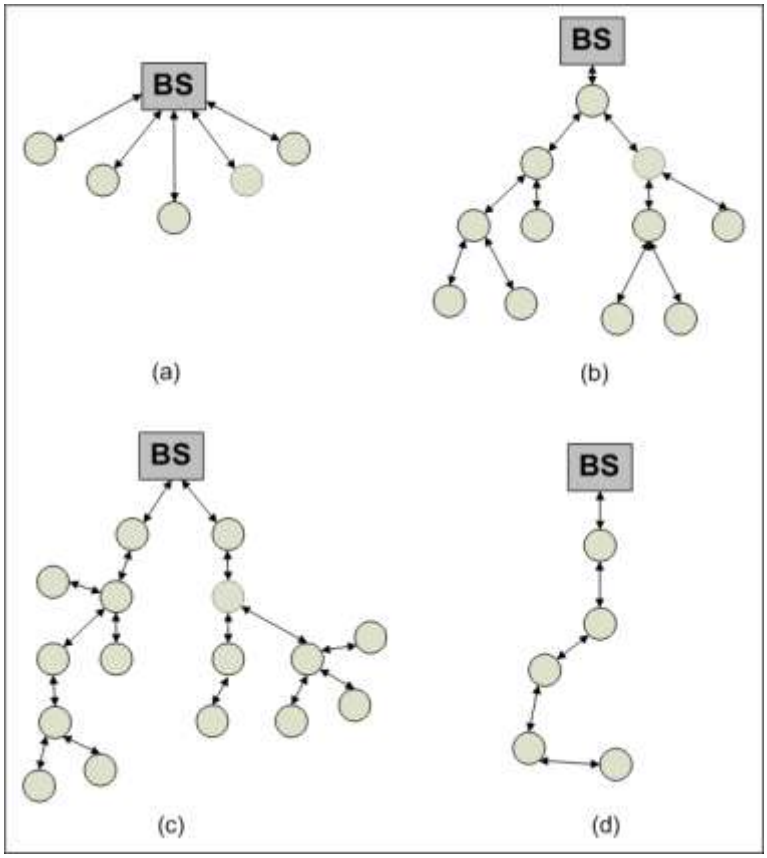

Figure 26. Examples of communication network topologies. BS represents the network base station. (a) horizontal, (b) fixed tree, (c) random tree, (d) vertical.

We chose to consider several topologies (Figure 26), which depend on the observation environment of the WVSN and on the network's application. For example, in a controlled environment the nodes can be placed in a specific pattern whereas in a hostile environment, the sensors may be randomly deployed. Also if the application requires the monitoring of a large area then a more widespread topology would be preferred whereas for specific monitoring of an airport gate for example, a horizontal topology might be better.

Intuitively, the topology of the WVSN naturally influences the life of the network. A WVSN is alive as long as it can perform its original function whereas when a WVSN cannot fulfill the goal it was built for, it is said to be dead. Because a network is but a set of nodes, its lifespan depends directly on the life of its cameras: by extension, the

network dies when its nodes die. However, all the nodes do not have to be dead in order for the WVSN to die. As expressed earlier, the network dies when it cannot perform its original application. This scenario can occur for example when all the nodes closest to the base station have depleted batteries. In such a case, even if the cameras further away are still alive, no transmission can reach the base station and therefore the network becomes useless; it has died. We define the life of the network, or lifespan, from a performance perspective as the time from the moment all the nodes are fully charged until so many of the network's nodes die that the WVSN cannot record any event anymore i.e., all paths to the base-station are broken. For example, a tree topology which presents a bottleneck close to the base station requires using the same nodes (the ones one-hop from the base station) for every data collecting process. Since these nodes will be used often, their batteries will deplete sooner and therefore the network's lifespan will be shorter compared to a more horizontal topology where there is no bottleneck at the base station.

In a similar way, it is intuitive that the network's topology can influence its steganographic security. For example, nodes spread in a wider area using a rather vertical distribution create more node links and therefore more potential breaches exploitable by an attacker. On the other hand, a shorter more horizontal network where all nodes would only be one-hop away from the base station, presents little interest and little opportunity for an attacker. The WVSN's topology is therefore an important parameter that needs to be considered in our investigation.

Empirical Studies

We study this matter in more detail and perform tests for a given communication topology using the following assumptions:

The WVSN's application is the monitoring of a predefined observation area and is used to detect the presence of intruders. The data is processed in three steps as illustrated in Figure 27.

Each node has a limited battery level of *Bt* units. Each data processing step requires the use of *k* units of battery. The transmission of raw data requires *Btrans* units of battery and each processing step reduces the power necessary for data transmission by *25%* as illustrated in Figure 27. Each processing step increases the steganographic security level of the data by *1 unit* (cf. Figure 27). The weakest security level is *0*, the highest is *3*.

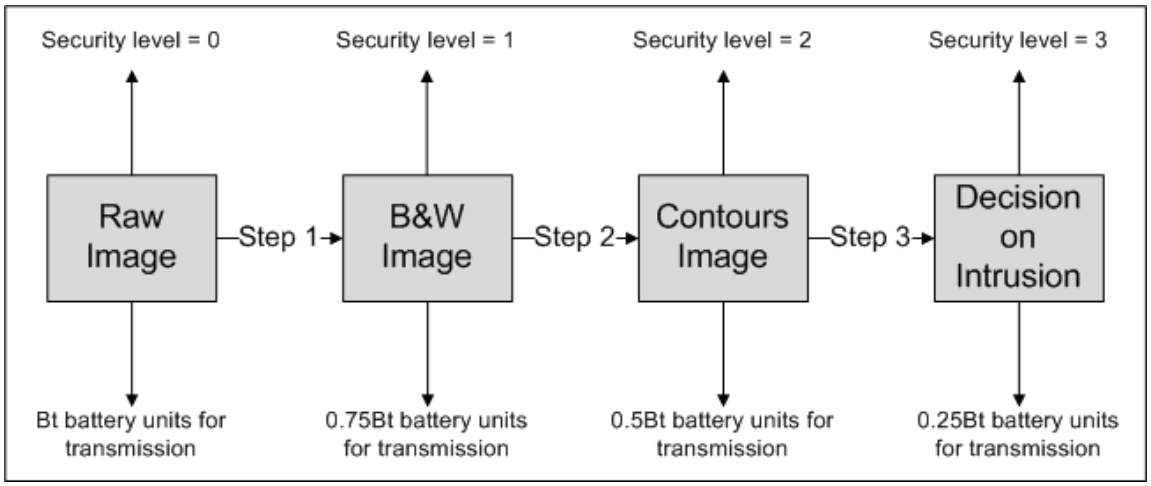

Figure 27. Data processing, security level and transmission battery usage models.

This model conveniently facilitates a focused study on steganalytic aspects of WVSNs while providing insight into interaction of security with communication, processing and battery power. Figure 27 illustrates the data processing chain that we consider for an intruder target-tracking surveillance WVSN. Each processing step reduces the amount of information in the images recorded by eliminating any irrelevant data: Step 1 eliminates the variations in grey levels, Step 2 eliminates the texture of potential objects present in the image, Step 3 eliminates any visual information since only the decision on the presence of an intruder remains. As mentioned previously the data processing chain chosen reduces the amount of information to be conveyed to the base station which also reduces its entropy and the volume of cover-media that can mask the steganography at the same time. With smaller entropy, and therefore a smaller redundancy, the data offers a reduced embedding capacity which leads to an increase in steganalytic security. Figure

27 points out also that the decrease in data volume for each data processing step reduces the amount of energy necessary for transmission to the next node.

Furthermore, in order to determine the effects of our preventative steganalysis approach on the overall WVSN, we also measure the success of the WVSN in detecting intrusion events. The overall surveillance process is illustrated in Figure 28. It consists of six main steps. The WVSN is monitoring an area of interest (Step 1). When an event is detected (Step 2), the camera records an image of the scene (Step 3) and forwards it to the base station (Step 4). Once the data arrives at the base station (Step 5) the network registers a successful record of event (Step 6). A recording is unsuccessful when the data does not arrive to the base station because of broken links. A broken link occurs if a node along the path to the base station is unresponsive because it has died due to lack of battery power or because the transmission of the data is blocked. Along the rate of successful recordings, we look at the average level of security achieved for each transmission. The level of security corresponds to the amount of processing steps the data has gone through: the more processing steps have occurred, the higher the security level of the data transmission. By looking at both the security levels achieved and the percentage of successful recordings of event, we can better understand how the increase in steganographic security can influence the WVSN performance. For example, distributing the data processing at the node level can make the nodes die faster which means the number of broken links would increase rapidly.

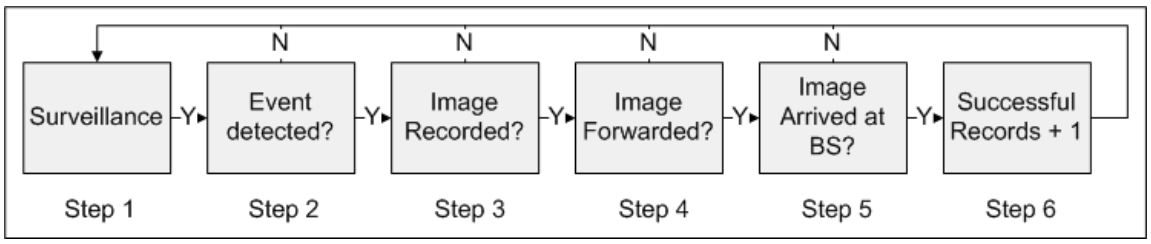

Figure 28. Overall surveillance process diagram.

The tests show the comparison of three data processing schemes in the three different network topologies cited earlier. For the simulations, we used a WVSN composed of 200 cameras. Each camera has a battery of 10 units to transmit the data to the base station and process it if necessary. Once the battery is depleted, the camera is virtually disconnected from the network and every successor becomes at the same time obsolete. Also, we set the amount of battery necessary for each data processing step to be k=0.5 units and the amount for transferring the raw data (full image) to be *Btrans=1 unit*. The results from the simulations are presented in Figures 29, 30, 31 and 32.

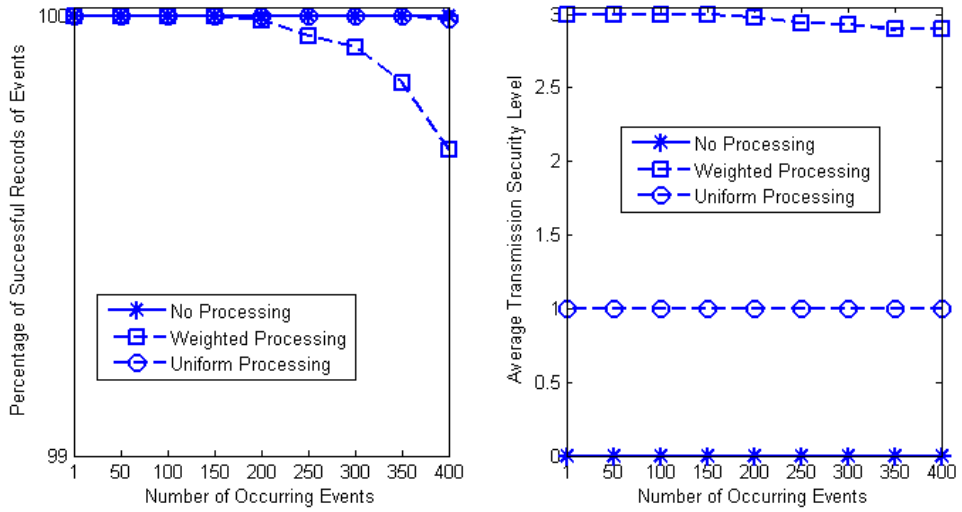

Figure 29. Horizontal WVSN of 200 cameras with Bt = 10 units of battery/node; (a) rate of successful records of events, (b) average transmission security level.

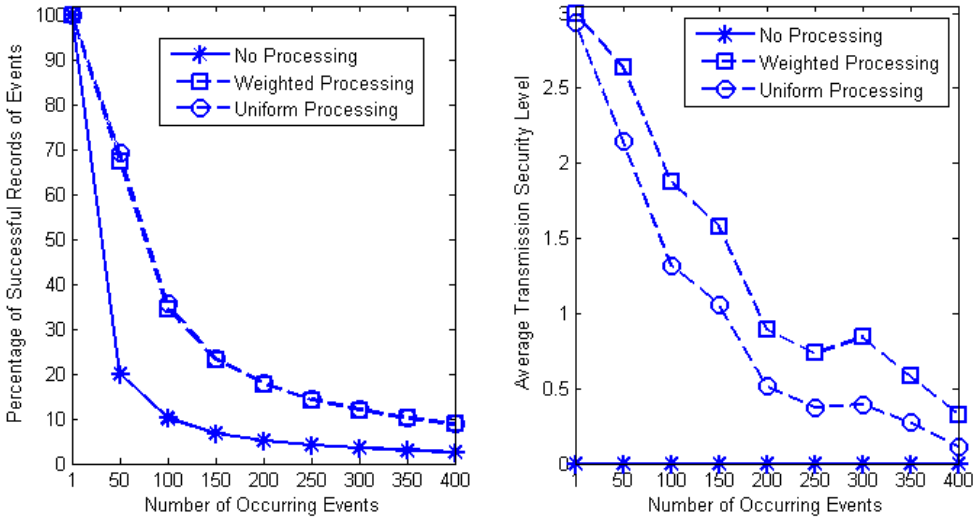

Figure 30. Vertical WVSN of 200 cameras with Bt = 10 units of battery/node; (a) rate of successful records of events, (b) average transmission security level.

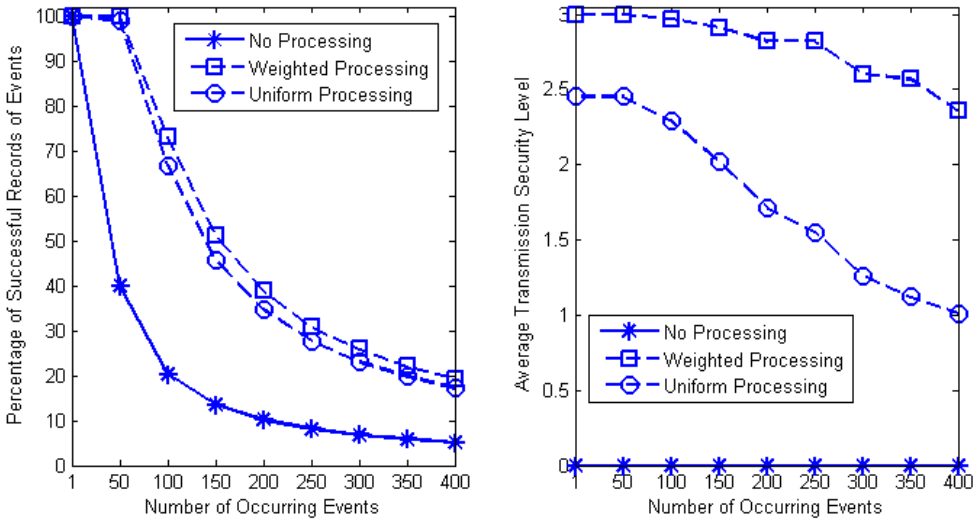

Figure 31. Fixed tree WVSN of 200 cameras with Bt = 10 units of battery/node; (a) rate of successful records of events, (b) average transmission security level.

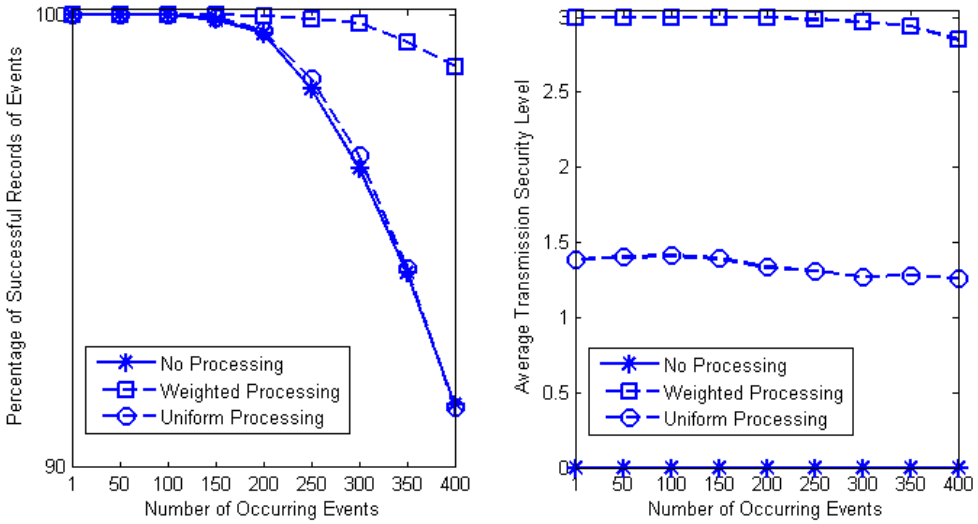

Figure 32. Random tree WVSN of 200 cameras with Bt = 10 units of battery/node; (a) rate of successful records of events., (b) average transmission security level.

Without any form of processing occurring at the camera, no measure of steganographic security exists; the average transmission security level is zero for these cases. However, as suspected, graphs in Fig.29 (b), Fig.30 (b), Fig.31 (b) and Fig.32 (b) prove that the use of a distributed data processing scheme adds a significant amount of security to the WVSN without adding more workload on its primary function by reducing the data entropy: the average transmission security level fluctuates between 1 and 3 in most cases except for the Fixed-Tree topology where it fluctuates between 0.25 and 3.

In addition, the distributed scheme helps liberate the network's bottleneck, formed by the nodes format close routing proximity to the base station, of heavy battery drainage by reducing the amount of data to be forwarded and therefore reducing the amount of battery power needed to transmit the information along. Our empirical results show that without distributed processing assignments, the percentage of successful records drops faster than in the other studied cases except in the case of the horizontal network. This is because in more vertical topologies, the nodes closest to the base-station are solicited with every data transmission toward the base station. Furthermore, without any kind of data processing at the nodes' level, the full raw data is handled. This results in the

workload of select nodes being high. By distributing the data processing, the amount of data handled by the nodes closest to the base-station is reduced therefore the workload is lightened and Fig.29 (a), Fig.30 (a), Fig.31 (a) and Fig.32 (a) prove that the number of successful events records is increased.

Moreover, the comparison between the results obtained for the Uniform and the Greedy architectures in the figures above confirm that the sooner the data is processed, the more secured the rest of the communication channels along the transmission path will be.
Additionally, these results show that choosing a more horizontal network topology can give some added steganographic security. However a purely horizontal network as shown in Figure 26 (d) does not cover a great deal of geographic space which limits the potential applications where such a network would be needed.
Overall, these simulations and their results provide us with insights and useful information to create a better, more efficient and more secured network against steganography: A horizontally spread network with a distributed scheme will improve the lifespan of the network and discourage attackers from corrupting the network by reducing the entropy of the data as early as possible on the communication path.

3.6. Single Point Preventative Steganalysis (SPPS)

Single Point Preventative steganalysis consists in a set of measures, preferably simple in their algorithm, which will ease the detection of steganography. The ultimate goal is to achieve extremely high true positive detection rate after the preventative solution has been applied and therefore discourage the attacker to even try using steganography. It is of the utmost importance that the critical data collected by the network remains untouched.

Single Point Preventative steganalysis takes place at the level of one single node. It consists in a single process that is only applied once at a strategic position within the network. The purpose of this solution is to transform the cover-media in such a way that the critical data is still intact but at the same time, any potential future steganographic

content would be so obviously apparent that the attacker would be discouraged to attempt corrupting the given network.

To answer the challenge that *SPPS* poses, the scenario focuses on a straight line link from a capturing node *C* to the base station *BS*. In this scenario, an event triggers the capture mechanism of node *C* which then records an image *I*. The same image is forwarded to the base station via a multi-hop link. A potential attack occurs at one of the forwarding node that could embed a secret message within the image *I*.

3.6.1. Theoretical Considerations

In the rest of this study, we consider only one node *C* from the WVSN but the theory can be applied to the whole network by extending the analysis to each of the network's capturing node. The scenes recorded by *C* can be seen as a movie with low sampling of frames per second. This low 'fps' rate implies that, from one frame to the next, some important changes can occur. However, there is usually still a high temporal redundancy between frames which can be exploited by the steganalyst. Indeed, in such conditions, it is possible to do a frame to frame comparison in order to extract the irregularities that might occur in the presence of steganography. It is important to wisely choose a reference frame in order to make these comparisons. For instance, if comparisons are made between two corrupted frames, the steganalytic decision might be different than the decision made from the comparison against a healthy frame. This is the reason why we propose to use the first, initial captured frames which can be collected during the calibrating process for example. This reference frame can be seen as the common background for any image capture and is denoted as *B*. This background frame *B* is an essential source of uncorrupted information for node *C* which can help the work of the steganalyst.

We first adopt a specific model for the images captured by the network. We initially restrict our study to the behavior of one single node N_i for simplification. To represent the entirety of the network, the process can then be extended on a node-by-node basis to

the entire network. Node N_i is set to capture frames, noted *I'*, which are modeled as the sum of three different components or subframes as shown in Equation 32.

$$I' = W + D + B \tag{32}$$

where:

- *W* is the watermark,
- *D* is the critical data,
- *B* is the frame background,
- *W* and *B* may be correlated,
- *W* and *D* are independent,
- *B* and *D* are independent.

3.6.2. Entropy and Uncertainty Factor

Using this deconstructive knowledge, the mutual information between the watermark *W* and the stego-frame *I'* can already be increased by looking at the conditional information between *W* and *I'* given the frame *B*:

$$I(W; I') \leq I(W; I'|B) \tag{33}$$

This motivates the need for maximizing the conditional mutual information instead of the simplistic $(W; I')$, which means that the steganalyst goal derived from Equation 33 is now to solve the following equation:

$$max[I(W; I'|B)] \; over \; steganalytic \; tools \tag{34}$$

As in Equation 33, Equation 34 states the ideal goal for the steganalyst which may not be achievable. However, if the maximum of $I(W; I'|B)$ is not reachable, the steganalyst should aim for a substantial increase in the mutual information $I(W; I'|B)$.

Equation 34 can be developed using entropy functions assuming the background B and the watermark W are independent:

$$\begin{aligned} I(W;I'|B) &= H(W|B) - H(W|I',B) \\ &= H(W) - H(W|I',B) \end{aligned} \tag{35}$$

To maintain initial generality of results, one often assumes as little as possible of the watermark, $H(W)$ cannot be controlled by the steganalyst to increase the result of Equation 35. Therefore increasing the conditional mutual information between W and I' given B means reducing the conditional entropy of W given I' and B:

$$\begin{aligned} &increasing\ I(W;I'|B) \\ &\Leftrightarrow reducing\ H(W|I',B) \end{aligned} \tag{36}$$

Reduction of the entropy above literally means reducing the uncertainty about the watermark when I' and B are known. This can be achieved when W is a function of I' and B. We also know from Equation 32 that $I' = I + W$ which further implies that the conditional entropy of Equation 36 can be reduced if I can be expressed as a function of B. Of course, the captured frame I usually includes more than just the background B. It conveys data and most likely some noise. This is the reason why we propose the following model:

$$I = B + D + N \tag{37}$$

where:

- B is the reference frame or common background,
- D is the data that bears interest to the network's purpose,

- N is some noise that includes the difference between the reference frame and the actual frame (excluding the data) as well as some potential additive watermark,
- B, D and N are assumed to be independent from one another.

In order for the two frames I and B to be similar, the conditional entropy of I given B must be minimized:

$$H(I|B) < \rho \ \ where \ \ \rho \to 0 \tag{38}$$

Since image I is assumed to be composed of three independent components in Equation 37, Equation 38 can be rewritten:

$$\begin{aligned} H(I|B) \ &\leq H(B|B) + H(N + D|B) \\ &\leq H(N|B) + H(D|B) \end{aligned} \tag{39}$$

The background and the data are important as they give information on the event happening at a specific location. On the other hand, the noise N should be removed as much as possible as they contain no important information and generally corrupt the integrity of the data. We propose to do so simply by filtering the noise in order to reduce its intensity or even erase it if possible.

The definition of preventative steganalysis implies that a simple analysis of the media I' gives as much information about the watermark W as possible such that the steganalysis can yield high success rates. This can be achieved by exploiting the uncertainty between W and I', or more specifically, the uncertainty about W from the observation of I'. This quantity can be measured via the uncertainty coefficient as described in [10]. The uncertainty coefficient between two variables X and Y is defined as:

(40)

$$U(Y|X) = \frac{I(X;Y)}{H(Y)}$$

It quantifies the amount of knowledge about *Y* that can be derived from *X*. The uncertainty coefficient takes value between *0* and *1*. It achieves *0* when *X* and *Y* are uncorrelated and reaches *1* when *Y* can be entirely predicted from *X*.

In our case, we are interested in evaluating the watermark *W* with the knowledge of the potentially corrupted frame *I'*. This involves computing the uncertainty coefficient $U_1(W,I')$:

$$U_1(W|I') = \frac{I(W;I')}{H(W)} \tag{41}$$

Our goal is to make $U_1(W|I')$ as close to *1* as possible so that the most information about *W* can be obtained from *I'*. This can be achieved if we manage to build a frame *I'* in such a manner that the following equality can be obtained:

$$I(W;I') = H(W) \tag{42}$$

Equation 42 does represent the steganalyst's ultimate goal since from the derivation given by Equation 43, Equation 42 implies that the conditional entropy of *W* given *I'* is *zero*, i.e. the knowledge of frame *I'* leads to the perfect knowledge of the watermark. Under such circumstances, the watermark is clearly identifiable and no covert communication can occur undetected.

$$I(W;I') = H(W) - H(W|I') \tag{43}$$

3.6.3. Data Preservation

While it is of the utmost importance to provide a steganalytic cover for the network, it is at least equally important that the WVSN can still performs its primary duty whether it involves data mining activities or area monitoring. In other word, the steganalysis must not interfere with the integrity of the data *D* collected by the network. In other words, when data is present in frame *I'*, it is important that *D* can be easily and preferably entirely predicted with the observation of *I'*. This can be expressed with another uncertainty coefficient $U_2(D|I')$:

$$U_2(D|I') = \frac{I(D;I')}{H(D)} \tag{44}$$

Our goal is to make $U_2(D|I')$ as close to *1* as possible which is possible if we can find a frame *I'* such that:

$$I(D;I') = H(D) \tag{45}$$

3.6.4. Common Outcome

In order to develop a preventative steganalytic solution that will both protect the network against covert communications and keep the collected data intact, Equations 42 and 45 must be satisfied. This implies that a common solution must be found for the following system:

$$find\ I' such\ that \begin{cases} I(W;I') = H(W) \\ I(D;I') = H(D) \end{cases} \tag{46}$$

Reasoning in two separate steps and using Equation 46, we first focus on the mutual information *I(W,I')* which can be further derived:

$$\begin{aligned} I(W;I') &= I(W;W+D+B) \\ &= H(W+D+B) - H(W+D+B|W) \\ &= H(W+B) + H(D) - H(D+B|W) \\ &= H(W+B) + H(D) - H(D) - H(B|W) \\ &= H(W+B) - H(B|W) \end{aligned} \tag{47}$$

For the previous equation to lead to the desired result offered by Equation 48, it is necessary to solve:

$$H(W+B) - H(B|W) = H(W) \tag{48}$$

This equality can be achieved in two trivial cases: when W and B are independent or when B is a known, temporally independent entity. On the case where W and B are independent, Equation 48 becomes:

$$\begin{aligned} H(W+B) - H(B|W) &= H(W) + H(B) - H(B) \\ &= H(W) \end{aligned} \tag{49}$$

Alternatively, if B is a known and temporally independent entity, B can be seen as a constant B_c for which we have:

$$\begin{aligned} H(W+B_c) &= H(W) \\ H(B_c|W) &= H(B_c) = 0 \end{aligned} \tag{50}$$

And Equation 48 would become:

$$\begin{aligned} H(W+B) - H(B|W) &= H(W) - 0 \\ &= H(W) \end{aligned} \tag{51}$$

Both solutions are satisfying. However, assuming W and B are independent is not realistic and goes against our initial set of assumptions. The steganalyst should expect the attacker to use elaborate steganographic techniques where the steganography would

not stand out in the background which implies a degree of correlation between W and B. Therefore, the solution where B is set as a known constant B_c is preferred.

Assuming that the background is a known entity such that $H(B_C) = 0$, we derive the second equality in Equation 8, i.e. the mutual information *I(D;I'),* which boils down to:

$$\begin{aligned} I(D;I') &= I(D;W+D+B_c) \\ &= H(W+D+B_c) - H(W+D+B_c|D) \\ &= H(W+B_c) + H(D) - H(W+B_c|D) - H(D|D) \\ &= H(W+B_c) + H(D) - H(W+B_c) \\ &= H(D) \end{aligned} \tag{52}$$

Equation 52 injected in Equation 44 gives an uncertainty coefficient $U_2(D|I')$ of *1* which by definition ensures that the data will remain identifiable when the frame *I'* is observed. Thus the substitution of the actual frame background B for a known and constant background B_c guarantees that the potential watermarking W appears more evidently in the frame *I'* and that the integrity of the data collected by the network is preserved.

3.6.5. Practical Considerations

Let B_k represent the first frame captured by the node N_i during the network's initialization phase. It is legitimate to assume that B_k is an uncorrupted image that does not contain steganography and critical information. Therefore frame B_k is a perfect candidate to replace the background of any future image recorded by N_i.

The frames *I'* that node N_i records can now be decomposed into four components: the data, the watermark, the background and some noise. Therefore Equation 37 becomes:

$$\begin{aligned} I' &= W + D + B \\ &= W + D + B_k + N \end{aligned} \tag{53}$$

where:

- B_k is the reference frame,
- W is the watermark,
- D is the critical data,
- N is some noise corresponding to the difference between B_k and B.

In order for I' to be of the desired form $I' = W + D + B_c$, it is necessary for N to be removed from I':

$$B_c = B_k \ \ and \ \ N = 0 \Rightarrow W + D + B_k + N = W + D + B_c \tag{54}$$

For illustration purposes, we use a sequence consisting of a backyard scene. The critical recorder data will be represented by a ball which will be thrown through the recorded backyard around mid-sequence. Figure 33(a) shows the frame which serves as the reference B_k. Figure 33(b) shows a random frame I' from the same sequence where the data D, i.e. the ball, has appeared.

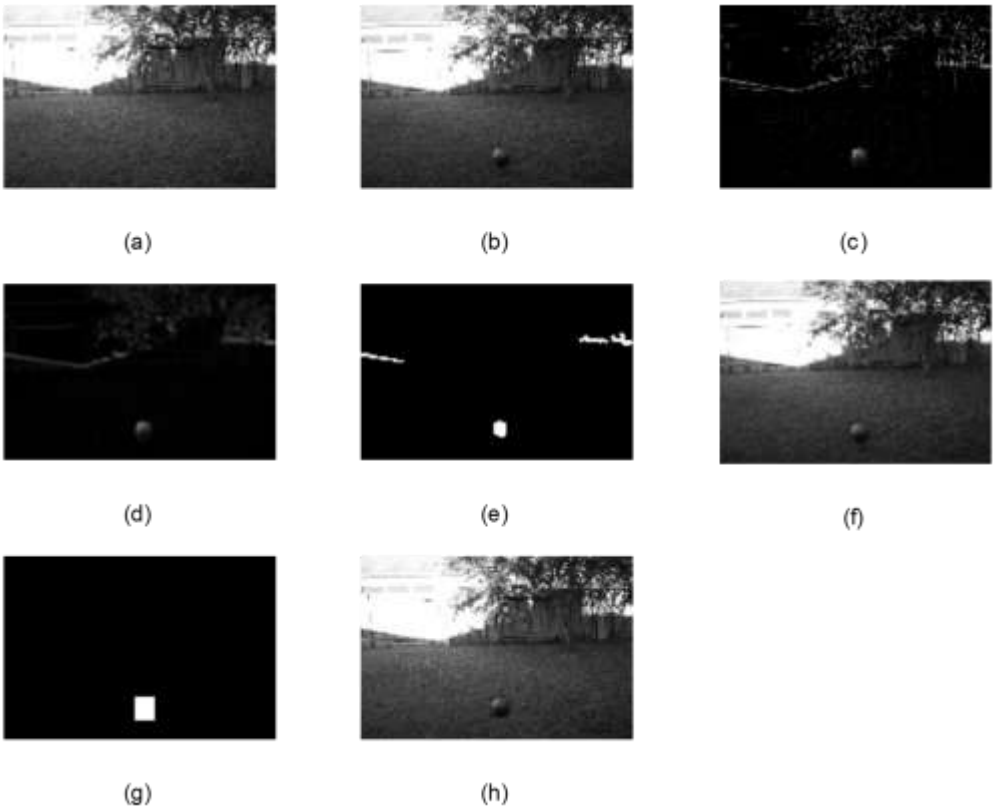

Figure 33. Illustrations of algorithm: (a) Reference frame B_k, (b) frame I' with intruder ball, (c) frame difference B_k-I', (d) average filtering of B_k-I', (e) mask obtained after thresholding, (f) reconstructed frame after filtering, (g) mask from area-based technique, (h) reconstructed frame from area-base algorithm with original data highlighted by rectangle.

To facilitate the denoising process, we can isolate N by subtracting B_k from I' so that only the noise and the data remain as illustrated in Figure 33(c) which shows the disparities between the actual background of I' and the reference frame B_k. The objective is to isolate the data and get rid of the other irrelevant discrepancies so that nothing left can negatively affect the steganalysis and D remains intact. We try two solutions to achieve the suppression of N based on the same assumptions: the data covers a large block of connected pixels whereas the noise N is zero-mean and sparsely distributed over I'. These two solutions respectively involve a simple denoising filter and an area -based masking technique.

3.6.5.1. Average filtering

Because N has the characteristics of a zero-mean noise, a simple denoising filter, e.g. an averaging filter, can at least considerably weaken N. After filtering the frame difference B_k-I', the data will of course be altered as well but as a large group of pixels of similar intensity, it will retain most of its shape after filtering whereas the rest of the smallest artifacts will mostly be erased. Figures 33(d) and 33(e) provide insight on the effect of using an averaging filter on the frame of Figure 33(c). As it can be seen, a great portion of the discrepancies present in Figure 33(c) are erased whereas the data is still visible in Figure 33(e). Figure 33(e) is used as a mask to reconstruct the final frame from Figure 33(f): black pixels in Figure 33(e) are replaced by pixels from the reference frame B_k and white pixels are preserved from the original frame I'.

Although the average filtering solution shows great results in this case, it is to be noted that depending on the size and shape of the data D, the denoising filter can have serious consequences on the integrity of D. For example, a square shape will become rounder with the use of an averaging filter. And although the network will probably still flag the presence of data in I', it might be more difficult for the user to identify the true nature of the data.

3.6.5.2. Area-based alternative

In order to avoid the problems that can be encountered with a filtering technique, we also try using an alternative technique based on area selection. This solution works in such a way that large clusters of connected pixels emerging from the difference I'-B_k are protected whereas the rest of the difference is cleared of any interferences by setting all unprotected pixels to black.

Figure 33(f) shows the mask obtained after the area selection algorithm has been applied to the frame in Figure 33(c). In this case, the white rectangle shows that the area

containing the large group of light pixels is the only remaining part of the original frame. As for the previous algorithm, the black pixels in Figure 33(f) are replaced by pixels from B_k and white pixels are replaced by pixels from the original *I'* frame. The result of this area-based technique is shown in the reconstructed frame of Figure 33(g). A rectangle is drawn around the area containing the original pixels from *I'* which clearly proves that the area surrounding the object constituting the data remains.

This method is more efficient in getting rid of any potential discrepancy between the actual frame background and the reference one B_k as shown in Figure 33(f). However, depending on how precise the selection of the concerned area is, the number of protected, or masked, pixels that do not belong to the data *D* varies. For example, Figure 33(g) indicates that a portion of the original frame inside the marked rectangle, although not part of the data *D* itself, manage to find its way in the reconstructed frame unchanged. In case the area masked is of large proportions, more noise might go through the cleaning process untouched. This could therefore diminish the efficiency of the preventative steganalysis.

3.6.5.3.Common outcome

In both cases, after processing, the steganalyst expects to find *I'* to be as close as possible as being the sum of only three parts:

$$I' \approx W + D + B_k \tag{55}$$

From this equality, it is clear that in the absence of watermark, the isolation of the data *D* becomes an easy task as B_k is a known entity. When steganography has occurred, because *W* and *D* have different distributions and B_k is known, the preventative steganalysis is expected to yield high success rates.

3.7. Simulations

To assess the validity of the proposed algorithm as a working proactive steganalytic system, several experiments are conducted. These experiments include:

- Simulations on the uncertainty coefficient,
- Simulations on data preservation,
- Simulations on steganography detection.

The simulations are conducted on a series of sequence of frames showing a child's playground similar to the one shown in Figure 33(b). In this setting, the goal of the network is to detect the presence of a ball in the playground by identifying round objects in the frame.

3.7.1. Uncertainty Coefficient

$U_1(W|I')$ is the uncertainty coefficient, computed for watermarked sequences where W is an additive white Gaussian noise. Tests are conducted on unprocessed sequences first, then on the same sequences when the actual background is replaced by the reference background B_k. The tests are also conducted on the sequences derived after the filtering process and the area selection technique have been applied after the background has been substituted for B_k. The watermark is embedded with a signal-to-noise ratio of 50 dB relative to the image.

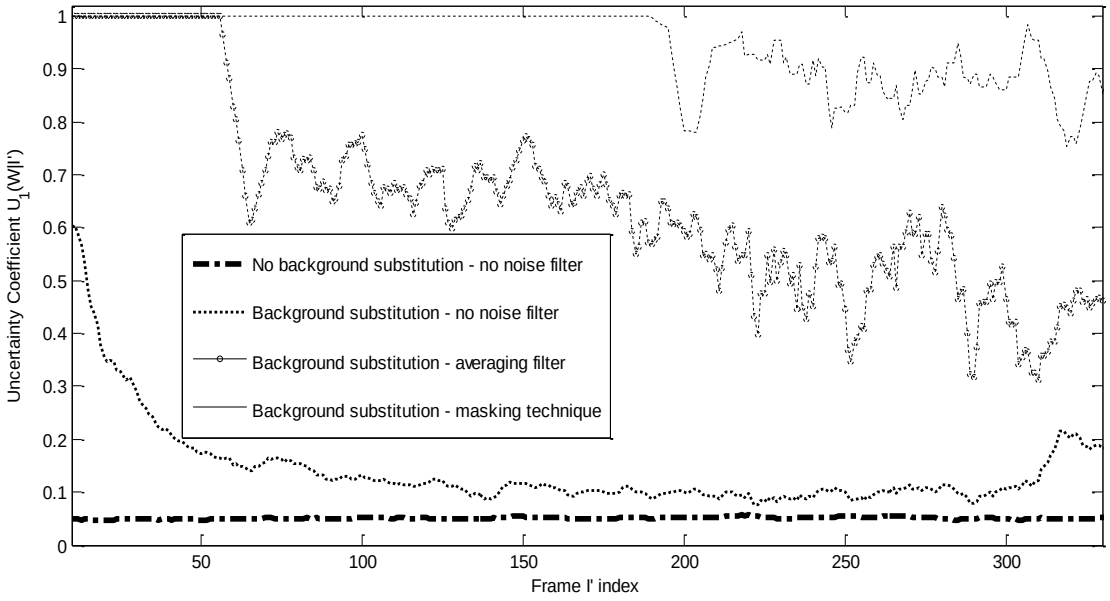

Figure 34. Uncertainty coefficient for frame sequence at various stage of processing with a watermark embedded with a SNR of 50dB.

Figure 34 shows that our proposed solution is effective in increasing the uncertainty coefficient $U_1(W|I')$ eventually making the watermark more easily detectable. When the actual background is replaced with B_k, it logically appears that the first frames in the sequence obtain the highest uncertainty coefficient due to the higher correlation with B_k. The uncertainty coefficient gets lower with time which can be explained by the appearance of the relevant data D in the frame and the eventual perturbation of the scenery has time goes by. This suggests that improvement could be achieved if the reference frame was refreshed at different points in time.

Overall, the computation results for the uncertainty coefficient $U_1(W|I')$ provides proof that the derived algorithms can greatly improve the knowledge of the steganalyst on the potential presence of watermarking.

3.7.2. Data Preservation

Although the primary objective of the steganalyst is to protect the network against covert communications, the steganalysis must not interfere with the network's main objective.

In monitoring applications, the network collects critical data which must still be identifiable after the steganalysis has taken place.

To quantify the effects of the steganalysis on the data, we test the efficiency of a data detection algorithm with the unmodified frame sequence and after our proposed solution has been applied.

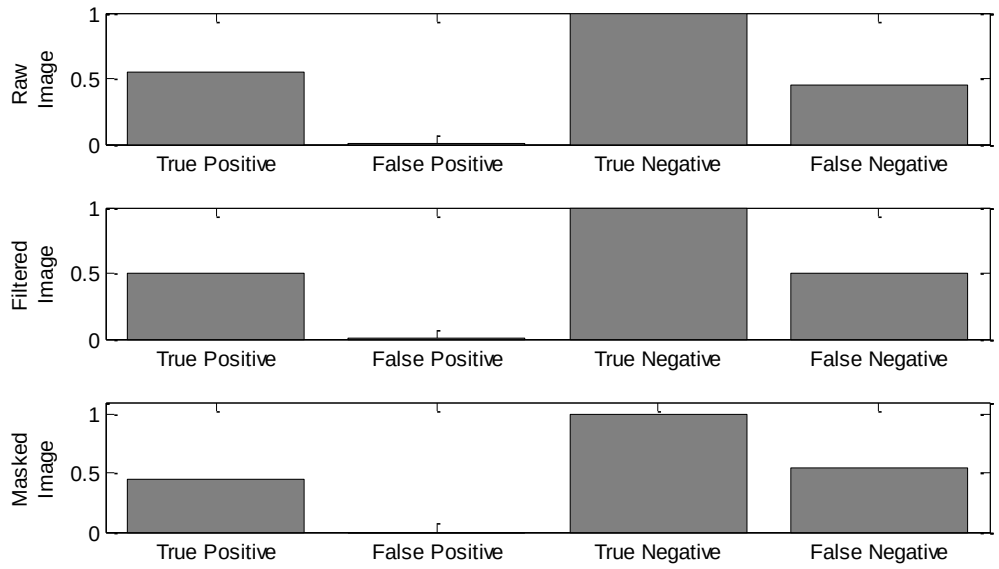

Figure 35. Data detection outcomes for uncorrupted frame sequences.

Figure 35 shows the success and error rates of the algorithm used to detect the presence of round objects. The results are drawn in the case where no steganography has occurred since purely the effects on the data are desired. What we are interested in is not the efficiency of the detection algorithm itself but rather in the changes in the detection outcome after the use of our proposed steganalytic solution. The results show a slight decrease in the rate of true positives but an important decrease in the rate of false positives. The results also show a rather similar true negative rate and a slight increase in the rate of false negatives. From these results we can infer that the data identification algorithm performs as well in each situation. Although the critical sensibility is slightly decreased, the specificity of the detection is improved. It is to be noted that parameters in

the steganalytic solutions can be tweaked to find the best compromise between steganalytic protection and data preservation depending on the desire of the network's users.

3.7.3. Steganography Detection

Because steganalysis is about protecting the sensor network against the possibility of covert communication, it is necessary to evaluate how well the preventative steganalysis performs. In order to verify if our preventative steganalytic approach leads to high success rates a simple threshold-based detector is derived. In this paper, we use an average block variance threshold-based detector.

The detector computes the average variance of blocks of pixels from the frame *I'*. Each frame is divided in a certain number of blocks where the variance is computed. The block variances obtained are then averaged over the whole frame. A block-based algorithm is suggested in order to reduce the effect that the presence of data might have on the final steganalytic decision. For example, if the steganalyst was to count the number of pixels originating from image aberration, the data *D*, as a large group of pixels, would have a great influence on false positive rate of the detector decision.

Frames are chosen randomly and corrupted with an additive white Gaussian watermark in both sequences obtained after the steganalytic solutions proposed have been applied to the original sequence. For the fairness of comparison, the same frames in both sequences are corrupted with identical watermark. The embedding is done using a signal-to-noise ratio of 30dB with regards to the original image. Computations of the average block variance are derived and presented in Figure 36.

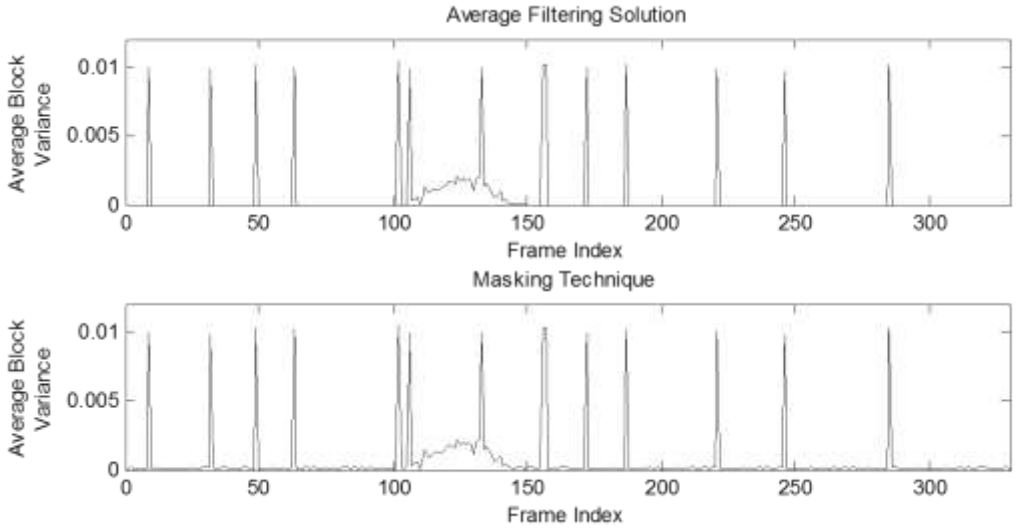

Figure 36. Average block variance in corrupted processed sequences.

Figure 36 shows similar results for both derived solutions. The presence of a watermark is easily identified due to the important increase of the average block variance when steganography has occurred in the frame. The presence of data, mostly visible from frame 100 to 150 in Figure 36, also generates an increase in the block variance but the scale is on another level compared to what watermarking contributes. This implies any confusion between the presence of data and the presence of watermarking is unlikely.

To get a better understanding on the merits of each method proposed, we compute the ratios of the obtained variance extrema. The results show that when with the averaging filter solution, the presence of watermarking increases the average block variance by a 1.6e+4 factor, whereas for the area-based solution, this factor increases to 5e+4, thus concluding that the area-based solution leads to better steganalytic results.

3.8. Conclusion

The simulations derived for the *SPPS* so far show promising results; the steganalysis clearly gains from the knowledge of the static background as expected. However, the assumption that the initial recording made by the camera is uncorrupted, although a valid

traditional assumption, can raise some questions. In case the first frame is corrupted, the proposed steganalysis would only forward the watermark repetitively to the base station.

4. IMPROVED PREVENTATIVE STEGANALYSIS IN SENSOR NETWORKS

Both the distributed and single point steganalyses show potential but come short in term of energy consumption and initial assumptions respectively. Therefore, we decided to look at a trade-off between both steganalyses.

4.1. A New Measure for Steganalysis

4.1.1. Motivation

In order to progress efficiently in the research for steganalytic solutions in sensor network, it is best to be able to quantify how difficult it can be to hide information in a given media. So far, we have looked at the entropy and the uncertainty factor to move in the desired direction but both measures fall somewhat short. The entropy certainly helps quantifying the uncertainty of the said media but fail to provide bounds on its value making it hard to concretely realize whether the steganalysis is efficient. The uncertainty factor does a better job but fails at capturing the nature and potential for watermarking of cover-images. With the uncertainty coefficient the image is seen as a whole single entity whereas we believe in looking at the image as a set of pixels prone to steganography.

4.1.2. Embedding Potential

When an image *I* is received by the steganalyst, it is very difficult to determine whether any hidden content is present. The content of the image is most of the time unknown to the steganalyst and a noisy image might just be a noisy image; not one where the steganography has introduced distortion. This is why regularly assumptions about distribution models are made to represent a natural image. Often, however, these assumptions although providing essential insight for research progress, are not realistic. As best as possible, efforts to minimize the amount of assumptions are necessary for a more complete and practical steganalytic solution.

Without any prior specifics, all that can be reasonably assumed about an image *I* is that each pixel has a maximum range value R_{max} of *0* to *255*.

$$R_{\max} = \{0,...,255\} \tag{56}$$

We denote S_{max} the set of pixels that have P_{max} for probability density function, with P_{max} being the pdf where each value in R_{max} is equiprobable

$$P_{max}(p = p_i) = \frac{1}{256}, \forall p_i \in R_{max} \tag{57}$$

The range for the pixel value is exactly $R_{max}$when the pixel is totally random. The range value is reduced when the steganalyst manages to gain partial knowledge that can help him or her estimate the given pixel. For example, if the image is known to be black and white, the range value reduces to $\{0,1\}$. In order for the attacker to hide information in an image successfully, the watermark would have to target the pixels that are not 'known' by the steganalyst. Only where the uncertainty is maximal can the attacker embed data without any precaution. From the steganalyst's point of view, any part of the image that cannot be fully predicted is a potential hiding place for covert-communications. Of course, the more uncertain the steganalyst is about pixel values, the higher the potential for steganography. In order to quantify the risk for steganography an image *I* can generate, we introduce a new measure, the embedding potential.

The embedding potential, *EP*, is a quantitative measure of how much data can be embedded in an image via the use of the entropy *H*. The entropy of a random variable X taking the values x_i with respective probability p_{x_i} has been defined as:

$$H(X) = -\sum_{i} p_{x_i} \cdot log2\left(p_{x_i}\right) \tag{58}$$

4.1.2.1.Definition - Embedding potential

For a *M-by-N* image *I*, the embedding potential *EP* is mathematically defined as follows:

$$EP = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} H\big(I(i,j)\big)}{H_{max}} \tag{59}$$

H_{max} represents the maximum entropy of a pixel which occurs when the pixel is part of the set S_{max}:

$$H_{max} = H\big(I(i,j)\big), \qquad \forall\, I(i,j) \in S_{max} \tag{60}$$

In other words:

$$\begin{aligned} H_{max} &= -\sum_{i=1}^{256} \frac{1}{256} \cdot log2\left(\frac{1}{256}\right) \\ &= 8 \end{aligned} \tag{61}$$

The value of *EP* ranges between *0* and *MN*. When $EP = 0$, the steganalyst has full knowledge of the image *I* and the attacker theoretically cannot hide any information stealthily:

$$EP = 0 \Leftrightarrow H\big(I(i,j)\big) = 0, \qquad \forall (i,j) \in [\![1, M]\!] \times [\![1, N]\!] \tag{62}$$

When $EP = MN$, the steganalyst has absolutely no knowledge of I and the attacker can take advantage of the full amount of pixels in *I* to embed any message.

$$EP = MN \Leftrightarrow H\big(I(i,j)\big) = H_{max}, \qquad \forall (i,j) \in [\![1,M]\!] \times [\![1,N]\!] \tag{63}$$

4.1.2.2. Assumptions

The role of the preventative steganalysis is to reduce as much as possible the embedding potential such that the possibility of covert-communication is also reduced.
The steganographic model is assumed to be additive such that:

$$I' = I + W \tag{64}$$

- I is the original image captured by the network.
- W is the potential watermark embedded by the attacker.

Naturally, the embedding potential of a watermarked image I' verifies:

$$EP(I') \geq EP(I) \tag{65}$$

Equality $\mathrm{EP}(\mathrm{I}') = \mathrm{EP}(\mathrm{I})$ occurs when the watermark replaces pixels from I with pixels with identical entropies.

Let's suppose the steganalyst manages to reduce the embedding potential of I so that $EP(I) = \alpha$. For practical reasons, let's further assume that this embedding potential describes the presence of α fully unknown pixels in I so that:

$$\begin{aligned} EP(I) &= \frac{1}{H_{max}}(\alpha \cdot H_{max} + (MN - \alpha) \cdot 0) \\ &= \alpha \end{aligned} \tag{66}$$

Then $EP(I')$ verifies:

$$EP(I') = EP(I) + EP(W) \\ = \alpha + EP(W) \tag{67}$$

From the steganalyst's point of view, W is completely unknown which means that the entropy of every pixel that W covers is going to equal H_{max}. Supposing that W covers β pixels, then we have:

$$EP(W) = \beta \tag{68}$$

And, by extension:

$$EP(I') = \alpha + \beta \tag{69}$$

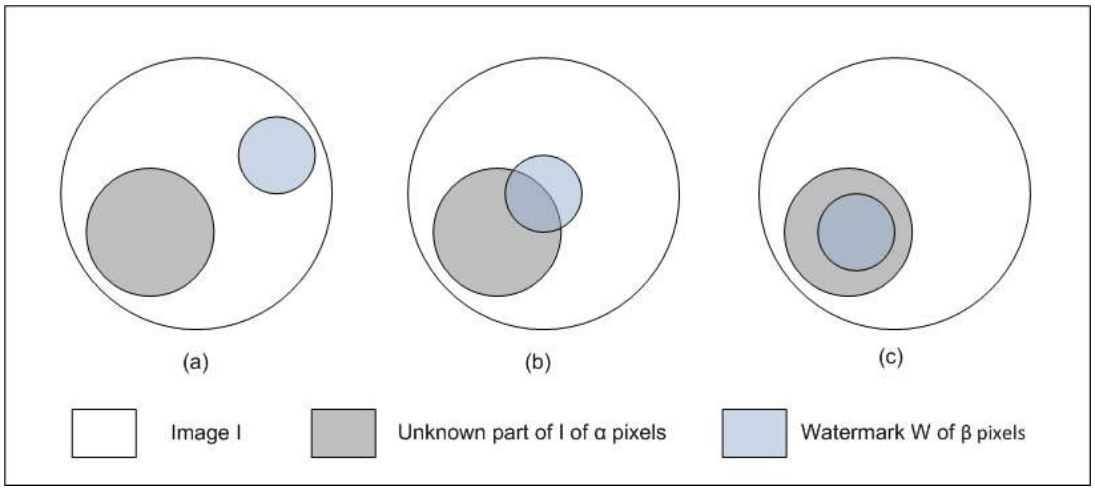

Figure 37. Representation of embedding possibilities in an image.

Let's assume that the watermark W covers δ pixels of the unknown part of I (Figure 37 (b)). Then we get:

$$EP(I') = \alpha + (\beta - \delta) \tag{70}$$

In order for $EP(I') = EP(I)$, we must have:

$$\alpha + (\beta - \delta) = \alpha \quad \Leftrightarrow \quad \beta - \delta = 0 \tag{71}$$

Therefore we have $EP(I') = EP(I)$ only when the watermark is completely included in the unknown part of I (Figure 37 (c)).

As mentioned, performing steganalysis in a sensor network setup presents a set of challenges. However there are also some advantages. For example, the network originally belongs to the trusted party and therefore gives the steganalyst some freedom as to the amount of processing that can be implemented to defeat steganography. It also allows for reasonable assumption on the content of the image captured by the cameras as the network has been setup in an area chosen by the trusted party.

4.2. Separation of Image Components

Because it is expected that the network will regularly pull data from each node, the steganalyst can count on elements of temporal correlation between samples to further help the battle against steganography.

4.2.1. Moving Background and Data

A typical image captured by the network is composed of two entities : the part composed of $mx \times nx$ pixels that is useless towards the network's goal and the one composed of $my \times ny$ pixels that is useful.

Figure 38. Example of separation of background and data in images.

For example, in a surveillance scenario, the useful part of the image is represented by the silhouette of the potential intruder. The 'useless' part of the image is mostly composed of the background. This basic definition of the camera's capture allows us to define a simple model for representing each m × n image recorded by the node N_k :

$$\mathrm{I}_{\mathrm{N_k,i}} = \mathrm{B'_i} \cup \mathrm{D'_i} \quad \text{and} \quad mx \times nx + my \times ny = m \times n \tag{72}$$

This equation can also be rewritten as an additive model for $\mathrm{I}_{\mathrm{N_k,i}}$:

$$\mathrm{I}_{\mathrm{N_k,i}} = \mathrm{B_i} + \mathrm{D_i} \tag{73}$$

Provided that:

$$\mathrm{B_i} = \mathrm{B'_i} \cup \mathbb{O}_{\mathrm{m,n}} \quad \text{and} \quad \mathrm{D_i} = \mathrm{D'_i} \cup \mathbb{O}_{\mathrm{m,n}} \tag{74}$$

Where $\mathbb{O}_{\mathrm{m,n}}$ is the all-zero matrix of size $m \times n$.

When the network is performing its assigned task, e.g. surveillance, it will at each suspicious event capture a screenshot of the monitored area and forward it towards the

base station so that further analysis can be made and the presence of an intruder can be assessed. It is a perfectly reasonable and realistic assumption to expect the intruder at each instance to appear in different zones of the screenshot: an intruder might come from one direction in one shot and another intruder might come from another direction in a future shot. Moreover, in order for the network to be able to trigger the capture of an image when some event arises implies that the network can recognize and isolate this event within the image.

Following the model adopted for the image capture by the node N_k, $\mathrm{I_{N_k,i}} = \mathrm{B'_i} \cup \mathrm{D'_i}$, and the statements made above, we can derive the following equation:

$$\bigcup_{\mathrm{i}=1}^{\mathrm{r}} \mathrm{B'_i} = \mathrm{B^{\#}} \text{ where } \mathrm{B^{\#}} \text{ is an } m \times n \text{ image} \tag{75}$$

This equation implies that after the node N_k makes a certain number of captures, the union of the backgrounds recorded, i.e. the image minus the data D_i , will span an $m \times n$ image. Practically this statement implies that it is possible to generate an $m \times n$ estimate of I_{N_k} where no trigger event is present, after observing enough captures from N_k. It also motivates the assumption that the background B_i and the data D_i of each image $I_{N_k,i}$ are independent and can be isolated from each other.

4.2.2. Static Background, Data and Noise

Because of the immobility of the cameras and therefore the great redundancy expected between visual samples, the background B_i can indeed be further divided into a static part B and a moving part N_i, where N_i represents the discrepancies between B_i and B:

$$\mathrm{B_i} = \mathrm{B} + \mathrm{N_i} \tag{76}$$

This leads to a more developed additive model of the images captured by node N_i is:

$$I_{N_k,i} = B + N_i + D_i \tag{77}$$

As previously stated, the data and background can be isolated if needed which gives us the freedom to perform separate analyses on $(B + N_i)$ and on D_i.

For practical reasons, let's assume that the distortions, or noise, are overtime statistically neutral such that:

$$\lim_{r\to\infty}\left(\sum_{i=1}^{r} N_i\right) = 0 \tag{78}$$

By making this usual assumption, we can further isolate the constant part of the background B from the rest of $I_{N_k,i}$ since after enough observation from the node N_k, we have:

$$\sum_{i=1}^{r}(B + N_i) \approx r \times B \tag{79}$$

4.3. Conditional Embedding Potential

It is to be noted that cameras in a sensor network are very likely to be static which undoubtedly has the effect of increasing the temporal correlation between each image captured by the same node.

$$\rho\left(I_{N_k,i}, I_{N_k,j}\right) \geq \varphi, \quad \forall(i,j) \in \mathbb{N}^2 \tag{80}$$

It would be a waste of resources for the steganalyst not to take advantage of this temporal correlation. This correlation will of course have an impact on the embedding potential and therefore will directly influence the risk for steganography to occur. By

using the redundancy between frames received from the same node, the embedding potential can be reduced. The conditional embedding potential is defined as:

$$\mathrm{EP}(\mathrm{I}|\mathrm{X}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{H}(\mathrm{I}(\mathrm{i},\mathrm{j})|\mathrm{X})}{\mathrm{H}_{\max}} \tag{81}$$

From this formula we can easily derive the inequality:

$$\mathrm{EP}(\mathrm{I}|\mathrm{X}) \leq \mathrm{EP}(\mathrm{X}) \tag{82}$$

The equality occurs when I and X are independent.

Proof: This inequality directly from the well-known definition:

$$\mathrm{H}(\mathrm{I}|\mathrm{X}) \leq \mathrm{H}(\mathrm{X}) \tag{83}$$

Suppose that we use the last received sample $I_{N_k,i-1}$ to compute the conditional embedding potential of $I_{N_k,i}$:

$$\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k,i}}\middle|\mathrm{I}_{\mathrm{N_k,i-1}}\right) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{H}\left(\mathrm{I}_{\mathrm{N_k,i}}(\mathrm{i},\mathrm{j})\middle|\mathrm{I}_{\mathrm{N_k,i-1}}(\mathrm{i},\mathrm{j})\right)}{\mathrm{H}_{\max}} \tag{84}$$

As shown before, it is possible for the steganalyst to separate the background from the data, both being independent, such that the conditional embedding potential becomes:

$$\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k},\mathrm{i}}\middle|\mathrm{I}_{\mathrm{N_k},\mathrm{i}-1}\right) = \frac{\sum_{s=1}^{mx}\sum_{t=1}^{nx}\mathrm{H}\left(\mathrm{B_i}(s,t)\middle|\mathrm{I}_{\mathrm{N_k},\mathrm{i}-1}(s,t)\right)}{\mathrm{H_{max}}} + \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny}\mathrm{H}\left(\mathrm{D_i}(s',t')\middle|\mathrm{I}_{\mathrm{N_k},\mathrm{i}-1}(s',t')\right)}{\mathrm{H_{max}}} \tag{85}$$

The data is independent from the rest of the image I and the data between two frames are assumed independent from one another. This gives a simpler formula for the conditional embedding potential:

$$\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k},\mathrm{i}}\middle|\mathrm{I}_{\mathrm{N_k},\mathrm{i}-1}\right) = \frac{\sum_{s=1}^{mx}\sum_{t=1}^{nx}\mathrm{H}\left(\mathrm{B_i}(s,t)\middle|\mathrm{I}_{\mathrm{N_k},\mathrm{i}-1}(s,t)\right)}{\mathrm{H_{max}}} + \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny}\mathrm{H}\left(\mathrm{D_i}(s',t')\right)}{\mathrm{H_{max}}} \tag{86}$$

Since we can also assume knowledge of the static background common to all sample, the conditional embedding potential $\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k},\mathrm{i}}\middle|\mathrm{B}\right)$ can be derived:

$$\begin{aligned}\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k},\mathrm{i}}\middle|\mathrm{B}\right) &= \frac{\sum_{s=1}^{mx}\sum_{t=1}^{nx}\mathrm{H}(\mathrm{B_i}(s,t)|\mathrm{B}(s,t))}{\mathrm{H_{max}}} + \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny}\mathrm{H}(\mathrm{D_i}(s',t')|\mathrm{B}(s',t'))}{\mathrm{H_{max}}} \\ &= \frac{\sum_{s=1}^{mx}\sum_{t=1}^{nx}\mathrm{H}(\mathrm{B_i}(s,t)|\mathrm{B}(s,t))}{\mathrm{H_{max}}} + \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny}\mathrm{H}\left(\mathrm{D_i}(s',t')\right)}{\mathrm{H_{max}}}\end{aligned} \tag{87}$$

4.4. Pixel Collaboration Considerations

The embedding potential gives us a tool to measure how weak against steganography an image is. Our goal as steganalyst is to reduce that weakness by using what is available in the working environment.

4.4.1. Temporal Dependency Considerations

The temporal correlation between two images is a pixel correlation.

$$\rho(I_t, I_{t+1}) = \tau \Leftrightarrow \rho\big(I_t(i,j), I_{t+1}(i,j)\big) = \tau \quad \forall (i,j) \in [\![1, M]\!] \times [\![1, N]\!] \tag{88}$$

We consider the correlation between two pixels to be represented by the pmf of $I_{t+1}(i,j)$ is symmetric around the pixel value of $I_t(i,j)$. For example, if $I_t(i,j)$ has an intensity value of *126*, then there is a 100τ percent chance that $I_{t+1}(i,j)$ has the same pixel value. We adopt a simple Gaussian model for the conditional pdf of $I_{t+1}(i,j)$ with mean the value of $I_t(i,j)$:

$$f_X(x|Y=y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-y)^2}{2\sigma^2}} \tag{89}$$

Or

$$f_X(x|Y=I_t(i,j)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-I_t(i,j))^2}{2\sigma^2}} \tag{90}$$

In order to relate to the correlation coefficient τ, we impose the conditional pdf to be maximize by τ such that:

$$\max_x\{f_X(x|Y=y)\} = \tau \tag{91}$$

This condition imposes a value on the standard deviation of $f_X(x|Y=y)$:

$$f_X(x|Y=y) = \tau \text{ for } x = y \Leftrightarrow \frac{1}{\sqrt{2\pi\sigma^2}} = \tau \tag{92}$$

This leads eventually to:

$$\sigma = \frac{1}{\sqrt{2\pi\tau^2}} \tag{93}$$

Finally the pdf of $I_{t+1}(i,j)$ is represented by the function:

$$f_X(x|Y = I_t(i,j)) = \tau \cdot e^{-\pi\tau^2\left(x - I_t(i,j)\right)^2} \tag{94}$$

Because we assume full knowledge of previously received frame within the network, $I_t(i,j)$ is expected to be fully known and thus:

$$f_X(x|Y = I_t(i,j)) = f_X(x) \tag{95}$$

Adopting a temporal correlation model proves useful as the pixel pdf goes from the uniform model $\mathcal{U}(0,255)$ to a normal model $\mathcal{N}\left(I_t(i,j), \frac{1}{2\pi\tau^2}\right)$ which undoubtedly increases the potential knowledge one can gather about the value of pixel $I_{t+1}(i,j)$. The conditional entropy $H\left(B_i(s,t)\middle|I_{N_k,i-1}(s,t)\right)$ defined earlier becomes:

\

$$\begin{aligned} &H\left(B_i(s,t)\middle|I_{N_k,i-1}(s,t)\right) \\ &= H(X) \text{ where } X \text{ is Gaussian } \mathcal{N}\left(I_{N_k,i-1}(s,t), \frac{1}{2\pi\tau^2}\right) \end{aligned} \tag{96}$$

The entropy of a random variable following a Gaussian distribution is given by:

$$\mathrm{H}(\mathrm{X}) = \frac{1}{2}(\ln(2\pi\sigma^2) + 1) \tag{97}$$

In the case we are studying, we obtain:

$$\begin{aligned} \mathrm{H}(\mathrm{X}) \quad &= \frac{1}{2}\left(\ln\left(2\pi\left(\frac{1}{\sqrt{2\pi\tau^2}}\right)^2\right) + 1\right) \\ &= \frac{1}{2}(-\ln(\tau^2) + 1) \end{aligned} \tag{98}$$

Using the conditional embedding potential derived earlier and the results obtained above, we can compute $\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k,i}} \middle| \mathrm{I}_{\mathrm{N_k,i-1}}\right)$:

$$\begin{aligned} \mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k,i}} \middle| \mathrm{I}_{\mathrm{N_k,i-1}}\right) \quad &= \frac{\sum_{s=1}^{mx}\sum_{t=1}^{nx} \mathrm{H}\left(\mathrm{B_i}(s,t) \middle| \mathrm{I}_{\mathrm{N_k,i-1}}(s,t)\right)}{\mathrm{H_{max}}} + \\ &\qquad \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny} \mathrm{H}\left(\mathrm{D_i}(s',t') \middle| \mathrm{I}_{\mathrm{N_k,i-1}}(s',t')\right)}{\mathrm{H_{max}}} \\ &= \mathrm{mx}\cdot\mathrm{nx}\cdot\frac{\frac{1}{2}(-\ln(\tau^2)+1)}{\mathrm{H_{max}}} + \frac{\sum_{s'=1}^{my}\sum_{t'=1}^{ny} \mathrm{H}\left(\mathrm{D_i}(s',t')\right)}{\mathrm{H_{max}}} \end{aligned} \tag{99}$$

If the data is independent from any other external source, then each pixel from the data is totally random and therefore the corresponding entropy is equal to H_{max}. With this piece of information, the conditional embedding potential can be rewritten as:

$$\mathrm{EP}\left(\mathrm{I}_{\mathrm{N_k,i}} \middle| \mathrm{I}_{\mathrm{N_k,i-1}}\right) = \mathrm{my}\cdot\mathrm{ny} - \mathrm{mx}\cdot\mathrm{nx}\cdot\frac{(\ln(\tau^2)-1)}{2\cdot\mathrm{H_{max}}} \tag{100}$$

4.4.2. Spatial Dependency Considerations

It is a common technique to introduce spatial correlation elements in the prediction of a pixel's value. Usually one considers a block of pixels surrounding the considered pixel as shown in Figure 39 for the case of a *3*x*3* pixel neighborhood.

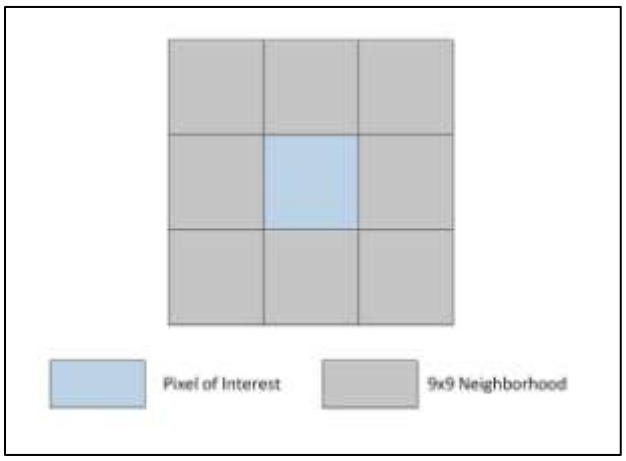

Figure 39. 3x3 block of pixels.

Spatial correlation can be used when it is assumed that there is no abrupt change of pixel value within a small neighborhood. This is usually a reasonable assumption which however has obvious shortcomings when the standard deviation of a block of pixels is large.

Spatial correlation combined with interpolation is useful to approximate the value of a given pixel. For example, it is very common to use a weighted average in order to estimate the value of the center pixel of a *3x3* block. The closest neighbors are given the most weight and the farthest the least weight.

In the same manner as with the temporal dependency, the knowledge of the neighboring pixels will give ample insight on what the center pixel value should be. We again assume that from this information, the pixel $I_{t+1}(i,j)$ will follow a Gaussian distribution whose mean is going to be the weighted average of the surrounding pixels' values:

$$E[I_{t+1}(i,j)] = \sum_{k=1}^{8} \alpha_k \cdot I_{t+1}(i,j,k) \tag{101}$$

Where $I_{t+1}(i,j,k)$ is the k^{th} neighbor of $I_{t+1}(i,j)$ and $\sum_{k=1}^{8} \alpha_k = 1$.

The variance is derived in the same manner as in the previous part where we consider a certain degree of dependency defined as the chance for $I_{t+1}(i,j)$ to be equal to its expectation $E[I_{t+1}(i,j)]$. Assuming this degree of dependency takes the value ω, the pdf of $I_{t+1}(i,j)$ is as follows:

$$f_X(x|B_{t+1}(i,j)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(x-\sum_{k=1}^{8} \alpha_k \cdot I_{t+1}(i,j,k)\right)^2}{2\sigma^2}} \tag{102}$$

Where $\sigma^2 = \frac{1}{2\pi\omega^2}$.

The conditional embedding potential could therefore be rewritten as:

$$EP\left(I_{N_k,i} \middle| I_{N_k,i-1}\right) = my \cdot ny - mx \cdot nx \cdot \frac{(\ln(\omega^2) - 1)}{2 \cdot H_{max}} \tag{103}$$

4.4.3. Combination of Temporal and Spatial Dependencies

The temporal and spatial dependencies both help reduce the uncertainty about the pixel $I_{t+1}(i,j)$ by providing a favorite estimate of its value. By considering both temporal and spatial correlations at the same time, we hope to further lessen the embedding potential and, as a consequence, improve the preventative steganalysis.

Combining both considerations is bound to have a positive, or null, effect on our objectives since it is well known that providing more information about a random variable will help reduce its entropy.

4.4.3.1 Combination of two previous models

In previous section, we considered, in the estimate of $I_{t+1}(i,j)$, using $I_t(i,j)$, pixel from the previous sample with same coordinates as $I_{t+1}(i,j)$, and $B_{t+1}(i,j)$, block of pixels from the same sample surrounding $I_{t+1}(i,j)$. Taking both $I_t(i,j)$ and $B_{t+1}(i,j)$ into account, we can derive the following entropy for $I_{t+1}(i,j)$:

$$\begin{aligned} &H(I_{t+1}(i,j)|I_t(i,j), B_{t+1}(i,j)) \\ &\leq\ min\ \{H(I_{t+1}(i,j)|I_t(i,j)), H(I_{t+1}(i,j)|B_{t+1}(i,j))\} \end{aligned} \tag{104}$$

As a consequence, this will have an effect on the embedding potential:

$$\begin{aligned} &EP(I_{t+1}(i,j)|I_t(i,j), B_{t+1}(i,j)) \\ &\leq min\{EP(I_{t+1}(i,j)|B_{t+1}(i,j)), EP(I_{t+1}(i,j)|I_t(i,j))\} \end{aligned} \tag{105}$$

The concerned pixel $I_{t+1}(i,j)$ can be expressed with a predictive model such that:

$$I_{t+1}(i,j) = f\left(\{x_i\}_{i\in[\![1,s]\!]}\right) + noise \tag{106}$$

Where $f\left(\{x_i\}_{i\in[\![1,s]\!]}\right)$ is an aggregate of the set of pixels $\{x_i\}_{i\in[\![1,s]\!]}$ and s is arbitrarily chosen depending on the number of pixels considered.

A predictive model representation is generally preferred when there is a strong correlation between the sample, in that case $I_{t+1}(i,j)$ and $\{x_i\}_{i\in[\![1,s]\!]}$. As previously

discussed, in the suggested sensor network application, high temporal correlation is to be expected and image processing studies agree that spatial dependency between adjacent pixels is very common. Therefore both temporal and spatial dependencies can be represented using the above predictive model.

Following the previous analyses, let's assume that, spatially speaking, $I_{t+1}(i,j)$ is a combination of the mean of its surrounding neighbors and a Gaussian noise of zero mean and variance $\frac{1}{2\pi\omega^2}$:

$$I_{t+1}(i,j) = mean\big(B_{t+1}(i,j)\big) + \mathcal{N}\left(0, \frac{1}{2\pi\omega^2}\right) \tag{107}$$

As for the temporal dependency, it can similarly be represented as the sum of the value of the pixel $I_t(i,j)$ and a Gaussian noise of zero mean and variance $\frac{1}{2\pi\tau^2}$:

$$I_{t+1}(i,j) = I_t(i,j) + \mathcal{N}\left(0, \frac{1}{2\pi\tau^2}\right) \tag{108}$$

Combining both predictive model, we can extend the expression of $I_{t+1}(i,j)$ to:

$$I_{t+1}(i,j) = f\big(\{x_i\}_{i\in[\![1,s]\!]}\big) + noise \tag{109}$$

Where $f\big(\{x_i\}_{i\in[\![1,s]\!]}\big) = \beta_1 \cdot I_t(i,j) + \beta_2 \cdot mean\big(B_{t+1}(i,j)\big),\ \beta_1 + \beta_2 = 1$ and $noise = \mathcal{N}\left(0, \frac{1}{2\pi\omega^2}\right) + \mathcal{N}\left(0, \frac{1}{2\pi\tau^2}\right)$.

This is equivalent to considering the following:

$$I_{t+1}(i,j) = \beta_1 \cdot X_t + \beta_2 \cdot X_s \tag{110}$$

Where $\beta_1 + \beta_2 = 1, X_t = I_t(i,j) + \mathcal{N}\left(0, \frac{1}{2\pi\tau^2}\right) and\ X_s = mean\big(B_{t+1}(i,j)\big) + \mathcal{N}\left(0, \frac{1}{2\pi\omega^2}\right)$.

Assuming the temporal and spatial spaces are independent, $I_{t+1}(i,j)$ boils down to the sum of two Gaussian independent random variables, which is also Gaussian. Then the obtained pdf for $I_{t+1}(i,j)$ given the temporal and spatial predictive models is:

$$f_{I_{t+1}(i,j)}(x|Y = I_t(i,j), B_{t+1}(i,j)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\hat{x})^2}{2\sigma^2}} \tag{111}$$

Where $\hat{x} = E[\beta_1 \cdot X_t + \beta_2 \cdot X_s]\ and\ \sigma^2 = E[(\beta_1 \cdot X_t + \beta_2 \cdot X_s - E[\beta_1 \cdot X_t + \beta_2 \cdot X_s])^2]$.

Deriving both the mean and variance, $\hat{x}\ and\ \sigma^2$, for the sum of two Gaussian random variables X_t and X_s, we get:

$$\begin{aligned} E[\beta_1 \cdot X_t + \beta_2 \cdot X_s] &= \beta_1 \cdot E[X_t] + \beta_2 \cdot E[X_s] \\ &= \beta_1 \cdot I_t(i,j) + \beta_2 \cdot \sum_{k=1}^{8} \alpha_k \cdot I_{t+1}(i,j,k) \end{aligned} \tag{112}$$

$$\begin{aligned} &E[(\beta_1 \cdot X_t + \beta_2 \cdot X_s - E[\beta_1 \cdot X_t + \beta_2 \cdot X_s])^2] \\ &= E\begin{bmatrix} {\beta_1}^2 \cdot {X_t}^2 + {\beta_2}^2 \cdot {X_s}^2 + 2\beta_1\beta_2 \cdot X_t\, X_s \\ -{\beta_1}^2 \cdot E^2[X_t] - {\beta_2}^2 \cdot E^2[X_s] + 2\beta_1\beta_2 \cdot E[X_t]\, E[X_s] \end{bmatrix} \\ &= {\beta_1}^2 \cdot E\big[{X_t}^2\big] + {\beta_2}^2 \cdot E\big[{X_s}^2\big] + 2\beta_1\beta_2 \cdot E[X_t] \cdot E[X_s] \\ &\quad -{\beta_1}^2 \cdot E^2[X_t] - {\beta_2}^2 \cdot E^2[X_s] - 2\beta_1\beta_2 \cdot E[X_t]\, E[X_s] \\ &= {\beta_1}^2 \cdot \big(E\big[{X_t}^2\big] - E^2[X_t]\big) + {\beta_2}^2 \cdot \big(E\big[{X_s}^2\big] - E^2[X_s]\big) \end{aligned} \tag{113}$$

The previous equation, once simplified, eventually leads to the final expression of the variance:

$$E[(\beta_1 \cdot X_t + \beta_2 \cdot X_s)^2] = \frac{1}{2\pi} \cdot \left[\left(\frac{\beta_1}{\tau}\right)^2 + \left(\frac{\beta_2}{\omega}\right)^2 \right] \tag{114}$$

Therefore with both the temporal dependency and the spatial dependency considered, the pdf of $I_{t+1}(i,j)$ becomes:

$$I_{t+1}(i,j) \sim \mathcal{N}\left(\beta_1 \cdot I_t(i,j) + \beta_2 \cdot \sum_{k=1}^{8} \alpha_k \cdot I_{t+1}(i,j,k), \frac{1}{2\pi} \cdot \left[\left(\frac{\beta_1}{\tau}\right)^2 + \left(\frac{\beta_2}{\omega}\right)^2 \right] \right) \tag{115}$$

Injecting the developed model into the formula for the conditional embedding potential finally leads to:

$$EP(I_{t+1}(i,j) | I_t(i,j), B_{t+1}(i,j)) = my \cdot ny + mx \cdot nx \cdot \frac{\left(1 + ln\left(\left[\left(\frac{\beta_1}{\tau}\right)^2 + \left(\frac{\beta_2}{\omega}\right)^2\right]\right)\right)}{2 \cdot H_{max}} \tag{116}$$

4.4.3.2. Extension

To extend the temporal and spatial correlation to a larger domain, we consider a rectangular box center around the desired pixel as illustrated in Figure 40.

The length and the width of the box corresponds respectively to the temporal span, also noted ts, and spatial span, also noted ss, of pixels included in the estimation of $I_{t+1}(i,j)$.

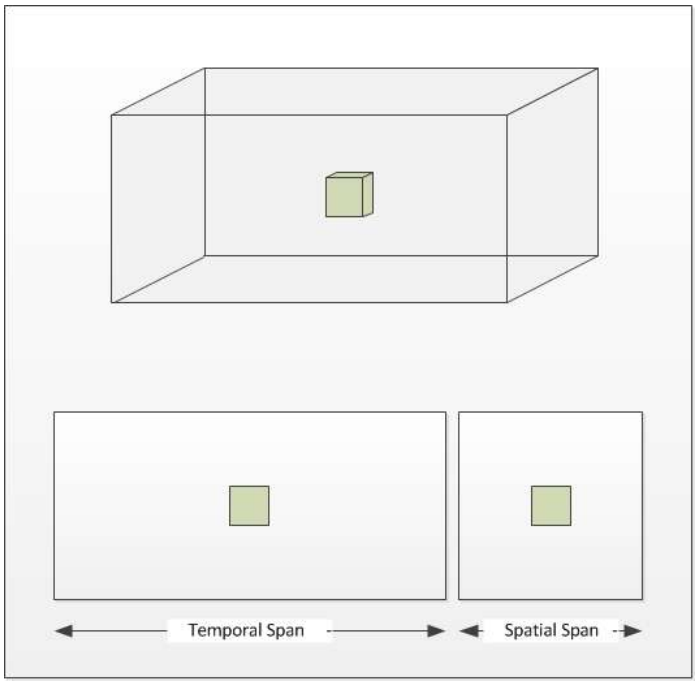

Figure 40. Temporal and spatial consideration illustrations.

Each pixel in the box will contribute with a specific strength factor in the estimation of $I_{t+1}(i,j)$, except $I_{t+1}(i,j)$ itself. In order to completely derive these factors, we choose a distance-based contribution model: the further a pixel is from $I_{t+1}(i,j)$, the smaller its collaborative factor is going to be. Each factor β_i will be normalized to that the sum of all collaborative factors adds to 1.

Let's start with the spatial contribution model and to do so let's consider a plane of $ss \times ss$ pixels, ts being odd. Because the box is centered around $I_{t+1}(i,j)$, the central position in the considered plane will yield the highest contribution factor. The other factors will decrease as the distance of their pixel from the central position increases. For this study, we compute the contribution factor around the central position (i,j), noted CF, using the following formula:

$$CF\big(I(x,y)\big) = \frac{1}{\sqrt{1+(x-i)^2+(y-j)^2}} \tag{117}$$

With the previous equation, for a 5-by-5 block of pixel, we get the spatial dependency model represented in Figure 41. Figure 42 and 43 show additional illustration of the same spatial dependency model for various spatial spans and confirm the symmetry of the model.

ρ/3	ρ/√(6)	ρ/√(5)	ρ/√(6)	ρ/3
ρ/√(6)	ρ/√(3)	ρ/√(2)	ρ/√(3)	ρ/√(6)
ρ/√(5)	ρ/√(2)	ρ	ρ/√(2)	ρ/√(5)
ρ/√(6)	ρ/√(3)	ρ/√(2)	ρ/√(3)	ρ/√(6)
ρ/3	ρ/√(6)	ρ/√(5)	ρ/√(6)	ρ/3

Figure 41. Potential distance based spatial dependency model.

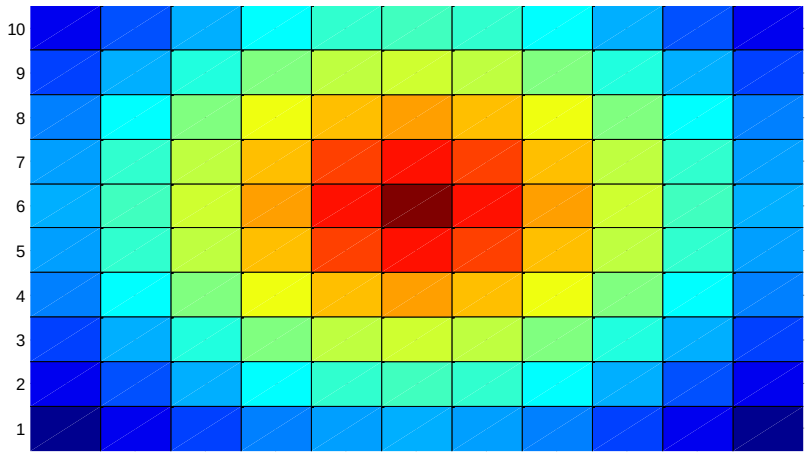

Figure 42. Illustration of the spatial dependency model using colors for a 9x9 pixel block.

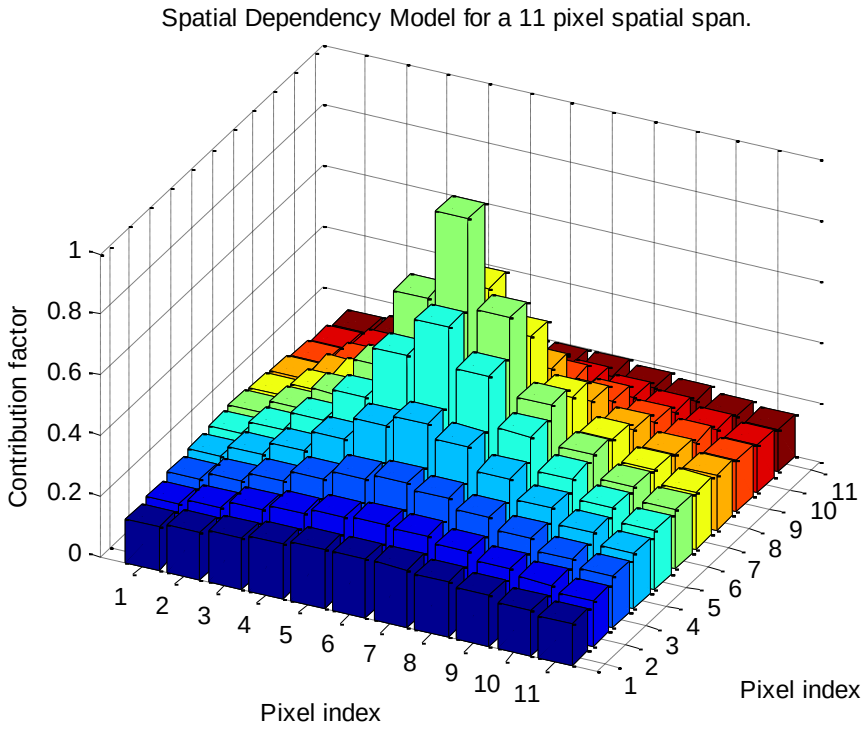

Figure 43. 3D illustration of the symmetric spatial dependency contribution model.

The temporal model is computed in the same manner as the spatial contribution model, using the same equation. We consider the set of pixels $\{I_{t+1-ts}(i,j), \dots I_{t+1+ts}(i,j)\}$ centered around $I_{t+1}(i,j)$. The highest contribution factor will, of course, be attributed to $I_{t+1}(i,j)$ (but will eventually be excluded from the computation of its estimate) such that the temporal contribution model illustrated in Figure 44 is obtained.

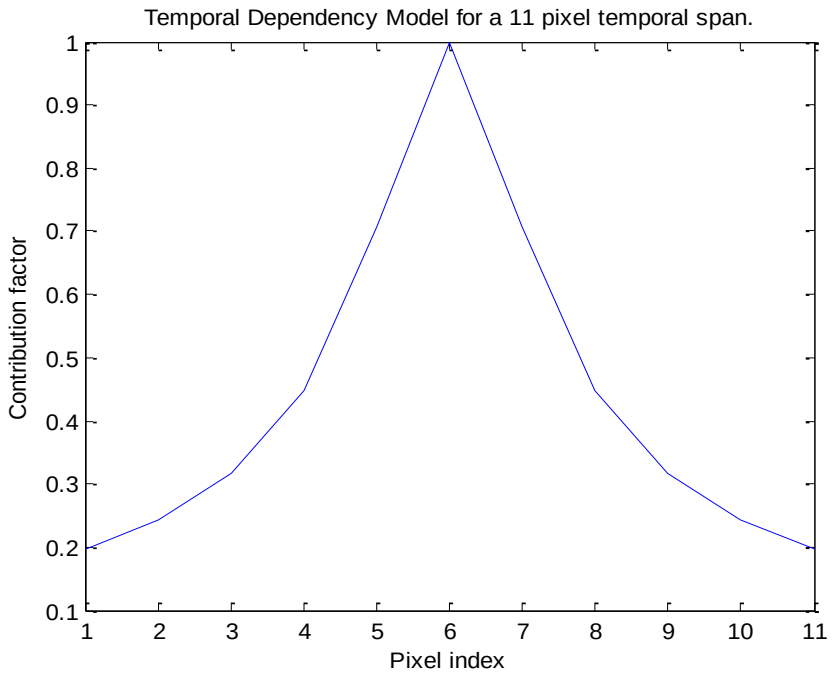

Figure 44. Illustration of the temporal dependency model.

In order to combine both models and derive the collaborating factors of every pixel, it is decided that each time sample, or plane, will follow the spatial model as defined earlier where the highest contribution factor will be replaced by the temporal contribution factor for that time sample. For example, let's suppose the temporal contribution factor for $t = T$ is τ, then the spatial model, for the sample captured at $t = T$ and a 5 pixel spatial span, can be seen in Figure 45.

τ/3	τ/√(6)	τ/√(5)	τ/√(6)	τ/3
τ/√(6)	τ/√(3)	τ/√(2)	τ/√(3)	τ/√(6)
τ/√(5)	τ/√(2)	τ	τ/√(2)	τ/√(5)
τ/√(6)	τ/√(3)	τ/√(2)	τ/√(3)	τ/√(6)
τ/3	τ/√(6)	τ/√(5)	τ/√(6)	τ/3

Figure 45. Spatial contribution model in the presence of temporal contribution τ.

We adopt a predictive model as was previously done such that the estimate of $I_{t+1}(i,j)$ becomes the combination of each contributing pixels, i.e. pixels within the considered box, and an associated noise. Using the previously used model, we can still represent $I_{t+1}(i,j)$ as:

$$I_{t+1}(i,j) = f\left(\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right) + noise \tag{118}$$

Where $f\left(\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right) = \sum_{i=1}^{ts*ss^2-1}\beta_i \cdot I_i, \sum_{i=1}^{ts*ss^2-1}\beta_i = 1$ and

$noise = \sum_{i=1}^{ts*ss^2-1}\mathcal{N}\left(0,\frac{1}{2\pi\tau_i^2}\right)$.

To simplify the notification, all contributing pixels in the box are identified as I_i. Because the box is of size $ts \times ss \times ss$, there are $ts \times ss \times ss - 1$ pixels to be considered in the estimation of $I_{t+1}(i,j)$, since $I_{t+1}(i,j)$ is not part of the estimate itself.

The computation of $I_{t+1}(i,j)$ boils down to the sum of Gaussian random variables respectively denoted as $X_i = \mathcal{N}\left(I_i, \frac{1}{2\pi\tau_i^2}\right)\ for\ i \in [\![1, ts \times ss^2 - 1]\!]$, which is also Gaussian. Then the obtained pdf for $I_{t+1}(i,j)$ given the temporal and spatial predictive models is:

$$f_{I_{t+1}(i,j)}\left(x \left| \{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right.\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\hat{x})^2}{2\sigma^2}} \tag{119}$$

Where $\hat{x} = E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right]\ and\ \sigma^2 = E\left[\left(\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i - E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right]\right)^2\right]$

Deriving both the mean and variance, $\hat{x}\ and\ \sigma^2$, we get:

$$\begin{aligned} E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right] &= \sum_{i=1}^{ts*ss^2-1} \beta_i \cdot E[X_i] \\ &= \sum_{i=1}^{ts+ss^2-1} \beta_i \cdot I_i \end{aligned} \tag{120}$$

$$\begin{aligned} &E\left[\left(\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i - E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right]\right)^2\right] = \\ &\sum_{i=1}^{ts*ss^2-1} \left(E[(\beta_i \cdot X_i)^2] - E^2[(\beta_i \cdot X_i)]\right) + \\ &2 \cdot \sum_{i=1}^{ts*ss^2-1} \sum_{j>i}^{ts*ss^2-1} \left(E\left[(\beta_i \cdot X_i) \cdot (\beta_j \cdot X_j)\right] - E[(\beta_i \cdot X_i)]\right. \\ &\left. \cdot E\left[(\beta_j \cdot X_j)\right]\right) \end{aligned} \tag{121}$$

The previous equation, once simplified, eventually leads to the final expression of the variance:

$$E\left[\left(\sum_{i=1}^{ts*ss^2-1}\beta_i\cdot X_i - E\left[\sum_{i=1}^{ts*ss^2-1}\beta_i\cdot X_i\right]\right)^2\right] = \frac{1}{2\pi} \cdot\left(\sum_{i=1}^{ts*ss^2-1}\left(\frac{\beta_i}{\tau_i}\right)^2 + 2\cdot\sum_{i=1}^{ts*ss^2-1}\sum_{j>i}^{ts*ss^2-1}\frac{\beta_i}{\tau_i}\cdot\frac{\beta_j}{\tau_j}\right) \quad (122)$$

Therefore with both the temporal dependency and the spatial dependency considered, the pdf of $I_{t+1}(i,j)$ becomes:

$$I_{t+1}(i,j)\sim\mathcal{N}\left(\sum_{i=1}^{ts*ss^2-1}\beta_i\cdot I_i\,,\frac{1}{2\pi}\cdot\left(\sum_{i=1}^{ts*ss^2-1}\left(\frac{\beta_i}{\tau_i}\right)^2 + 2\cdot\sum_{i=1}^{ts*ss^2-1}\sum_{j>i}^{ts*ss^2-1}\frac{\beta_i}{\tau_i}\cdot\frac{\beta_j}{\tau_j}\right)\right) \quad (123)$$

Injecting the developed model into the formula for the conditional embedding potential finally leads to:

$$EP\left(I_{t+1}(i,j)\,\middle|\,\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right) = my\cdot ny + mx\cdot nx \cdot\frac{\left(1+ln\left(\sum_{i=1}^{ts*ss^2-1}\left(\frac{\beta_i}{\tau_i}\right)^2 + 2\cdot\sum_{i=1}^{ts*ss^2-1}\sum_{j>i}^{ts*ss^2-1}\frac{\beta_i}{\tau_i}\cdot\frac{\beta_j}{\tau_j}\right)\right)}{2\cdot H_{max}} \quad (124)$$

4.5. Insight for Steganalysis

Developing the equation for the embedding potential allows us to quantify with boundaries the risk for steganography to occur but it also provides a way to quantify how effective an estimate of the original data (in our case, $I_{t+1}(i,j)$) can be.

More clearly, if the conditional embedding potential $EP(X|Y)$ is deemed low enough according to specifics set by the network's users, then Y is a legitimate candidate for estimating X and can therefore be used for comparison purposes.

For example, if the steganographic detector consists in evaluating the correlation between the original data X and its estimate to a threshold λ, and if $EP(X|Y) \leq \theta$, then it can be recommended to look at the following equation in order to decide on the presence of steganography:

$$Is \;\; \rho(X,Y) \leq \lambda \,? \tag{125}$$

In the previous section, Y was a weighted average of past image samples represented as Gaussian noises:

$$Y = \mathcal{N}\left(E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right], E\left[\left(\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i - E\left[\sum_{i=1}^{ts*ss^2-1} \beta_i \cdot X_i\right]\right)^2\right]\right) \tag{126}$$

Simulations are later presented in Section 4.6. to assess the legitimacy of the proposed estimate *Y*.

4.6. Simulations

4.6.1. Embedding Potential vs. Temporal and Spatial Considerations

In order to assess the behavior of the embedding potential when surrounding pixels and frames are considered in the estimation of the current frame, we have implemented a simulation in the Matlab environment. The experiments are conducted on a set of 100x100 frames. Without loss of generality, we assume that only the current frame (the one we compute the embedding potential for) includes a data part. In the case where we would assume other past frames contain data, we would only need to exclude the concerned pixels from the computation of the embedding potential.

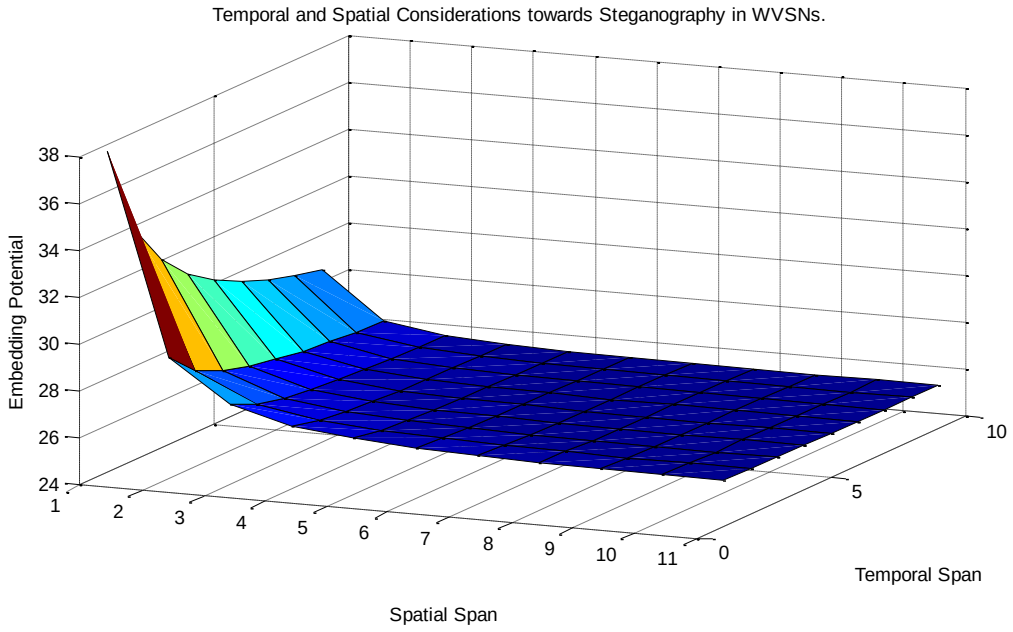

Figure 46. Effects of temporal and spatial considerations on the embedding potential.

What Figure 46 shows also is that there is no considerable improvement when too many temporal or spatial samples are considered in the estimation of the current frame's pixels.

The experiment shows that considering five previous samples and a 5x5 neighborhood around the desired pixel is more than enough to reduce the embedding potential efficiently.

4.6.2. Steganalysis Contribution

Using the knowledge gained from the previous simulation, we aim to see concretively what improvement could be gained from the computation of the embedding potential towards steganography. To do so, we consider that the steganalyst uses the correlation between the studied frame and its estimate to decide whether steganography is present in the media.

For comparison purposes, we choose two scenarii:

> Scenario 1. We compute the correlation between frame at time t (corrupted and steganography-free) and the previous frame at time t-1.
> Scenario 2. We compute the correlation between frame at time t (corrupted and steganography-free) and the estimate obtained after spatial and temporal contributions have been considered.

Simulations are conducted to find the correlation coefficient, derived from the image background, between the frame of interest (assumed to occur at time t) and its estimate obtained according to both scenarii. We use video sequences with a low sampling rate to simulate captures from the network and put aside the foreground to compute the correlation between the backgrounds of the studied frame and its estimate. Results obtained from the experiments are represented in the following figures.

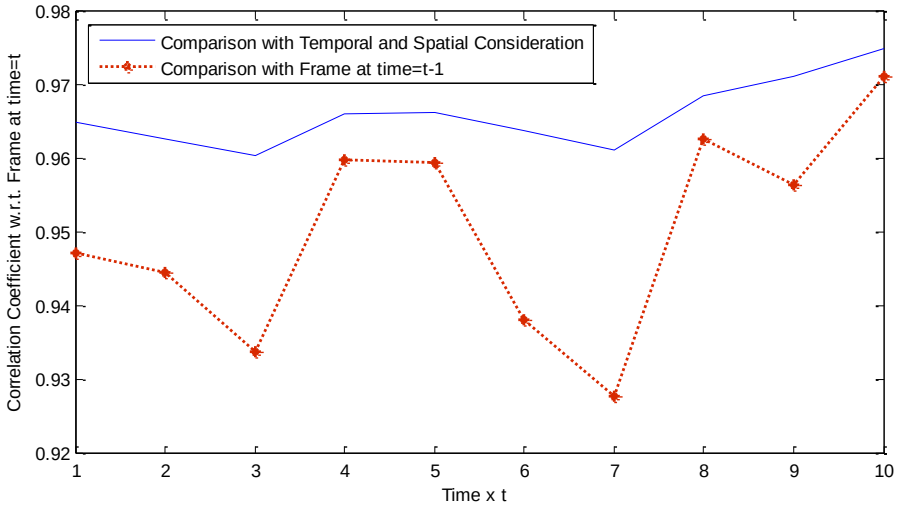

Figure 47. Average frame correlation between uncorrupted frame at time *t* and its estimate in scenarii 1 and 2. The average is computed over the set of sequences used for the experiment.

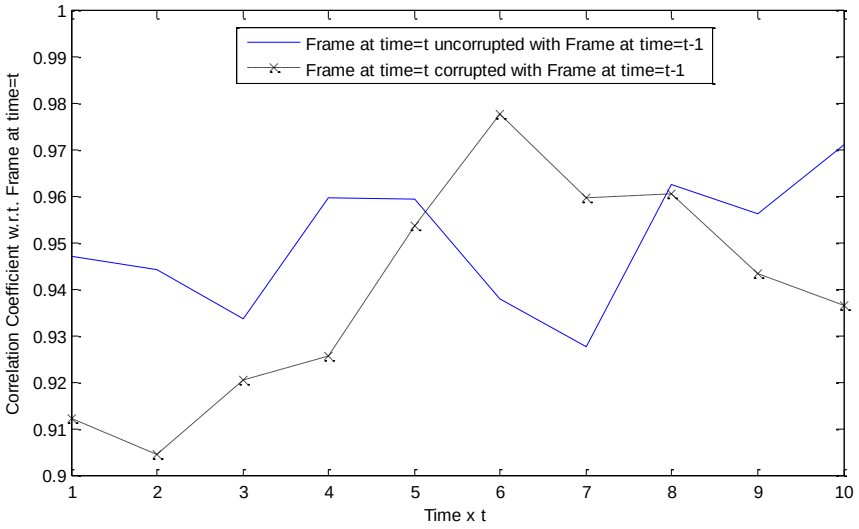

Figure 48. Average correlation coefficient between frame at time *t* and frame at time *t-1* in scenario 1.

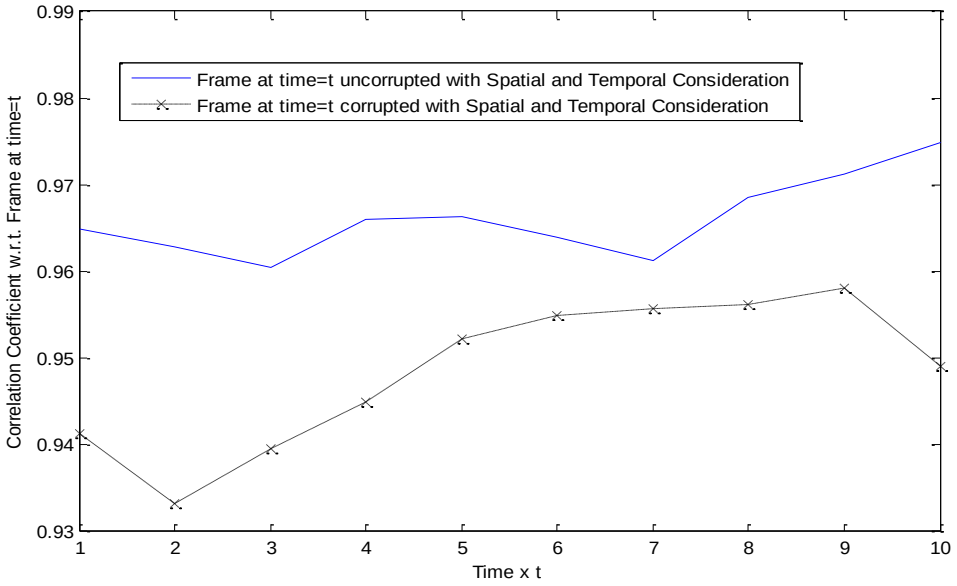

Figure 49. Average correlation coefficient between frame at time *t* and the estimated frame obtained when spatial and temporal contributions are considered in scenario 2.

First, Figure 47 shows the slight improvement in correlation that is obtained when the spatial and temporal pixel contributions are involved: although frame at time t and frame at time t-1 are consecutive, their correlation is less than the one obtained with the weighted average estimate. But more importantly it shows also that variations of the correlation coefficient are smaller in Scenario 2. Such improvement is important to reduce the chance of false positive and false negative rates.

Figure 48 shows what the output of the correlation based detector would be in Scenario 1. Clearly, both cases of corrupted or steganography-free frames vary importantly leading to obvious false decisions from the steganalyst. Figure 49 on the other hand shows the output of the correlation based detector in Scenario 2. In this latter scenario, the cases of corrupted and steganography-free frames are clearly separated which translates in better detection performances for the steganalyst.

Globally what the simulations show is that the embedding potential can therefore be used as a tool to quantify the risk for steganography to occur but also as a selector of better estimate for the purpose of steganalytic comparison. Whatever panel of solutions $\{Y_i\}$ the steganalyst comes up with in order to estimate the current frame X, computing the embedding potential for each proposed Y_i, $EP(X|Y_i)$, will help select the better candidate which should correspond to Y_{best} :

$$Y_{best} \ \ such\ that \ \ EP(X|Y_{best}) = \min_i(EP(X|Y_i)) \qquad (127)$$

4.7. Performance Improvement

So far, we have shown that sensor networks provide a very advantageous ground for steganalysis. Indeed, using the embedding potential as a measure of steganographic threats, we have shown that over time, the expected high redundancy between samples helps reduce the potential for steganography. Even if it is obvious after analysis that incorporating past image samples in the computation of the embedding potential decreases further the risk for steganography to occur, we should aim at improving the steganalysis even more.

Our initial goal is a trade-off between our previously detailed multi-point and single-point steganalyses. Combining both algorithms in the computation of the embedding potential proves to have a positive steganalytic effect.

Indeed, we showed earlier that it is possible to differentiate the data, or foreground, from the background thanks to the redundancy between captured frames. With both parts distinctly separated it is possible to apply our distributed processing scheme on the data only while keeping the background as it is. Doing so will have two advantages:

- The embedding potential of the image section containing the data can be reduced while minimizing the amount of processing to a small part of the image.

- The computation embedding potential coming from the background remains highly efficient without processing since it is based on frame comparison and individual pixel distributions.

By choosing this solution, it is possible to slightly improve on the embedding potential derived from the data while the embedding potential from the background remains identical. Therefore the initial embedding potential equations can be rewritten as follows:

$$EP\left(I_{t+1}(i,j)\left|\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right.\right) = \frac{\sum_{s\prime=1}^{my}\sum_{t\prime=1}^{ny}\mathrm{H}\left(\mathrm{D_i}(s',t')\right)}{\mathrm{H_{max}}} + mx\cdot nx\cdot\frac{\left(1+ln\left(\sum_{i=1}^{ts*ss^2-1}\left(\frac{\beta_i}{\tau_i}\right)^2+2\cdot\sum_{i=1}^{ts*ss^2-1}\sum_{j>i}^{ts*ss^2-1}\frac{\beta_i}{\tau_i}\cdot\frac{\beta_j}{\tau_j}\right)\right)}{2\cdot H_{max}} \quad (128)$$

Where:

$$\frac{\sum_{s\prime=1}^{my}\sum_{t\prime=1}^{ny}\mathrm{H}\left(\mathrm{D_i}(s',t')\right)}{\mathrm{H_{max}}} = my\cdot ny\cdot\frac{H_{processing}}{H_{max}} \quad (129)$$

For example, let's assume that the processing transforms the grayscale data into its black and white version. Then we would have:

$$H_{max} = log2(256) = 8 \;\; and \;\; H_{proc} = log2(2) = 1 \quad (130)$$

This means that the embedding potential derived from the data would decrease from $my \cdot ny$ to $\frac{my \cdot ny}{8}$ and therefore, whatever processing is done on the transmission path from the capturing node to the base station, we get:

$$EP\left(I_{t+1}(i,j)\left|\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right.\right)_{w/o\ proceesing} \leq EP\left(I_{t+1}(i,j)\left|\{I_i\}_{i\in[\![1,ts*ss^2-1]\!]}\right.\right)_{with\ \ processing} \tag{131}$$

From the attacker's point of view, two choices are presented: either embed the message in the background with greyscale pixel values and high variance or embed the data in the foreground (data) which represents the most unknown part of the image for the steganalyst but also represents the part with the least entropy globally since it would have been processed.

The choice remains the attacker's but if he, or she, do decide to hide the data in the background, thanks to the high correlation in the background between frames, the steganalyst will have an easier job at identifying the presence of steganography.

If, however the attacker chooses to hide the secret data in the foreground of the cover-image, their effective covert communication bandwidth would be greatly limited from the reduced size of the foreground first of all but mainly due to the prior processing.

The reason why we chose not to process the background also is that there is no assurance that the embedding potential related to the background will improve if we do so. In the worst case scenario it could indeed become worse and therefore be an hindrance to the steganalysis. Indeed, if it has been proven that:

$$H(X|Y) \geq H(f(X)|Y) \tag{132}$$

It is not possible to extend it to the following inequality:

$$H(X|Y) \overset{\geq}{?} H(f(X)|f(Y)) \quad (133)$$

The reason is that processing Y can remove knowledge about X and therefore reduce the conditional entropy:

$$H(X|Y) \leq H(X|f(Y)) \quad (134)$$

From simulations on the Matlab platform, the correlation between frames of the same sequence actually is significantly reduced between grayscale images and black & white images. We considered the several sequences of 60 frames and averaged the correlation factor between each frame and the 30th frame of each sequence. Results are presented in Figure 50.

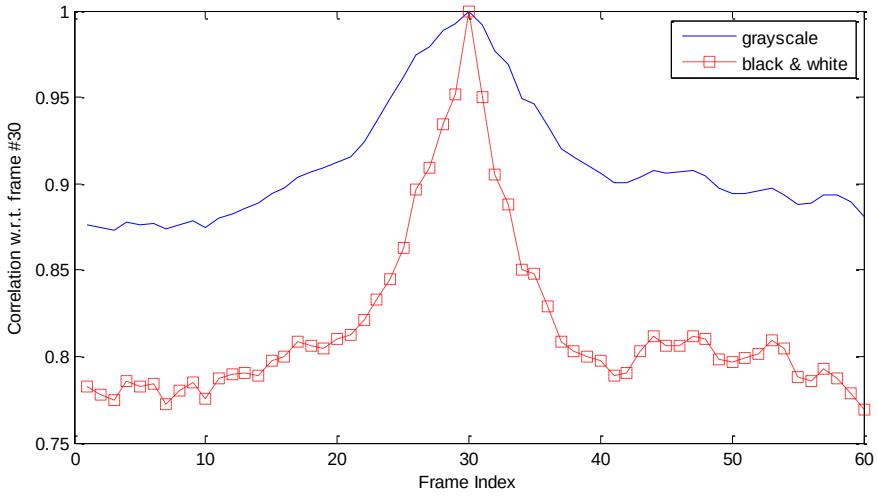

Figure 50. Correlation coefficient with respect to the 30th frame in 60-frame-long sequences.

Figure 50 shows that processing the image, although reducing the bandwidth for embedding can quite significantly reduce the correlation between frames and therefore be a hindrance when the steganalyst uses frame comparison towards steganalysis.

This is the reason why we chose to keep the background, which represents a fertile support for comparative purposes, as it is while processing the data part of the image.

4.8. Conclusion

In this section, we have tackled the novel problem of covert-communications in sensor networks. We preferred an approach that would discourage a steganographic attack even before it occurs by introducing the concept of preventative steganalysis. The idea is to gather as much information as possible and give the steganalyst enough tools so that if steganography occurs, it will be easily detected. We propose to do so by using the natural behavior of sensor networks, especially the high redundancy between captured samples to develop a steganalysis that improves with time and offer three steganalytic solutions, aiming for the steganalyst to be as passive as possible in order not to disturb the primary function of the network. We measure the risk of embedding content based on the embedding potential which represents how much 'un-knowledge' the network has about the captured samples. We further develop the steganalytic solution by processing the part of the data that is most unknown to the steganalyst in order to reduce the potential embedding bandwidth of the attacker.

5. OVERALL CONCLUSION

Although steganalysis in still images has been around for some time now and has been the subject of many research papers, other fields are still in high demand for protection against covert-communications. In this dissertation, we tried to fill some of the existing by developing steganalytic measures for videos and sensor networks.

We first looked at video steganalysis believing that the high correlation between successive frames should be exploited in order to detect the presence of watermark. This is the foundation of our algorithm MoViSteg which uses motion vectors for frame interpolation and estimation. The particularity of MoViSteg compared to other algorithms is that it does not borrow from existing still image steganalytic techniques but takes fully advantage of the cover-media being a video. The decision on whether data is embedded in the video is not made frame by frame but by considering a subset of the video and utilizing the high frame correlation to do so. MoViSteg has been developed for a company specializing in security but has also been published and well received by the community.

The excitement surrounding the field of sensor networks made us curious about sensor network steganalysis. The subject had not been specifically treated but the obvious interest these networks would represent for malicious activity showed how needed a steganalytic solution was. We introduced the concept of preventative steganalysis in order to discourage the potential for steganography. We did so by showing how limited the capacity for embedding can be in sensor networks with or without additional processing from the steganalyst.

Both video and visual sensor network images provide high redundancy and high bandwidth making these cover-media very attractive to steganographic attacks. Our developed steganalytic solutions provide a good basis for future studies and satisfying protection for today's malicious activities. However, attacks and covert communication techniques evolve every day and become more elaborate. It is obvious that steganalysis

and steganography fight an endless battle that continuously escalates in complexity. Therefore the steganalyst is always challenged and no solution he, or she, derives, however efficient it is at the time it was formulated, can expect to deter attacks forever. Thus, we hope that our work will prove to be useful in deriving future steganalytic solutions against new embedding techniques.

REFERENCES

[1] Adams J., "The next world war: computers are the weapons and the front line is everywhere", Simon and Schuster, 1998.

[2] Wang H., Wang S., "Cyber warfare: steganography vs. steganalysis". Communications ACM 47, vol. 10, pp. 76-82, October 2004.

[3] Carr J., "Inside cyber warfare: mapping the cyber underworld" (1st ed.), O'Reilly Media, Inc., 2009.

[4] Chertoff, M., "The cybersecurity challenge", Regulation & Governance, pp. 480–484, 2008

[5] Assante M. J., Tobey D. H., "Enhancing the cybersecurity workforce", IT Professional 13, vol.1, pp. 12-15, January 2011.

[6] Janczewski L. J., Colarik A. M., "Cyber warfare and cyber terrorism" (1st ed.). IGI Publishing, Hershey, PA, USA, 2007.

[7] Kolata G., "Veiled messages of terrorists may lurk in cyberspace", New York Times, October 2001.

[8] Kelly, J., "Terror groups hide behind web encryption", USA Today, http:// www.usatoday.com /life/cyber/tech/2001-02-05-binladen.htm, 2 May 2001.

[9] Gallagher S., "Steganography: how al-Qaeda hid secret documents in a porn video", Arstechnica.com, May 2012.

[10] Federal Plan for Cyber Security and Information Assurance Research and Development, National Science and Technology Council, April 2006.

[11] Ahsan K., Kundur D., "Practical data hiding in TCP/IP", Proceedings of the Workshop on Multimedia Security at ACM Multimedia '02, December 2002.

[12] Murdoch S. J., Lewis S., “Embedding covert channels in TCP/IP”, Proceedings of the 7th International Workshop on Information Hiding, June 2005.

[13] Buchegger S., Boudec J.-Y.L., “Nodes bearing grudges: towards routing security, fairness and robustness in mobile ad hoc networks”, Proceedings of the Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp.403-410, 2002.

[14] Karlof C., Wagner D., “Secure routing in wireless sensor networks: attacks and countermeasures”, IEEE International Workshop on Sensor Network Protocols and Applications, pp. 113-127, 2002.

[15] Rogers, R., Devost, M., “Hacking a terror network: the silence threat of covert channels”, Syngress Editions, 2005.

[16] Owens, M., “A discussion of covert channels and steganography”, SANS Institute whitepaper, 2002.

[17] Wayner, P., “Disappearing cryptography, information hiding: steganography and watermarking” (2nd Edition), Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002.

[18] Cachin, C., "An information-theoretic model for steganography”, Information Hiding: 2nd International Workshop. Lecture Notes in Computer Science, Vol. 1525. Springer-Verlag, Berlin Heidelberg New York, pp. 306–318, 1998.

[19] Johnson N. F., Jajodia S., “Steganalysis: the investigation of hidden information”, IEEE Information Technology Conference, Syracuse, New York , USA, pp. 113-116, 1-3 September 1998.

[20] Chandramouli, R., Kharrazi, M. & Memon, N., “Image steganography and steganalysis: concepts and practice”, Proceedings of the 2nd International Workshop on Digital Watermarking, October 2003.

[21] Herodotus, “The History”, translated by David Grene, University of Chicago Press, 1987.

[22] Fabien A., Peticolas P., Anderson R. J., Kuhn M. G., "Information hiding - a survey”, IEEE special issue on Protection of Multimedia Content, vol. 87, No. 7, pp. 1062-1078, July 1999.

[23] Johnson N. F., Duric Z., Jajodia S., “Information hiding: steganography and watermarking : attacks and countermeasures”, 2001.

[24] Katzenbeisser S., Petitcolas F. A., “Information hiding techniques for steganography and digital watermarking”, Artech House, Inc., Norwood, MA, 2000.

[25] Kipper G., “Investigator's guide to steganography”, CRC Press, Inc., Boca Raton, FL, 2003.

[26] Johnson N., “Steganography tools and software”, online resource, http://www.jjtc.com/Steganography/toolmatrix.html, 2002.

[27] Westfeld A., Pfitzmann A., “High capacity despite better steganalysis”, Proceedings of the 4th International Workshop on Information Hiding, vol. 2137, 2001.

[28] Westfeld A., “F5 – a steganographic algorithm”, Proceedings of the 4th International Workshop on Information Hiding, pp. 289–302, 2001.

[29] Provos N., “Defending against statistical steganalysis”, Proceedings of the 10th USENIX Security Symposium pp. 323–336, August 2001.

[30] Fridrich J., Goljan M., Hogea D., “Attacking the Outguess”, Proceedings of the ACM Workshop on Multimedia and Security., vol. 71, December 2002.

[31] Fridrich J., Goljan M., Hogea D., “Steganalysis of JPEG images : Breaking the F5 algorithm”, 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, pp. 310–323, October 2002.

[32] Westfeld A., Pfitzmann A., “Attacks on steganographic systems”, Proceedings of the 3rd International Workshop on Information Hiding, 1999.

[33] Zhang L., Wang H., Wu R., “A high-capacity steganography scheme for JPEG2000 baseline system”, Transactions on Image Processing, vol. 18, pp. 1797-1803, August 2009.

[34] Sarreshtedari S., Ghaemmaghami, S., "High capacity image steganography in wavelet domain", 7th IEEE Consumer Communications and Networking Conference (CCNC), pp.1-5, 9-12 January 2010.

[35] Al-Ataby A. A., Al-Naima F. M., "High capacity image steganography based on curvelet transform", Developments in E-systems Engineering (DeSE), pp.191-196, 6-8 December 2011.

[36] Chang C. C., Chou Y. C., Kieu T. D., “An information hiding scheme using Sudoku”, Proceedings of the Third International Conference on Innovative Computing, Information and Control, June 2008.

[37] Roshan S. B. R., Rohith J., Mukund V., Rohan H., Shanta R., “Steganography using Sudoku puzzle”, Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCOM '09), IEEE Computer Society, Washington, DC, USA, pp. 623-626, 2009.

[38] Sanmitra I, Shivananda P., Shrikant B., Usha B. A., “Image steganography using Sudoku puzzle for secured data transmission”, International Journal of Computer Applications, New York, USA, vol. 48, pp. 31-35, June 2012.

[39] Hao B., Zhao L.-Y., Zhong W.-D., "A novel steganography algorithm based on motion vector and matrix encoding," IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 406-409, 27-29 May 2011

[40] He X., Luo Z., "A novel steganographic algorithm based on the motion vector phase", Proceedings of the 2008 International Conference on Computer Science and Software Engineering, IEEE Computer Society, Washington, DC, USA, vol. 3, pp. 822-825, 2008

[41] Westfeld, Pfitzmann, "Attacks on steganographic systems", Proceedings of the 3rd International Workshop on Information Hiding, 1999.

[42] Fridrich J., Goljan M., Du R., "Reliable detection of LSB steganography in color and grayscale images", Proceedings of the ACM Workshop on Multimedia and Security, pp. 27–30, 2001.

[43] Fridrich J., Goljan M., Du R., "Steganalysis based on JPEG compatibility", Proceedings of the SPIE Multimedia Systems and Applications IV, pp. 275–280, August 2001.

[44] Lyu S., Farid H., "Detecting hidden messages using higher-order statistics and support vector machines", Proceedings of the 5th International Workshop on Information Hiding, 2002.

[45] Avcibas I., Memon N., Sankur B., "Steganalysis using image quality metrics", IEEE Transactions on Image Processing, vol. 12, no. 2, pp. 221–229, 2002.

[46] Sullivan K., Madhow U., Chandrasekaran S., Manjunath B. S., "Steganalysis of spread spectrum data hiding exploiting cover memory", Proceedings of the 17th Annual Symposium on Electronic Imaging Science Technology, pp.38-46, January 2005.

[47] Sullivan K., Madhow U., Chandrasekaran S., Manjunath B. S., "Steganalysis for Markov cover data with applications to images", IEEE Transactions on Information Forensics and Security, vol. 1, issue 2, pp.275-287, June 2006.

[48] Chamorro A. G. H., Miyatake M. N., "A new methodology of image steganalysis including for JPEG steganography", Electronics, Robotics and Automotive Mechanics Conference (CERMA), pp. 434-438, 28 September – 1 October 2010.

[49] Zong H., Liu F.-L., Luo X.-Y., "Blind image steganalysis based on wavelet coefficient correlation", Digital Investigation, vol. 9, issue 1, pp. 58-68, June 2012.

[50] Budhia U., Kundur D., "Video steganalysis using collusion sensitivity", Proceedings of the SPIE, Sensors, Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense, Orlando, Florida, vol. 5403, pp. 210-221, April 2004.

[51] Budhia U., Kundur D., Zourntos T., "Digital video steganalysis exploiting statistical visibility in the temporal domain", IEEE Transactions on Information Forensics and Security, vol. 1, issue 4, pp. 502-516, 2006.

[52] Jainsky J. S., Kundur D., Halverson D. R., "Digital video steganalysis", research paper written for Sagem Morpho Inc., July 2007.

[53] Jainsky J. S., Kundur D., Halverson D. R., "Towards digital video steganalysis using asymptotic memoryless detection," Proceedings of the ACM Multimedia and Security Workshop, Dallas, Texas, pp. 161-168, September 2007.

[54] Thu Thu Htet and Khin Than Mya, "Video steganalysis using statistical features and Bayes classifier", International Journal of Research and Reviews in Computer Science (IJRRCS), vol. 3, no. 2, pp. 1590-1592, April 2012.

[55] Su Y., Zhang C., Wang L., "A new video steganalysis based on mode detection", Proceedings of the International Conference on Audio, Language and Image Processing, pp. 1507– 1510, Shanghai, China, July 2008.

[56] Pankajakshan V., Ho A. T. S., "Improving video steganalysis using temporal correlation", Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, vol. 1, pp. 287-290, November 2007.

[57] Turner M., “Sensor-network startup has customers, unveils products”, Sacramento Business Journal, December 2006.

[58] Huang M.-Y., Jasper R. J., Wicks T. M., “A large scale distributed intrusion detection framework based on attack strategy analysis”, Computer Networks Journal, vol.31, pp. 2465-2475, December 1999.

[59] da Silva A. P. R., and Martins M. H. T., Rocha B. P. S., Loureiro A. A. F., Ruiz L. B., Wong H. C., “Decentralized intrusion detection in wireless sensor networks”, Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks, pp. 16-23, 2005.

[60] Agah A., Das S. K., Basu K., Asadi M., “Intrusion detection in sensor networks: a non-cooperative game approach”, 3rd IEEE International Symposium on Network Computing and Applications, pp. 343-346, 2004.

[61] Mainwaring A., Culler D., Polastre J., Szewczyk R., Anderson J., “Wireless sensor networks for habitat monitoring”, Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88-97, 2002.

[62] Law Y. W., van Hoesel L., Doumen J., Hartel P., Havinga P., “Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols”, Proceedings of the 3rd ACM workshop on Security of Ad Hoc and Sensor Networks, pp. 76-88, 2005.

[63] Manzo M., Roosta T., Sastry S., “Time synchronization attacks in sensor networks”, Proceedings of the 3rd ACM workshop on Security of Ad Hoc and Sensor Networks, pp. 107-116, 2005.

[64] Xu W., Ma K., Trappe W., Zhang Y., “Jamming sensor networks: attack and defense strategies”, IEEE Network Magazine, pp.41-47, May-June 2006.

[65] Lo B. P. L., Wang J. L., and Yang G.-Z, “From imaging networks to behavior profiling: Ubiquitous sensing for managed homecare of the elderly", Adjunct

Proceedings of the 3rd International Conference on Pervasive Computing, Munich, Germany, pp. 101–104, May 2005.

[66] Fidaleo D. A., Nguyen H.-A., and Trivedi M., "The networked sensor tapestry (NeST): A privacy enhanced software architecture for interactive analysis of data in video-sensor networks", Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks, New York, USA, pp. 46–53, October 2004.

[67] Wickramasuriya J., Datt M., Mehrotra S., Venkatasubramanian N., "Privacy protecting data collection in media spaces", Proceedings of the 12th annual ACM International Conference on Multimedia, New York, USA, pp. 48–55, October 2004.

[68] Kundur K., Luh W., Okorafor U.N., Zourntos T., "Security and privacy for distributed multimedia sensor networks," Proceedings of the IEEE Special Issue on Distributed Multimedia, vol. 96, no. 1, pp. 112-130, January 2008.

[69] Feng J., Potkonjak M.,"Security in sensor networks: watermarking techniques", Proceedings of the SPIE, Security and Watermarking of Multimedia Contents V, vol. 5020, pp. 391-402, June 2003.

[70] Zheng J., Li J., Lee M. J., Anshel M., "A lightweight encryption and authentication scheme for wireless sensor networks", International Journal of Security and Networks, vol.1, Issue 3, pp. 138-146, December 2006.

[71] Arazi B., Elhanany I., Arazi O., Qi H., "Revisiting public-key cryptography for wireless sensor networks", IEEE Computer Journal, vol. 38, issue 11, pp. 103-105, November 2005.

[72] Gaubatz G., Kaps J.-P., Sunar B., "Public key cryptography in sensor networks – revisited", 1st European Workshop on Security in Ad-Hoc and Sensor Networks, 2004.

[73] Karlof C., Sastry N., Wagner D., "TinySec: a link layer security architecture for wireless sensor networks", 2nd CAN Conference on Embedded Networked Sensor Systems, November 2004.

[74] Oliveira L. B., Aranha D., Morais E., Daguano F., Lopez J., Dahab R., "TinyTate: identity-based encryption for sensor networks", Cryptology ePrint Archive, 2007.

[75] Martinovic I., Pichota P., Schmitt J. B., "Jamming for Good: A Fresh Approach to Authentic Communication in WSNs", Proceedings of the 2nd ACM Conference on Wireless Network Security, Zurich, Switzerland, pp. 161-168, March 2009.

[76] Blaß E., Wilke J., Zitterbart M., "Relaxed authenticity for data aggregation in wireless sensor networks", Proceedings of the 4th international Conference on Security and Privacy in Communication Networks, Istanbul, Turkey, pp. 1-10, September 2008.

[77] Van Trees H. L., "Detection, estimation, and modulation theory. Part I, detection, estimation, and linear modulation theory", Wiley, 2001.

[78] Lehmann E. L., Romano J. P., "Testing statistical hypotheses", 3rd edition, Springer Texts in Statistics, 2005.

[79] Halverson D. R., Wise G. L., "A detection scheme for dependent noise processes", Journal of the Franklin Institute, vol. 309, pp. 287-300, May 1980.

[80] Halverson D. R., Wise G. L., "Asymptotic memoryless detection of random signals in dependent noise", Journal of the Franklin Institute, vol. 312, pp. 13-29, July 1981.

[81] Bertsekas D. P., "Constrained optimization and Lagrange multiplier methods", Academic Press, 1982.

[82] Online resource, "mathworld.wolfram.com/GaussianDistribution.html".

[83] Kancherla K., Mukkamala S., "Video steganalysis using motion estimation", Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN'09). IEEE Press, Piscataway, NJ, USA, pp. 3129-3134, 2009.

[84] Pevny T., Fridrich J., "Merging Markov and DCT features for multiclass JPEG steganalysis", Proceedings SPIE, Electronic Imaging, Security, Steganography, and

Watermarking of Multimedia Contents IX, vol. 6505, pages 3 1–3 14, San Jose, CA, 29 January - 1 February 2007.

[85] Jain M. K., “Wireless sensor networks: security issues and challenges”, International Journal of Computer and Information Technology (IJCIT), vol. 2, issue 1, 2011.

[86] Xu N., Sun Y., Huang B., Yu J., "An energy-efficient cross-layer framework for security in wireless sensor networks," Fourth International Symposium on Knowledge Acquisition and Modeling (KAM), pp.121-124, 8-9 October 2011.

[87] Han Y., Li H., Qiu J., "The analysis and summary about energy saving technologies of wireless sensor network", International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), pp.883-885, 12-14 August 2011.

[88] Ji K., Kim C., Kim S., "Implementation of energy-efficient node management in wireless sensor networks", Second International Conference on Future Generation Communication and Networking, 2008, pp.324-327, 13-15 December 2008.

[89] Pensas H., Valtonen M., Vanhala J., "Wireless sensor networks energy optimization using user location information in smart homes", International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 351-356, 26-28 October 2011.

[90] Chiasserini C.-F, Magli E., “Energy consumption and image quality in wireless video-surveillance networks”, Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 5, Lisbon, Portugal, pp. 2357-2361, September 2002.

[91] Jainsky J. S., Kundur D., “Visual sensor network processing and preventative steganalysis", chapter for the book: Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications, October 2010.

[92] Jainsky J. S., Kundur D., "Preventative steganalysis in wireless sensor networks: challenges and solutions", IEEE International Conference on Multimedia & Expo – Content Protection and Forensics Workshop, Barcelona, Spain, July 2011.

[93] Jainsky J. S., Kundur D., "Towards preventative steganalysis in wireless visual sensor networks", International Journal of Multimedia Technology, accepted June 2012.

[94] Cover, T.M. and Thomas, J.A., "Elements of information theory". John Wiley & Sons, New York, NY, 1991.