

Spring 2011

Data Hiding in Digital Video

Abdullah Cay
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Cay, Abdullah. "Data Hiding in Digital Video" (2011). Doctor of Philosophy (PhD), dissertation, Engineering and Technology, Old Dominion University, DOI: 10.25777/kmnq-nd55
https://digitalcommons.odu.edu/ece_etds/56

This Dissertation is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

DATA HIDING IN DIGITAL VIDEO

by

Abdullah Cay

B.S. August 1992, Turkish Army Academy, Turkey

M.S. December 1999, Naval Postgraduate School

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

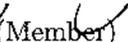
ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY

May 2011

Approved by:


Dimitrie C. Popescu (Director)


Jiang Li (Member)


Yuzhong Shen (Member)


Dean J. Krusienski (Member)

ABSTRACT

DATA HIDING IN DIGITAL VIDEO

Abdullah Cay

Old Dominion University, 2011

Director: Dr. Dimitrie C. Popescu

With the rapid development of digital multimedia technologies, an old method which is called steganography has been sought to be a solution for data hiding applications such as digital watermarking and covert communication. Steganography is the art of secret communication using a cover signal, e.g., video, audio, image etc., whereas the counter-technique, detecting the existence of such as a channel through a statistically trained classifier, is called steganalysis.

The state-of-the art data hiding algorithms utilize features, such as Discrete Cosine Transform (DCT) coefficients, pixel values, motion vectors etc., of the cover signal to convey the message to the receiver side. The goal of embedding algorithm is to maximize the number of bits sent to the decoder side (embedding capacity) with maximum robustness against attacks while keeping the perceptual and statistical distortions (security) low. Data Hiding schemes are characterized by those three conflicting requirements: security against steganalysis, robustness against channel associated and/or intentional distortions, and the capacity in terms of the embedded payload. Depending upon the application it is the designer's task to find an optimum solution amongst them.

The goal of this thesis is to develop a novel data hiding scheme to establish a covert channel satisfying statistical and perceptual invisibility with moderate rate capacity and robustness to combat steganalysis based detection. The idea behind the proposed method is the alteration of Video Object (VO) trajectory coordinates to convey the message to the receiver side by perturbing the centroid coordinates of the VO. Firstly, the VO is selected by the user and tracked through the frames by using a simple region based search strategy and morphological operations. After the trajectory coordinates are obtained, the perturbation of the coordinates implemented through the usage of a non-linear embedding function, such as a polar quantizer where both the magnitude and phase of the motion is used. However, the perturbations made to the motion magnitude and phase were kept small to preserve the semantic

meaning of the object motion trajectory.

The proposed method is well suited to the video sequences in which VOs have smooth motion trajectories. Examples of these types could be found in sports videos in which the ball is the focus of attention and exhibits various motion types, e.g., rolling on the ground, flying in the air, being possessed by a player, etc. Different sports video sequences have been tested by using the proposed method. Through the experimental results, it is shown that the proposed method achieved the goal of both statistical and perceptual invisibility with moderate rate embedding capacity under AWGN channel with varying noise variances. This achievement is important as the first step for both active and passive steganalysis is the detection of the existence of covert channel.

This work has multiple contributions in the field of data hiding. Firstly, it is the first example of a data hiding method in which the trajectory of a VO is used. Secondly, this work has contributed towards improving steganographic security by providing new features: the coordinate location and semantic meaning of the object.

I would like to dedicate this thesis to my wife Mubcrra, my precious son Bartu and my mentor Dr. Zia-ur Rahman who unexpectedly passed away before this work was completed.

ACKNOWLEDGMENTS

First of all I would like to express my deepest sorrow for the unexpected recent loss of my principle advisor Dr. Zia-ur Rahman. He was truly a mentor from whom I benefited a lot in many ways. His personality, strong research skills and academic posture will always be remembered by me. I wish he saw this moment for which we vigorously looked at different aspects of the problem, had many fruitful discussions not only on data hiding but other possible research problems in digital image processing.

I would like to take this opportunity and extend my gratitudes to Dr. Dimitrie Popescu who became my advisor later on. I definitely enjoyed working with him who has very strong research skills. His approach to the complex problems by breaking down them into simple steps have nurtured me in the course of my research. His guidance and support until the completion of degree helped me stay on track and maintain my motivation.

I am grateful to my committee members Drs. Yuzhong Shen, Jiang Li, Dean Krusienski and Dr. Glenn Hines of NASA for their valuable comments and suggestions to improve the quality of this work.

I also want to take this opportunity to thank Prof. Dr. Levent Onural of Bilkent University Electrical Engineering Department for his support and academic guidance during my studies at Bilkent University in Turkey. Courses I took from him and my interactions with him helped me build up the background and foundations for my thesis research. Special thanks go to Dr. Oktay Baysal and Dr. Osman Akan for their help, support and coordination efforts for Turkish Armed Forces education program.

I am so grateful to Muberra and Bartu, my wife and son, for their love, encouragement and sacrifices they made during our tough times and my studies at Old Dominion University. Without their love and support I could never accomplish this. And finally I would like to thank my parents for their continuing support and loving care throughout all my endeavors in my life.

TABLE OF CONTENTS

	Page
LIST OF TABLES	ix
LIST OF FIGURES	xii
 CHAPTER	
I INTRODUCTION	1
I.1 Thesis Motivation and Objectives	1
I.2 Problem Statement and Objectives	3
I.3 Thesis Organization and Contributions	3
II BACKGROUND	5
II.1 Information Hiding Framework	5
II.2 Information Hiding Constraints	9
II.2.1 Design Criteria	10
II.3 An Overview of Information Hiding Applications	12
II.3.1 Data Hiding or Steganography	12
II.3.2 Digital Watermarking	13
II.3.3 Applications	14
II.3.4 Steganography versus Watermarking	15
II.4 Previous work in Digital Video Information Hiding	15
II.5 Steganalysis	19
II.6 Chapter Summary	22
III POLAR QUANTIZATION AND PERFORMANCE ANALYSIS	24
III.1 Quantization	24
III.2 Polar Quantization	25
III.2.1 Phase-Only Polar Quantizer	28
III.2.2 Magnitude Quantizer	28
III.2.3 Numerical Form of Distortion	29
III.3 Derivation of Error Probability	32
III.3.1 Magnitude or Phase-Only Quantization	33
III.3.2 Union Bound on Probability of Error for Polar Quantization	35
III.3.3 An Exact Result for Symbol Error Probability for Polar Quantization	40
III.3.4 Bessel Function Approximation	41
III.3.5 Marcum Q-function	42
III.3.6 Probability of Symbol Error	42
III.3.7 Experimental Results	43
III.4 Chapter Summary	45
IV VIDEO OBJECT TRACKING	48
IV.1 Object Based Representation	48
IV.2 VO Tracking	51
IV.2.1 Image Plane Kalman Filter Tracking	53

IV.2.2	Mean Shift (MS)	54
IV.2.3	Particle Filter (PF)	55
IV.2.4	Feature Based Methods	57
IV.3	Tracking Video Objects in Sports Videos	57
IV.4	Ball VO tracking	59
IV.4.1	Ball Detection:	59
IV.4.2	Experimental Results	62
IV.5	Performance Evaluation of Object Tracking	63
IV.6	Research Foundations	71
IV.7	Chapter Summary	72
V	TRAJECTORY PERTURBATION DATA HIDING	75
V.1	Conveying Synchronization Information to the Decoder	77
V.2	Trajectory Perturbation	86
V.3	Motion Compensation and Recomposition of the Video Sequence	88
V.4	Decoder Side Tracking and Detection	92
V.5	Chapter Summary	93
VI	EXPERIMENTAL RESULTS AND DISCUSSIONS	95
VI.1	VO Selection, Preprocessing and Tracking	95
VI.1.1	VO Tracking	98
VI.1.2	Data Embedding and Decoding	100
VI.2	Comparison Of the Proposed Method With Other Methods	115
VI.3	Chapter Summary	117
VII	CONCLUSIONS AND FUTURE PERSPECTIVES	118
VII.1	Future Directions	119
VII.1.1	Near Term Focus	119
VII.1.2	Long Term Focus	119
	REFERENCES	121
	APPENDICES	
A	Q FUNCTION	130
B	STEGANALYSIS	131
B.1	Measures Based on Pixel Differences	131
B.1.1	Minkowsky Measures:	131
B.1.2	Correlation Based Measures:	132
B.2	Spectral Measures	132
B.3	Perceptual Measures	132
B.3.1	Peak Signal-to-Noise Ratio (PSNR):	132
B.4	Histogram Measures	133
B.4.1	K-L Divergence:	133
B.4.2	χ^2 (Chi-Square) Metric:	133
VITA		134

LIST OF TABLES

Table	Page
1 BER Performance of “Table Tennis” Sequence for Magnitude Quantization Based Embedding under AWGN with Varying Variance.	102
2 BER Performance of “Table Tennis” Sequence for both Magnitude and Phase Quantization Based Embedding under AWGN with Varying Variance.	109
3 BER Performance of “Soccer Ball” Sequence for Magnitude Quantization Based Embedding under AWGN with Varying Variance.	114
4 BER Performance of “Soccer Ball” for both Magnitude and Phase Quantization Based Embedding under AWGN with Varying Variance.	114

LIST OF FIGURES

Figure	Page
1 An Example of LBS Based Information Hiding (An Image in another Image) [1].	2
2 A Generic Information Hiding Framework. a. Non-Oblivious Data Hiding b. Oblivious Data Hiding.	6
3 Linear Non-Oblivious Information Embedding and Detection.	7
4 a. Base Quantizer b. Quantizer for Embedding $m = 0$. c. Quantizer for Embedding $m = 1$	9
5 Trade-Off Between Design Criteria [2].	12
6 Uniform and Nonuniform Quantization.	25
7 Polar Quantizers Magnitude and Phase Partitions and Reconstruction Levels.	27
8 Distortion as a Function of Levels L and Number of Cells N for a Source with Density Function $f(r) = re^{-\frac{r}{2}}$. a. Distortion vs. Varying number of Magnitude Quantization Levels. b. Distortion vs Varying number of Cells at fixed level L=10.	30
9 Distortion as a Function of Levels L and Number of Cells N for a Source with Density Function $f(r) = \frac{1}{b-a}, a \leq r \leq b$ a. Distortion vs. Varying number of Magnitude Quantization Levels. b. Distortion vs Varying number of Cells at fixed level L=8.	31
10 Polar Quantizers a. Restricted Nonuniform b. Restricted uniform c. Non-restricted Nonuniform [3].	32
11 Some Examples of Signal Constellations.	33
12 Signal Constellation of Magnitude Only Polar Quantizer with $\Delta = 2r$. Reconstruction Points are shown as dots whereas \mathcal{R}_j 's represent corresponding decision regions.	34
13 Probability of Bit Error vs SNR for Polar Quantizer with 4 Levels and Magnitude Quantization Only. In this case $P_b \approx \frac{3}{4}Q(\sqrt{\frac{4E_b}{21N_0}})$	36
14 a. Signal Constellation of a Uniform Polar Quantizer with $\Delta = 2r$. b. Detail of One Quantization Cell where d_{min} is the distance between a_i and a_j , and $\frac{d_{min}}{2}$ is the distance to the side of decision boundary.	38
15 Left. Union bound estimate of symbol error probability: Plot of (40) when $r=1$ fixed and σ is varied. Right. Symbol error probability as a function of r when $\sigma=0.1$	39
16 Bessel Function and Marcum-Q function Approximations to Exact Error Probability.	44
17 The Probability Of Symbol Error For Nonuniform Signal Constellation Obtained by Using a Restrictive Nonuniform Polar Quantizer As Shown in Fig.10.a.	45
18 An Example of Non-uniform Signal Constellation.	46

19	Symbol Error Probability For the Non-uniform Signal Constellation Shown in Fig. 18.	47
20	Video Sequence Representation.	49
21	A Frame from “News” Sequence VO Segmentation Results.	51
22	Correspondence Problem and Trajectory Estimation.	52
23	Object Representation After [4].	53
24	MS Tracking of Ball VO. Left: Frame 15 and Right: Frame 44. Underlying problem of MS for fast moving objects and a homogeneous background which has lots of the same pixels (noise) as that of VO.	56
25	PF Tracking of Ball VO. Left: Frame 1 , Middle: Frame 9 and Right: Frame 37 [5].	57
26	Example of Ball Size, Shape and Color Variations in Soccer Video. (After [6]).	59
27	Binary Image Obtained After Thresholding Pixel Values in the Bounding Box of the Object as $O(i, j) \geq 0.8637$. Connected Component Analysis is done on this Binary Image to Label the Pixel Belonging to the Same Object.	61
28	Example Frames From “Table Tennis” and “Soccer” Sequences.	62
29	Table Tennis Sequence “Ball VO” Trajectory.	64
30	Ground Truth and Computed Trajectory Centroid Coordinates. Top: X coordinate Bottom: Y Coordinate. The difference between the frames 33 and 42 is due to the occlusion of the ball by the player.	65
31	Frames 1 to 6 from Table Tennis. From top to bottom: noisy frames with $\sigma_n^2 = 0.001$ to 0.3.	66
32	Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.1$. Top right: Difference in Y coordinate of the ground truth and the tracking output. Bottom Left: Difference in X coordinate of the ground truth and the tracking output.	67
33	Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.05$. Top right: Difference in Y coordinate of the ground truth and the tracking output. Bottom Left: Difference in X coordinate of the ground truth and the tracking output.	68
34	Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.01$. Top right: Difference in Y Coordinate of the Ground Truth and the Tracking Output. Bottom Left: Difference in X Coordinate of the Ground Truth and the Tracking Output.	69
35	Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.001$. Top right: Difference in Y Coordinate of the Ground Truth and the Tracking Output. Bottom Left: Difference in X Coordinate of the Ground Truth and the Tracking Output.	70
36	An example of VO Partial Trajectory Embedding.	73
37	An example of Multiple VO Partial Trajectory Embedding.	73
38	Block Diagram of the Proposed Method.	76
39	Synchronization Data Embedding Mechanism.	78

40	JPEG Quantization Table (Luminance Quantization).	80
41	Zig Zag Ordering of DCT Coefficients in an 8x8 Block.	82
42	Sync Data Embedded and Extracted Frames. Statistical Invisibility of the Stego Frame Based on Pixel Measures Presented in Appendix A. Image Fidelity=1.0 and MSE=0.6122. Difference Frame between the original and the stego frames shows the randomized locations of the 8x8 blocks.	83
43	Additive Noise Effect. Sync Data Embedded and Extracted Frames. .	84
44	Gaussian Blurr Effect. Sync Data Embedded and Extracted Frames. .	85
45	Uniform Polar Quantizer for Motion Magnitude Quantization.	88
46	Patch Based Motion Compensation Example Frames. Top Left and Right:Frames 7, 14. Bottom:Frame 34.	90
47	Inpainting Region and isophote directions [7].	91
48	Inpainted Frames from 21 to 24.	92
49	Ball VO Centroid Coordinates.	96
50	Movie Editor for Frame-by-Frame Analysis of the Video Sequence. . .	97
51	Table Tennis Sequence “Ball VO” Trajectory.	99
52	Histogram and Density Function Analysis of “ Bal VO”.	100
53	Decoded Binary Data.	102
54	Stego and Original Frames.	104
55	Steganalysis Results of “ Bal VO” Perturbation Based Data Hiding. .	105
56	Steganalysis Results of “ Bal VO” Perturbation Based Data Hiding. .	106
57	Angle Quantization Scheme [8].	107
58	Magnitude and Angle Quantized Frames. The Semantic Meaning of the VO is lost.	108
59	The Results of the Steganalysis Measures for Magnitude and Phase Quantized VO Stego Frames.	109
60	The Results of the Steganalysis Measures for Magnitude and Phase Quantized VO Stego Frames.	110
61	Soccer Ball VO Motion Magnitude.	112
62	Noise Added 1st Frame.Top-Left, Top-Right, Bottom-Left and Bottom-Right represent the noise variance in order $\sigma_n^2 =$ [0.001, 0.01, 0.05, 0.1].	113
63	PSNR and Spectral Distortion Results.	115
64	Pixel Wise and Histogram Based Steganalysis Results of “Soccer Ball” VO Motion Magnitude and Angle Perturbation Based Data Hiding. .	116
65	<i>Q function</i>	130

CHAPTER I

INTRODUCTION

Information hiding is the art of hiding a message signal in a host signal without any perceptual degradation to the host signal. Its roots can be traced back to ancient times. Various examples of early information hiding reported in history were primarily focused on military applications.

With the rapid development of digital multimedia technologies and rapid progress in Internet, information hiding has gained momentum receiving attention from the research community. Its popularity can be attributable to advances in storage, reproduction, editing and distribution of digital multimedia which have associated pirating, copyright protection and illegal distribution problems. For instance, over the last decade, the movie industry has moved from analog to digital media, bringing DVD devices and distributing movies on DVD's to consumers at low cost. Due to illegal copying, the movie industry has become interested in techniques to prevent loss of profit due to these illegal activities. The digital watermarking techniques, which are just one instance of information hiding, have been proposed as a solution to the copyright protection problem. Other applications of different types of information hiding techniques are discussed in Section II.3.

The simplest and most commonly used information hiding method is the Least Significant Bit (LSB) modification in which first i^{th} LSB of the cover data is replaced with the secret data. An example of LSB based information hiding is illustrated in Fig. 1. Some widely used image steganographic methods includes JSteg, F5, Outguess, etc.

I.1 THESIS MOTIVATION AND OBJECTIVES

In this thesis, a method to embed a secret message in digital video sequences aiming at establishing a steganographic channel between two parties has been proposed. The overall goal behind devising such a data hiding algorithm is mainly for defense applications. The motivation of the thesis is discussed with an application example which shows the practicality of the research as follows. Imagine a scenario in which the sender, using the proposed method, embeds the data which could be a secret message, into a video sequence and posts it on a public Internet web page or sends it

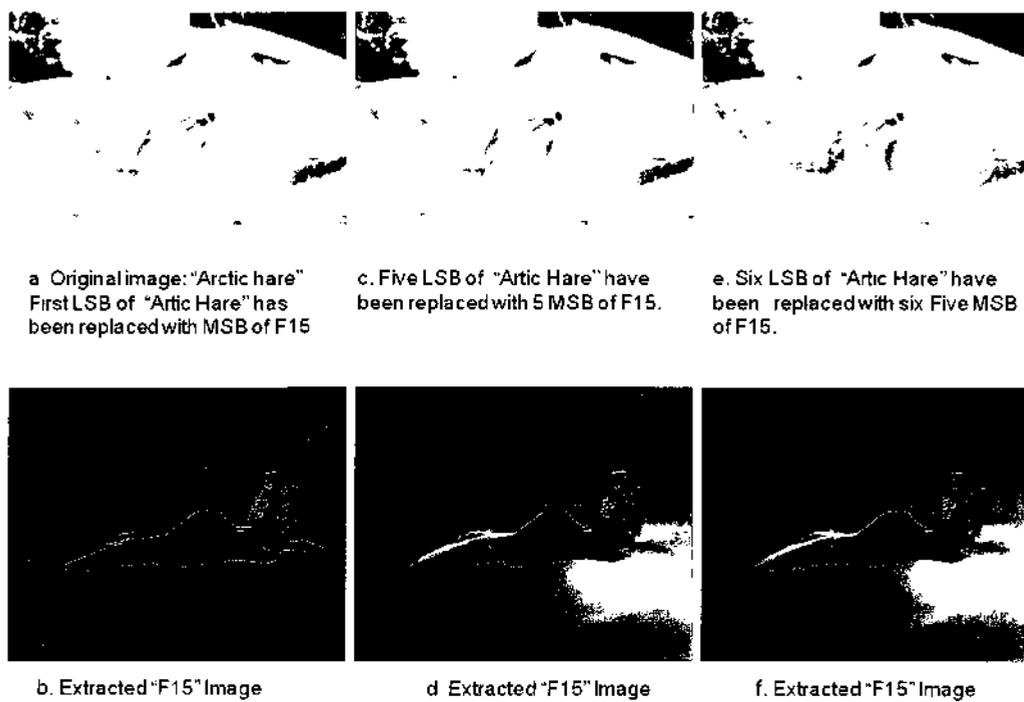


FIG. 1: An Example of LBS Based Information Hiding (An Image in another Image) [1].

as an attachment to an email. The video sequence could be previously known to both the sender and the receiver or be a newly recorded sequence by the sender. Since both sides know the embedding and decoding algorithm, the receiver can decode the embedded data by accessing the video sequence through the Internet from anywhere in the world. The benefits provided by the proposed method for this generic scenario are:

1. There is no need for a dedicated communication channel or device between the two parties,
2. Accessibility from anywhere utilizing Internet access,
3. Very low probability of detection as establishing the covert channel arouses almost no suspicion.

I.2 PROBLEM STATEMENT AND OBJECTIVES

In this thesis, the information hiding problem and its application in the field of digital video will be investigated. Specifically, the focus will be on a steganographic method aiming at sending a secret message using a digital video sequence as cover data to the receiving side. The proposed method will satisfy two major requirements which are the case for any steganographic application:

1. Perceptual Invisibility
2. Statistical invisibility

These requirements are discussed in detail in a general context in Chapter II.

I.3 THESIS ORGANIZATION AND CONTRIBUTIONS

This thesis is organized into seven Chapters. Chapter I introduces the problem statement and objectives behind this work. Chapter II provides background for a reader on information hiding by presenting a general information hiding framework with relevant taxonomy and the state-of-the art techniques with some application examples. Chapter III is mainly dedicated to the polar quantization scheme which is used as a non-linear embedding function. In this chapter, detailed analysis of distortion and symbol error probability are carried out for different cases. Analytical results

with corresponding simulation results are also presented. Video Object (VO) tracking basics and tracking methods are discussed in detail in Chapter IV. The proposed method is presented in Chapter V. Chapter VI focuses on the experimental results and discussion on the results. And finally, the thesis concludes with Conclusion and Future Perspectives in Chapter VII.

Note that the work in Chapter V was presented at [9] and the work in Chapter III was submitted for presentation at the IEEE Globcom2011 Conference.

CHAPTER II

BACKGROUND

This chapter is designed as a precursor to the proposed method helping the reader understand the fundamentals of information hiding. The general information hiding framework includes an Encoder, Channel and Decoder, forming the basis for different analytical approaches, such as information theoretic or game theoretic analysis, to a data hiding problem.

In what follows, the information hiding framework components (embedding, decoding functions, channel etc.) together with different design constraints and some important information hiding application types will be presented. And finally, the so-called steganalysis, which is the countering mechanism, will also be defined and the distinction between two types which usually create confusion will be discussed briefly.

II.1 INFORMATION HIDING FRAMEWORK

A generic information hiding framework is illustrated in Fig. 2. The information hiding process consists of an embedder \mathbf{E} and a decoder D . The host data signal vector $\mathbf{c} \in \mathfrak{R}^N$ is typically obtained from a host image, video or audio signal. When the host signal is assumed to be available at the decoder side, the method is called “non-blind” or “escrow” whereas the situation in which the host signal \mathbf{c} is not available at the decoder side is called “blind” or “oblivious”. Depending upon the application, the host signal could represent spatial, temporal or transform domain features of the host signal.

Message set \mathbf{m} is produced by the alphabet \mathcal{M} which can be encrypted and/or error correction encoded before embedding to further increase the security. Side information K , such as a cryptographic key or side information about the host signal, is assumed to be available to both the encoder and the decoder but not to the attacker. The role of the side information K is two fold. First, it may introduce randomness in information embedding locations through the use of look-up tables. Second, it can provide side information about the host data signal such as partial information about the host signal features, hash values of the host signal, location of the watermarks, cryptographic key and the seeds for modulating pseudo-noise

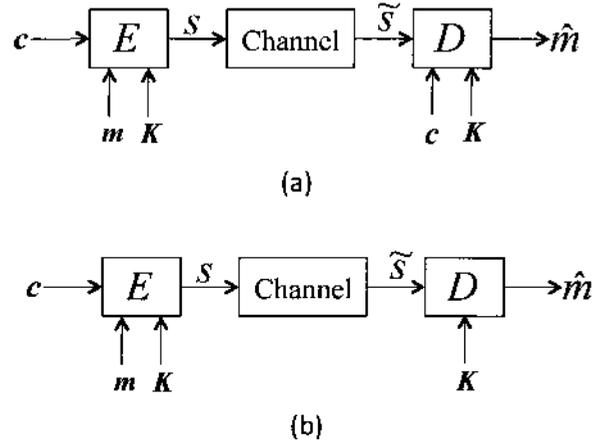


FIG. 2: A Generic Information Hiding Framework. a. Non-Oblivious Data Hiding b. Oblivious Data Hiding.

sequences in spread spectrum systems [10]. At the information hiding stage, the host signal c , side information K and the message are passed through the embedding function \mathbf{E} resulting in stego data s which can be defined by:

$$s = \mathbf{E}(c, m, K) \quad (1)$$

In the model, the embedder and the decoder may be linear or non-linear functions operating on scalar or vector variables, and are not necessarily inverses of each other. In the first category, the embedding methods in which embedding function \mathbf{E} adds message sequence linearly to c and D decodes m from \tilde{s} by simple subtraction if c exists at the decoder, or otherwise by a correlation based detection called Type-I. Addition can be performed in a specific domain, such as spatial or transform, or on specific features, such as transform domain coefficients, pixel values, texture, edge, motion vector, centroid coordinate etc. The Type-1 embedding and detection schemes are illustrated in Fig. 3. Considering the embedding of only one bit, the difference between stego data s and original cover data c is a function of b , i.e., $c - s = f(b)$. Although it is possible to detect b directly from s , c can be considered as noise, knowledge of which will enhance the detection performance [11]. An example of Type-I embedding is the spread spectrum watermarking which has the embedding function of the form

$$s'_i = c_i + b' \alpha_i w_i \quad (2)$$

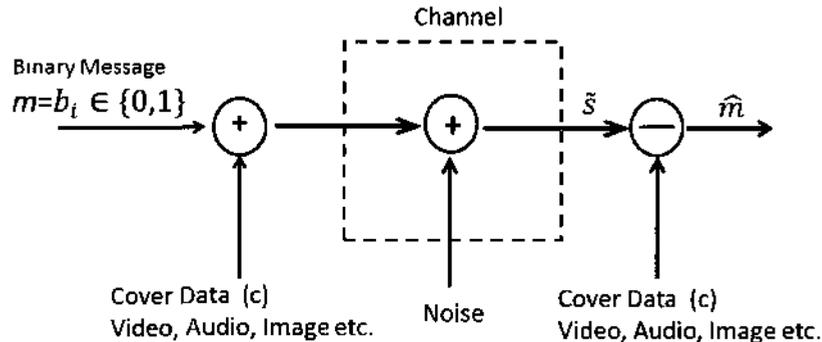


FIG. 3: Linear Non-Oblivious Information Embedding and Detection.

where c_i are the host signal features, i.e. pixel values, transform coefficients etc, s'_i are the marked coefficients and $b' \in (-1, +1)$ is the antipodal modulation mapping of $m[i]$, the bit to be embedded, α_i is the gain which is used to adjust the embedding strength and w_i is the spread spectrum signal obtained by using m-sequences. Reader should refer to Cox [12] and to Perez [13] for examples of Type-I embedding methods.

In the second category, the embedding \mathbf{E} and the decoding functions \mathbf{D} are non-linear. The embedding domain features are mapped by a function \mathbf{E} to obtain \mathbf{s} so that the relationship of $b = \mathbf{E}(c)$ is deterministically enforced. Contrary to the first type, the detector for this category does not require the knowledge of original cover data as the information regarding b is solely carried by s . Another important characteristic of non-linear methods is their ability to suppress noise due to the original content, or self-noise [14], and to encode one bit in a comparatively smaller number of cover data features, hence high capacity. A simple example is the so-called odd-even embedding, which is a special case of the look-up table (LUT) embedding which uses a LUT to determine mapping between cover data and the data to be embedded, where a quantized feature is enforced to an even number for binary “0” and an odd number for bit “1”. Another pioneering work for non-linear techniques is the quantization based data hiding methods. Chen et al. in [15] provide a more formal treatment of data hiding techniques that use the quantization index to embed bits (methods that force the quantized indices to take a desired value, depending on the information signal to be embedded) [14]. In recent data hiding literature, the data hiding methods employing quantization are referred to as Type-II methods.

A key aspect of the design of QIM based information hiding methods involves the

choice of practical quantizer ensembles. One commonly used quantizer scheme, an extension of QIM, is the so-called dithered quantization, which has the characteristic that the quantization cells and reconstruction points of any given quantizer in the ensemble are shifted versions of the quantization cells and reconstruction points of any other quantizer in the ensemble [15]. The stego signal is generated by quantizing the host signal with the corresponding dithered quantizer as

$$s = Q_{\Delta}(c + W_m) - W_m \quad (3)$$

where Q_{Δ} is the high-dimensional base quantizer with step size Δ , and W_m is the watermark signal corresponding to message indexed by m , $1 \leq m \leq M$, where each component W_{mi} , $1 \leq i \leq N$ of W_m is a representation from a set $\Upsilon \in \mathfrak{R}$.

For embedding one bit, $m \in \{0, 1\}$, in a real-valued sample, $s \in \mathfrak{R}$, the dithered quantizers are defined as

$$Q_i(s) = Q(s - d_i) + d_i, i = 0, 1 \quad (4)$$

where $Q(s) = \Delta \text{round}(s/\Delta)$ and $\text{round}(\cdot)$ denotes rounding to nearest integer, $d_0 = -\Delta/4$ and $d_1 = \Delta/4$. The reproduction levels of quantizers Q_0 and Q_1 forming two lattices Λ_0 and Λ_1 are shown in Fig. 4 where the lattices are defined as

$$\begin{aligned} \Lambda_0 &= -\Delta/4 + \Delta Z, \\ \Lambda_1 &= \Delta/4 + \Delta Z \end{aligned} \quad (5)$$

where ΔZ is the set of integers.

At the receiver side, one possible choice for the decoder is a minimum-distance decoder which finds the quantizer point closest and outputs the estimated message as $\hat{m} = \underset{m}{\operatorname{argmin}} \| \tilde{s} - s \|$.

The least-significant-bit (LSB) based embedding is another spatial domain non-linear approach in which the LSB plane of the host signal is overwritten with the secret bit stream. For LSB based methods see [16, 17, 18, 19].

The output of the embedding module \mathbf{s} should be perceptually similar to the original host signal \mathbf{c} according to a perceptual distortion measure denoted by $d(\cdot, \cdot)$ such that embedding function should satisfy the distortion constraint $d(s, c) \leq d_1$, where d_1 is the maximum allowable perceptual distortion beyond which distortions will be visible. This requirement is known as transparency or perceptual invisibility.

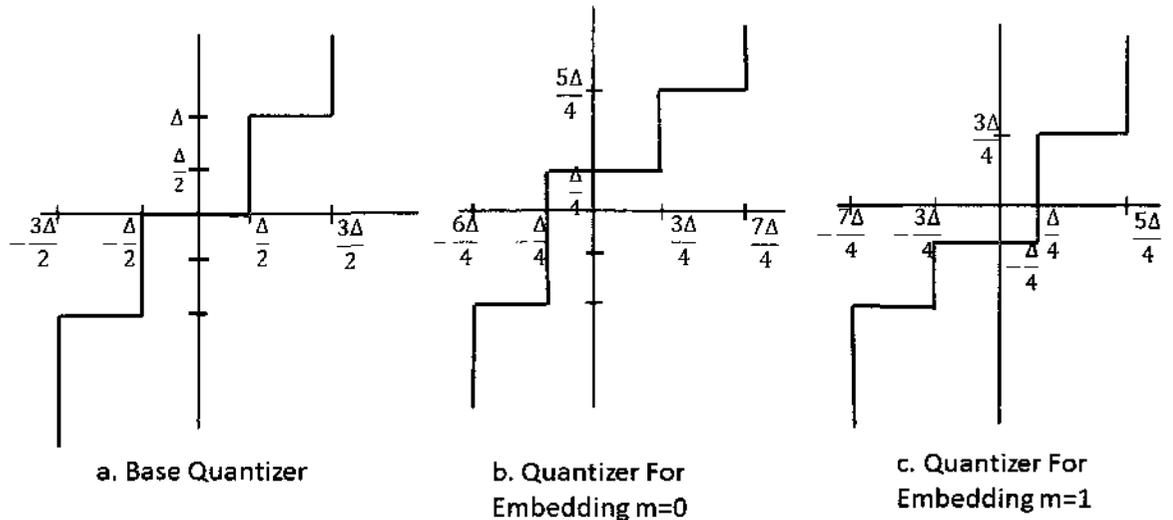


FIG. 4: a. Base Quantizer b. Quantizer for Embedding $m = 0$. c. Quantizer for Embedding $m = 1$.

A rudimentary but common choice is the squared Euclidean metric which can be defined as

$$d(s, c) = \|s - c\|^2 \quad (6)$$

And finally, the stego data \mathbf{s} is communicated to the receiver through a public channel which can be monitored by an active or passive steganalysist attempting to detect covert communication or to remove/modify the embedded information. Additionally the stego data may undergo common signal processing manipulations such as lossy compression, filtering, noise addition etc., which eventually result in distortion in the stego data [15]. All of the intentional or unintentional channel associated distortions should also be bounded by the distortion constraint $d(s, \tilde{s}) \leq d_2$.

II.2 INFORMATION HIDING CONSTRAINTS

The resource of the communication between the embedder and the decoder is the distortion introduced to host signal during embedding [20]. In general, information hiding techniques are mostly evaluated by design criteria that conflict (see Fig.5). Additionally, the designers of information embedding algorithms are required to make proper trade-off between these criteria based on the the requirements of the specific

application. For instance, for an application specific minimum payload requirement and a maximum acceptable level of distortion criterion the designer tries to achieve robustness by tuning the distortion in a controllable fashion until the distortion requirement is met.

II.2.1 Design Criteria

Fidelity

Fidelity is related to the distortion introduced by the embedding mechanism. The perceptual quality of the host signal should be preserved up to a certain level based on the distortion criterion. Referring back to generic framework in Fig. 2, this requirement can be stated as $c \approx s \approx \tilde{s}$. Perceptual quality of the resultant stego signal will eventually be assessed by the human viewer whose visual system is much more complicated than any analytical distortion metric such as Peak Signal-to-Noise Ratio (PSNR), L_1 , L_2 etc. Therefore, researchers generally incorporate HAS (Human Auditory System) and HVS (Human Visual System) properties based on JND (Just Noticeable Differences) models into the embedding stage to satisfy perceptual quality requirements. Reader should refer to [21, 22, 23] for details of HAS/HVS based information embedding methods.

Robustness

The robustness of an information embedding scheme can be described as the ability of the detector to extract hidden message from the received stego data which might possibly have undergone some unintentional channel associated perturbations and/or intentional attacks (cropping, rotation, volumetric changes, frame dropping etc.) during transmission. In the general information hiding framework this constraint can be stated as $m = D(s) = D(\tilde{s})$.

Capacity

Capacity is the number of bits that can be hidden in a given host signal when the stego signal undergoes intentional and unintentional attacks in the channel that can be usually modeled by Gaussian probability distribution function. In [10] it is shown that the data hiding capacity is the value of mutual-information among the encoder, decoder and the attacker. The channel associated noise/attack is generally modeled

as additive noise but, in reality, the receiver may not know the exact attack channel model and the associated channel parameters. Thus it would be impossible to define capacity without defining robustness criterion [11].

In addition to the embedding capacity, another inherent problem discussed in [11] is the uneven embedding capacity in the host signal in which the amount of embedded data varies significantly from region to region. Wu et al. proposed shuffling embedding regions before an embedding operation as a solution to tackle the uneven embedding capacity problem.

Security

This requirement defines the cryptographic aspect of the information hiding problem by stating the inability of an unauthorized users to access, remove, read or write the hidden message.

Statistical Invisibility

The embedding mechanism should leak limited information under steganalysis to protect the existence of the hidden message against active or passive warden. This requirement is discussed in detail in Section II.5 of this Chapter.

Discussion

There is a strong trade-off among these requirements which needs to be considered in the design phase. Fig. 5 illustrates conflicting relationships between design criteria. If the capacity is increased, this might yield perceptually visible degradations in the cover data. If robustness is needed by means of increasing embedding strength, this might also lead to visual artifacts. On the other hand, the capacity requirement is inversely related with robustness. It is clear that as the number of bits in the hidden message is increased, it will be more difficult to extract the hidden message without any bit error if an active warden launches attacks on the channel or even channel associated distortions such as noise exist.

The optimum compromise between these requirements is application specific. While the number of hidden bits, in some applications, such as broadcast monitoring, should be sufficient to differentiate all broadcasts from each other, some others, such as copyright protection, might require only one bit of hidden information, which

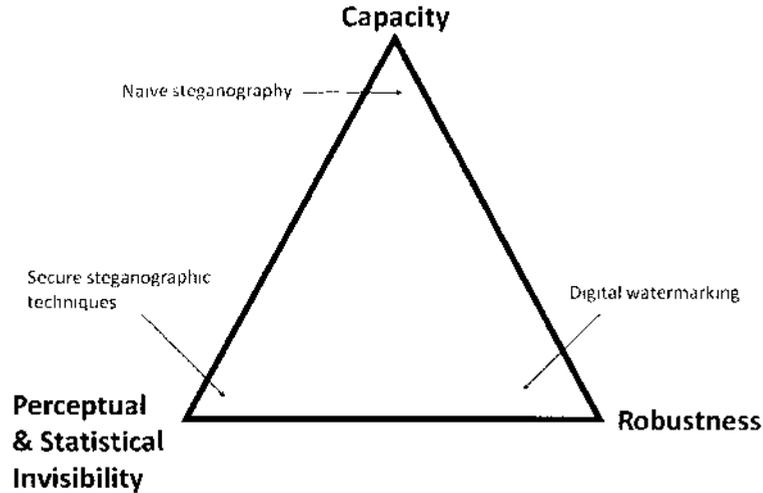


FIG. 5: Trade-Off Between Design Criteria [2].

indicates the whether the multimedia has been tempered with or not. An information embedding mechanism should be based on the balance between this trade-off to achieve the ultimate design goal.

II.3 AN OVERVIEW OF INFORMATION HIDING APPLICATIONS

The categorization of information hiding techniques is still an ongoing debate among research community due to its still being in infancy and broadening application domain. In this thesis, these applications are broadly classified into two categories as described in Section II.3.1 and II.3.2.

II.3.1 Data Hiding or Steganography

Steganography is defined as the art of hidden communication in which the goal is to conceal the presence of communication. It is an instance of data hiding in which the goal is to decode an embedded message with less probability of error. This can be considered as a decoding problem [24] which has borrowed theories from communication and information theories in defining the theoretical foundations such as finding the total information embedding capacity in a game-theoretic framework [10, 25, 26].

At this point it would be helpful for the reader to realize the distinction between classical cryptography and steganography. The goal in cryptography is to map plain

text into an unreadable cypher text using a secret key that is shared between the sender and the receiver. For an adversary to defeat the system, he or she must break the secret key by searching over the entire key-space which is a time consuming and complex undertaking. On the other hand, steganography offers security in a different manner in that it hides the existence of information exchange from the adversary. So the distinction between cryptography and steganography can be summarized as encryption protects the content of the message whereas steganography prevents discovery of the existence of a communication. Using steganography in connection with the cryptography doubly protects the information.

II.3.2 Digital Watermarking

Digital watermarking refers to the embedding of secondary data within the host data. The embedded data, usually called watermarks, can be used for various purposes, each of which is associated with different robustness, security, and embedding capacity requirements [11]. In this set up, the watermark data (copyright data) is known at the receiving end. The receiver is designed to verify reliably the existence (or non-existence) of hidden data in the received data. Hence, this problem is a verification problem. Furthermore, the problem can be viewed as a hypothesis testing problem using tools from detection theory. Most of the proposed information hiding schemes in the literature fall under this category [24]. The principal advantage of digital watermarking compared to other solutions such as encryption is its ability to associate secondary data with the host data in a seamless way. For example, compared with cryptographic encryptions, the embedded watermarks can travel with the host data and assume their protection functions even after decryption. With only the exception of visible watermarks, the watermark data are expected to be imperceptible [11].

Over the past two decades, numerous information embedding methods have been developed for applications such as authentication and temper detection, ownership protection, fingerprinting or labeling, copy control and cover communication. For different examples of applications, readers should refer to [27, 28, 29, 30, 31]. A brief description of some common applications and the associated design criteria are discussed below.

II.3.3 Applications

Covert Communication:

This application falls under stenography, providing different implementations for military and intelligence areas. The goal is to hide the very presence of the transmission over the channel from an adversary. For this type of information hiding application, statistical as well as perceptual invisibility is more important than robustness as the active/passive wardens generally launch statistical attacks to verify the existence of secret communication.

Ownership protection:

A watermark labeling the ownership is embedded in the host signal. The watermark is expected to be robust against intentional attacks or unintentional channel associated perturbations to demonstrate the ownership. The detection should have as little ambiguity and false alarm as possible and the embedding capacity does not usually have to be high [11]. Protection of digital multimedia (digital audio clips, digital video etc.) is a common representative of this class of application.

Authentication or Tamper Detection:

To verify the authenticity of the host data, usually without the knowledge of the original host, also called *blind detection*, secondary data is embedded in the host signal. Forging a valid authentication watermark in unauthorized or tampered host data is prevented [11]. These applications generally require high embedding capacity. Some example applications could include authentication of electronic business documents, medical records, military documents etc.

Copy Control and Access Control:

The embedded data represents certain copy control or access control policies. A detector is generally built-in in a recording/playback system. When detected, the policy is enforced by directing certain hardware or software actions, e.g. disabling recording. The robustness against removal and blind detection are the common requirements [11].

Fingerprinting or labeling:

The data is embedded for customer tracing type applications where the owner of the multimedia issues a user ID for each valid customer. In case of illegal copying the originator or recipients of a particular copy of the multimedia can be easily identified. This application requires robustness against removal.

Annotation:

The embedded data may describe a signature of the originator of the host signal. Robustness against attacks is not required whereas blind detection is required.

II.3.4 Steganography versus Watermarking

Data hiding and watermarking are different in the sense that watermarking is mainly a detection problem where the receiver has to decide whether or not a certain watermark has been embedded in the host signal. On the other hand data hiding is a decoding/communication problem where the receiver assumes that the sender is transmitting some message and the goal is to decode the embedded bits. The latter problem is more difficult as there is no reference sequence for comparison. For watermarking applications, the reference sequence that is being detected is available at the receiver [24].

II.4 PREVIOUS WORK IN DIGITAL VIDEO INFORMATION HIDING

Having discussed the definitions and application specific requirements for information hiding in Section II.3, the state-of-the art information hiding methods developed for the digital video domain will be discussed next. The information embedding methods employing digital video can be classified into two broad categories:

1. *Compressed (Bitstream) Domain*
2. *Raw-Video Domain*

Different methods that fall under these categories are given in detail below.

Compressed Bit-Stream

In compressed (bit stream) domain information hiding, the host signal is in the form of bit stream that has been compressed by using a standard video codec (coder-decoder) such as MPEG-1/2/4, H.264 AVC, etc. For this type of application, generally the quantized DCT coefficients or motion vectors (MV)s are utilized for information embedding. The classical and flexible approach is to partially decode the compressed video bit stream to extract the aforementioned features. The embedder should have access to compression specifics such as quantization parameters, Group Of Picture (GOP) structure, etc., to improve perceptual quality, robustness and capacity, and must ensure that the bit-stream obtained after embedding is a syntactically valid bit-stream that can be decoded by the corresponding decoder [32]. The reader should refer to [30] for a brief survey on information hiding and different applications.

Some examples of compressed video domain digital watermarking applications will be discussed next. Hartung et al. [33] proposed a compressed domain watermarking method which partially decodes MPEG-2 coded video bit-stream to obtain the DCT coefficients of each frame and inserts a watermark in them. Their method includes drift compensation and rate control mechanism to adjust the bit rate. As another related compressed domain work, Langelaer et al. [34] have presented a video watermarking method which embeds data by manipulating some selected set of DCT coefficients based on the difference between a set of high frequency and low frequency DCT coefficients.

Parallel with the developments in object based coding of video such as MPEG-4, video watermarking methods aiming at protecting individual VOs have become popular. Some exemplary work in this context are the methods proposed in Alattar [32] and Barni [21]. The rationale behind these methods is their ability, provided by the coding standard to a user, to access and manipulate individual VOs in the scene. Those methods can be considered as a solution to protecting the ownership of VOs. Barni in [21] presented another example of compressed domain digital video watermarking in which the watermark is embedded in each VO by imposing a particular relationship between the DCT coefficients in the luminance blocks of pseudo-randomly selected macro blocks using a secret key. A masking parameter modulating the watermark amplitude is incorporated to limit the visual artifacts and to improve the robustness. In the decoding phase, the compressed watermarked

video bit-stream is decoded and the watermark is extracted using the secret key.

In another compression domain method proposed by Alattar [32], the MPEG-4 compressed bit stream partially decoded the embedded watermark. Their method includes a block-by-block local gain based on the motion vector (MV) data and the DCT coefficients to adjust the watermark strength. The spatial spread-spectrum watermark is embedded directly by modifying the DCT coefficients. To deal with watermark leakage into successive frames, they incorporated a drift compensation module. The detection is performed in the spatial domain via a linear correlator after establishing and maintaining detector synchronization.

Another interesting application of data hiding is presented in Chen et al. [35], who proposed a QIM based data hiding method for error concealment of intra-coded frames in H.264/AVC. At the encoder side, the MV of a macroblock (MB) is encoded and imperceptibly embedded into another MB within the same frame. If an MB is found missing at the decoder, the embedded information is retrieved from the corresponding MB.

In [36], Wong et al. propose a complete video quality preserving data hiding method where the information is embedded into a compressed video by simultaneously manipulating quantization scale (Mquant) and quantized DCT coefficients, which are the significant parts of Moving Picture Experts Group (MPEG) and H.26x-based compression standards. Reverse Zerotrun Length (RZL) is proposed for achieving high embedding efficiency. The problem of video bit-stream size increment caused by data embedding is also addressed, and two independent solutions are proposed to suppress this increment.

Raw Video Domain

Commonly used raw video domain methods employ various transforms as tools to provide features where the data is embedded via coefficient modification. In transform domain methods, the host signal is transformed into a different domain such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and the data is embedded in selected coefficients by employing linear or non-linear embedding/decoding functions. One advantage of using transform domain information hiding is the ability to incorporate HVS properties into the embedding mechanisms as many of today's video coding standards use those properties in the transform domain.

Hartung et al. [33] proposed a spread spectrum watermarking technique where the data is embedded into the non-zero DCT coefficients of the cover video. The detection is done by means of a correlation based receiver in which the original cover video is not required. Swanson et al. proposed a wavelet based raw video watermarking scheme based on DWT [37]. They applied the DWT along the temporal axis, resulting in a multi resolution description of the frames which is then used for embedding. Wu et al. [11] introduced a DCT based multilevel embedding scheme in which low and mid-frequency DCT coefficients are used for data embedding.

Another example of transform domain approach is presented by Mukherjee et al. [38]. They proposed a DWT based data hiding method in which hidden data is source-coded by vector quantization, and the indices obtained in the process are embedded in the host video using orthogonal transform domain vector perturbations constituting the channel codebook.

Although transform and spatial domains have been utilized as the primary operating domains for information embedding, there are some methods devised for information embedding in the temporal domain of the video [39, 40, 41, 42, 43, 44]. These methods are basically aimed at modifying selected subset of MVs to convey the data to the receiver side. The selection of MVs is based on their magnitude and direction and in some cases their position with respect to a reference grid [42]. For instance, Zhang et al. [43] proposed a video watermarking method that embedded information into selected MVs based on just the magnitude of motion vectors: Large vector magnitudes indicate fast moving objects in which case, human eyes cannot perceive motion vector perturbations. These distortions are much more perceivable for the smaller magnitude MVs. Key issues that have to be considered with MV alteration based schemes are:

1. Whether or not the embedded data are still persistent after decompression in which case the motion information is no longer present.
2. Effects of MV modification on the semantic meaning of the video sequence as these methods generally do not take into consideration the smoothness and semantic meaning of the motion at all, i.e. a semantic video object consists of several major segments having different movements where a single motion model cannot represent the whole object motion appropriately.

Most of the video watermarking methods consider video frames as still images and

apply embedding schemes originally developed for image watermarking in a frame-by-frame manner. In contrast the methods proposed in Koz [22], Degul [45] and Liu [46] take the temporal dimension of the video into account. Koz [22] exploited the temporal dimension for video watermarking by means of utilizing temporal sensitivity of the HVS. The proposed method utilizes the temporal contrast thresholds of HVS to determine the maximum strength of the watermark, which still gives imperceptible distortion after watermark insertion. Liu et al. [46] proposed a new video watermarking algorithm based on the 1D-DFT and the Radon transform. They introduced the 1D-DFT along the temporal direction for watermarking. This enabled the robustness against video compression. The Radon transform-based watermark embedding and extraction produces the robustness against geometric transformations.

II.5 STEGANALYSIS

The term “*steganalysis*” refers to collection of techniques that are designed to detect secret communication. The overall idea is similar to the classical detection problem in the sense that distinguishing between cover data and stego data results in deciding on the existence or non-existence of the hidden message. The detection mechanism is based on the fact that hiding information in digital media alters some inherent statistics of the cover media which eventually results in degradation. The deviated perceptual and statistical properties of the stego data can be considered as indicators that can be utilized to prove the existence of a hidden message. Even if the presence of the hidden message is known, revealing its content is not always easy due to the usage of secondary encryption methods. It may also be unnecessary to decode the message. Once detected, rendering and corrupting the stego data will defeat the ultimate goal of steganography.

In terms of detection and attacking the stego data the steganalysis can be classified into two categories.

1. Passive Warden Steganalysis: Detect the existence or absence of a secret message in an observed stego data or identify the type of embedding algorithm.
2. Active Warden Steganalysis: Estimate/extract some properties of the message or the embedding algorithm. For example, extract a (possibly approximate) version of the secret message from a stego message.

Until recently, the research in information hiding focused on analyzing or evaluating the hiding algorithms' robustness against various attacks, aiming at removal or destruction of the embedded data. The detection of the presence of hidden data for different applications such as steganography and digital watermarking is the first requirement before manipulation or removal. This need has received intensive attention from the research community and has led to the development of steganalysis methods. Some exemplary steganalysis methods are discussed next.

Avcibas et al. [47] proposed a novel steganalysis framework based on a set of image quality measures to capture the perceptual and statistical signatures from the stego data which are then used in regression analysis to indicate the presence of the hidden data.

Gul et al. [48] investigated a Singular Value Decomposition (SVD) based blind image steganalysis method to detect, especially in the spatial domain, steganographic methods. Some other examples of steganalysis techniques proposed for detection of spatial domain methods based on least significant bit (LSB) modification are presented in Pevny [49], Yang [50], Dumitrescu [51] and Ker [52].

In [53] a novel SVD based steganalysis method against JPEG image based perturbed quantization (PQ) data hiding method is proposed. Authors showed that JPEG based PQ data hiding alters linear dependencies of rows and columns of pixel values which are used for training a classifier for differentiating cover and stego images.

Yet another steganographic scheme (YASS), a DCT domain QIM based method using repeat accumulate (RA) codes for error correction, proposed by Sarkar et al., was designed to resist blind steganalysis methods by embedding data in randomized locations. Bin et al. [54] proposed a novel steganalytic method to attack the YASS algorithm by partially accessing and extracting the features from JPEG quantized DCT domain where YASS algorithm embeds hidden message.

Budhia et al. [55] presented a steganalysis technique for digital video sequences based on an inter-frame collusion attack exploiting the temporal statistical visibility of hidden message. They claimed that steganalysis algorithms based on frame-by-frame analysis are suboptimal and proposed a method which uses redundant information present in the temporal domain to detect hidden messages embedded using spread spectrum techniques.

In another work [43], authors proposed a video steganalysis method based on

aliasing detection. They showed that by analyzing the aliasing in the probability mass function of the frame difference signal, frame-by-frame, additive type steganographic method can be detected.

Motion coherency has recently been identified as a desirable property for steganographic methods developed for video to resist temporal frame averaging. In [56] authors proposed a novel oracle to detect whether a video sequence contains any motion incoherent component or not. The proposed oracle exploits some features extracted from error frames after motion compensation. Through experimental results they showed that the proposed method can be used to detect the existence of a hidden message in compressed and uncompressed video streams.

Information hiding in video is commonly considered as data hiding in a sequence of still images. Hence, for video applications image based steganographic methods are usually employed. Doer et al. [57] investigated this strategy and proved that such frame-by-frame approach could not resist collusion based steganalysis attacks. They proposed alternative embedding strategies exhibiting superior performance against collusion attacks.

In a further related study, Su et al. [58] studied the linear collusion analysis of digital video sequences and presented a definition of statistical invisibility in which the presence of a collusion resistant watermark is not revealed using statistical tools. They proposed two video watermark design principles robust to linear collusion. In the first principle, they claim that the watermark strength should be adjusted according to some function of the image variance both at global and local scales. In the second principle, they propose that the correlation of the watermarks embedded into each pair of video frames should be matched to the host frames themselves. This implies that highly correlated video frames should be watermarked with highly correlated watermark patterns and vice versa [58].

And finally, in [23], the author examined the relationship between motion of an object in a video sequence and the perceptual visibility of hidden data (or watermark in their case) and investigated the so called “dirty window problem” originating from artifacts due to frame-by-frame data hiding approach resulting in visibility of the hidden data in video frames. Based on the experimental and analytical studies he proposed a new method as a solution to the perceptual visibility problem by taking motion magnitude and direction into account to adjust the embedding strength.

II.6 CHAPTER SUMMARY

In this chapter a generic model of the information hiding framework which can be used to describe any information hiding algorithm is presented. To make an analogy, the framework can be viewed as a classical communication channel in which the cover data plays the role as the carrier or part of the channel. To familiarize the reader, different types of data hiding algorithms classified into two broad categories namely additive and quantization are provided. From the designer's perspective, the central challenge of any information hiding algorithm is to balance conflicting requirements which manifest themselves as design criteria. For instance, it would be arguable to claim a data hiding system will supposedly provide high fidelity and robustness simultaneously, as these two criteria do conflict with each other. Satisfying one design criterion will exclude the other and vice versa. So it is the designer's task to set an optimum compromise among these conflicting criteria to meet the application specific goals.

Different applications and specifically the distinction between two commonly used data hiding applications, steganography versus watermarking, is elaborated.

Section II.4 covers data hiding applications for digital video which provide the linkage between previous work in this field and the proposed method. And finally, in other section, the so-called steganalysis techniques that are devised to detect the very presence of hidden channel are presented. The underlying assumption for this battery of tools is that any data hiding mechanism will eventually alter some features of the cover data which are scrutinized to indicate any perceptual and/or statistical deviations. Next, some exemplary known state-of-the-art steganalysis techniques are presented.

In summary the generic information hiding model building blocks, constraints and rationale behind design criteria and specific applications are presented in detail. From this perspective, this chapter sets the scene for the discussions for the proposed method by providing a general overview of the data hiding framework and providing linkage between previous work in video domain and the proposed method.

In the context of the general information hiding framework the embedding function could be linear or non-linear, as discussed before. For instance, in the commonly used dither modulation, two scalar quantizers are used. In Quantization Index Modulation (QIM) data hiding methods, an ensemble of uniform linear quantizers is used. In the proposed method a polar quantization scheme is used in place of an embedding

function. Therefore the next chapter will elaborate on the polar quantization basics, including distortion analysis of different polar quantization schemes and symbol error performance analysis of signal constellations obtained after using polar quantization of a source signal.

CHAPTER III

POLAR QUANTIZATION AND PERFORMANCE ANALYSIS

Different quantization schemes could be used in non-linear data embedding methods where the message is conveyed to the decoder side via quantized feature values e.g. Transform Coefficients, pixel values, motion vectors, etc. Polar quantization is an example of such schemes which allows us to quantize both the magnitude and phase of a joint variable.

As the proposed method utilizes both the magnitude and the phase of the VO motion, a polar quantizer is considered to be the ideal option for quantization. To provide the basics with the reader, general description of the quantization operation and then a detailed analysis of the polar quantization will be discussed next.

Note that, during the literature survey, an unsolved problem has been discovered on the exact probability of error analysis for 2-D non-uniform constellations. This problem was attacked in two approaches and the results are provided, showing an improvement in BER computations. The derivations of the proposed solution for the exact error probability computations and the results will also be presented later on.

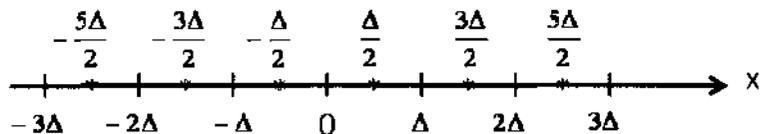
III.1 QUANTIZATION

Quantization is the division of a quantity into a discrete number of small parts, often assumed to be integral multiples of a common quantity. More generally, a quantizer can be defined as a set of disjoint intervals or *partitions* $\mathcal{S} = \{S_i; i \in \mathcal{I}\}$, where the index set \mathcal{I} is ordinarily a collection of consecutive integers beginning with 0 or 1, together with a set of reproduction levels $\mathcal{C} = \{y_i; i \in \mathcal{I}\}$, so that the overall quantizer is represented as function $\mathcal{Q} : \mathcal{S} \rightarrow \mathcal{C}$ where $\mathcal{Q}(x) = y_i$ if $x \in S_i$ where the function \mathcal{Q} is often called the *quantization rule*.

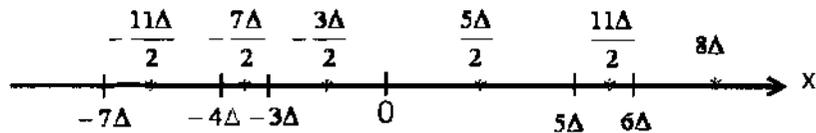
In scalar quantization, \mathcal{S} is a partition of the real line where the partitions are disjoint and exhaustive. In that case, the partitions can be represented as $S_i = (a_{i-1}, a_i]$.

A quantizer is said to be uniform if the partitions are equispaced, i.e. Δ apart, and the reproduction levels are midway between adjacent decision levels. In the case of non-uniform partitions the quantizer is called *non-uniform*.

An example of a uniform quantizer with step size Δ and nonuniform quantizer is



a. Uniform Quantizer



b. Nonuniform Quantizer

FIG. 6: Uniform and Nonuniform Quantization.

shown in Fig.6.

The distortion resulting from reproducing r as \hat{r} can be measured by a distortion measure $d(r, \hat{r})$. The most common distortion measure is the so-called squared error $d(r, \hat{r}) = |r - \hat{r}|^2$. In practice, average distortion is used to measure the quality in which the average distortion becomes an expectation $D(Q) = E[d(r, Q(r))]$.

III.2 POLAR QUANTIZATION

In polar quantization, a two dimensional random vector is quantized in terms of its phase θ and magnitude r . Polar quantizers are considered to be the natural choice for 2-D data with circularly symmetric density where the magnitude and angle components are independent.

Let $\mathbf{X}=(x_1, x_2)$ be the sample vector which can be represented by its polar coordinates (r, θ) . The magnitude is given by $R = \sqrt{x_1^2 + x_2^2}$ and distributed on $[0, \infty)$ while the phase is given by $\theta = \tan^{-1} \frac{x_2}{x_1}$, which is uniformly distributed on $[0, 2\pi)$. More generally one can write $z = re^{j\theta}$. Since the magnitude and phase are independent, the joint probability density function (pdf) of the sample vector can be written as

$$\begin{aligned} f_{\mathbf{X}}(r, \theta) &= f_{\theta}(\theta) \cdot f_R(r) \\ &= \frac{1}{2\pi} f_R(r) \end{aligned} \quad (7)$$

where $f_R(r)$ and $f_\theta(\theta)$ are the marginal pdfs of the magnitude and phase variable, respectively. Also note that if x_1 and x_2 are independent and identically distributed (IID) Gaussian random variables, then R is Rayleigh.

A polar Quantizer \mathcal{Q} with N cells is shown in Fig.7. The magnitude range is partitioned into L magnitude levels indexed by $i=1,2,\dots,L$ in which the magnitude decision levels and reconstruction levels are given as: $r_i = (i - 1)\Delta$ where $1 \leq i \leq (L + 1)$ and $\hat{r}_i = (i - \frac{1}{2})\Delta$ where $1 \leq i \leq L$. Note that Δ represents the step size. The boundaries of the amplitude levels are

$$r_0 = 0 < r_1 < r_2 < \dots < r_L \quad (8)$$

Let P denote the number of phase levels at each magnitude level \hat{r}_i such that the step size of the uniform phase quantizer is $\Delta_\theta = \frac{2\pi}{P}$. Then each magnitude ring is partitioned into P_i phase regions. The phase regions for the amplitude level r are

$$0 < \frac{2\pi}{P} < 2\frac{2\pi}{P} < \dots < (P - 1)\frac{2\pi}{P} < 2\pi \quad (9)$$

If $\phi_{i,j}$ and $\phi_{i,j+1}$ are the phase decision levels and $\hat{\phi}_{i,j}$ is the j^{th} phase reconstruction level for the i^{th} magnitude ring, then the decision and reconstruction levels for the phase quantizer can be written as

$$\phi_{i,j} = (j - 1)\frac{2\pi}{N} \text{ and } \hat{\phi}_{i,j} = (2j - 1)\frac{\pi}{N} \quad (10)$$

where $j = 1, 2, \dots, P_{i+1}$.

The polar quantizer in which the magnitude and phase are quantized independently is called ‘‘Restricted (or Strict) Polar Quantizer’’ where the total number of the quantization cells is $N = LP$. For this type of polar quantizer, the magnitude and phase rates are given by $\mathcal{R}_L = \log_2 L$ and $\mathcal{R}_\theta = \log_2 P$ respectively. The overall rate is $\mathcal{R} = \frac{1}{2}(\mathcal{R}_L + \mathcal{R}_P)$.

In an unrestricted polar quantizer, the phase is quantized after the magnitude and the phase quantizer varies with the quantized magnitude. In this case, the total number of quantization cells is $N = N_{\theta,1} + \dots + N_{\theta,N_L}$ where L is the number of magnitude levels and $\mathcal{R} = \frac{1}{2}\log_2 N$.

In nonuniform polar quantization, the magnitude is quantized with an arbitrary scalar quantizer. Fig.10 illustrates some examples of different polar quantizers.

The mean-square error (MSE) of the polar quantizer in which the quantized point for the cell S is defined by the magnitude and phase regions can be expressed as

$$D = E(|re^{j\theta} - \hat{r}e^{j\hat{\theta}}|^2) \quad (11)$$

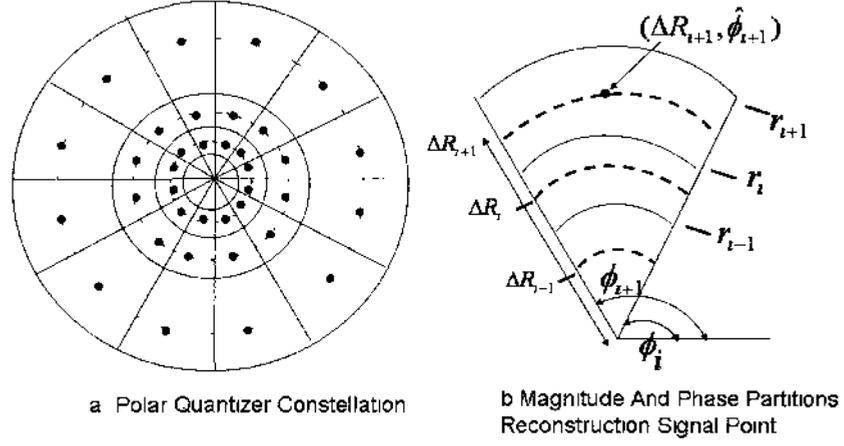


FIG. 7: Polar Quantizers Magnitude and Phase Partitions and Reconstruction Levels.

Assuming that the probability density functions for r and θ are known and they are statistically independent, then Mean Squared Error (MSE) can be simplified to

$$D = E(r^2) - 2E(r\hat{r})E(\cos(\theta - \hat{\theta})) + E(\hat{r}^2) \quad (12)$$

Since \hat{r} is a function of r and since r and \hat{r} are positive quantities, the minimization of D requires maximization of the term $E(\cos(\theta - \hat{\theta}))$. The estimator $\hat{\theta}$ that maximizes $E(\cos(\theta - \hat{\theta}))$ is also the estimator that minimizes the MSE of a phasor quantizer. In other words as described by Voran in [59] this can be stated as

$$D_\theta = E(|e^{j\theta} - e^{j\hat{\theta}}|^2) \quad (13)$$

It is interesting to note that when the phase quantization is infinitely finer than the term $E(\cos(\theta - \hat{\theta})) \rightarrow 1$ then MSE of a polar quantizer simplifies to the MSE of a magnitude quantizer. That is

$$D = E((r - \hat{r})^2) \quad (14)$$

It is well known that in a case where a uniform scalar quantizer with step size Δ is used, MSE can be approximated as $D \cong \frac{\Delta}{12}$.

III.2.1 Phase-Only Polar Quantizer

If we quantize only the phase and not the magnitude, then MSE becomes

$$D = E(r^2) - 2\hat{r}E(\cos(\phi - \hat{\phi}))E(r) + \hat{r}^2 \quad (15)$$

It can be easily shown that \hat{r} that minimizes MSE is $\hat{r} = E(r)E(\cos(\phi - \hat{\phi}))$. Then the MSE of a phase only quantizer is

$$D = E(r^2) - E(\cos(\phi - \hat{\phi}))^2 E^2(r) \quad (16)$$

When the number of phase levels N_θ increases and the step size Δ_Q decreases the term $E(\cos(\phi - \hat{\phi})) \rightarrow 1$. As a result of this $D = E(r^2) - E(r)^2 = \text{var}(r)$. So the smallest MSE for the phase-only quantization is the variance of the magnitude probability density function. If the phase is uniformly distributed on $(0, 2\pi]$ then the optimum phase quantizer is a uniform quantizer and $E(\cos(\phi - \hat{\phi})) = \text{sinc}(\frac{\pi}{P_i})$.

III.2.2 Magnitude Quantizer

As seen before, the phase quantizer design is independent of the magnitude. On the other hand, magnitude quantizer design depends upon the phase quantizer. For the sake of simplicity let $\alpha = E(\cos(\phi - \hat{\phi}))$. Recall that distortion (MSE) of a polar quantizer is defined as

$$D = E(r^2) - 2(r\hat{r}) + E(\hat{r}^2) \quad (17)$$

The solution for \hat{r} that minimizes distortion could be found easily by

$$\begin{aligned} D &= E(r^2) - 2\alpha E(r\hat{r}) + E(\hat{r}^2) \\ \frac{dD}{d\hat{r}} &= -2\alpha E(r) \int_{-\infty}^{\infty} \left(\frac{d}{d\hat{r}}\hat{r}\right) f_{\hat{r}}(\hat{r}) d\hat{r} + \int_{-\infty}^{\infty} \left(\frac{d}{d\hat{r}}\hat{r}^2\right) f_{\hat{r}}(\hat{r}) d\hat{r} \\ 0 &= -2\alpha E(r) \underbrace{\int_{-\infty}^{\infty} f_{\hat{r}}(\hat{r}) d\hat{r}}_1 + 2 \underbrace{\int_{-\infty}^{\infty} \hat{r} f_{\hat{r}}(\hat{r}) d\hat{r}}_{E(\hat{r})} \\ 0 &= -2\alpha E(r) + 2E(\hat{r}) \\ \alpha E(r) &= E(\hat{r}) \end{aligned} \quad (18)$$

which implies that $\alpha r = \hat{r}$

III.2.3 Numerical Form of Distortion

As shown for different polar quantization schemes, analyzing MSE in terms of expectation gives approximate results. In what follows we will derive exact numerical results for the distortion in terms of magnitude and phase levels which in turn will allow to computation of the resulting distortion in terms of number of quantization cells. To do this analysis we took the approach described in [60] as follows. Let the average distortion of a polar quantizer be rewritten as

$$\begin{aligned}
D &= E(|re^{j\theta} - \hat{r}e^{j\hat{\theta}}|^2) \\
&= \sum_{i=1}^L \sum_{j=1}^{P_i} \int \int_{S_{i,j}} |re^{j\theta} - \hat{r}e^{j\theta_Q}|^2 f_X(r, \theta) dr d\theta \\
&= \sum_{i=1}^L \sum_{j=1}^{P_i} \int_{r_i}^{r_{i+1}} \int_{\phi_{i,j}}^{\phi_{i,j+1}} [r^2 + \hat{r}_i^2 - 2r\hat{r}_i \cos(\theta - \hat{\phi}_{i,j})] \frac{f(r)}{2\pi} dr d\theta \\
&= \sum_{i=1}^L \sum_{j=1}^{P_i} \int_{r_i}^{r_{i+1}} \left[(r^2 + \hat{r}_i^2)(\phi_{i,j+1} - \phi_{i,j}) - 2r\hat{r}_i [\sin(\phi_{i,j+1} - \hat{\phi}_{i,j}) \right. \\
&\quad \left. + \sin(\phi_{i,j} - \hat{\phi}_{i,j})] \right] \frac{f(r)}{2\pi} dr d\theta \tag{19}
\end{aligned}$$

Using the fact that $\phi_{i,j+1} - \phi_{i,j} = \frac{2\pi}{P}$ and $\phi_{i,j} - \hat{\phi}_{i,j} = -(\phi_{i,j-1} - \hat{\phi}_{i,j}) = \frac{\pi}{P}$ and also $\text{sinc}(x) = \frac{\text{sn}(x)}{x} \approx 1 - \frac{1}{6}x^2 + \epsilon(x)$ we simplify (19) as

$$\begin{aligned}
D &= \sum_{i=1}^L \sum_{j=1}^{P_i} \int_{r_i}^{r_{i+1}} \left[(r^2 + \hat{r}_i^2) \frac{2\pi}{P_i} - 2r\hat{r}_i (\sin(\frac{\pi}{P_i}) + \text{sinc}(\frac{\pi}{P_i})) \right] \frac{f(r)}{2\pi} dr \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \left[(r^2 + \hat{r}_i^2) - 2r\hat{r}_i (\text{sinc}(\frac{\pi}{P_i})) \right] f(r) dr \\
&= \frac{r_{max}^2}{12L^2} + \sum_{i=1}^L \left[\frac{\hat{r}_i^2 \pi^2 f(r_i) r_{max}}{3P_i^2 L} \right] \tag{20}
\end{aligned}$$

where we have used the approximation $\int_{r_i}^{r_{i+1}} r f(r) dr \approx \hat{r}_i f(r_i) \Delta_L$ and $\Delta_L = \frac{r_{max}}{L}$ is the step size.

Peric et al. in [60] have studied optimization problems for polar quantizer by forming the equation $J = D + \lambda P_i$ where λ represents the Lagrangian multiplier. After solving for the optimum number of phase levels they showed that the distortion given in (20) can be simplified to

$$D = \frac{r_{max}^2}{12L^2} + \frac{L^2 \pi^2}{3N^2 r_{max}^2} \left(\int_0^{r_{max}} \sqrt[3]{r^2 f_X(r)} dr \right)^3 \tag{21}$$

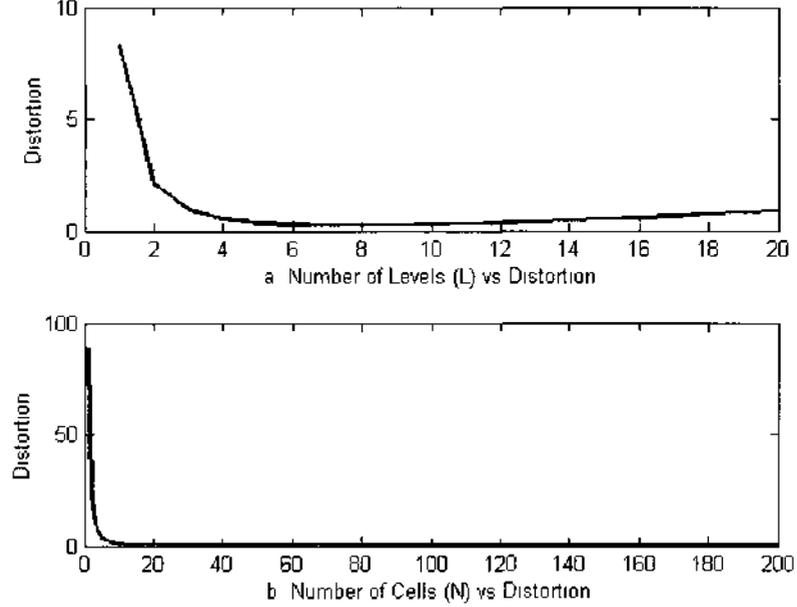


FIG. 8: Distortion as a Function of Levels L and Number of Cells N for a Source with Density Function $f(r) = re^{-\frac{r^2}{2}}$. a. Distortion vs. Varying number of Magnitude Quantization Levels. b. Distortion vs Varying number of Cells at fixed level $L=10$.

where N represents the number of cells.

Example-1: Consider a source with Gaussian Density Function $f(r) = re^{-\frac{r^2}{2}}$.

After plugging in the density function $f(r)$ in (21) and simplifying we can get

$$D = \frac{r_{max}^2}{12L^2} + \frac{9L^2\pi^2}{N^2r_{max}^2} \left(1 - e^{-\frac{r_{max}^2}{6}}\right)^3 \quad (22)$$

Fig. 8 shows a plot of (22) as a function of varying levels L and in Fig. 8.b as a function of the number of cells keeping the level fixed at level $L=10$.

Example-2: Consider magnitude distribution with a Uniform probability density function

$$f(r) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq r \leq b, \\ 0 & \text{else.} \end{cases}$$

Using (21) and simplifying after substituting the distortion for the polar quantizer

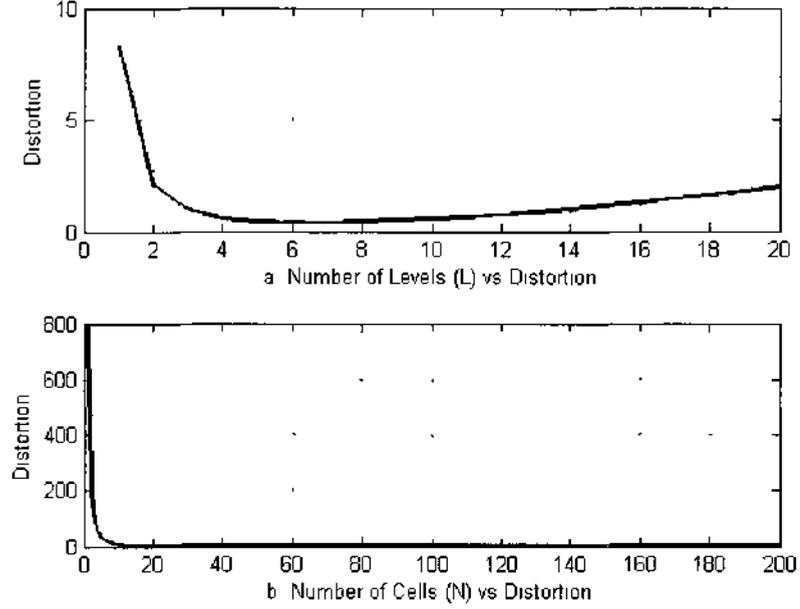


FIG. 9: Distortion as a Function of Levels L and Number of Cells N for a Source with Density Function $f(r) = \frac{1}{b-a}$, $a \leq r \leq b$ a. Distortion vs. Varying number of Magnitude Quantization Levels. b. Distortion vs Varying number of Cells at fixed level $L=8$.

can be found as

$$D = \frac{r_{max}^2}{12L^2} + \frac{L^2\pi^2}{5N^2} \quad (23)$$

Fig. 9 shows a plot of (23) as a function of varying levels L and in Fig. 9.b as a function of the number of cells keeping the level fixed at level $L=10$.

It is obvious that the number of levels can be found from the Distortion-Level figures approximately or analytically by solving the equation using the necessary condition i.e., $\frac{\partial D}{\partial L}$. If the latter approach is taken, then the optimal distortion can further be simplified as given in [60]

$$D^{opt} = \frac{\pi}{3N} \left(\int_0^{r_{max}} \sqrt[3]{r^2 f_X(r)} dr \right)^{\frac{3}{2}} \quad (24)$$

And finally the optimum distortion of a polar quantizer for different density functions can be found by numerically evaluating (24).

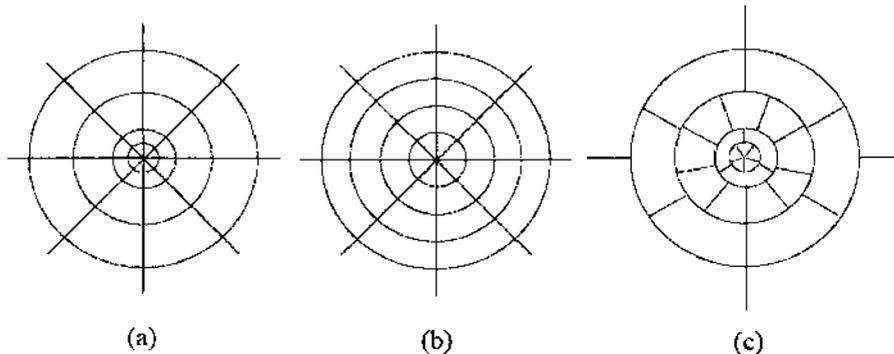


FIG. 10: Polar Quantizers a. Restricted Nonuniform b. Restricted uniform c. Non-restricted Nonuniform [3].

III.3 DERIVATION OF ERROR PROBABILITY

The error analysis of a polar quantization scheme, by considering the stego data to be sent through an Additive White Gaussian Noise Channel (AWGN), is analyzed. In order to help the reader understand the error probability derivations, some basics of signal constellations and related error probability definitions will be explained next. Some examples of basic signal constellations are shown in Fig.11.

An N -dimensional *signal constellation*, \mathcal{A} , can be denoted by

$$\mathcal{A} = \{a_i, 1 \leq i \leq M\} \quad (25)$$

where $a_i \in \mathbb{R}^N$ is the *signal point*, and M represents the number of signal points. Some basic 1-D and 2-D signal constellations are illustrated in Fig. 11.

In the decoding stage, the received signal $y = x + n$, where n is the noise, is mapped into an estimate of the transmitted signal sequence under a decision rule such as *minimum-distance*, which can simply be stated as given y choose $\hat{a}_j \in \mathcal{A}$ such that $\|y - \hat{a}_j\|^2$ is minimized among all $\|y - a_j\|^2$ $a_j \in \mathcal{A}$. The decision regions are obtained by partitioning the real N -D space \mathbb{R}^N into M regions \mathcal{R}_j , $1 \leq j \leq M$ where \mathcal{R}_j includes the received vectors y which are closer to a_j than any other point in \mathcal{A} .

$$\mathcal{R}_j = \{y \in \mathbb{R}^N : \|y - a_j\|^2 \leq \|y - a_{j'}\|^2 \forall j' \neq j\} \quad (26)$$

Detection error occurs when the noise causes the received data to fall into a wrong

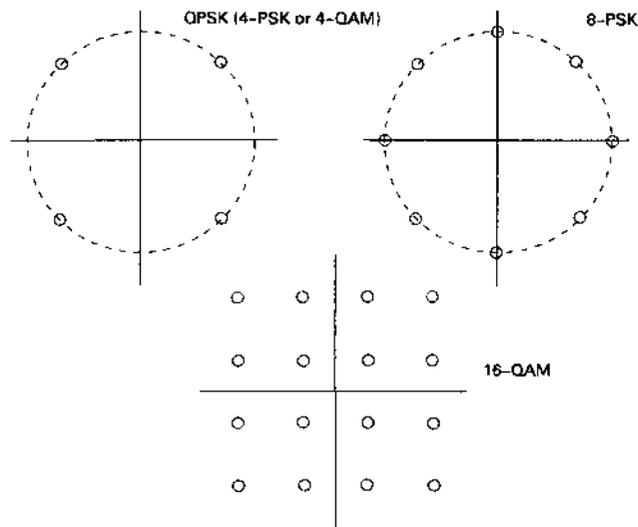


FIG. 11: Some Examples of Signal Constellations.

detection region. The probability of error can be explained simply as given the signal a_j is transmitted the probability that the received signal y fall outside the decision region \mathcal{R}_j whose center is located at a_j . In the next section, the error probability derivations of three different polar quantization schemes will be examined.

III.3.1 Magnitude or Phase-Only Quantization

In the case of magnitude only quantization, the magnitude is quantized with a scalar quantizer. For this quantization scheme, the probability of error analysis will be similar to that of Pulse Amplitude Modulation (PAM) scheme. Assume that a 2-bit polar quantizer with step size $\Delta = 2r$ is used to quantize the magnitude with dynamic range $(0, r_{max}]$. This quantization scheme will result in the the signal space representation with the corresponding decision regions as shown in Fig.12. Note that this representation is similar to well known 4-PAM modulation scheme in digital communications.

Let the signal x go through an AWGN channel, yielding the received signal $y = x+n$ where AWGN n is Gaussian distributed with zero mean and variance $\sigma^2 = N_0/2$.

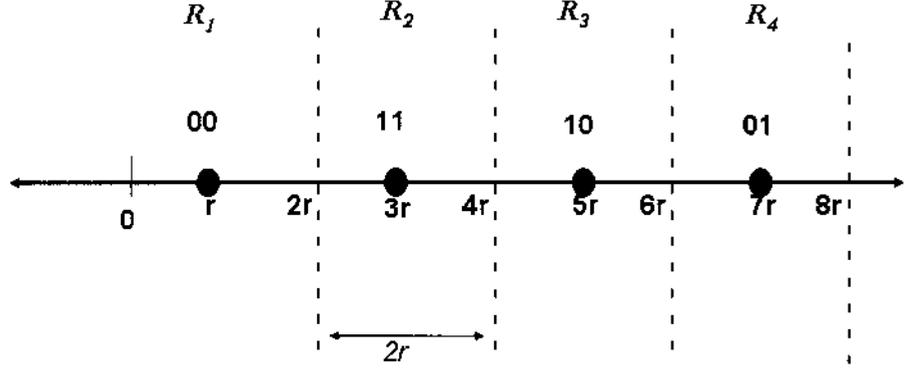


FIG. 12: Signal Constellation of Magnitude Only Polar Quantizer with $\Delta = 2r$. Reconstruction Points are shown as dots whereas \mathcal{R}_j 's represent corresponding decision regions.

By using conditional error probabilities the symbol error probability P_e can be defined by

$$\begin{aligned} P_e &= \sum_{i=0}^3 P(e|s_i)P(s_i) \\ &= P(e|00)P(00) + P(e|01)P(01) + P(e|10)P(10) + P(e|11)P(11) \end{aligned} \quad (27)$$

where s_i represents the symbols and $P(s_i)$ is the probability of occurrence of the individual symbols. A symbol error occurs whenever the received signal y does not fall into the corresponding decision regions. Individual symbol errors are computed as follows.

$P(e|00) = P(y_1 \notin Z_1)$ and $P(e|01) = P(y_4 \notin Z_4)$ are the same due to the symmetry in the signal space diagram. And therefore the symbol error for both cases can be written as

$$P(y_4 \notin Z_4) = P(y_1 \notin Z_1) = \int_{2r}^{\infty} \frac{1}{\sqrt{2\pi N_0/2}} e^{-\frac{1}{2} \left(\frac{y-r}{\sqrt{N_0/2}} \right)^2} dy \quad (28)$$

letting $z = \frac{y-r}{\sqrt{N_0/2}}$ and taking the derivative $\frac{dz}{dy} = \frac{1}{\sqrt{N_0/2}}$, after reorganizing the equation can be simplified to

$$P(y_4 \notin Z_4) = \int_{r/\sqrt{N_0/2}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z^2)} dz = Q\left(\frac{r}{\sqrt{N_0/2}}\right)$$

Assuming equal symbol probabilities the average energy of the signal can be written as $E = \frac{1}{4}[r^2 + 9r^2 + 25r^2 + 49r^2] = 21r^2$ so that $r = \sqrt{\frac{E}{21}}$. Substituting back in the result one can obtain $P(y_4 \notin Z_4) = P(y_1 \notin Z_1) = Q\left(\sqrt{\frac{2E}{21N_0}}\right)$

Probability of symbol error for the other two cases ($P(y_3 \notin Z_3) = P(y_2 \notin Z_2)$) can be computed as

$$P(y_3 \notin Z_3) = \int_{-\infty}^{2r} \frac{1}{\sqrt{2\pi N_0/2}} e^{-\frac{1}{2}\left(\frac{y-3r}{\sqrt{N_0/2}}\right)^2} dy + \int_{4r}^{\infty} \frac{1}{\sqrt{2\pi N_0/2}} e^{-\frac{1}{2}\left(\frac{y-3r}{\sqrt{N_0/2}}\right)^2} dy$$

Following the same approach taken in the previous case, the result can be found as

$$P(y_3 \notin Z_3) = P(y_2 \notin Z_2) = 1 - Q\left(\frac{-r}{\sqrt{N_0/2}}\right) + Q\left(\frac{r}{\sqrt{N_0/2}}\right) = 2Q\left(\frac{r}{\sqrt{N_0/2}}\right) \quad (29)$$

Assuming equal symbol probabilities the probability of a symbol error is given by

$$P_e = \frac{3}{2}Q\left(\frac{r}{\sqrt{N_0/2}}\right) = \frac{3}{2}Q\left(\sqrt{\frac{2E}{21N_0}}\right) \quad (30)$$

Taking into consideration that the average energy per bit $E_b = \frac{1}{2}E$, the above expression can be written in terms of bit error probability approximately as:

$$P_b \approx \frac{3}{4}Q\left(\sqrt{\frac{4E_b}{21N_0}}\right) \quad (31)$$

Using the above approximation the probability of bit error for varying $\frac{E_b}{N_0}$ is plotted in Fig.13.

III.3.2 Union Bound on Probability of Error for Polar Quantization

Union bounds are widely used in computing the probability of error for a variety of signal constellations. It can be stated as $Pr(E|a_j)$ is upper bounded by the sum of the pairwise error probabilities to all other signals such that

$$Pr(E|a_j) \leq \sum_{a_j \neq a'_j} Q\left(\frac{d(a_j, a'_j)}{2\sigma}\right) \quad (32)$$

Let \mathcal{S} denote the set of distances between the signal points in the constellation so than Union Bound Estimate can be written as

$$Pr(E|a_j) \leq \sum_{d \in \mathcal{S}} \kappa_d(a_j) Q\left(\frac{d}{2\sigma}\right) \quad (33)$$

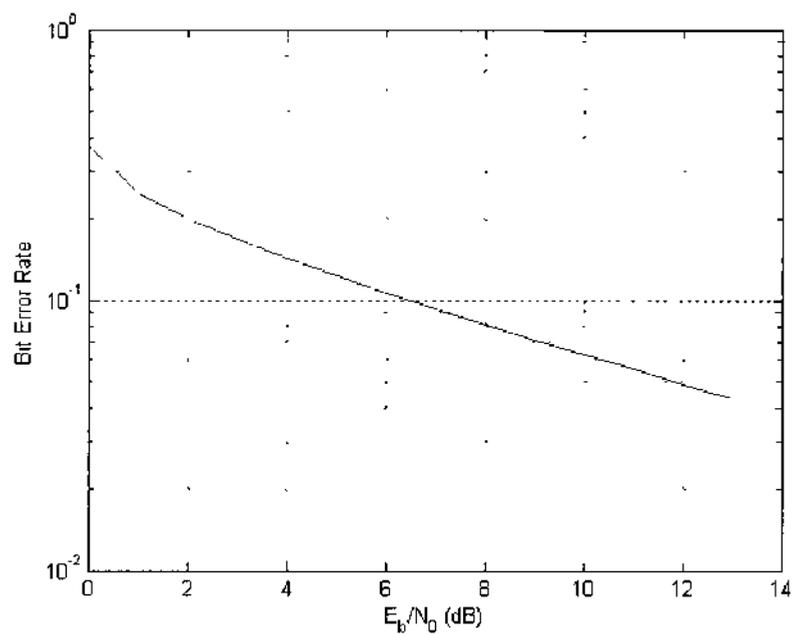


FIG. 13: Probability of Bit Error vs SNR for Polar Quantizer with 4 Levels and Magnitude Quantization Only. In this case $P_b \approx \frac{3}{4}Q\left(\sqrt{\frac{4E_b}{21N_0}}\right)$.

where $\kappa_d(a_j)$ is the number of signals at distance d from a_j . In a particular constellation, there are a few of these distances that are significantly smaller than the others in which case they dominate the sum due to the following property of the Q function. Q function is defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy$ which decreases exponentially due to the term $e^{-\frac{y^2}{2}}$. Consequently its value will be largest for the minimum Euclidean distance. Interested readers should refer to Appendix-A for properties of Q function.

If there are $\kappa_{min}(a_j)$ neighbors at distance d_{min} from the signal point a_j , then

$$Pr(E|a_j) \approx \kappa_{min}(a_j) Q\left(\frac{d_{min}}{2\sigma}\right) \quad (34)$$

Readers should note that this estimate is valid only if the next nearest neighbors are at a significantly greater distance and there are not too many of them. If these assumptions are violated, then further terms should be used in the estimate. A bound on the total probability of error is obtained by weighting each $Q\left(\frac{d_{min}}{2\sigma}\right)$ by its probability of occurrence i.e.

$$Pr(E) \approx \kappa_{min} Q\left(\frac{d_{min}}{2\sigma}\right) \quad (35)$$

where κ_{min} is the average number of signal points at distance d_{min} .

Example:

The probability of symbol error of the signal constellation constellation given in Fig.15 will be analyzed next. Assume that the magnitude values only lie in the first quadrant. From the signal constellation geometry, the smallest distance from any signal point to its decision boundary is $d_{min}/2 = \hat{r} \sin \theta$ where \hat{r} is the reconstruction level (quantized magnitude) and it occurs one time for outer signal points and twice for the inner points. And the next such large distance is the distance between two quantized magnitudes, \hat{r}_i and \hat{r}_{i-1} , where the distance is $2r$. For the outer signal points, it occurs twice whereas for inner signal points, it occurs three times. For the sake of simplicity assume that $r_{max} = 8r$ and $P = 4$ (Phase Quantization Levels) and therefore $P_i = \pi/8$, resulting in $d_{min}/2 = \hat{r} \sin(\pi/8)$. The magnitude is quantized with the uniform scalar quantizer illustrated in Fig.12.

After examining the constellation it can be seen that $d_{min}/2 = r$ occurs 24 times, which can be found intuitively by $2(L - 1)P$. Total number of cells in the first quadrant is equal to 16. So the sum of symbol error probability for those signal points is simply

$$Pr_r(S) = \frac{24}{16} Q\left(\frac{r}{\sigma}\right) \quad (36)$$

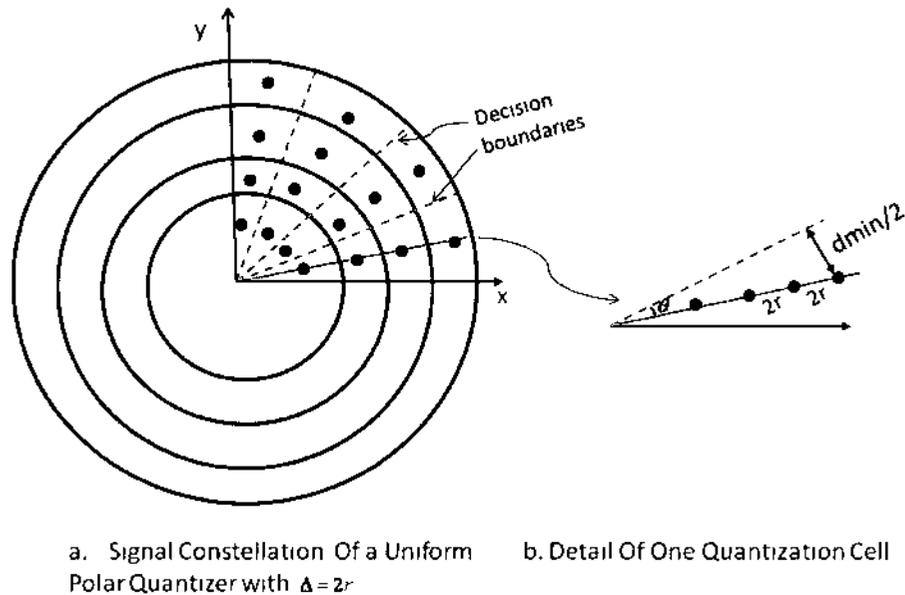


FIG. 14: a. Signal Constellation of a Uniform Polar Quantizer with $\Delta = 2r$. b. Detail of One Quantization Cell where d_{min} is the distance between a_i and a_j , and $\frac{d_{min}}{2}$ is the distance to the side of decision boundary.

In a same manner, for the signal points where $\hat{r} = 7r$, the distance $d_{min}/2 = \hat{r} \sin(\pi/8)$ to the side of the decision boundaries occur 6 times. Also, this can be obtained by inspection or by intuition from $2(P-1)$ where P is the number of phase levels. So the total symbol error probability for those signal points is

$$\begin{aligned}
 Pr_{7r}(S) &= \frac{6}{16} \left[Q\left(\frac{7r \sin(\pi/8)}{\sigma}\right) \right] \\
 &= \frac{6}{16} \left[Q\left(\frac{2.678r}{\sigma}\right) \right]
 \end{aligned} \tag{37}$$

By the same token the total error probability for signal points at $\hat{r} = 5r$ and $\hat{r} = 3r$ can be found as

$$\begin{aligned}
 Pr_{5r}(S) &= \frac{6}{16} \left[Q\left(\frac{5r \sin(\pi/8)}{\sigma}\right) \right] \\
 &= \frac{6}{16} \left[Q\left(\frac{1.9134r}{\sigma}\right) \right]
 \end{aligned} \tag{38}$$

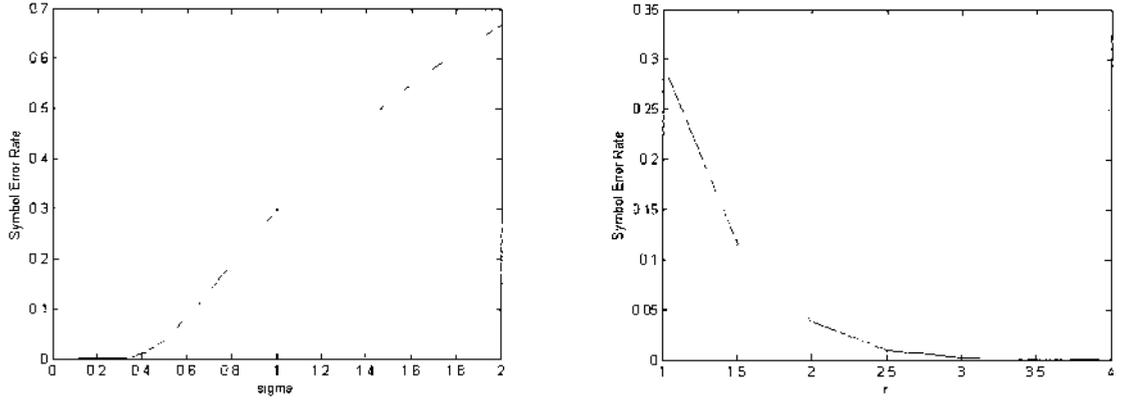


FIG. 15: Left. Union bound estimate of symbol error probability: Plot of (40) when $r=1$ fixed and σ is varied. Right. Symbol error probability as a function of r when $\sigma=0.1$.

$$\begin{aligned}
 Pr_{3r}(S) &= \frac{6}{16} \left[Q\left(\frac{3r \sin(\pi/8)}{\sigma}\right) \right] \\
 &= \frac{6}{16} \left[Q\left(\frac{1.1481r}{\sigma}\right) \right]
 \end{aligned} \tag{39}$$

Union bound on the symbol error probability can be obtained by summing (36), (39), (38) and (37) so that

$$\begin{aligned}
 Pr(E) &\leq Pr(S) \\
 &\leq Pr_r(S) + Pr_{3r}(S) + Pr_{5r}(S) + Pr_{7r}(S) \\
 &\leq \frac{24}{16} Q\left(\frac{r}{\sigma}\right) + \frac{6}{16} \left[Q\left(\frac{1.1481r}{\sigma}\right) + Q\left(\frac{1.9134r}{\sigma}\right) + Q\left(\frac{2.678r}{\sigma}\right) \right]
 \end{aligned} \tag{40}$$

Union bound estimate of the symbol error probability for fixed r and varying σ is illustrated in Fig.15. As expected, when r is increased, the probability of error decreases due to the increased distance between the signal points in the signal constellation.

III.3.3 An Exact Result for Symbol Error Probability for Polar Quantization

In this section, a novel approach for the derivation of exact error probability is given. Let $\mathbf{X}=(x_1, x_2)$ be the 2-D signal vector which can be represented by its polar coordinates (r, θ) . The magnitude is given by $R = \sqrt{x_1^2 + x_2^2}$ which is distributed on $[0, \infty)$ and the phase is given by $\theta = \tan^{-1} \frac{x_2}{x_1}$ which is uniformly distributed on $[0, 2\pi)$. From Fig.7 and the quantization cell notations provided in the previous section, it is obvious that R corresponds to the ΔR . Hereafter R and ΔR will be used interchangeably. As the magnitude and phase are independent, joint probability density function (pdf) can be written as

$$f_{\mathbf{X}}(R, \theta) = \frac{1}{2\pi} f_R(R) \quad (41)$$

where $f_R(R)$ is the marginal pdf of the magnitude variable. If x_1 and x_2 are independent and identically distributed (IID) Gaussian random variables, then R is Rayleigh distributed:

$$f_R(R) = \frac{R}{\sigma^2} e^{-\frac{R^2}{2\sigma^2}} U(R) \quad (42)$$

Let the received signal be written as $Y = X + N$ where N is the 2-D noise with AWGN on each component (n_1, n_2) each having zero mean and variance σ_n^2 . Hence, the received signal radius $A = \sqrt{(x_1 + n_1)^2 + (x_2 + n_2)^2}$ is Rician distributed with pdf of the form

$$f_A(A, R) = \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) U(A) \quad (43)$$

where $R = \sqrt{x_1^2 + x_2^2}$ and I_0 is the zeroth order modified Bessel function of the first kind.

The received signal phase will be uniformly distributed on $[0, 2\pi)$. As the magnitude and phase are independent, the pdf of the received signal can be written as

$$\begin{aligned} f_{\mathbf{Y}}(A, R, \theta) &= \frac{1}{2\pi} f_A(A, R) \\ &= \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) U(A) \end{aligned} \quad (44)$$

Now let us begin the analysis of the symbol error probability by first computing the probability of a correct decision P_c when an arbitrary symbol is transmitted.

Under a decision rule such as the *maximum-likelihood*, if the received signal Y falls in the decision region \mathfrak{D}_i , the receiver decides that the symbol s_i is transmitted. Hence, the probability of correct decision for a specific quantization cell can be calculated as follows:

$$\begin{aligned}
P_c(s_i) &= \sum_{i=1}^L \sum_{j=1}^{L_i} \int_{r_i}^{r_{i+1}} \int_{\phi_i}^{\phi_{i+1}} f_Y(A, R, \theta) d\phi dA \\
&= \sum_{i=1}^L \sum_{j=1}^{L_i} \int_{r_i}^{r_{i+1}} \int_{\phi_j}^{\phi_{j+1}} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) d\phi dA \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \sum_{j=1}^{L_i} \int_{\phi_j}^{\phi_{j+1}} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) d\phi dA \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \sum_{j=1}^{L_i} \frac{2\pi}{P_i} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} P_i \frac{2\pi}{P_i} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA
\end{aligned} \tag{45}$$

where we have used the fact that $\phi_{i,j+1} - \phi_{i,j} = \frac{2\pi}{P_i}$ and the total number of phase levels at magnitude level L_i is P_i i.e. $\sum_{j=1}^{L_i} P_j = P_i$. Note that there is no closed form solution for the above expression, and two approaches based on approximations will be shown next.

III.3.4 Bessel Function Approximation

In this approach the Bessel function can be approximated by

$$I_\alpha(x) \approx \frac{1}{\sqrt{2\pi x}} e^x \quad \text{if } x \gg \left| \alpha^2 - \frac{1}{4} \right| \tag{46}$$

Using this approximation $P_c(s_i)$ at (45) simplifies to

$$\begin{aligned}
P_c(s_i) &= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \frac{A}{\sigma_n^2} e^{-\frac{(R^2 + A^2)}{2\sigma_n^2}} \frac{\sigma_n}{\sqrt{2\pi AR}} e^{\frac{AR}{\sigma_n^2}} dA \\
&= \sum_{i=1}^L \int_{r_i}^{r_{i+1}} \frac{1}{\sigma_n} \sqrt{\frac{A}{2\pi R}} e^{-\frac{(A-R)^2}{2\sigma_n^2}} dA
\end{aligned} \tag{47}$$

III.3.5 Marcum Q-function

The probability of correct decision in (45) can also be evaluated numerically by utilizing Marcum's Q-Function which is available in common software packages. The expression of the first order Marcum's Q-function is

$$Q_1(a, b) = \int_b^\infty x e^{-\frac{(a^2+x^2)}{2}} I_0(ax) dx \quad (48)$$

Using (48) and noting that the difference between phase boundaries at each magnitude level is $\phi_{i,j+1} - \phi_{i,j} = \frac{2\pi}{P}$, then we can further simplify $P_c(s_i)$ as

$$\begin{aligned} P_c(s_i) &= \sum_{i=1}^L \sum_{j=1}^{L_i} \int_{r_i}^{r_{i+1}} \int_{\phi_i}^{\phi_{i+1}} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2+A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) d\phi dA \\ &= \sum_{i=1}^L \sum_{j=1}^{L_i} \left[\int_{r_i}^\infty \frac{2\pi}{P_i} \frac{1}{2\pi} \frac{A}{\sigma_n^2} e^{-\frac{(R^2+A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \right. \\ &\quad \left. - \int_{r_{i+1}}^\infty \frac{1}{2\pi} \frac{2\pi}{P_i} \frac{A}{\sigma_n^2} e^{-\frac{(R^2+A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \right] \\ &= \sum_{i=1}^L \left[\int_{r_i}^\infty \frac{P_i}{P_i} \frac{A}{\sigma_n^2} e^{-\frac{(R^2+A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \right. \\ &\quad \left. - \int_{r_{i+1}}^\infty \frac{P_i}{P_i} \frac{A}{\sigma_n^2} e^{-\frac{(R^2+A^2)}{2\sigma_n^2}} I_0\left(\frac{AR}{\sigma_n^2}\right) dA \right] \\ &= \sum_{i=1}^L \left[Q_1\left(\frac{R}{\sigma_n}, \frac{r_i}{\sigma_n}\right) - Q_1\left(\frac{R}{\sigma_n}, \frac{r_{i+1}}{\sigma_n}\right) \right] \end{aligned} \quad (49)$$

III.3.6 Probability of Symbol Error

Once a closed-form expression for (45) is found using one of the approximations mentioned above, the symbol error probability for s_i can be found as $P_e(s_i) = 1 - P_c(s_i)$. Assuming that $P(s_i)$ is the probability that signal s_i is transmitted, the average probability of symbol error P_s for a set of \mathcal{M} signal points is

$$P_s = \sum_{i=1}^{\mathcal{M}} P(s_i) P_e(s_i) \quad (50)$$

Note that, in practice, when the geometry of the signal constellation is rather complex, the usual approach is to find an upper bound such as the Union Bound

on the symbol error probability. Union bounds are based on the distance between the signal point and the decision boundary, but instead of using all the distances, only a particular set of smallest distances is used. For a given signal point, the symbol error probability is upper bounded by $Q(\frac{d_{min}}{2\sigma_n})$, and the union bound on the total probability of error is obtained by weighting each $Q(\frac{d_{min}}{2\sigma_n})$ by its probability of occurrence, that is

$$P_s \approx \kappa_{min} Q\left(\frac{d_{min}}{2\sigma_n}\right) \quad (51)$$

where κ_{min} is the average number of points in the signal constellation that are at distance d_{min} .

III.3.7 Experimental Results

In this section, experimental results illustrating the symbol error probability obtained using the approach are presented. Readers should note that, since the focus of this section is the analysis of error probability and not the development of a new constellation optimization method or the design of a polar quantizer, it was assumed that the 2D signal constellations considered were obtained by using any of the existing constellation design algorithms.

Fig. 16 illustrates the symbol error probability obtained using the Bessel function approximation and the Marcum-Q function approach for a non-uniform constellation with three magnitude reconstruction levels and 12 phase partitions at each level. Note that two expressions result in values of the symbol error rate that are consistent with each other up to about 20 dB, when they start to diverge because of the differences in the approximation. It can be easily seen that, for SNR values beyond 25 dB the Marcum-Q function approach implies more meaningful symbol error values than the Bessel Function approximation which appears to flatten out.

Fig. 17 plots the symbol error probability corresponding to the signal constellation shown in Fig.10.a., obtained after using a restricted uniform polar quantizer. It is assumed that there are $L = 3$ magnitude levels with the magnitude decision and reconstruction levels are $r_i = \{0, 4, 10, 13\}$ and $\Delta R = \{2, 7, 11\}$ respectively. Furthermore, each magnitude ring, $\Delta R_{i+1} - \Delta R_i$, is partitioned into $P_i = 12$ phase sub-partitions. These parameters result in a nonuniform signal constellation for which the smallest distance from any signal point to its decision boundary is $d_1 = r_1 \sin(\pi/12) = 0.51$ and occurs 24 times for the most inner circle. The next such larger distance is $d_2 = 1$ and exists 12 times for the inner-most circle. After examining the

other signal points in the constellation in a similar manner, the union bound on the symbol error is found to be

$$P_s = \frac{24}{36} \left[Q\left(\frac{2 \sin(\pi/12)}{\sigma_n}\right) \right] + \frac{24}{36} \left[Q\left(\frac{7 \sin(\pi/12)}{\sigma_n}\right) \right] + \frac{12}{36} \left[Q\left(\frac{1}{\sigma_n}\right) \right] + \frac{12}{36} \left[Q\left(\frac{2}{\sigma_n}\right) \right] \quad (52)$$

The union bound is also shown in Fig.17, and note that the proposed method for calculation of the symbol error rate implies an improvement in E_s/N_o that is between 5 and 8 dB. This result is significant when the exact error probability is used in signal constellation optimization or performance evaluation.

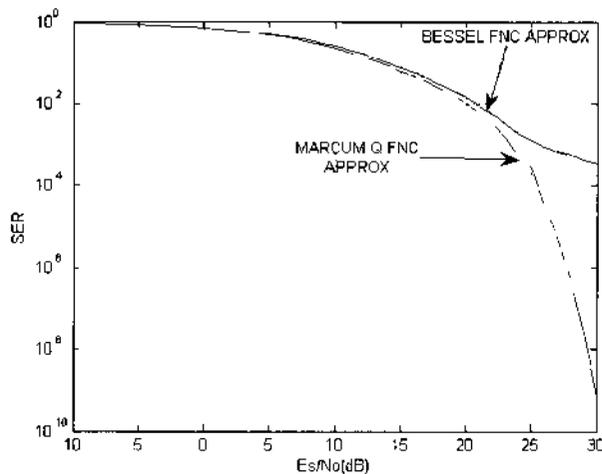


FIG. 16: Bessel Function and Marcum-Q function Approximations to Exact Error Probability.

The next example is the (4,11,17) 32-QAM constellation shown in Fig.18 with symbols having different probabilities of transmission $P_{s-i} = \{1/8, 11/31, 17/32\}$. The constellation contains three circular constellations where the inner, central and the outer circles contain 4, 11, and 17 symbols respectively. The following magnitude and phase levels are assumed: $L = 3$, $P_i = (4, 11, 17)$, and the reconstruction levels are $\hat{r} = \Delta R = (1, 4, 8)$ with decision boundaries $r_i = (0, 3, 6, 10)$. The union bound estimate for the symbol error probability of this constellation is found by using the analytical expressions given in [61, pp. 7-21]. The symbol error rate for this constellation are shown in Fig.19, for which it is noted that, similar to the previous example, that the proposed expressions for error probability method imply a similar improvement (of approximately 8 dB) in E_s/N_o .

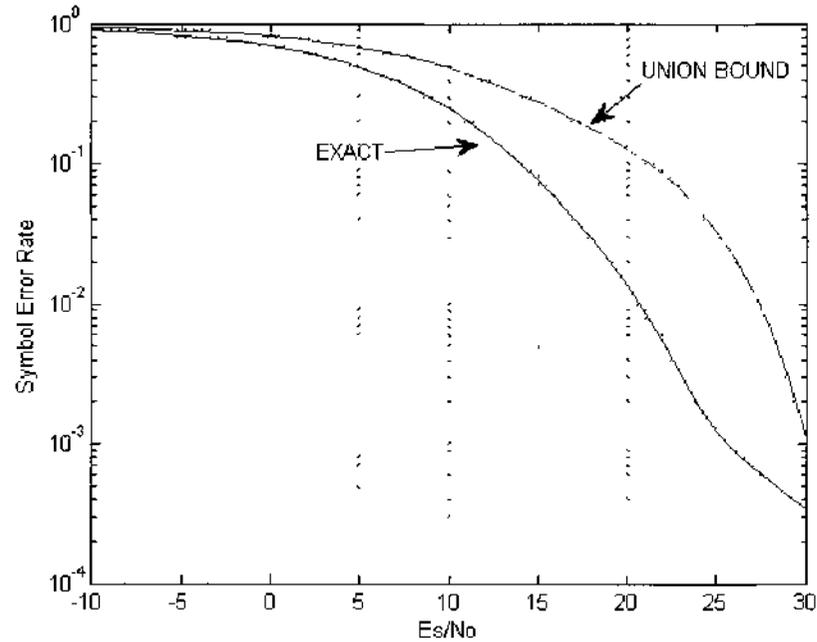


FIG. 17: The Probability Of Symbol Error For Nonuniform Signal Constellation Obtained by Using a Restrictive Nonuniform Polar Quantizer As Shown in Fig.10.a.

III.4 CHAPTER SUMMARY

This chapter provides a detailed study of polar quantization by focusing on quantization distortion and error probability analysis. MSE measure is used for distortion analysis. A closed form expression for optimum distortion and the number of phase and magnitude quantization levels are derived and some examples with distributions are presented. For the error probability analysis, different quantization scenarios are investigated. Specifically phase or magnitude only quantization schemes are analyzed similar to PAM error analysis. For the specific case of magnitude and phase quantization, the union bound estimate of the error probability is derived and an example is given.

The next chapter is devoted to the discussion of VO tracking to provide enough background for the reader since the proposed data hiding method aims at modifying the VO trajectory segments.

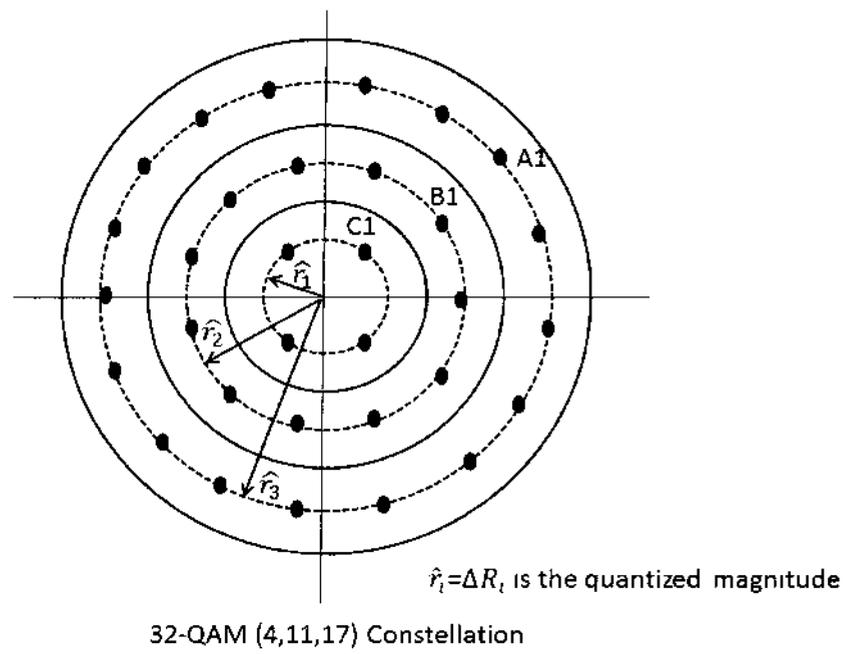


FIG. 18: An Example of Non-uniform Signal Constellation.

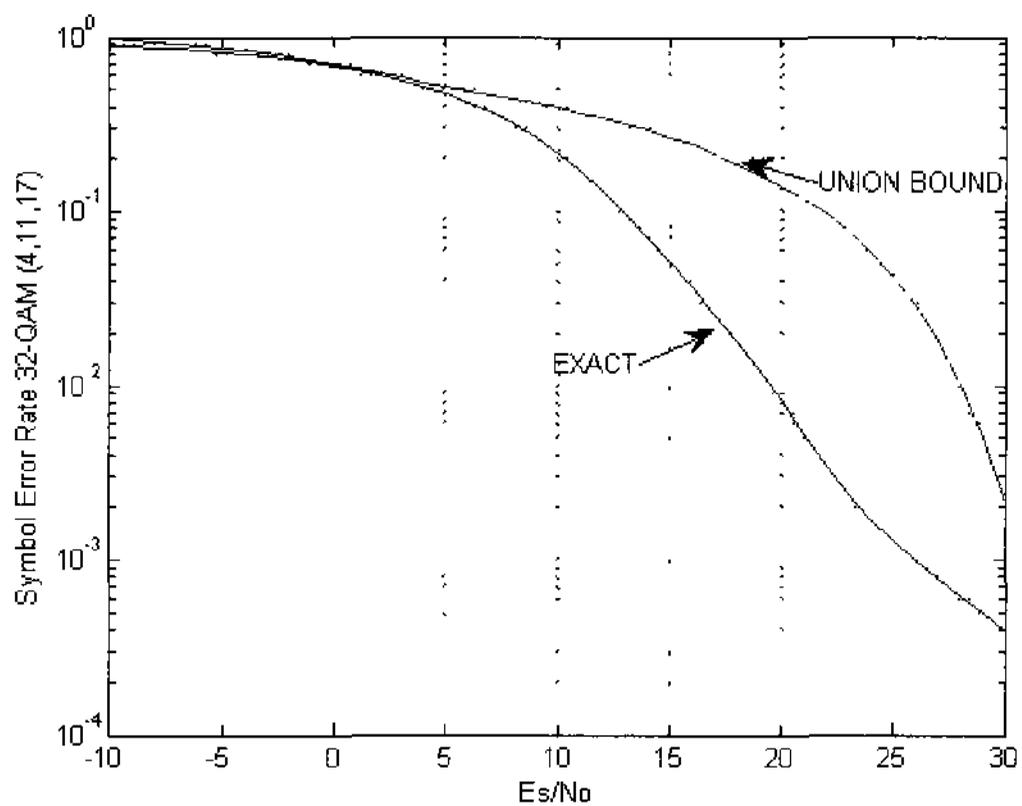


FIG. 19: Symbol Error Probability For the Non-uniform Signal Constellation Shown in Fig. 18.

CHAPTER IV

VIDEO OBJECT TRACKING

A video sequence can be considered as a series of still images taken over time, as illustrated in Fig. 20. The number of still images per unit of time of the video is called its frame (or picture) rate, typically ranging from six or eight frames per second (fps) to 120 or more fps. A commonly accepted minimum frame rate to achieve the illusion of motion without jitter is about 15 fps. In practice video sequences are stored and transmitted in compressed format using the current international video compression standards such as MPEG-1/2/4, H.261/3/4, etc. Computational complexity and speed limiting factors in the real life implementations of these standards vary based on the technology used in different blocks of the encoder/decoder, such as motion estimation algorithms, transforms employed (DCT, DWT, etc.), format of the bit stream, Group of Picture (GOP) format, etc. Typically, motion estimation consumes 60% of encoding time, whereas motion compensation consumes 11% [11]. The bit rate of the encoded video bit stream is closely related to the frame size and the rate as shown by an example for MPEG-1 bit stream below.

Maximum number of pixels/line: 720
 Maximum number of lines/picture: 576
 Maximum number of pictures/sec: 30

$$\begin{aligned} \text{MPEG - 1 Maximum Bit rate} &= 720 \frac{\text{pixels}}{\text{line}} \times 576 \frac{\text{lines}}{\text{picture}} \times 30 \frac{\text{pictures}}{\text{sec}} \\ &= 1.86 \text{ Mbps} \end{aligned}$$

As will be discussed in detail in Chapter IV shortly, the essence of the method proposed in this thesis lies in utilization of VO motion trajectory to convey the hidden message to the receiver side. In order for a reader to better understand the proposed method, some of the fundamental concepts pertaining to VO representation and tracking will be briefly discussed next.

IV.1 OBJECT BASED REPRESENTATION

In an object based video coding framework that employs object based representations, such as MPEG-4, each scene is described as a composition of VOs having homogeneous regions with respect to a criterion such as shape, motion or texture.

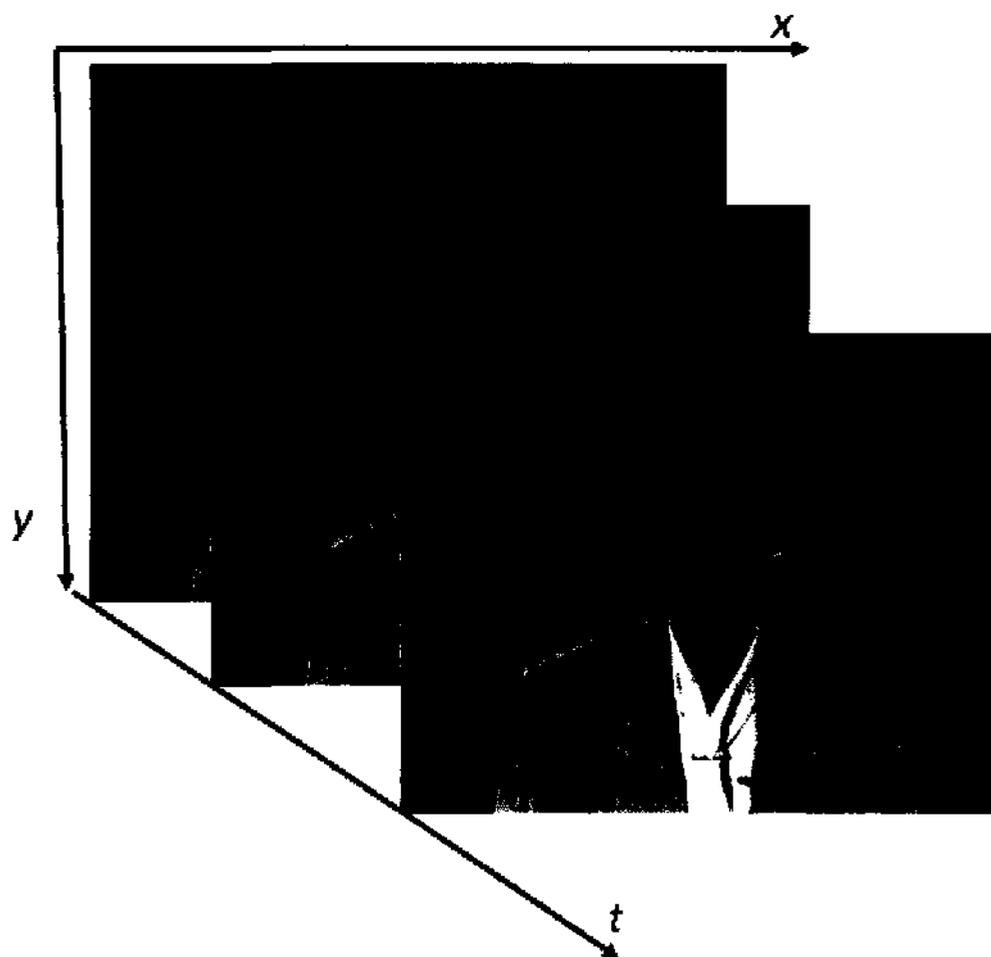


FIG. 20: Video Sequence Representation.

This object based representation is a key property for a variety of multimedia applications, allowing a user to access and manipulate objects within the video sequence. Each frame of a video sequence is segmented into a number of arbitrarily shaped regions called Video Object Planes (VOPs), and the shape, motion and texture information of the VOPs belonging to the same VO are coded into a separate Video Object Layer (VOL). Its primary goal is to support coding of pre-segmented video sequences and to allow separate reconstruction and manipulation of VOs at the decoder side. Successive VOPs are clustered in a group to form Group of VOPs (GOV). The first VOP of GOV is intraframe coded whereas each subsequent VOP in the GOV is interframe coded (P-VOP or B-VOP) using prediction. An illustrative example of VOs in a video scene is depicted in Fig. 21 which can be extracted by using an existing segmentation method (Also note that MPEG-4 standard does not specify any segmentation method in the standard). In the figure, three extracted VOs are illustrated by using their shape contour in red. At the coding stage, the object's shape, texture and motion information is coded separately as representation of the VO.

Extracting VOs from digital video is still a complex undertaking due to the ill-posed nature of the object segmentation problem. The VO segmentation algorithms can be roughly classified into two groups: spatial-based and temporal-based. In spatial based methods, also known as intra-frame methods, each frame is partitioned into homogeneous regions with respect to color, intensity or texture. Temporal-based methods utilize motion information to extract video objects. For examples of different state-of-the art segmentation algorithms the reader should refer to [62, 63, 64, 65].

In terms of user intervention, VO segmentation methods can be categorized as either supervised, requiring user involvement in defining the object in the first frame, or as automatic which does not require any user intervention.

The segmentation of VO is the pre-requisite for object based tracking algorithms. These two tasks interact with each other in a way that segmentation in the consecutive frames is done using tracking information from the previous frame and the tracking information is updated based on the refined segmentation results in the current frame.

Having discussed object based representation briefly, the next section discusses object tracking methods from a general perspective, by looking into common methods and their definitions and applications, and provides transition into more specific



FIG. 21: A Frame from “News” Sequence VO Segmentation Results.

application: tracking in sports videos. This essential transition is required to tie in object based representation, tracking with the proposed data hiding method.

IV.2 VO TRACKING

Object tracking is a common task in the fields of computer vision and multimedia technologies. It is used for diverse applications such as human-computer interaction, surveillance, smart rooms, content based indexing, automatic sports analysis, summarization, retrieval etc.

Let a VO trajectory T_j be represented as $T_j = \{(x_j^i, y_j^i), i = 1, 2, \dots, N_j\}$ where (x_j^i, y_j^i) is the centroid coordinate of the j th object, N_j is the number of trajectory points and $j = 1, 2, \dots, J$ is the number of trajectories. Tracking can generally be defined as the problem of estimating the trajectory of an object in the image plane by both determining and establishing the correspondence between object positions over time. The correspondence problem and resolved trajectories of three objects are illustrated in Fig. 22.

Tracking objects can be complex due to occlusions in the scene, noise, complex object motion, deformation of objects, scene illumination changes, etc. Numerous approaches for object tracking have been proposed to overcome these problems. In

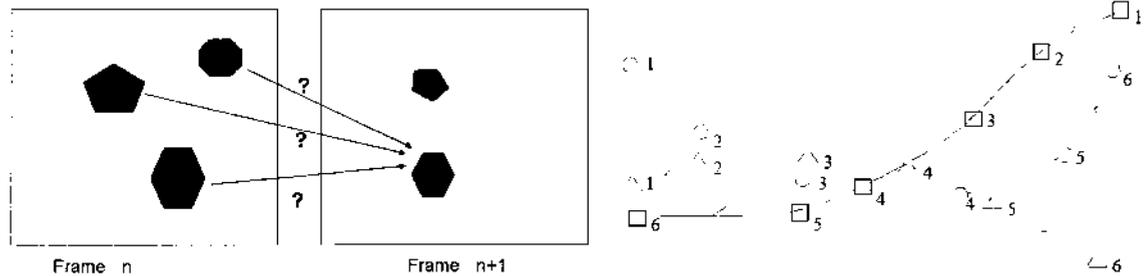


FIG. 22: Correspondence Problem and Trajectory Estimation.

that respect, the suitability of a particular tracking algorithm may depend on various constraints such as the type of application in which the track information is used, object appearances, object shapes, number of objects, object and camera motions, and illumination conditions

The first step in any tracking algorithm is the suitable representation of the object. The objects can be represented by their shapes and appearances. Commonly used object shape representations are illustrated in Fig. 23.

In the appearance based representation, the object can be represented by templates, parametric (Gaussian or mixture of Gaussian) or non-parametric representations, as well as active and multi-view appearance models.

The second step in the process is the selection of unique features that distinguish the objects from each other in the feature space. The commonly used features are color, texture, motion, edge-map etc.

The third step is the detection of the object, represented using the shape or appearance models, in every frame after the object first appears in the video. A common approach for object detection is to use information in a single frame (background subtraction, segmentation etc.). Other object detection methods employ motion and temporal information in the form of frame differencing computed from a sequence of frames to decrease the number of false detections.

And finally, given the object regions in each frame, the tracking algorithm establishes object correspondence from one frame to the next to generate the tracks. The commonly used tracking methods will be discussed briefly next.

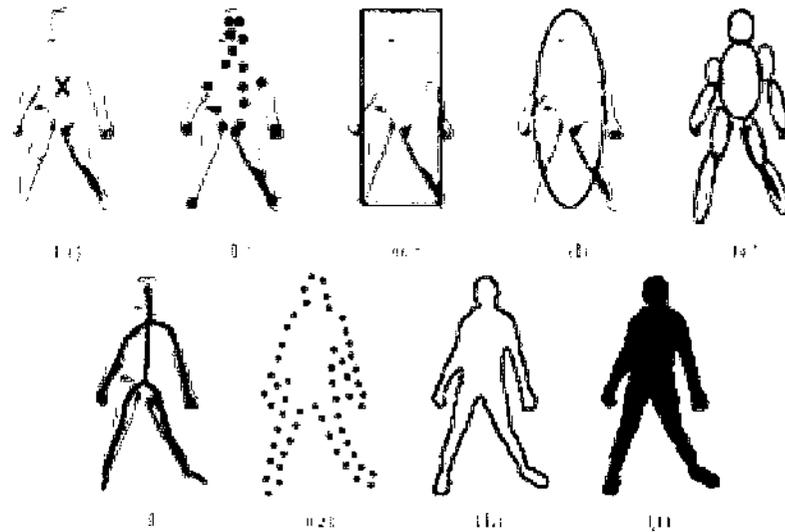


FIG. 23: Object Representation After [4].

IV.2.1 Image Plane Kalman Filter Tracking

A common approach for object tracking is borrowed from estimation theory which states that given all the measurements up to the current frame, the object's position can be estimated using predictive filtering and statistics of the object's color and position. When the measurement noise is assumed to be Gaussian, the Kalman filter is seen to be the optimal solution. In a typical Kalman filter, the state transition and measurement equations are simply defined as

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{w}(k) \quad (53)$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (54)$$

where \mathbf{w} and \mathbf{v} are the process and measurement noise respectively. The process noise term is a Gaussian random variable with zero mean, covariance Q and PDF $p(w) \sim N(0, Q)$ assumed to be independent of state $x(k)$. Similar to the process noise, the measurement noise is also assumed to be Gaussian with $p(v) \sim N(0, R)$ with covariance R , \mathbf{A} is the state transition matrix and \mathbf{H} is the measurement matrix defined by

$$A = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \quad (55)$$

$$H = \begin{bmatrix} \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \quad (56)$$

where ΔT is equal to the inverse of frame rate, \mathbf{I}_2 and \mathbf{O}_2 represent 2×2 identity and zero matrices respectively. The state \mathbf{x} and measurement \mathbf{z} are defined as:

$$\mathbf{x} = \begin{bmatrix} r_0 & c_0 & \dot{r}_0 & \dot{c}_0 & \Delta r_1 & \Delta c_1 & \Delta r_2 & \Delta c_2 \end{bmatrix}^T \quad (57)$$

$$\mathbf{z} = \begin{bmatrix} r_0 & c_0 & r_1 & c_1 & r_2 & c_2 \end{bmatrix}^T \quad (58)$$

where (r_0, c_0) is the centroid of the bounding box, (\dot{r}_0, \dot{c}_0) is the velocity, (r_1, c_1) and (r_2, c_2) are the top-left and bottom-right coordinates of the bounding-box, and finally $(\Delta r_1, \Delta c_1)$ and $(\Delta r_2, \Delta c_2)$ are the relative positions of the two opposite corners of the bounding-box with respect to each other.

The Kalman filter has two steps: prediction and correction. In the prediction step, the current state is projected to obtain an estimate $x^-(k)$. In the correction step, actual measurement is incorporated into the estimate as feedback to obtain an improved estimate $x^+(k)$ by using the Kalman gain.

One drawback of the Kalman filter is the Gaussian assumption for state variables. The filter performance is degraded in cases where Gaussian distribution assumption is no longer valid. Kalman filter is commonly used for tracking vehicles, tracking balls in sports videos, etc.

IV.2.2 Mean Shift (MS)

The MS algorithm [5, 66, 67] is an iterative, kernel-based, deterministic procedure which converges to a local maximum of the measurement function under certain assumptions on the kernel behaviors. It tries to find the image window which is most similar to the object color histogram in the current frame by iteratively carrying out a kernel-based search. The main idea of the mean-shift tracker is the computation of

an offset from the position \hat{y}_0 to a new position \hat{y}_1 according to the mean shift vector

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i K(\hat{y}_0 - x_i)}{\sum_{i=1}^{n_h} w_i K(\hat{y}_0 - x_i)} \quad (59)$$

where K is a suitable kernel function (Epanechnikov, Uniform, Gaussian etc.), w_i is the weight obtained by taking the square root of log-likelihood of the location (or pixel) x_i inside the kernel, and n_h is the number of pixels inside the kernel. Given the target distribution \hat{q} of the target model, to estimate the location \hat{y}_1 of the target in the current frame, the mean-shift tracker measures the distribution around the object's previous location \hat{y}_0 and evaluates the similarity between two distributions using the Bhattacharyya coefficient as

$$\rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (60)$$

where $\hat{p}(y)$ is the distribution of the candidate object. The algorithm then iteratively computes similarity between two distributions by shifting the location \hat{y}_1 until a convergence is met according to a threshold.

MS tracking is employed in some real time tracking applications such as face detection due to its simplicity and speed. However, the tracking success of MS mostly depends upon the discriminating power of the object's color histograms. Although MS is a low complexity algorithm that provides a general and reliable solution independently from the features representing the target, it fails in tracking small and fast-moving targets and in recovering a track after a total occlusion [5]. This problem is illustrated with some examples frames from MS tracking of "Ball VO" in Fig. 24. Since the ball velocity is so much between the frames, resulting in displacements larger than the kernel size, no part of the ball falls under the kernel at its previous position violating the major underlying assumption of the MS tracker and the effectiveness of the MS tracker is decreased. Another problem with this sequence that diminishes the performance is the fact that the ball is white while the homogeneous background is also cluttered with white pixels which can be considered as noise.

IV.2.3 Particle Filter (PF)

PF [68, 69, 70, 71] is a parametric method solving non-linear and non-Gaussian state estimation problems in the Bayesian framework and can deal with multi-modal PDFs. Let $x_{0:k} = \{x_i, i = 0, 1, \dots, k\}$ and $z_{1:k} = \{z_j, j = 1, \dots, k\}$ represent the state

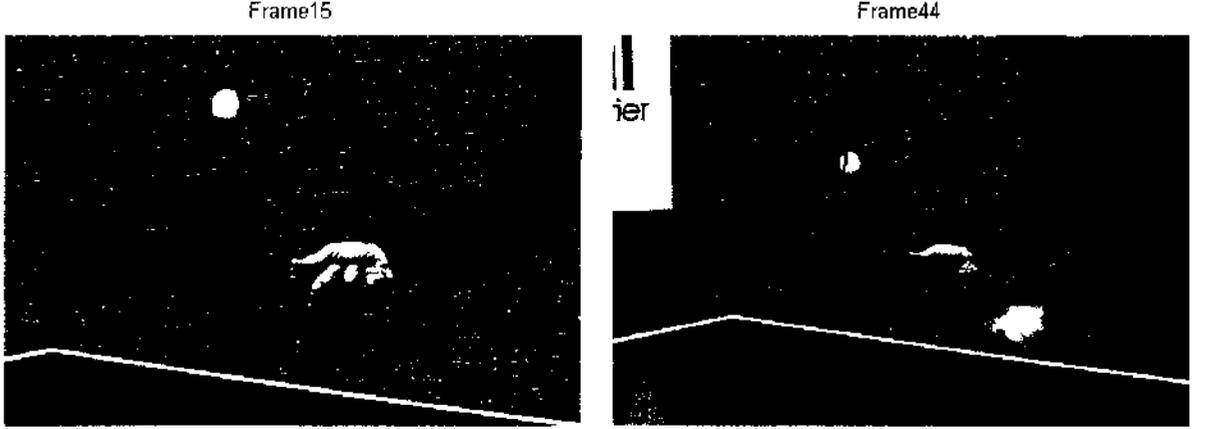


FIG. 24: MS Tracking of Ball VO. Left: Frame 15 and Right: Frame 44. Underlying problem of MS for fast moving objects and a homogeneous background which has lots of the same pixels (noise) as that of VO.

and observations up to time k . The idea behind the particle filter is to estimate the object state x_k from the observations by $p(x_k|z_{1:k})$ by using

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (61)$$

with a weighted particle set described as $\mathbf{x} = \{x_k^i, w_k^i\}_{i=1}^{N_s}$ where N_s is the number of particles and the weights w_i , $\sum_{i=1}^{N_s} w_i = 1$. The evolution of the particle set is achieved by propagating according to a linear model (a constant velocity model) such as $x_k = Ax_{k-1} + v_{k-1}$ in which A is the deterministic system model and v_{k-1} is a random vector drawn from the noise distribution of the system. The major limitation of the PF is its requirement of a large number of samples to be drawn from state space to describe the underlying probability density function efficiently.

PF has applications in video surveillance, human face tracking etc. Its feature that allows it to recover from lost tracks makes PF one of the most popular methods. PF is more complex than MS and heavily dependent on its parameter settings, which in turn, depend on the scene content. If the parameters are set optimally, it can track fast small objects. However, the number of particles needed to model the variations of the underlying PDF increases exponentially with the dimensionality of the state space, thus dramatically increasing the computational load [5]. Example frames from PF tracking of the Table Tennis sequence are given in Fig. 25.

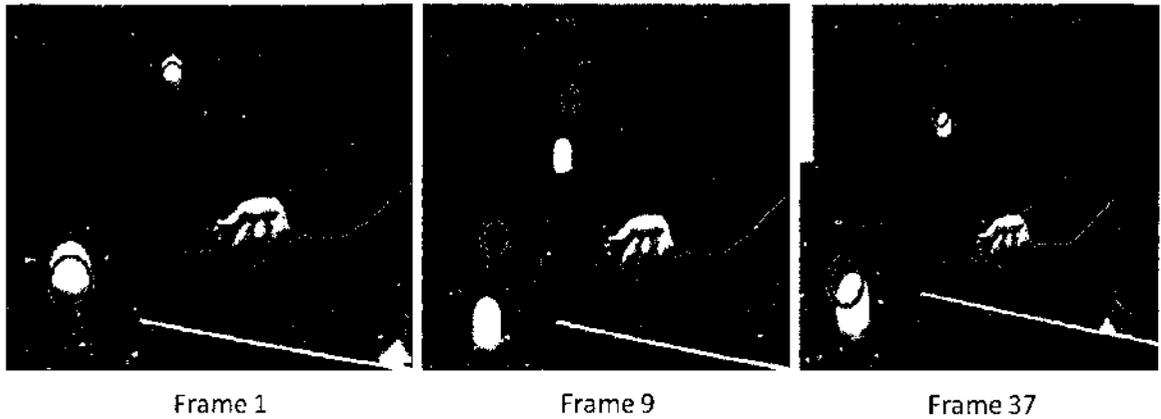


FIG. 25: PF Tracking of Ball VO. Left: Frame 1 , Middle: Frame 9 and Right: Frame 37 [5].

IV.2.4 Feature Based Methods

Feature based tracking methods employ features such as shape, color distributions, and shape and color together, to differentiate between objects via trained classifiers. One example is the commonly used method in which the difference image obtained by subtracting the background image from the current frame leaves possible target objects which are then input into the trained classifier.

IV.3 TRACKING VIDEO OBJECTS IN SPORTS VIDEOS

Automatic analysis, content based indexing, and the summarization and retrieval of sports videos have become popular during the past decade. An important component of any video summarization and indexing system is an event-detection mechanism which is triggered by an object's actions in the scene. Locating and tracking players in each video frame plays a crucial role in automatic comprehension of sports videos. For instance, the location of a ball provides information on the ball possession rates, segmentation of soccer video into play and break sequences, and detection of semantic events (goal, kick, etc.).

Within this context, research is focused on developing methods for ball games such as soccer, football, basketball, baseball, etc. As an exemplary work, Chen et al. [72] proposed a physics based tracking method for broadcast basketball video

taking the physical characteristics of ball motion into consideration. Chu et al. [73] presented a trajectory based framework for automatic ball tracking and pitching evaluation in broadcast baseball videos. In ball position prediction and trajectory extraction, they analyzed the 2D distribution of ball candidates and exploited the characteristic that the ball trajectory is a near parabolic curve in the video frames.

Ball detection and tracking in broadcast soccer videos (BSV) was investigated in [74, 6, 75, 39]. Yu et al. [74] proposed a trajectory-based algorithm for ball detection and tracking for broadcast soccer videos. In their method the ball size is first estimated from salient objects (goalmouth and ellipse), and different sieves (shape, size, aspect ratio etc.) are used to detect ball candidates. The true trajectory is extracted from potential trajectories of the ball candidates by a verification procedure based on the Kalman filter.

In a similar vein, Ren et al. [6] presented a real-time method for detection and 3-D tracking of a ball in BSV captured by multiple fixed and calibrated cameras. Size, color, and speed are features that discriminate the ball from other moving objects. Temporal filtering of the ball-likelihood has also proved to be essential in robust ball detection and tracking. They model the ball trajectory as curve segments in consecutive virtual planes and use geometric reconstruction techniques to estimate the 3-D ball position from a single view. They also introduced high-level ball phase transition information to aid low-level tracking.

Although the players can be successfully detected and tracked, ball detection and tracking with high accuracy is still challenging for the following reasons:

- the size of the ball is relatively small, especially when the camera is in far view, when compared to players,
- the shape, size and color of the ball exhibits variations due to motion and movement of the camera. (see examples in Fig. 26),
- there are many false alarms: objects such as players' head that are similar to the ball (ball-like objects),
- there are possible occlusions: possessions by players and merging with lines or players in the same frame.

These reasons are the key factor for the fundamental difference between ball detection-and-tracking and general object detection-and-tracking methods in the

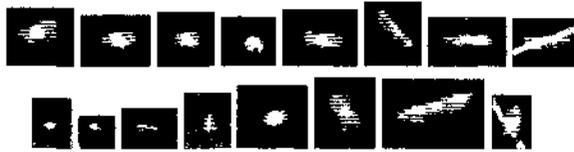


FIG. 26: Example of Ball Size, Shape and Color Variations in Soccer Video. (After [6]).

sense that there is no global ball representation applicable to each video frame to differentiate the ball from other false alarms. To increase the precision of ball tracking any method should take these constraints into account.

IV.4 BALL VO TRACKING

Different tracking methods are discussed in Chapter III. As can be concluded from the discussions in that chapter each method has its pros and cons such that there is no global tracking method applicable to any tracking problem. In other words, one method working for a particular problem such as human or vehicle tracking may not be suitable for another tracking problem, for instance ball or player tracking. Therefore, it is concluded that a simple VO tracking method satisfying the requirement of accurate tracking will be sufficient for this work.

IV.4.1 Ball Detection:

First of all, as discussed earlier, initial to every tracking algorithm is the identification of the object. Due to the deformation of the ball and camera conditions (zoom-in, zoom-out etc.) there is no universal ball model applicable to every frame. Therefore, the tracking algorithm developed in this work utilizes a set of sieves based on the ball properties such as size, color and shape to identify the ball in each frame.

Given a ball object O , let $AR(O)$ denote the aspect ratio ($\frac{height}{width}$) of the ball object O and $A(O)$ represent the area respectively. Following sieves, which are based on object properties, are defined to identify the ball after filtering out candidate objects, in which case the remaining object is considered to be the ball object.

- **Ball-Size Sieve:** Although the ball size changes due to the ball deformation and camera conditions (zoom in and out, tilt etc.), it should fall within a

specific range. If the object size is not in the range $R_{min} \leq A(O) \leq R_{max}$ then the object is discarded. To mitigate ball size variations, the limits for the ball size are defined as $R_{min} = A(i, j)(1 - \lambda)$ and $R_{max} = A(i, j)(1 + \lambda)$ where λ is empirically determined constant set to $\lambda = 0.5$ and $A(i, j)$ represents the ball size whose bounding box is located at the (i, j) coordinate.

- **Ball Color Sieve:** Ball color is a good discriminator for ball like objects. Define $O(i, j) \geq \beta$ denoting pixel wise thresholding of an object to remove the non-ball objects.
- **Shape Sieve:** Remove the objects whose aspect ratios are not in the range $0.8 \leq AR(O) \leq 1.7$

As will be discussed broadly in Chapter V under Section V.1, both the encoder and decoder are synchronized with the data which include the starting and ending frame numbers as well as the bounding box coordinates of the object. These coordinate locations are determined manually by the user. This supervised approach solves the problem of initial ball recognition aiming at identifying the ball VO in the first frame.

The first step in applying the above sieves is the Ball Color Sieve where the threshold is $\beta = 220/255 = 0.8627$. Note that, for the gray scale frames, a pixel value of 255 corresponds to the *white* pixels whereas 0 indicates *black* pixels. Output of this thresholding is basically a binary image as shown in Fig. 27. In the binary image, the pixels that are set to 1 belong to the foreground whereas the ones set to 0 belong to the background. At the end of this stage a ball-like object is identified for further morphological operations and blob analysis.

To identify the ball object in the binary image, a 4 connected-component labeling search, which is available in Matlab, is done. After obtaining labeled regions, Matlab's *regionprops* function, which computes a set of properties such as Area, Major, Minor Axis Length, Bounding Box, Centroid etc., is used to compute Ball-Size and Shape Sieve.

Object O is identified as a ball if its properties match with the sieves defined above and the centroid coordinate c_x and c_y is stored in the object trajectory represented by $T(c_x, c_y) = [(c_{x1}, c_{y1}), (c_{x2}, c_{y2}), \dots, (c_{xi}, c_{yi})]$.

To identify the ball in the consecutive frames, a simple Region Of Interest (ROI) based search strategy, as proposed by Wong in [76] is employed. The rationale behind

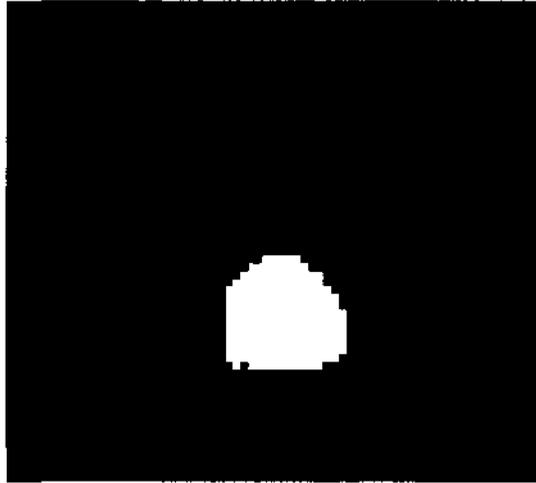


FIG. 27: Binary Image Obtained After Thresholding Pixel Values in the Bounding Box of the Object as $O(i, j) \geq 0.8637$. Connected Component Analysis is done on this Binary Image to Label the Pixel Belonging to the Same Object.

this approach is

- To decrease complexity which is inherent in even the mostly used methods such as Kalman Filter Tracking, which involves computations for prediction and correction stages. When a ball moves so quickly in a video scene, the physical behavior of the ball motion still demonstrates smooth trajectory segments in consecutive frames for ball's both freely moving on the ground and flying in the air cases. Therefore, based on this valid assumption, a simple linear motion with a region of interest in which the object highly likely exists is used for identifying the ball in consecutive frames.
- To help eliminate making assumptions on the noise and other motion state model parameters which can seriously effect the performance of the tracking algorithm.
- To facilitate the demonstration of the data hiding algorithm.

The goal of this work is neither to develop new tracking algorithms nor to implement tracking algorithms which have intelligence in identification of semantic meanings such as ball possession, out of field, ball rolling, or flying but to demonstrate

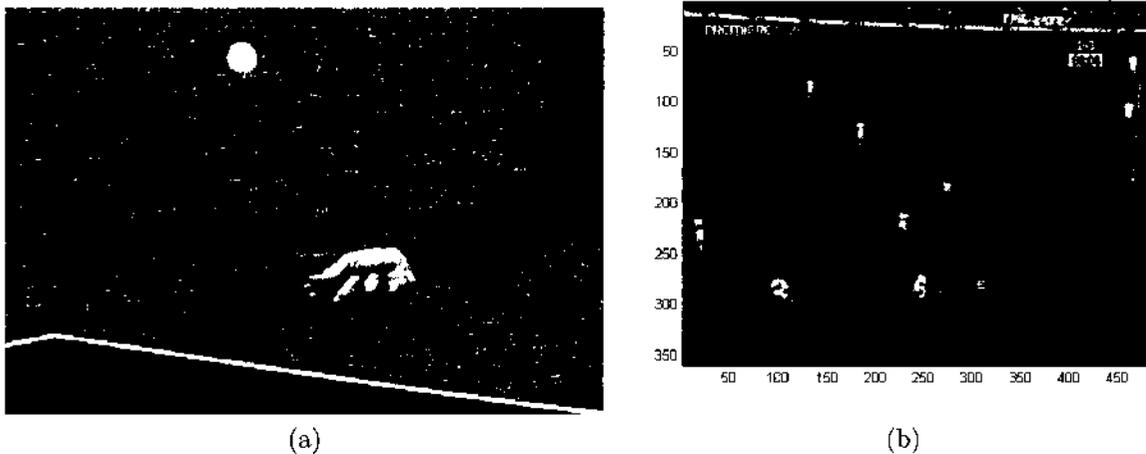


FIG. 28: Example Frames From “Table Tennis” and “Soccer” Sequences.

the idea that in smooth trajectory segments of the ball VO, a simple linear smooth-motion assumption is considered to be sufficient.

To estimate the ball location in the current frame, a search region (a window) of $N \times M = 30 \times 30$ pixels demonstrating the maximum expected displacement of the ball between the frames is defined. This window of search region is centered at the centroid location of the VO in the previous frame. Once again this is a valid assumption as the motion of the ball between consecutive frames is considered to be smooth.

After the search region is identified, the morphological operations and sieves discussed before are used to detect the ball and the centroid coordinate in the current frame. This procedure continues until the end of the video sequence.

IV.4.2 Experimental Results

The tracking algorithm described in the previous section is used to find the trajectory of the “Ball Video Object (VO)” in “Table Tennis” and “Soccer” video sequences. An example frame from each of the sequences is illustrated in Fig.28. The Table Tennis sequence includes the Ball VO with complex motion e.g., flying freely in the air, occlusion, and fast motion change, whereas the soccer sequence includes the Ball VO rolling on the ground with smooth motion and partial occlusion due to the possession of the ball by the player.

Trajectory obtained for the Ball VO in the Table Tennis video sequences is shown in Fig.29.

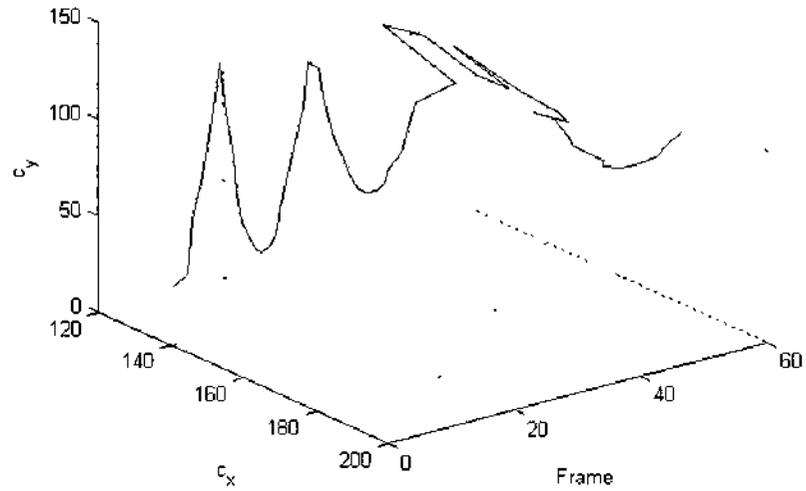
A common approach to measuring the performance of a tracking algorithm is the comparison of the results of the tracking algorithm with those of the ground truth coordinate locations. For this, the actual centroid x and y coordinates are manually obtained via frame by frame analysis of the Ball VO. Fig.30 illustrates the ground truth and the tracking algorithm computed centroid coordinates together. As can be seen from the figures, clearly the tracking algorithm gives very consistent results except for the frames in which the ball is occluded by the player's hand. In these frames the ball almost disappeared in the hand which color is similar to the ball color range.

IV.5 PERFORMANCE EVALUATION OF OBJECT TRACKING

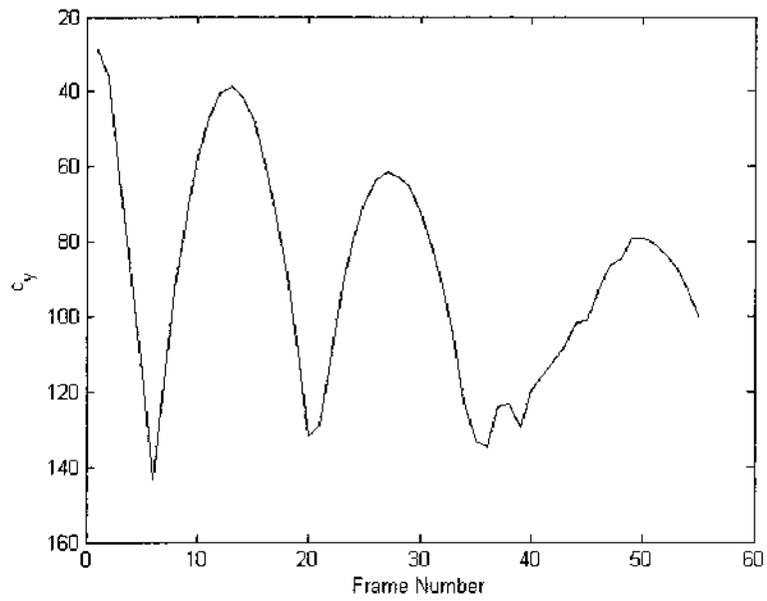
In order to evaluate the performance of different tracking algorithms, a simple and fast distance-based metric is employed. First, the ground truth object position, which is manually obtained, is represented in terms of the object's centroid position. Then for each ground truth track and algorithm generated track, the centroid distance, which is the Euclidean distance between their centroids, is computed i.e., $dist(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$. The difference is then considered to be the tracking error for which the essential rule is the smaller the distance the better the performance. Note that, in addition to Euclidean distance, examining the individual errors in each dimension of the VO motion may be used for performance evaluation as well.

To analyze the performance of the tracking algorithm under noisy channel conditions, Gaussian white noise with zero mean and varying variance is added to the video frames (AWGN Channel). Examples of noisy frames are illustrated in Fig.31. Through visual examination of the noisy frames, it can be concluded that when $\sigma_n^2 > 0.1$ the visual quality of the frames get worse and it becomes difficult to perceive the content of the frame. Hence, it is valid to state that the threshold for noise variance beyond which it is obsolete to track is $\sigma_n^2 = 0.1$.

The results of the error between tracked and the ground truth trajectory both in Euclidean distance and errors in each individual axis are shown in Fig.32, Fig.33, Fig.34 and Fig.35. It can be clearly observed from the results that for up to $\sigma_n^2 = 0.05$ the tracking errors are approximately in the range of ± 2 pixels except for the fact that a peak stands out around 40th frame. The reason for this large error is due to



(a) Ball Motion in X-Y Dimension



(b) Ball Motion in Y Dimension

FIG. 29: Table Tennis Sequence "Ball VO" Trajectory.

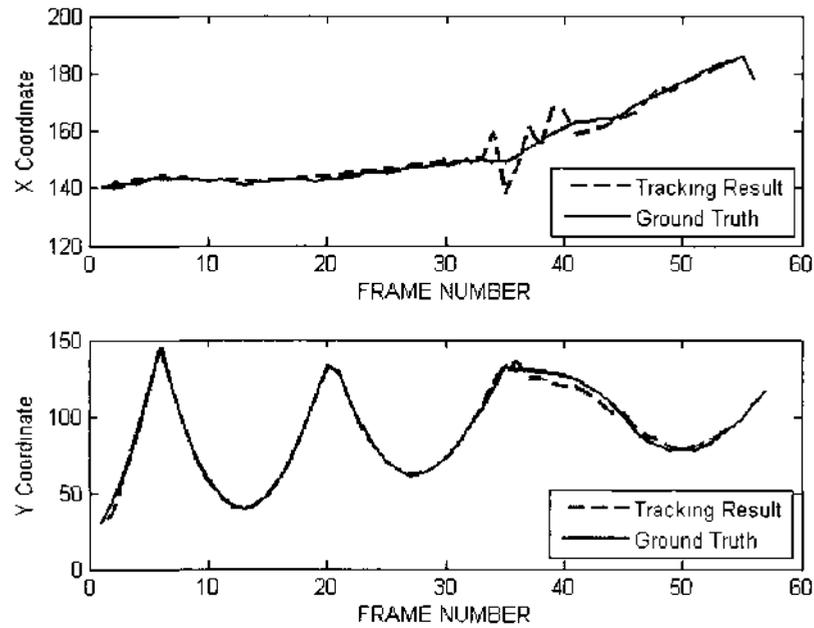


FIG. 30: Ground Truth and Computed Trajectory Centroid Coordinates. Top: X coordinate Bottom: Y Coordinate. The difference between the frames 33 and 42 is due to the occlusion of the ball by the player.

the tracking algorithm's not having any occlusion detection mechanism. Once the player holds the ball, the tracking algorithm presumably tracks the hand instead of the ball as it finds color and shape match in the hand of the player. This problem might be solved by incorporating advanced features to the algorithm by introducing occlusion detection, split, merge, ghost and reappear type mechanisms.

AWGN Sigma=0.001



AWGN Sigma=0.01



AWGN Sigma=0.05



AWGN Sigma=0.1



AWGN Sigma=0.2



AWGN Sigma=0.3



FIG. 31: Frames 1 to 6 from Table Tennis. From top to bottom: noisy frames with $\sigma_n^2 = 0.001$ to 0.3.

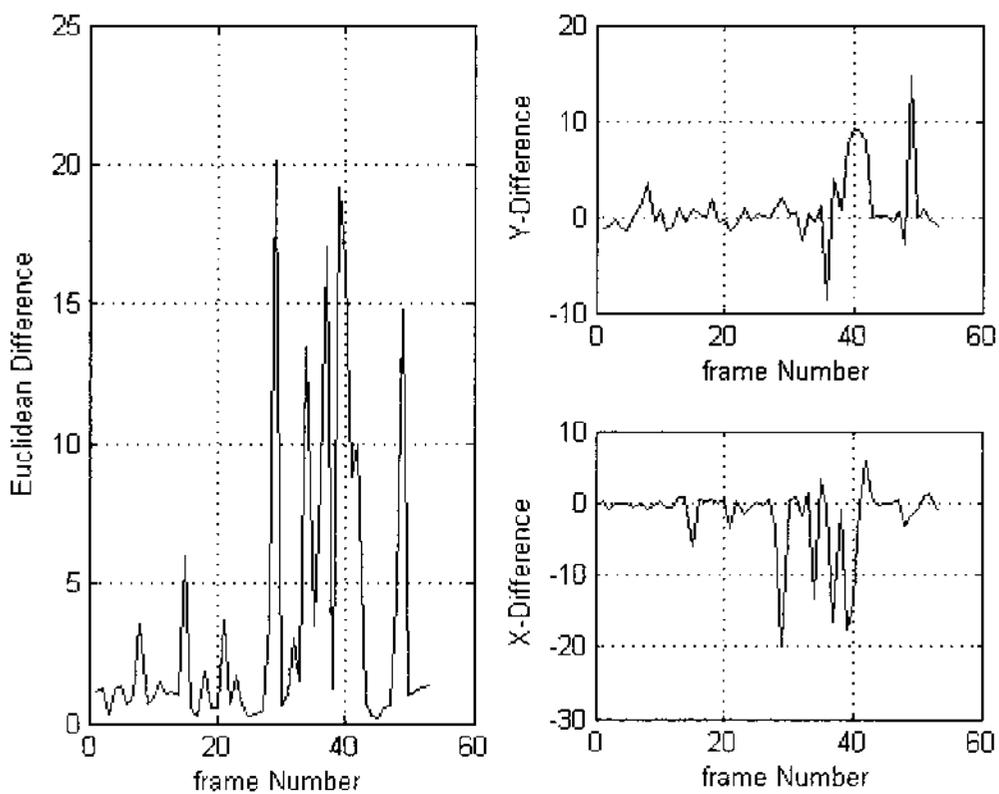


FIG. 32: Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.1$. Top right: Difference in Y coordinate of the ground truth and the tracking output. Bottom Left: Difference in X coordinate of the ground truth and the tracking output.

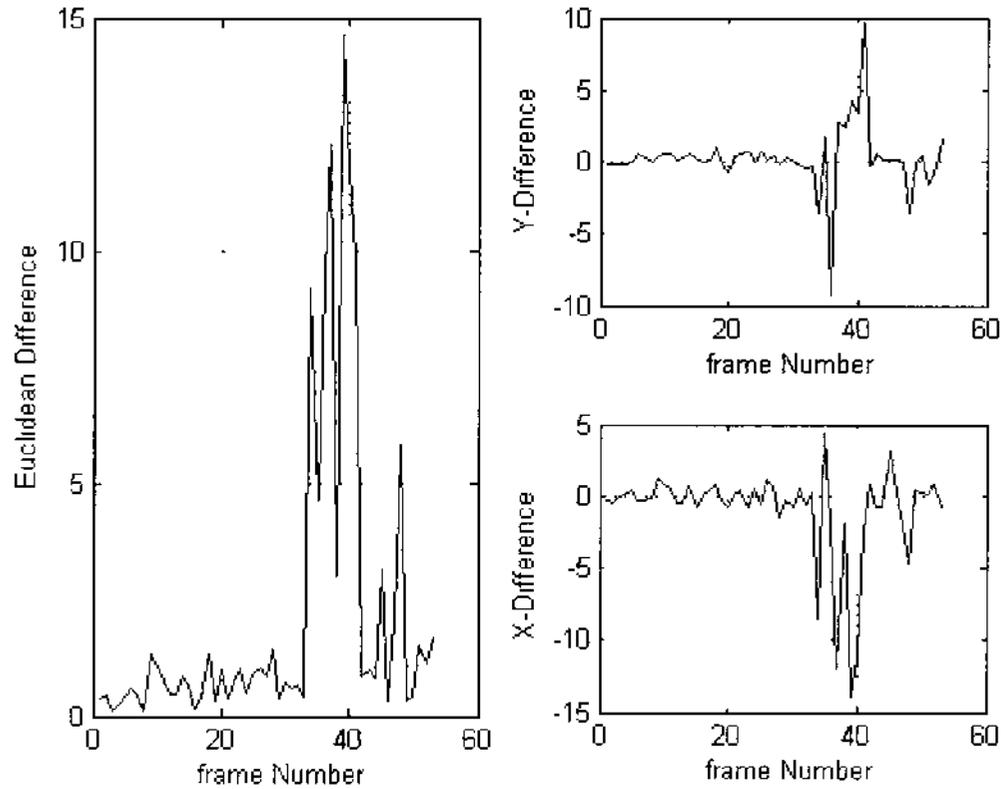


FIG. 33: Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.05$. Top right: Difference in Y coordinate of the ground truth and the tracking output. Bottom Left: Difference in X coordinate of the ground truth and the tracking output.

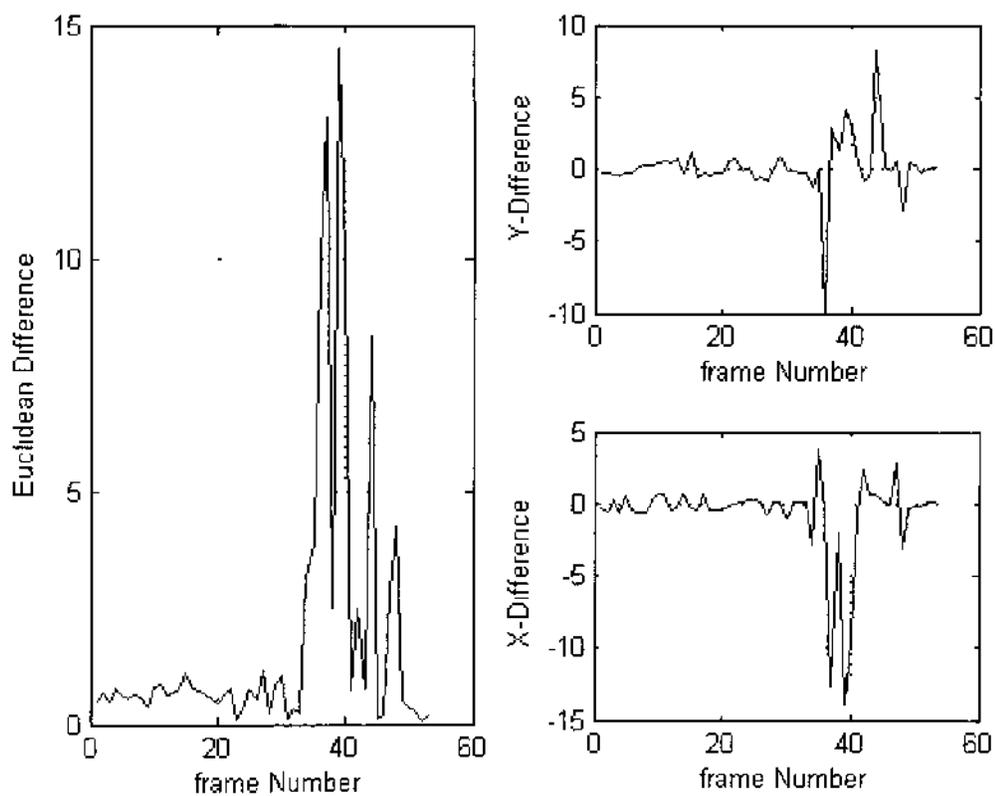


FIG. 34: Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.01$. Top right: Difference in Y Coordinate of the Ground Truth and the Tracking Output. Bottom Left. Difference in X Coordinate of the Ground Truth and the Tracking Output

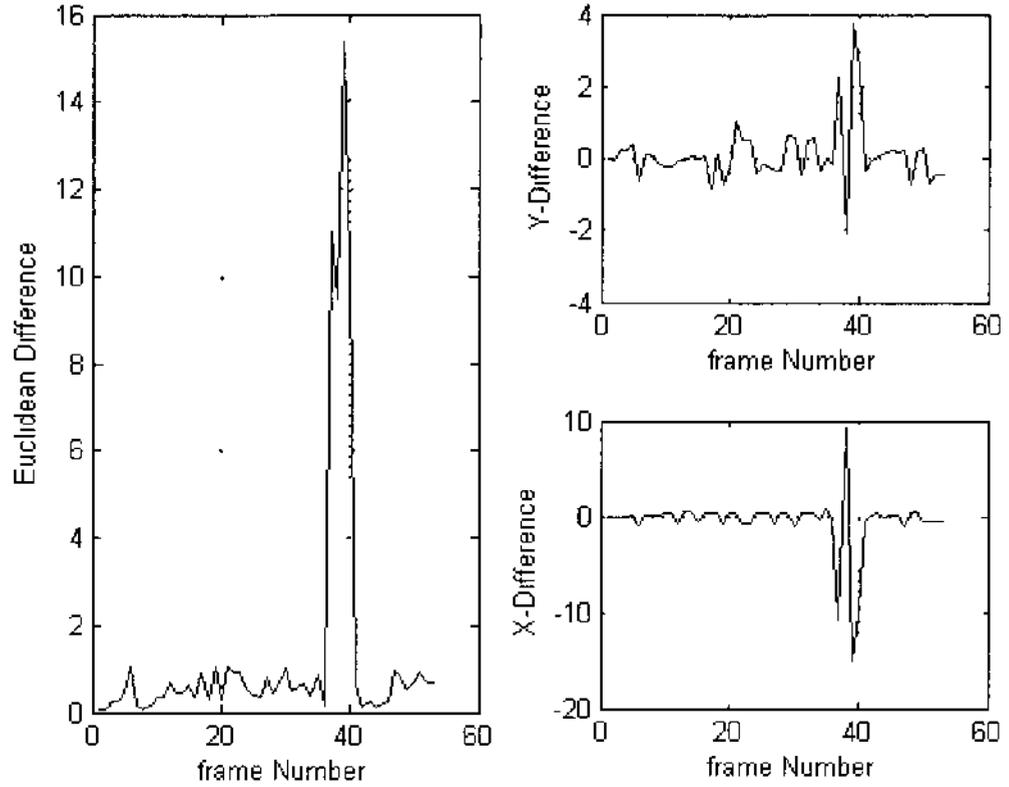


FIG 35: Left: Euclidean Distance between the Ground Truth and Tracking Result when $\sigma_n^2 = 0.001$. Top right: Difference in Y Coordinate of the Ground Truth and the Tracking Output. Bottom Left: Difference in X Coordinate of the Ground Truth and the Tracking Output.

IV.6 RESEARCH FOUNDATIONS

Having discussed the commonly used VO tracking methods, and tracking in sports videos with its inherent problems, the key points which the idea of trajectory based data hiding method have been derived from is presented next.

Despite the fact that tracking the ball in any ball game video is a challenging task due to several constraints, it is arguable on the contrary that the ball trajectory can be a feature to embed data in an invisible way as follows.

As discussed earlier, the ball is the focus of attention in many sports videos in which the ball size and shape varies due to its motion, as shown in Fig. 26. Its size, its single homogeneous region with a common motion-compared to a player whose body parts might have different motion directions-and a semantic meaning make ball trajectories attractive for manipulations. The changes made to the trajectories will result in video sequences with almost no visible and statistical artifacts when both global image quality and statistical measures such as PSNR, MSE, and histogram based metrics are used. This is a valid argument due to the fact that in a ball trajectory perturbation based method, modifications will affect only a small portion of the whole frame as opposed to motion or block based data hiding methods that may alter the features in larger regions leaving comparatively significant signatures after data embedding. Any modifications that preserve the semantic meaning of the ball, or even a player, trajectory in the stego scene will be imperceivable for a viewer.

The data hiding mechanism discussed in general above can be extended further to include multiple players, in addition to the ball, to increase the embedding capacity. In general, every information hiding algorithm uses a secret key to encrypt the message as illustrated in the generic data hiding framework in Chapter II Fig. 2. The secrecy provided by the encryption mechanism can be strengthened further by embedding data in random trajectory segments of the ball or a player VO in the video sequence.

This idea is illustrated in Fig. 36. As shown in the figure at the information embedding side, the trajectory based embedding mechanism utilizes trajectory segments that can be selected in either random or deterministic fashion. This information must be known by the receiver as ancillary information, which can be also embedded in the video frames using conventional techniques such as YASS, Matrix embedding or any DCT based data hiding method. Extending trajectory based embedding to multiple VO partial trajectories is depicted in Fig. 37. In the figure, line segments represent

the occurrence of each individual VO in the video sequence, whereas the red intervals represent partial trajectory segments that are utilized for embedding secret data.

In this scenario, frame synchronization between the encoder and the decoder is very crucial since the embedded data can only be extracted only if the decoder identifies the trajectory segments that are used for information embedding. To facilitate correct recovery, a simple protocol between information embedder and decoder that includes the (START FRAME, END FRAME) of the partial trajectory, VO CENTROID COORDINATES, and BOUNDING BOX WIDTH and HEIGHT can be employed. These parameters can be exchanged between the sender and receiver secretly in a separate covert channel, or encrypted as the secret data and embedded in the first frame only, first few frames or in any frame chosen randomly in a repetitive manner to compensate for the channel associated errors or intentional attacks. Thus, the entire video will be partitioned into sequences of the form $\{C_0, C_1, C_2, \dots, C_n, S_0, S_1, S_2, S_3, \dots, S_n, C_{n+1}, C_{n+2}, \dots\}$ where C_i , S_i represent the cover frame and stego frame respectively. When the duration of the ball video sequences (i.e. 90 min for soccer) is considered, and the utilization of partial trajectory segments based on the selection strategy defined above, it will be very difficult for an observer to find all the trajectory segments where the data is embedded. In that respect, this scheme will serve two purposes: providing an additional layer of security in addition to trajectory based embedding itself and combating against frame reordering, frame dropping and frame addition.

The decoder receiving the video sequence will only be interested in analyzing stego frames that contains the hidden data. The decoder first checks the locations where the encrypted data are embedded and decides whether it is a valid video sequence for decoding or not. Any degradation to video sequence frames such as frame dropping will result in elimination of the video sequence and a request for resending the whole stego sequence again.

IV.7 CHAPTER SUMMARY

In summary, this chapter aims at providing basics of both VO representation by providing common taxonomy, examples of different object tracking methods and elaborating on the mechanics involved in tracking. The scope of the discussions are limited to three most commonly used tracking methods since the goal of this thesis is neither the development of a new tracking nor improvement of an existing

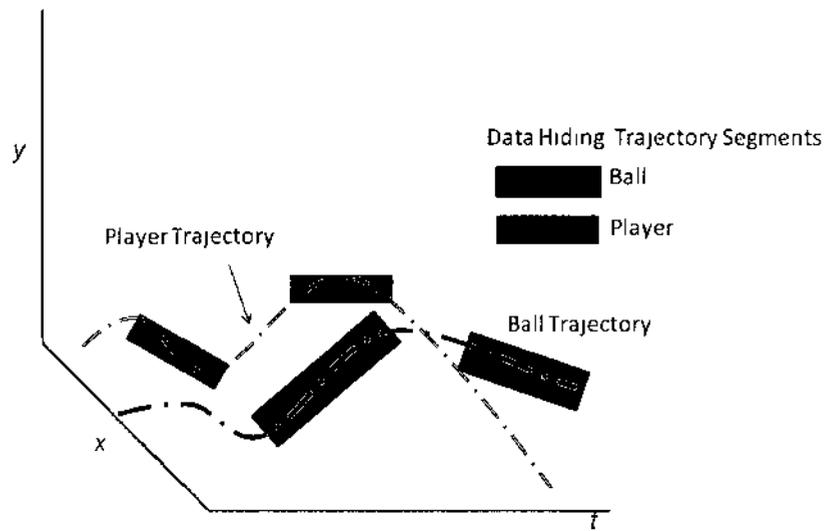


FIG. 36: An example of VO Partial Trajectory Embedding.

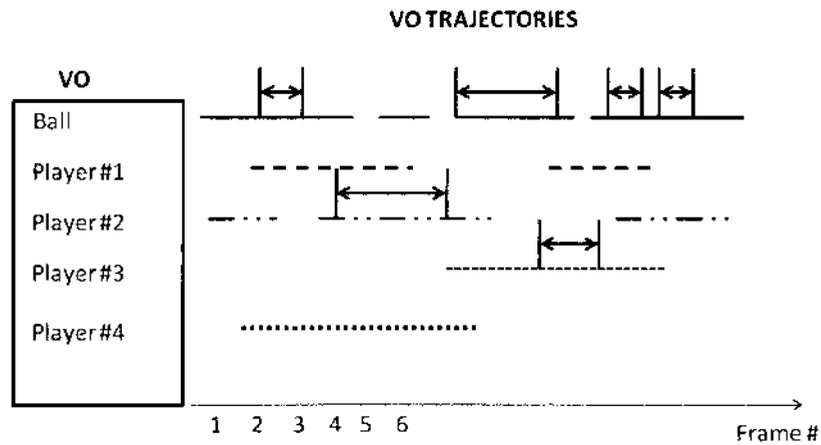


FIG. 37: An example of Multiple VO Partial Trajectory Embedding.

method. The idea behind investigation of different methods is the identification of a fit-for-purpose yet a simplistic tracking method for the application at hand. Different metrics allow us to test the performance of a particular algorithm against a specific challenge. More research is required to improve robustness of the tracking algorithm against common problems noise, illumination change, occlusion etc.

Furthermore, discussions are concentrated on common tracking methods developed for sport videos since the proposed method utilizes sport video sequences as the cover data. From the discussion it may be concluded that there is no global method that is applicable to any tracking problem for different sport videos. In general, the designers come up with ad-hoc methods to fulfill the requirements for a specific application problem e.g., tracking basketball, soccer ball, gold ball, baseball, etc. And finally, the discussion of the proposed method research foundations is summarized in Section IV.6.

It is claimed that the ball VO trajectory could be used for data hiding contrary to the fact that there are inherent problems associated with ball tracking. Then the fundamental idea of perturbing trajectory segments of single and multiple VOs together with the novel synchronization scheme is presented.

CHAPTER V

TRAJECTORY PERTURBATION DATA HIDING

Digital video presents many challenges that need to be carefully considered in the design stage of the data hiding method. These challenges may put limitations on the application specific requirements discussed in Chapter II under Section 2.

The reader may ask the natural questions of why video sequences are used for data hiding and what are the benefits and constraints as opposed to the other multimedia cover signals? The answers to these types of questions will be sought by presenting constraints and advantages of video domain data hiding. This approach will help the reader understand the relationship between the proposed method performance, application specific requirements and steganalysis.

First of all, digital video compression standards such as MPEG-1, MPEG-2 and MPEG-4 are considered to be a form of attack against information hiding techniques in the sense that the compression algorithm may damage or remove hidden data.

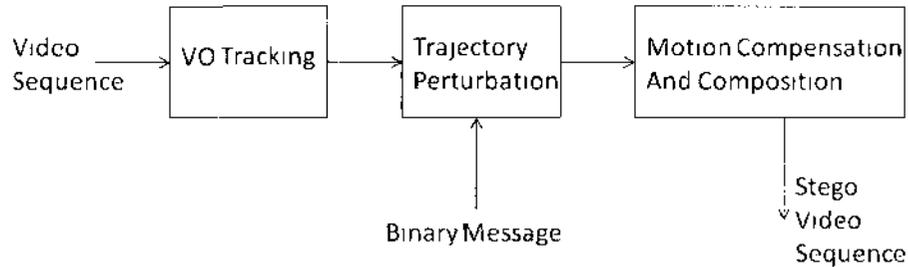
Successive video frames are highly correlated, and hence it is possible to align and average them to obtain a perceptually similar video stream. Data hiding methods employing methods originally developed for still images do not efficiently take into account the temporal dimension of the video. Therefore, the temporal dimension of the video should be considered in designing data hiding techniques to resist against collusion type attacks. Motion along the temporal axis and temporal sensitivity of the HVS may provide potential improvements in designing efficient algorithms.

The computational cost and coupling with a specific compression standard limiting the portability of the embedding mechanism could be other constraints for real time applications.

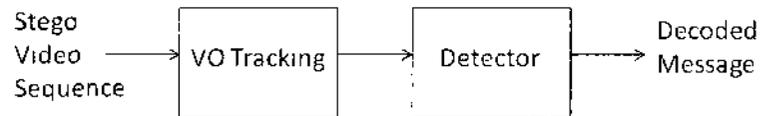
Having discussed the video domain constraints and advantages above, a novel data hiding steganographic algorithm using VO trajectories to embed data in a statistically and perceptually invisible way is presented next. It is assumed that the secret data is encrypted by using a well-known cyrptosystem (e.g. AES) before the embedding stage, which is the case for most secret communication systems. Hence, even though the adversary monitoring the channel can somehow access the secret data, he will not be able to decrypt the actual plain text message without having the appropriate key.

The proposed method is an oblivious, quantization based data hiding method

in spatio-temporal domain. The block diagram of the proposed method is given in Fig. 38. The idea behind the method is to perturb the motion trajectory of a video object (VO) in a fashion that preserves the smoothness of the trajectory while modifying the direction and the magnitude at each feature point.



(a) Embedding Mechanism



(a) Extraction Mechanism

FIG. 38: Block Diagram of the Proposed Method.

The first step of the data embedding stage is the manual selection of the VO bounding box, as well as the start and the end frame numbers of the partial trajectory. This side information can be considered as a synchronization signal for the decoder as it will not otherwise know which frame and VO are used for perturbation to embed data. From the security perspective, this selection will provide another layer of security against active or passive wardens who are monitoring the channel. Without having the exact synchronization signal it will be impossible for them to both detect and attack the partial trajectories that are used for sending information to the decoder side. The parameters for synchronization can be selected by using a user interface which will allow frame by frame analysis with features such as fast forward, skip, pause etc. This type of user interface will allow a user to interactively select synchronization side information parameters in a considerably shorter time.

The tracking module then takes the video sequence and the synchronization information as input to track the VOs. The VOs which are selected for trajectory perturbation are tracked only in the interval defined by the synchronization signal.

The result of the tracking module is the centroid coordinates of the tracked VOs.

As stated earlier, the secret message is in the binary form obtained from the output of a cryptographic system. This binary message and the trajectory information are used as input to the trajectory perturbation module to obtain the new centroid coordinates of the VOs. In essence, the secret message is conveyed to the receiver through perturbations (modulation of trajectory data within the binary message) introduced into the trajectory centroid coordinates of the VOs. In the trajectory perturbation phase, the direction and the magnitude of the motion are quantized using two different set of quantizers. Finally the new centroid coordinate is used to motion compensate the VO.

At the decoder side, the decoder first extracts the synchronization signal so that it can determine the bounding box, start, stop frames from which the partial trajectory segment is obtained. After decoding of the synchronization signal the tracking module takes the video sequence to generate the partial tracks of the VOs by returning the centroid coordinates of the bounding box of the VOs. The message is then decoded by using the same set of quantizers for which the range and step sizes can be agreed between the sender and the receiver beforehand. Also note that the decoder quantizer parameters may be obtained through analysis of an ensemble of video sequences in which the object demonstrate common trajectory types (c.g., ball rolling on the ground with different speeds and directions etc.) by using a classifier/future extractor.

Until now the data hiding mechanism and overall functionality of each module are explained in general. Each module in the proposed method will be discussed with examples in detail next.

V.1 CONVEYING SYNCHRONIZATION INFORMATION TO THE DECODER

As discussed in Chapter IV Section IV.6, the synchronization information is required between the data embedder and the decoder to determine the video segments which are used for data hiding and which VOs in those partial video segments are used for trajectory perturbation. This side information has to be conveyed to the decoder either by using other conventional secret communication techniques or by embedding the data in the video sequence. Contrary to the general case which the availability of the side information at the decoder side is assumed, a method which can be used

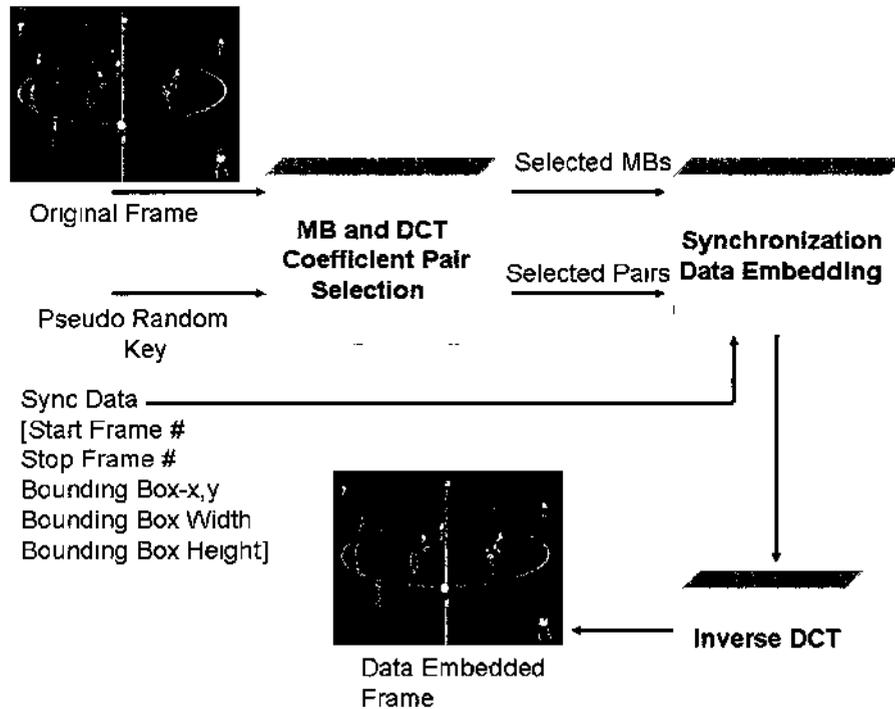


FIG. 39: Synchronization Data Embedding Mechanism.

to convey additional side information to the decoder is presented below.

The synchronization data embedding algorithm proposed in this thesis embeds one bit of data by imposing specific relationships in a way similar to the method proposed in [21]. Specifically, a relationship is imposed, if not naturally occurring, between some pairs of mid band DCT coefficients in selected MBs, both of which are selected pseudo-randomly. The overall scheme of the proposed watermarking system is shown in Fig. 39. The synchronization information consists of [Start Frame No., End Frame No., VO Bounding Box Top-Left x Coordinate, Bounding Box Top-Left y coordinate, Bounding Box Width, Bounding Box Height] resulting in 48 bits when eight bit representation is used for each parameter. The synchronization data embedding is based on a simple idea of embedding data in random locations within a frame, which makes it difficult for the steganalysist to get a good estimate of the cover frame features via the self-calibration process.

At the start of each video sequence a pseudo-random sequence is generated, based on a secret key, for randomly selecting those MBs in which the synchronization data

has to be embedded via the modification of the coefficient pairs. It is assumed that both the sender and the receiver share the same seed for the random number generator which determines the location of the MBs. Each MB contains 8x8 pixel blocks which are then DCT transformed to produce DCT blocks $F(u, v)$. The MB size can be set to the standard JPEG 8x8 grid size or BxB where $B > 8$. In case of $B > 8$ smaller 8x8 blocks can be selected pseudo randomly from the BxB blocks as done in [77].

Note that first frame is used to embed synchronization data. To provide redundancy as protection against attacks such as frame dropping, cropping of the regions in the first frame where the MBs are utilized for data embedding and random noise addition to the frames, the synchronization data can be added in multiple frames of the same sequence. This can be simply implemented by using the mechanism shown in Fig. 39 iteratively over the selected number of frames.

After random selection of MBs, the next step is the alteration of pairs of DCT coefficients based on energy difference. The DCT coefficients are zig-zag scanned to obtain a vector of coefficients as illustrated in Fig. 41. Note that the coefficient residing at index $i = 0$ corresponds to the DC coefficient, whereas coefficient index $i = 1$ to 5 corresponds to the low frequency components and 6-53 mid frequency and 54-63 are the high frequency components. The modifications made to low frequency components are more noticeable than those in the high frequency components. Hence, these coefficients need to be left untouched to preserve perceptual quality of the original video.

For video compression, every standard uses different quantization tables for different quality factors to quantize DCT coefficients. An example of a quantization table is depicted in Fig. 40. The quantized coefficients are obtained by dividing the DCT coefficients with the corresponding entries in the quantization table so that $\hat{F}(u, v) = F(u, v) / Q(u, v)$ where $Q(u, v)$ and $\hat{F}(u, v)$ represent the quantization step and quantized coefficient respectively. As can be easily seen from the quantization table in Fig.40, the high frequency components will mostly likely be rounded towards zero after quantization, since the quantization table entries, i.e the quantization step sizes, are big compared to the entries corresponding to the mid and low frequencies which have smaller quantization step sizes. To achieve a trade-off between the requirements of invisibility and robustness against compression, the rule of thumb is to use only the DCT coefficients $F(u, v)$ belonging to a mid-frequency range for

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

FIG. 40: JPEG Quantization Table (Luminance Quantization).

embedding synchronization information. These coefficients are shown as shaded in Fig. 41.

The embedding algorithm hides one bit of the synchronization data in a pair of DCT coefficients belonging to the mid-frequency region of a set of selected MBs in a fashion as follows.

Let a consecutive pair of coefficients be denoted by $F(u_i, v_j)$ and $F(u_{i+1}, v_{j+1})$ and the difference between a pair of DCT coefficients is denoted as

$$\Delta_F = F(u_i, v_j) - F(u_{i+1}, v_{j+1}), \text{ if } i, j = 6, 7, \dots, 21$$

where the index i, j correspond to the mid-frequency coefficients. The selection of the coefficient pair could also be made random as the block selection, but in our case we chose to pick successive pairs as this simple selection strategy will meet the requirements at hand. The embedding rule is described in pseudo code in Algorithm 1.

Finally, to obtain the synchronization data embedded video sequence, the inverse DCT of the altered frame(s) is computed.

Also note here that using error correction coding framework in addition to repetitive embedding of the synchronization data will provide robustness against distortion

Algorithm 1 Synchronization Data Embedding

```

1: Note:  $\mathcal{F}$ =2D DCT
2: Inputs:
3:  $I_C$ =CoverFrame
4:  $I_S$ =StegoFrame
5: Data=SyncData
6: Key
7:  $m$ =length(Sync Data)
8: for  $i = 1$  to  $m$  do                                ▷ Select the block randomly based on the key
9:   block=key(i)
10:   $f$ = $\mathcal{F}$ (block)
11:   $f'$ =zigzagscan( $f$ )                                  ▷ Select Coeffs from 6 to 21
12:   $b$ =dec2bin(Data(i))
13:  for  $j = 1$  to 8 do
14:    if  $b(j) = 1$  then
15:      if  $f'(2j - 1) \geq f'(2j)$  then
16:        Keep the Coefficients
17:      else
18:        Swap the coefficients
19:         $f'(2j - 1) \leftarrow f'(2j)$ 
20:      end if
21:    else                                             ▷  $b(j) = 0$ 
22:      if  $f'(2j - 1) \leq f'(2j)$  then
23:        Keep the coefficients
24:      else
25:        Swap the coefficients
26:      end if
27:    end if
28:  end for
29: end for
30:  $f_{new}$ =inverse zigzagscan( $f'$ )
31:  $f_{new} = \mathcal{F}^{-1}(f_{new})$ 
32: return  $f_{new}$ 

```

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

FIG. 41: Zig Zag Ordering of DCT Coefficients in an 8x8 Block.

constrained attacks which an active warden can launch.

At the end of synchronization data embedding stage, the first frame or first few frames will contain the pointers (start, end frames etc.) to the video sequence segments and the corresponding VOs in these segments that are utilized for information hiding. An example of synchronization data embedded frame for Table Tennis sequence “Ball VO” is shown in Fig. 42. The frame size is 240×352 pixels resulting in a total of 1320 8×8 DCT blocks. In this example the synchronization data include start frame=27, stop frame=34, bounding box $[x, y]$ coordinates, and width and height are (134, 105, 25, 25) respectively. For illustration purposes first frame, synchronization data embedded frame, the difference between unmarked original and data embedded frames and the start frame obtained after decoding the synchronization data are displayed in the Fig. 42. By visual inspection of the original and altered frame it can be concluded that the perceptual qualities of both frames are the same without any visual artifact.

The stego video sequence may undergo unintentional channel associated degradations, two of which are noise and filtering. To investigate the performance of the

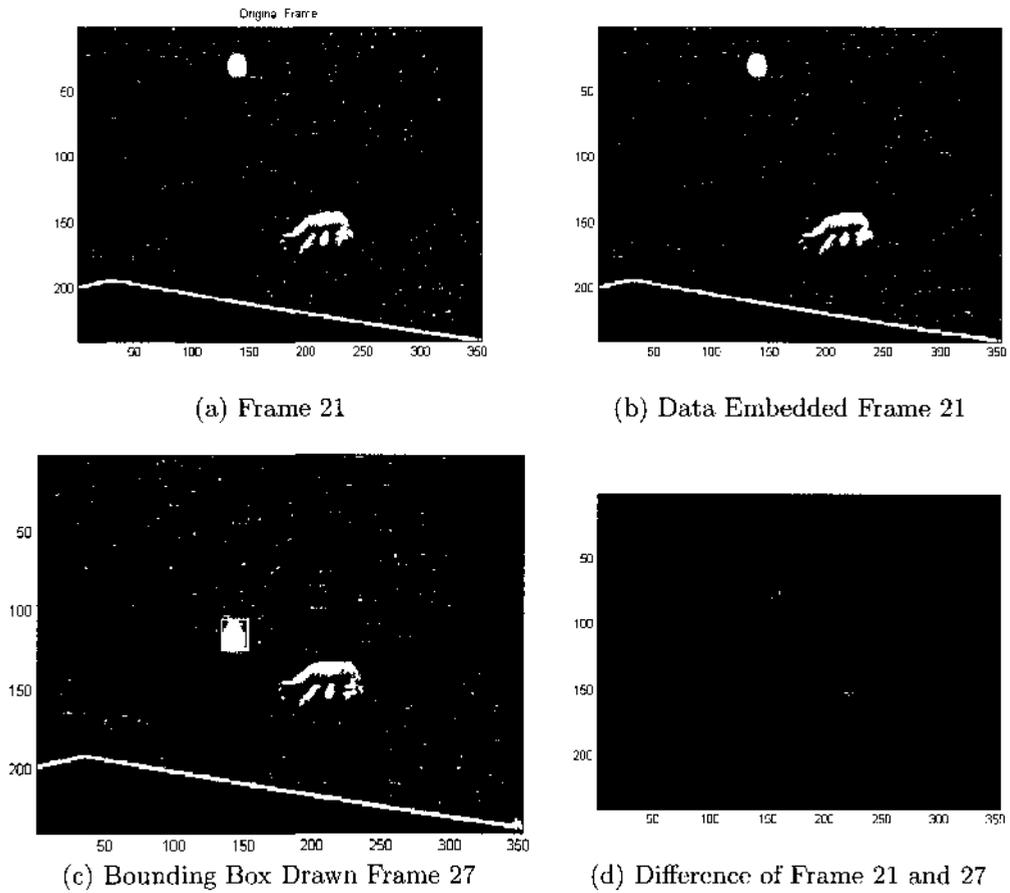


FIG. 42: Sync Data Embedded and Extracted Frames. Statistical Invisibility of the Stego Frame Based on Pixel Measures Presented in Appendix A. Image Fidelity=1.0 and MSE=0.6122. Difference Frame between the original and the stego frames shows the randomized locations of the 8x8 blocks.

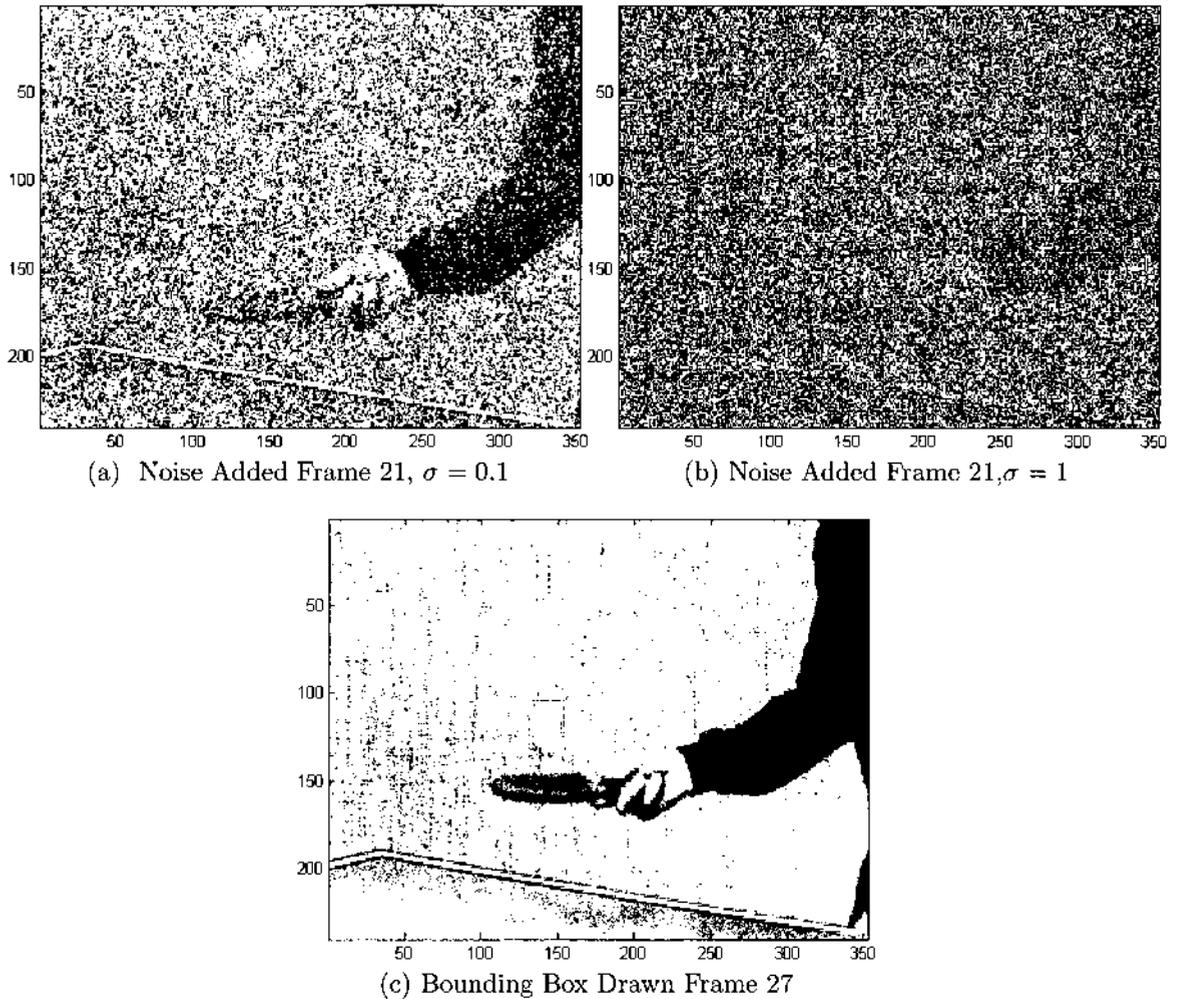


FIG. 43: Additive Noise Effect. Sync Data Embedded and Extracted Frames.

synchronization data extraction (decoding) mechanism under channel associated additive noise, noise with $N(0, 0.1)$ and $N(0, 1)$ is added to the frame in which the synchronization data is embedded. The performance of the exact data extraction at the decoder side must be satisfied to guarantee correct decoding of the secret message. The noise added frames only and with the extracted bounding box result are shown in Fig. 43. For both cases the synchronization data is extracted correctly.

An intuitive explanation for this can be provided as follows. The DCT is a linear transform, meaning $\mathcal{F}(x + n) = \mathcal{F}(x) + \mathcal{F}(n)$. During energy difference based binary data extraction, the noise coefficients will not contribute much to the difference so the DCT coefficients will determine the bit value.

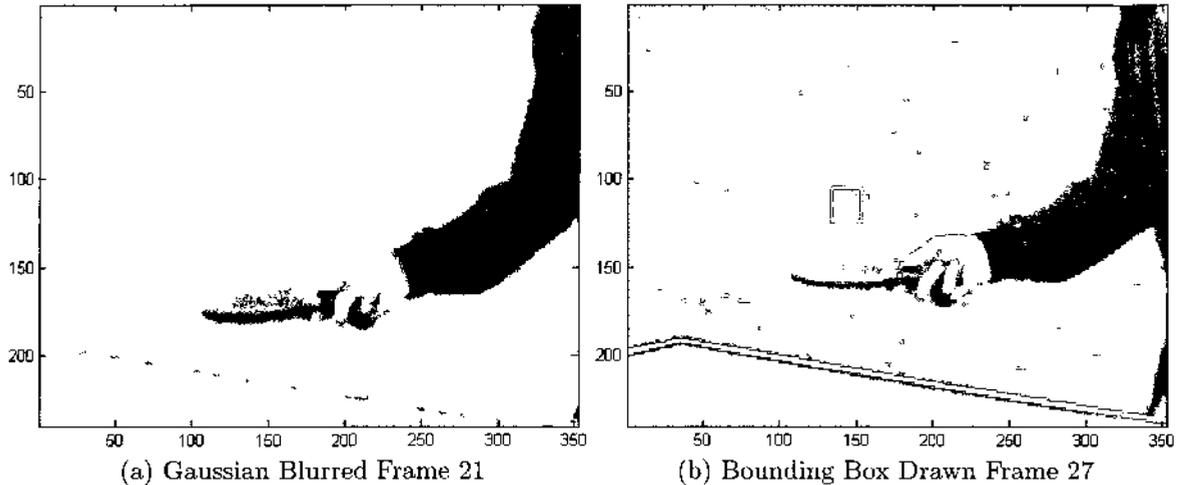


FIG. 44: Gaussian Blurr Effect. Sync Data Embedded and Extracted Frames.

Similarly, the reference frame is filtered with a Gaussian blurring function, shown in Fig. 44, having size 5×5 and $\sigma = 4$. The decoder side successfully extracted the synchronization data and the bounding box drawn at the initial frame of the partial trajectory is shown in the same figure as well. Intuitive explanation for the result is the fact that Gaussian blur eliminates the high frequency components which are not used during binary embedding.

From the above results, it can be concluded that a simple embedding technique, based on energy difference of the mid frequency band DCT coefficient pairs, survives common channel associated degradations.

When an active warden attacks the stego data, using techniques such as frame dropping, frame cropping, and format changes, the immunity of the embedding mechanism will be as follows. For the frame dropping case, as commented earlier, a simple redundancy approach provided by embedding the same synchronization data in a window of frames will render the attack obsolete.

In case of a frame cropping attack, having the data embedded in randomized locations decreases the likelihood of having corrupted blocks to almost zero. Also redundant embedding in randomized frames, such as embedding synchronization data in odd frames starting from a random frame number, will help remedy frame cropping, assuming active warden will not know the start frame and crop the same locations where the bits are embedded in every redundant frame embedding scenario.

And finally, for the case of compression format changes, the embedding mechanism will still survive the attack as we employ a commonly used transform in many coding standards and specifically use mid frequency DCT coefficients which are kept together with low frequency and DC coefficients during the quantization step. From this perspective, it will be a valid assumption to expect perfect recovery of the synchronization data after different compression ratios and formats.

V.2 TRAJECTORY PERTURBATION

The trajectory perturbation module consists of a polar quantizer that utilizes the centroid coordinates of the VO. Recall from the previous discussions that the trajectory for a particular VO is represented by

$$T\{(c_x, c_y)\} = \{(c_{x_1}, c_{y_1}), \dots, (c_{x_n}, c_{y_n})\} \quad (62)$$

where the pair (c_x, c_y) represents the bounding box coordinates obtained by using the tracking algorithm presented before.

The message is conveyed to the decoder side through the perturbations of the trajectory coordinates via the usage of a nonlinear embedding function such as a polar quantizer. In particular, the term *perturbations* refers to differences between the reconstruction level of the quantizer at a specific partition and the original motion magnitude. Note that this difference is also widely known as the quantization noise. The output of the quantization is the new motion magnitude and the phase from which the new centroid coordinates of the bounding box can easily be obtained.

If one has to make the analogy between this scheme and the existing digital modulation techniques, the uncoded Pulse Amplitude Modulation would be the best example. In both cases, the encoder selects one of possible $M = 2^J$ amplitude levels based on the input message. Or in a similar vein, in a quantization scheme the binary codewords are assigned to each individual reconstruction level, while in the perturbation based scheme presented here, the magnitude and/or phase levels are selected based on the binary message. This explains the process of how the motion magnitude and/or angle are modulated with the input binary message.

Let the magnitude of the VO motion in x-y coordinates between consecutive frames be defined in terms of centroid coordinates as $\Delta c_x = (c_{x_i}, c_{x_j})$ and $\Delta c_y = (c_{y_i}, c_{y_j})$ where i, j is the frame number and $i > j$. Then the motion magnitude and angle can be represented in polar coordinates such as

$$\theta = \arctan \frac{\Delta c_y}{\Delta c_x} \quad r = \sqrt{(\Delta c_x)^2 + (\Delta c_y)^2} \quad (63)$$

The message $m[i]$ is assumed to be a binary bit stream with $i \in \{0, 1\}$. In each frame, L-bit binary code words $\{b_1, b_2, \dots, b_L\}$ are embedded. The first bit, b_1 , is embedded through angle quantization and the remaining L-1 bits are embedded through magnitude quantization as discussed below. The first bit of the L-bit binary code word is embedded in motion angle θ by using following the quantization scheme [8]

$$\begin{aligned} \theta_{\mathcal{Q}} &= \mathcal{Q}(\theta, \Delta_{\theta}, m[i]) \\ &= \Delta_{\theta} \text{round}\left(\frac{\theta + m[i] \frac{\Delta_{\theta}}{2}}{\Delta_{\theta}}\right) + m[i] \frac{\Delta_{\theta}}{2} \end{aligned} \quad (64)$$

To embed the remaining L-1 bits of the binary code word, motion magnitudes are quantized with a uniform polar quantizer as follows. First a range (r_{min}, r_{max}) that represents the motion for the VO in the entire video sequence is defined. This value can be made a priori or can be determined by histogram analysis of the VO displacements for consecutive frames. The dynamic range obtained by this setting is divided into disjoint adjacent rings in the polar quantizer as illustrated in Fig. 45. The magnitude values that fall into corresponding disjoint ring are quantized with uniform polar quantizer $\mathcal{Q} = \{q_1, q_2, \dots, q_M\}$ with a step size Δ . The rationale behind partitioning the range into disjoint quantizers is to decrease the distortion introduced by the quantization. As an example, when the magnitude r_1 falls in the range $[0, r_1]$ the first quantizer is used, and when $r_1 < r_2$ the quantizer with the range defined by the ring $(r_1, r_2]$ is used and so on. With this approach the distortions due to the quantization of the motion magnitude will be smaller than the distortion resulting from using single global scalar quantizer with the whole motion dynamic range.

This quantization strategy allows us to build up look-up tables on both sides. The only side information that needs to be communicated to the decoder side are the binary code word length L and disjoint rings. Moreover, if we use uniform partitioning of the motion magnitude dynamic range, R, we only need to send L, R and the number of disjoint rings. After quantization, new centroid coordinates can be obtained by converting $(r_{\mathcal{Q}}, \theta_{\mathcal{Q}})$ back to Cartesian coordinates such as

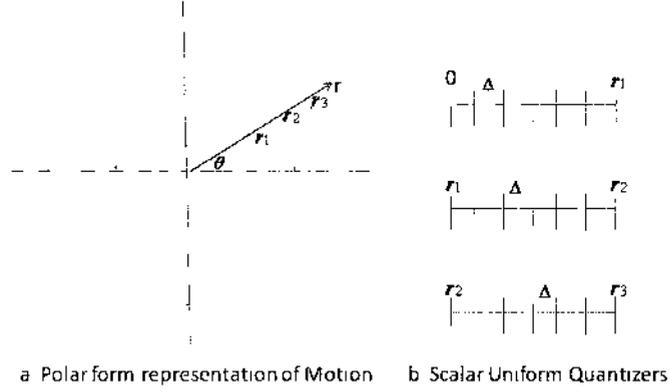


FIG. 45: Uniform Polar Quantizer for Motion Magnitude Quantization.

$$\begin{aligned}
 c_{x_{new}} &= r_Q \cos(\theta_Q) \\
 c_{y_{new}} &= r_Q \sin(\theta_Q)
 \end{aligned} \tag{65}$$

These new centroid coordinates essentially determines the new location of the “Ball VO”. The motion compensation process that allows these new coordinates to place the ball in the new location and video sequence recomposition will be discussed in detail next.

V.3 MOTION COMPENSATION AND RECOMPOSITION OF THE VIDEO SEQUENCE

A block matching algorithm is a commonly used technique in many video coding standards for determining the motion vectors. In this method the video frame is divided into non-overlapping blocks of size $M \times N$ where a global displacement vector is assigned for each block. Blocks are formed in a region without overlapping each other. Every block in a frame is compared to the corresponding blocks in the reference frame by sliding pixel positions in x and y direction within a search window and the displacement that gives smallest error with respect to an error measure such as “Sum Of Absolute Differences (SAD)” is defined as the motion of the corresponding block.

Once the motion vectors are found for each block, only those motion vectors and the resulting error are transmitted so that interframe redundancy is reduced. The reference frame and motion vectors would be sufficient to regenerate the frame

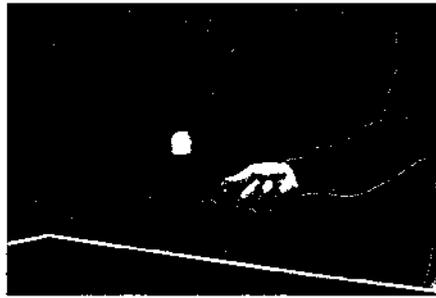
at the decoder side. This process is known as “Motion Compensated (MC)” prediction. Various search strategies have been developed for motion estimation and compensation, such as the Fixed-Size Block Matching, Object Based Block Matching and Variable Size Block Matching. The reader should note that the development of computationally efficient block based motion estimation methods is still an active research topic in which the researchers try to find solution to the problems associated with selection of optimum block size such as blockiness, computational complexity, and loss of local motion.

Two approaches are implemented for motion compensation of the original bounding box of the VO. The first is a practical approach similar to classical block based motion compensation described above. The second is a more advanced method called inpainting. For the first approach, let $x_{new}, y_{new}, x_{old}$ and y_{old}, w, h define the new and old top-left x-y coordinates, the width and the height of the VO bounding box respectively. First, a patch $\mathfrak{A}(i : i + w, j : j + h)$ belonging to the background is defined manually by the user only at the first frame. For the other frames where there will be no scene changes and the background will stay the same, this patch will basically be used for replacing pixels in the original position of the bounding box. Otherwise, a scene change detection and re-initialization of this background patch must be done. Next, the region where the ball object is located (old ball position) at, for example $O(x_{old} : x_{old} + w, y_{old} : y_{old} + h)$, is cropped automatically. Then the previously defined background patch is assigned to the old ball region, making the region the same as the background. And finally, the pixels at the cropped ball region are assigned to the new coordinates as

$$\begin{aligned} I(x_{old} : x_{old} + w, y_{old} : y_{old} + h) &= \mathfrak{A}(i : i + w, j : j + h) \\ I(x_{new} : x_{new} + w, y_{new} : y_{new} + h) &= O(x_{old} : x_{old} + w, y_{old} : y_{old} + h) \end{aligned} \quad (66)$$

where I represents the original frame. Fig.46 demonstrates three frames of the “Motion Compensated Ball VO” obtained via this method.

To evaluate the perceptual quality of the method, the patch-based motion compensated frames played consecutively. The blockiness effects have been observed from the play back frames for the homogeneous background areas as well as the areas where two objects, the ball and the player hand, come closer to each other. These examples have lead to the conclusion that this simple patch-based region replacing



(a)



(b)



(c)

FIG. 46: Patch Based Motion Compensation Example Frames. Top Left and Right:Frames 7, 14. Bottom:Frame 34.



FIG. 47: Inpainting Region and isophote directions [7].

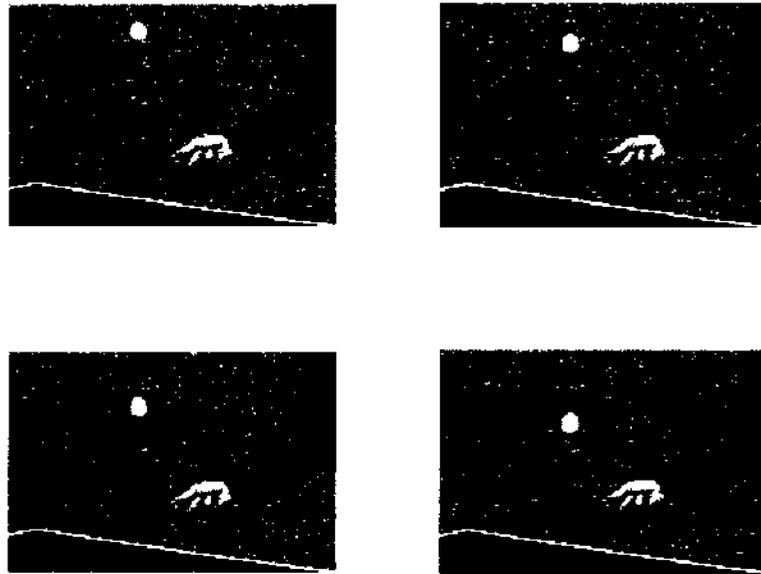
strategy might work for homogeneous backgrounds up to a certain perceptual quality. But, in the case of a non-homogeneous background, such as a soccer field with vertical dark and light shades of green, this patch-based method will not work at the border lines between these two regions.

To solve the blockiness problem and have more natural looking video frames after the motion compensation, an inpainting algorithm is used. Let Ω represent the hole to be inpainted and $\partial\Omega$ be its boundary. The basic idea behind in inpainting is to smoothly propagate the information surrounding Ω in the direction of the isophotes entering $\partial\Omega$. Both gray pixel values and isophote directions are propagated inside the region, as shown in the Fig.47. This propagation is done by numerically solving the Partial Differential Equation (PDE) [7] and [78]

$$\frac{dI}{dt} = \nabla(\Delta I) \bullet \nabla^\top I \quad (67)$$

where ∇ , Δ , and ∇^\top stand for the gradient, Laplacian, and orthogonal-gradient (isophote direction) respectively. This equation is solved only inside Ω , with proper boundary conditions in $\partial\Omega$ for the gray values and isophote directions [7]-[78]. Interested readers could refer to [7]-[78] for more information on inpainting.

The inpainting approach discussed above is implemented with the code provided by Bhat in [79]. Some example frames obtained by using the inpainting method are shown in Fig.48. Clearly from these results and the visual inspection of the inpainted video sequence, it can be concluded that the inpainting method gives better results than the simple block based background patch replacement in terms of visual perceptual quality.



Motion Compansated (Inpainted) Frames 21 to 24

FIG. 48: Inpainted Frames from 21 to 24.

Note that the processing time of the inpainting algorithm for one frame is approximately 7 seconds when the program runs on an IBM T42p Laptop having 1 GB RAM with Intel Pentium 2 GHz Processor. This is a fairly reasonable amount of time for an off-line video sequence processing. Since the encoder side implements video motion compensation and recomposition off line, the time it takes to process the video sequence should not be a pressing issue. But, in any case the per frame processing time could be decreased by increasing the computational resources.

V.4 DECODER SIDE TRACKING AND DETECTION

The input to the decoder side is the stego video sequence obtained by applying the proposed method to the original video sequence. The decoder first checks the synchronization information indicating exactly from which frame to start with and the location of the object. Recall from Section V.1 that the synchronization information is embedded in randomized 8x8 DCT blocks of the cover frames where both the encoder and the decoder are assumed to have the same seed for the random number generator. So from the practical application point of view, the decoder will only

check the DCT blocks where the synchronization information is embedded.

Once the decoder is synchronized with the encoder in terms of start frame and the location of the VO, the tracking module starts tracking the VO to get the centroid coordinates. An example of the decoder's synchronizing itself to the start frame and right VO is depicted in Fig.43.

Next, the motion magnitude and the angle is computed from the centroid coordinates similar to what is done in the encoding stage. And finally the motion magnitude and/or motion angle is compared with a codebook, whose entries are obtained by using the same quantizer as that of the embedding quantizer, to read out the binary message.

V.5 CHAPTER SUMMARY

This chapter discusses the block diagram of the proposed method with an overall process flow of the intermediate steps. Note that the tracking algorithm is already discussed in detail in Chapter IV Section IV.4 and IV.5. Therefore, the tracking module of the proposed method is skipped in this chapter. First, the following key question inherent to every blind data hiding mechanism is answered: "how should the synchronization between embedder and the decoder be established?". In addition to the classical loose assumption that the decoder has the side information or the synchronization info is sent through a separate channel, a practical DCT based approach is proposed as a solution to the synchronization problem. The proposed method is tested for different channel associated conditions such as filtering and noise and the results demonstrated that *when there is a passive steganalysist who is only monitoring the channel, DCT energy difference based method will survive the channel degradations.*

In the case of an active steganalysist who could launch a battery of attacks such as frame cropping, frame dropping, rotation etc., aiming at breaking the synchronization between the encoder and sender, a redundant embedding of the same synchronization data in randomized frames, such as embedding synchronization data in odd frames starting from a random frame number or resending the same data again, will mitigate the problem.

Next, how the message is conveyed to the decoder side via the perturbations of the trajectory coordinates, i.e. the process of motion magnitude and/or angle modulation with the input binary message through the usage of a uniform polar

quantizer, is explained in detail.

The goal of the motion compensation and video sequence recomposition step is to reproduce video frames which do not raise suspicion, which is important from the perceptual invisibility constraint perspective. To accomplish this task, two practical approaches are presented. The first is the the block based motion compensation approach, which has resulted in blockiness artifacts even in the homogeneous regions. Second, the inpainting based approach, which gave more natural looking, perceptually less-distracting results. After the motion compensation stage, the individual video frames are re-written in the same video sequence to get the stego sequence.

Once the stego video sequence is obtained it could be transmitted to the receiver in many ways, such as an attachment to an email or a posting to a web page. Assuming that the sender and the receiver have established a channel, the steps involved in the message decoding phase will be similar to the encoding phase. The tracking algorithm will first obtain the synchronization information to start tracking and then track the VO for the trajectory segments indicated by the synchronization data. Once the trajectory centroid coordinates are obtained, the polar quantization scheme which is used in the encoding phase will be used to decode the binary message.

Having discussed the proposed method in detail, the experimental results of the proposed method will be presented in Chapter VI next.

CHAPTER VI

EXPERIMENTAL RESULTS AND DISCUSSIONS

The modules of the proposed method that perform embedding and decoding are introduced in Chapter V. This chapter presents the experimental results and discussion on the performance of the proposed method. The experiments were conducted by using two different sports video sequences, “Table Tennis” and “Soccer Ball” both consisting of smooth linear and sudden motion change. The aim was to demonstrate the practical applicability of the proposed method as a proof of concept. The “Table Tennis” sequence has 65 frames which have a frame size of 352 x 240 and a frame rate of 30 fps. The “Soccer Ball” sequence has 80 frames which have a frame size of 360x480 with a frame rate of 30 fps. In the experiments a random binary string was generated and used as the message to be sent to the decoder side.

The results of each step of the proposed method will be presented in detail next.

VI.1 VO SELECTION, PREPROCESSING AND TRACKING

Recall from the discussions in Chapter IV and V that the proposed algorithm utilizes a user defined VO and its trajectory points as the features to be modified for sending the data to the receiver side.

As discussed earlier, VO segmentation could be unsupervised, meaning fully automatic segmentation done by a segmentation algorithm, or supervised where a user interaction is needed. The first type is suitable for object based coding standards. A supervised VO definition was selected as this option fit-for-purpose for the application presented in this work. The VO selection is accomplished by first reading the first frame of the Table Tennis sequence and displaying it on the monitor to select the VO. Next, the centroid coordinates of the “Ball VO” are obtained manually as shown in the Fig.49.

The rationale behind using the ball trajectories for perturbation data embedding is re-iterated as follows. Although the ball has small size and its shape varies due to its motion, having a single homogeneous region with a common motion, compared to a player whose body parts might have different motion directions, and a semantic meaning make ball trajectories attractive for manipulations. Because the changes made to the trajectories by the proposed perturbation method will only modify a

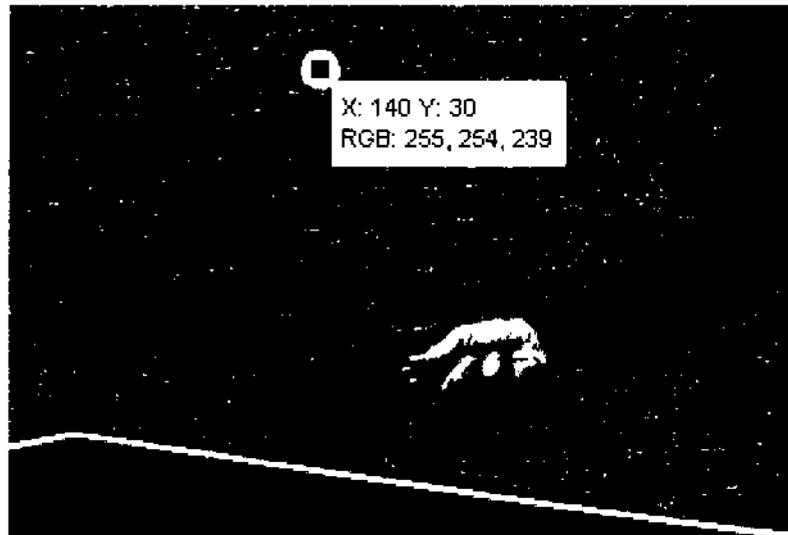


FIG. 49: Ball VO Centroid Coordinates.

small area of the video frame due to the ball size. These small modifications will result in video sequences with almost no visible and statistical artifacts when both pixel wise global image quality (e.g. PSNR, MSE) and statistical measures (e.g., histogram based metrics) are used. Any modifications that keep the semantic meaning of the ball, or even a player, trajectory in the stego scene will be imperceivable for a viewer.

As discussed earlier VO's full trajectory (trajectory from first frame to the last frame) is not suitable for perturbation because the tracking algorithm does not have the features to identify player possessions, full and/or partial occlusions which usually exist in more advanced tracking algorithms such as in [6]. Additionally, deformation of the ball due to fast motion smear (see Fig.26 in Chapter IV) has not been dealt with at the moment. Those two issues were left for future research topics as a continuation of the work presented here.

To identify the trajectory segments that could be used for perturbation, the video sequence is analyzed with an editor shown in the Fig.50. Once the trajectory segments are identified with user supervision, the synchronization data (start, end frame numbers and VO location) is formed.

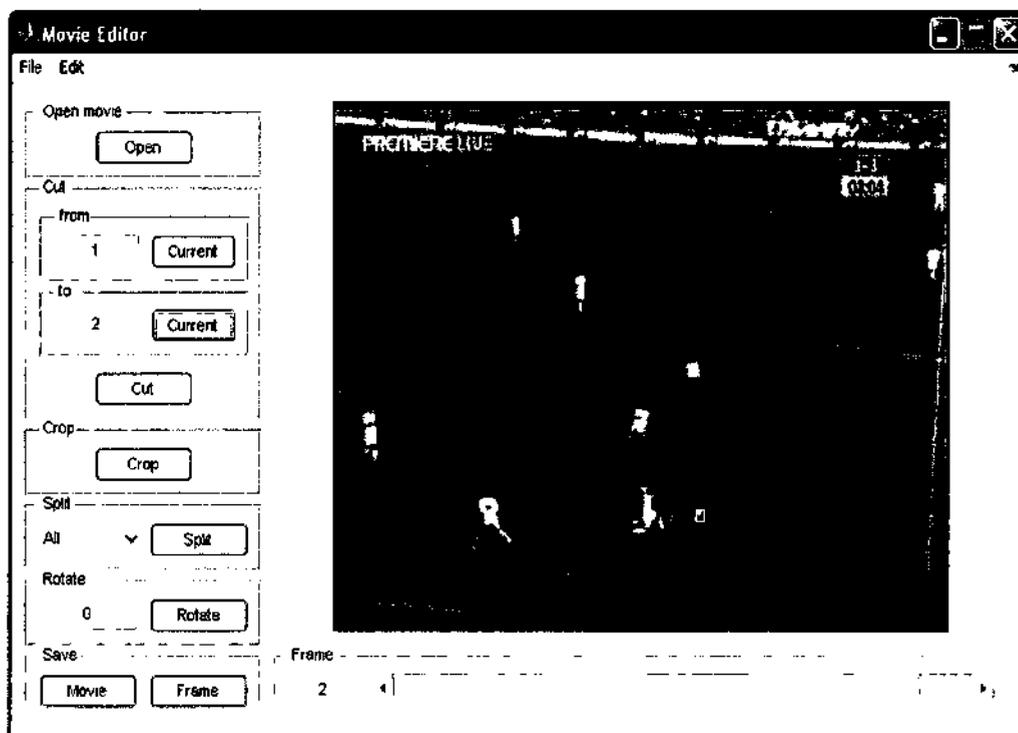


FIG. 50: Movie Editor for Frame-by-Frame Analysis of the Video Sequence.

VI.1.1 VO Tracking

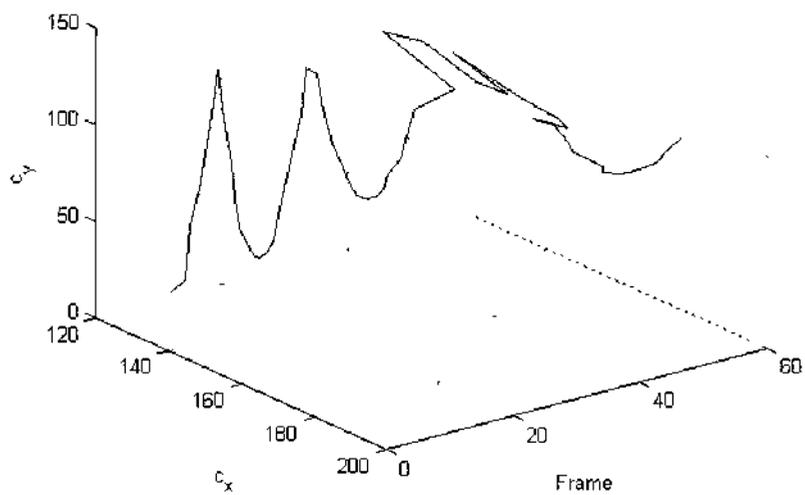
The tracking result for the Ball VO, obtained by using the method described in Chapter IV Section IV, is illustrated in Fig.51.

The Ball VO motion magnitude histogram and probability density fit to the data are also shown in Fig. 52. From the histogram analysis, it was concluded that small to large displacements between the frames were due to the reduced and increased acceleration of the ball during free fall or upwards direction. In Fig. 52.b the probability density function (pdf), which is Rician with $\sigma = 9.94153$ and non-centrality parameter $s = 0.00279$, is given. This probability density estimate was obtained by fitting a curve to the motion magnitude histogram data. Note that the pdf plot is included for illustrative purposes and would otherwise be different for different motion data obtained for tracking other VO.

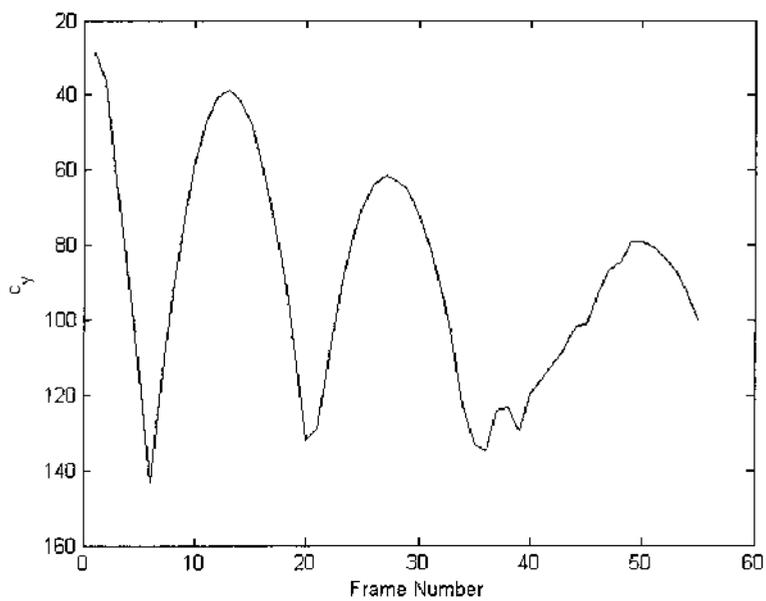
Recall that the proposed method is a quantization based one which parameters depend upon the motion magnitude and angle. To set up the parameters for the quantizer, the ball motion magnitude histogram was analyzed as follows. First, the dynamic range (r_{max}, r_{min}) of the motion magnitude was determined as a result of histogram analysis and found to be $r_{min} = 0$ and $r_{max} = 30$. Based on the histogram analysis, the dynamic range was divided into disjoint quantization rings as $r_1 = [0, 8]$, $r_2 = (8, 16]$, $r_3 = (16, 24]$, $r_4 = (24, 32]$ with step size $\Delta = 2$ (Refer to the Chapter V Section V.2 for explanation of how motion parameters are quantized.). A 4-level scalar quantizer resulting in 2 bits per quantized magnitude was selected for each disjoint magnitude interval.

Again, the reason behind using disjoint rings as opposed to one scalar quantizer for the whole dynamic range is to decrease the distortion and preserve the semantic meaning of the VO motion which would otherwise result in big motion magnitude differences between the actual and the quantized values.

For a quantization mechanism like this, the only side information needed on the decoder side is the disjoint ring intervals and the step size(s) used at the encoder side. This side information could also be transmitted either in a separate channel or embedded in the cover data using a synchronization data embedding technique. But for this implementation the decoder is assumed to have these parameters.

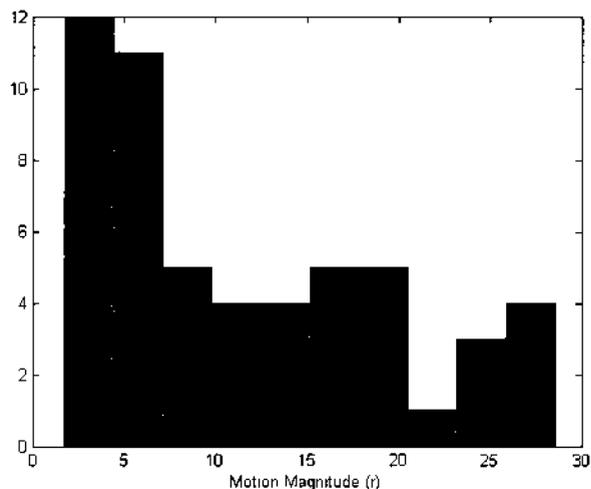


(a) Ball VO Trajectory

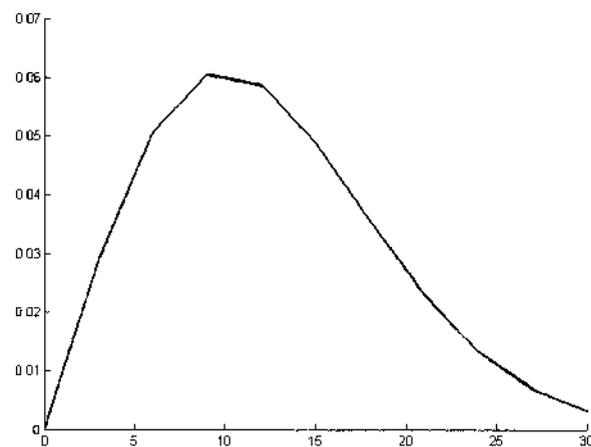


(b) Ball Motion in Y Dimension

FIG. 51: Table Tennis Sequence "Ball VO" Trajectory.



(a) Histogram of the Ball Motion Magnitude



(b) Probability Density Function Fit

FIG. 52: Histogram and Density Function Analysis of “Bal VO”.

VI.1.2 Data Embedding and Decoding

For the first experiment, the first 33 frames of the “Table Tennis” sequence were used for perturbation based data hiding. The reason for not using all of the frames in the sequence was because the sequence has occlusions (player catching and holding the ball), split of the objects (the ball and the hand of the player) and the deformation of the ball due to fast motion after the player hits the ball. As mentioned previously, the tracking algorithm does not have features to detect occlusion, split, merging and deformation of the objects. Thus, only the trajectory segment, in which no occlusion and/or drastic deformation exists, was used for illustration of the idea behind the

proposed method. On the other hand, the segment that was used for data hiding also has complex motion (ball flying freely in the air with acceleration and change of motion direction, moderate deformation due to speed of the ball, etc.) so, in that respect, the selection of these frames was considered to be sufficient for demonstration of the proposed method.

Extension of the proposed method to the scenes with more complex motion requires a tracking algorithm having intelligence through the use of occlusion detection, non-rigid body tracking features, etc.

Once the trajectory points, centroid coordinates of the VO, were computed, the embedding function modulates those coordinates with the binary data using the polar quantization mechanism described before.

The first experiment conducted was the magnitude-only quantization of the VO motion magnitude. Since the number of frames that were used for data embedding was 33, a binary string of 66 bits (2 bits per frame), representing random data as the output of a cryptological algorithm, was used as the input to the perturbation based data embedding function (Refer to Fig.38.)

The new centroid coordinates were found basically by mapping the 2 bit code-words to new reconstruction levels in the polar quantizer. The motion compensation was done via inpainting and recomposition of the video sequence was done by creating an “avi” file and writing the modified (stego) frames into it.

The decoding process was implemented in the way presented in Chapter V Sections V.3 and V.4. An exemplary output of the decoded binary information is illustrated in Fig.53.

For performance measurements under AWGN, Bit Error Rate (BER) of the proposed algorithm was measured for varying noise variances. The results of the BER for magnitude only quantization are tabulated in Table.1. The reader should note that the BER performance for noise variances up to 0.1 was tested. Because, as can be seen from the Fig. 31 in Chapter IV Section IV.5, as the noise variance is increased beyond $\sigma_n^2 = 0.1$, the visual quality of the video frames deteriorates dramatically. And from the previous tracking performance experiments, it was concluded that the tracking module could not track the object reliably beyond $\sigma_n^2 = 0.1$. Therefore, the experiment was performed for values up to $\sigma_n^2 = 0.1$.

The BER results are consistent with the performance of the tracking algorithm presented in Chapter IV Section IV.5. As can be observed from the results, as

```

Edit  Debug  Distributed  Desktop  Window  Help
Current Directory C:\Documents and Settings\Administrator\My Documents\MATLAB
Command Window
>>
1 0 1 0 1 1 0 0 1 0 1 1 1 1 0 0
Columns 17 through 32
1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 1
Columns 33 through 48
0 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0
Columns 49 through 64
0 0 0 1 0 1 0 1 0 1 1 1 0 1 1 1
Columns 65 through 80
1 1 0 0 0 1 1 0 1 1 0 1 1 0 1 1
Columns 81 through 96
1 1 1 1 0 1 0 0 1 1 0 0 1 0 0 1
Columns 97 through 108
0 0 0 1 1 0 1 0 0 0 1 1
>> decode_msg(x,y,54)
ans =
1010110010111100110100100001000101001101101011100001010101110111110001011011011111010011001001000110100011

```

FIG. 53: Decoded Binary Data.

TABLE 1: BER Performance of “Table Tennis” Sequence for Magnitude Quantization Based Embedding under AWGN with Varying Variance.

σ_n^2	BER
0.001	0
0.01	13.23%
0.05	14.70%
0.1	19.11%

the noise variance increases, the tracking error increases, which yields errors in the decoding side. This is due to the fact that noise deteriorates the performance of the object locating phase of the tracking in which color, size and aspect ratios are used through morphological operations for filtering non-ball objects. The deviations from the actual quantized coordinate locations (on the encoder side) result in the detected magnitude on the decoder side to fall under different disjoint quantization rings and/or a different partition in the same disjoint ring.

One possible solution to mitigate the effects of noise would be the usage of a tracking method, which is invariant to noise or could decrease the effect of noise such

as the one proposed by Porikli et al. in [80]. They propose a simple and elegant algorithm to track nonrigid objects using a covariance-based object description and an update mechanism which effectively adapts itself to the undergoing object deformations and appearance changes. The power of their method is that the covariance tracking method does not make any assumption on the measurement noise and the motion of the tracked objects and provides the global optimal solution. Through the experimental results, it was shown that the covariance based tracking method was capable of tracking objects in AWGN with very high success rates.

Discussion on Perceptual and Statistical Invisibility

The visual perceptual quality assessment of the resultant stego frames was done primarily by subjective assessments of three researchers who were experts in the field and two non-experts. Also, the stego video sequences were shown to an expert audience of 10 people in a conference. The randomly selected stego and original video frames are shown in Fig.54 for illustrative purposes. From the subjective quality assessments, it was concluded that the visual quality of the stego frames were almost identical to that of the original ones. There were no local artifacts, such as blockiness, or global artifacts, such as the change in the semantic meaning of the motion which could otherwise raise suspicion.

To further justify the claim that the proposed method did not degrade the perceptual quality of the frames, pixel wise steganalysis measures presented in Appendix B were used to compute the differences between the original and the stego frames. The reader should note that it is possible to get an estimate of the original frame, by an averaging method such as collusion, from which a comparison with the suspected stego frame may reveal the existence of the stego channel under steganalysis.

The steganalysis results of the “Table Tennis” video sequence are given in Fig.55 and 56. High per frame PSNR values shown in Fig.55a also support the high perceptual quality of the perturbed frames. Note that the gaps in per frame PSNR values indicate infinite values as MSE is almost zero for those frames. Measures that are based on pixel wise differences such as MSE and MAE also justify the claim that stego frames are almost identical to the original ones.

In order to verify whether or not the data embedding algorithm has altered any statistical property of the original video frame, the stego frames were tested against commonly used histogram and correlation-based metrics given in Appendix

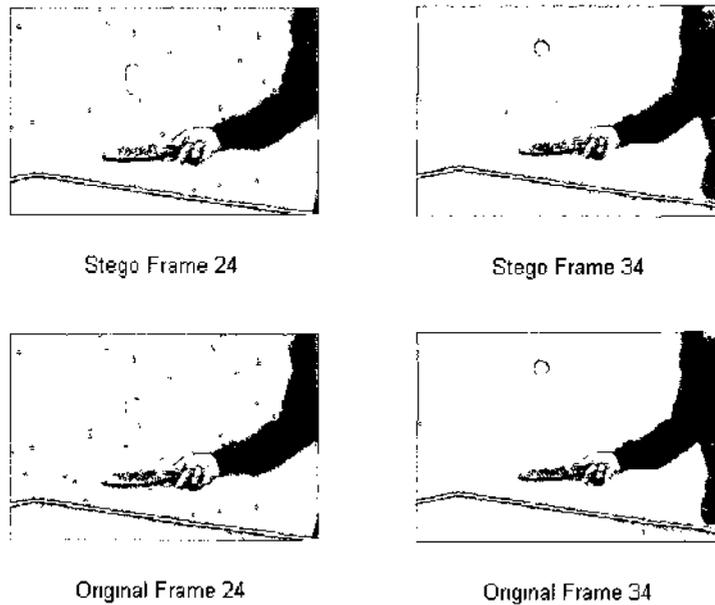
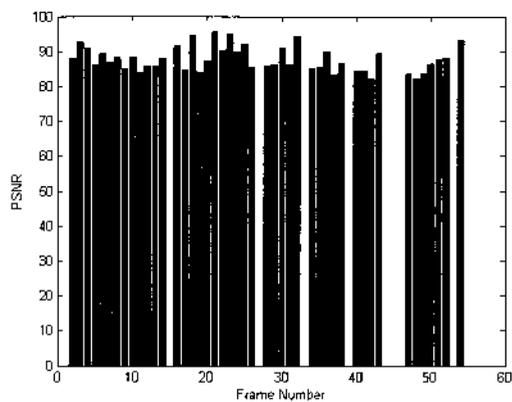


FIG. 54: Stego and Original Frames.

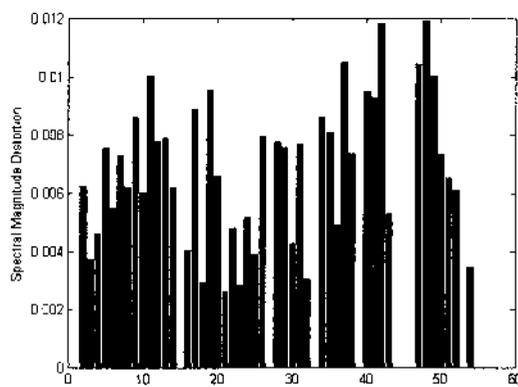
B. Among the histogram metrics, the χ^2 metric was shown to be the most sensitive histogram differencing metric. It is used to compare two binned data sets to identify whether or not they are drawn from the same distribution. If the two distributions are identical, then difference $d_{hist} = \chi^2(H_o, H_s) = 0$ and it will increase towards 1 as the histograms differ more.

The results illustrated in Fig.56b were found to be significant as they are on the order of 10^{-4} and 10^{-6} for histogram based differences and approximately 1 for normalized cross correlation indicating low statistical evidence of the perturbation. To investigate the effects of perturbations in the frequency domain per frame spectral difference between the original frames and the perturbed frames were computed. The results illustrated in Fig.55b also prove that the degree of distortion in DFT domain is almost negligible.

In the next experiment, both the phase and the magnitude of the VO were quantized to embed a total of three bits, one bit for phase quantization and two bits for magnitude quantization. The magnitude quantization levels were kept the same as before. For the phase angle quantization, the scheme shown in Fig.57 was used. The reason behind using such a quantization scheme rather than the general uniform

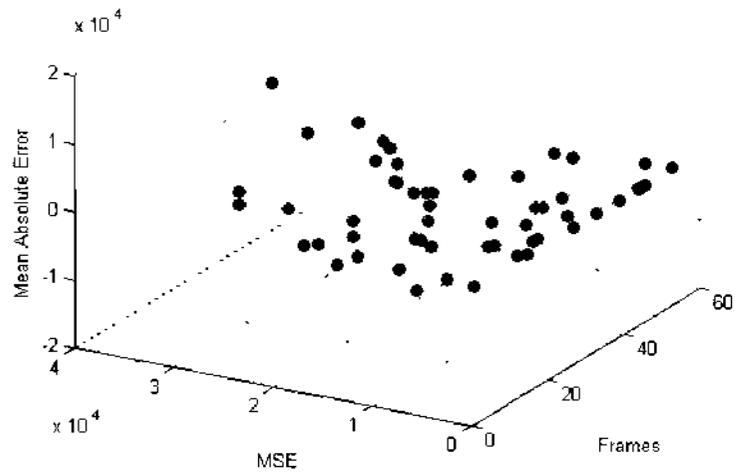


(a) Per Frame PSNR

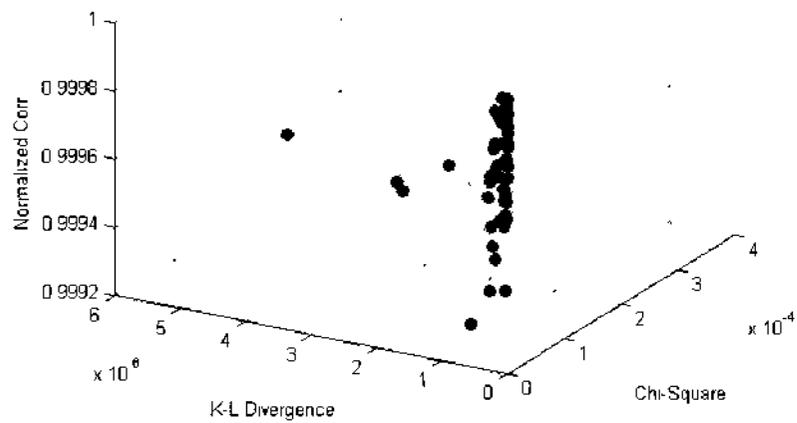


(b) Per Frame Spectral Magnitude Distortion Analysis

FIG. 55: Steganalysis Results of “Bal VO” Perturbation Based Data Hiding.



(a) Pixel Wise Distortion Based Steganalysis



(b) Histogram Based Steganalysis Results

FIG. 56: Steganalysis Results of “Bal VO” Perturbation Based Data Hiding.

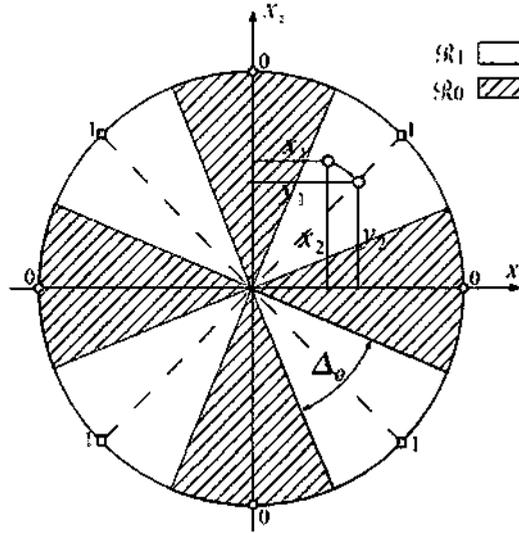


FIG. 57: Angle Quantization Scheme [8].

or non-uniform polar quantizer with P phase levels (Refer back to Chapter III) is the fact that in the polar quantization scheme the phase may be assigned to any of quantization cells based on the codeword which might result in sudden motion change (e.g. VO motion angle having motion in 45° direction gets quantized to 135° then 60° and so on). Instead, phase changes were limited only to the neighborhood of the original motion direction with $\pm\Delta_\theta$. This quantization strategy was chosen to implement small perturbations and at the same time to preserve the semantic meaning of the VO motion. To better explain the effects of phase quantization on the semantic meaning of the VO motion, some example frames in which the phase of the motion was quantized using the strategy in Fig.57 with $\Delta_\theta = \pm 30^\circ$ are shown in Fig.58. By visual inspection of the frames, it can be deduced that the semantic meaning of the ball is lost because the ball no longer moves along a smooth trajectory compared to the original trajectory.

For joint quantization of the motion phase angle and the magnitude, the phase quantization step size was empirically set to $\Delta_\theta = 2^\circ$ whereas the magnitude quantization levels were kept the same as before. With this set-up a total of 99 bits (3 bits/frame and a total of 33 frames) could be transmitted to the decoder side. The performance of the decoder for joint magnitude and phase angle quantization under AWGN is tabulated in Table 2.

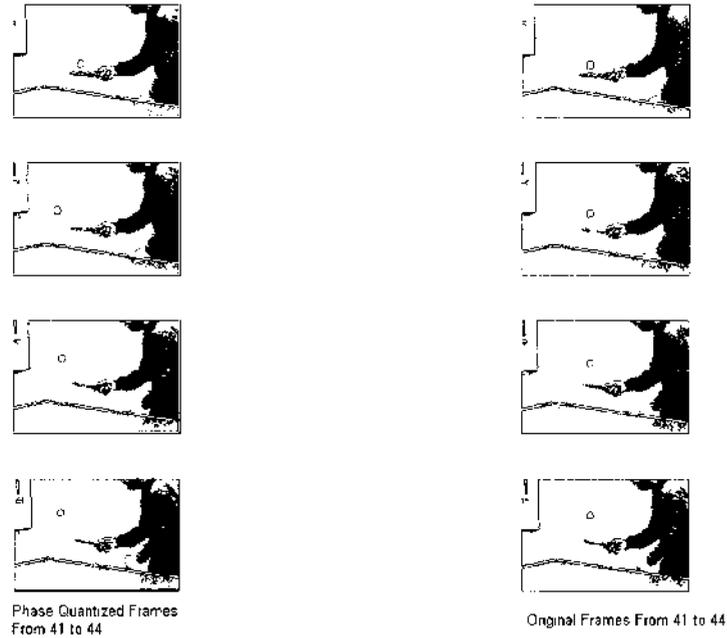


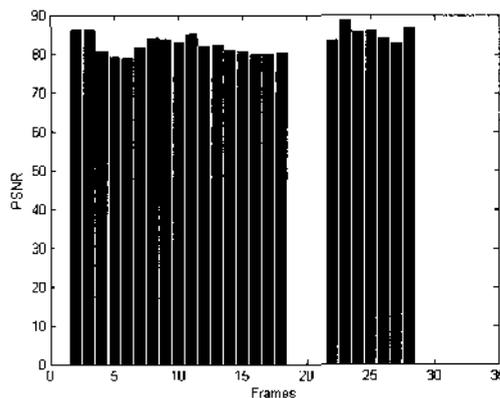
FIG. 58: Magnitude and Angle Quantized Frames. The Semantic Meaning of the VO is lost.

Another observation made from the experimental results was that the BER performance of joint quantization of both the magnitude and phase got worse as compared to the magnitude-only case. This makes sense because as opposed to the magnitude-only case, the tracking error now effects the phase as well as the magnitude which results in the increase in BER. One remedy against the effect of noise would be increasing the magnitude and phase quantization step sizes. However, in this case, the visual perceptual quality of the object motion might be degraded seriously.

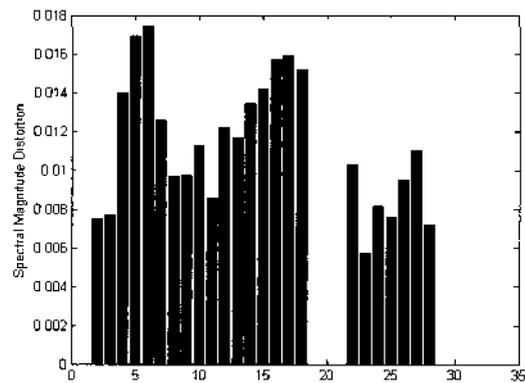
As it has been done for the magnitude-only case the perceptual and statistical invisibility tests were done using the steganalysis metrics given in Appendix B. The results are illustrated in Fig.59 and 60. After examining the results, it can be concluded that the phase quantization, if the quantization step size is kept small, does not degrade the statistical properties of the original frames. The results of the per frame histogram based metrics illustrated in Fig.60b were found to be significantly low since they are on the order of 10^{-4} and approximately 1 for normalized cross correlation indicating low statistical evidence of the perturbation. To investigate the effect of perturbations in the frequency domain as in the case of magnitude only

TABLE 2: BER Performance of “Table Tennis” Sequence for both Magnitude and Phase Quantization Based Embedding under AWGN with Varying Variance.

σ_n^2	BER
0.001	9.37%
0.01	22%
0.05	21.2%
0.1	28.3%



(a) Per Frame PSNR

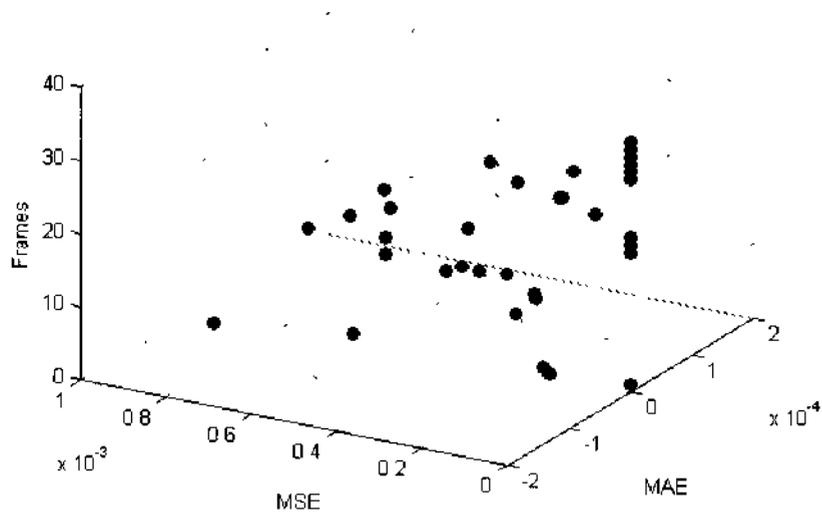


(b) Per Frame Spectral Magnitude Distortion Measure

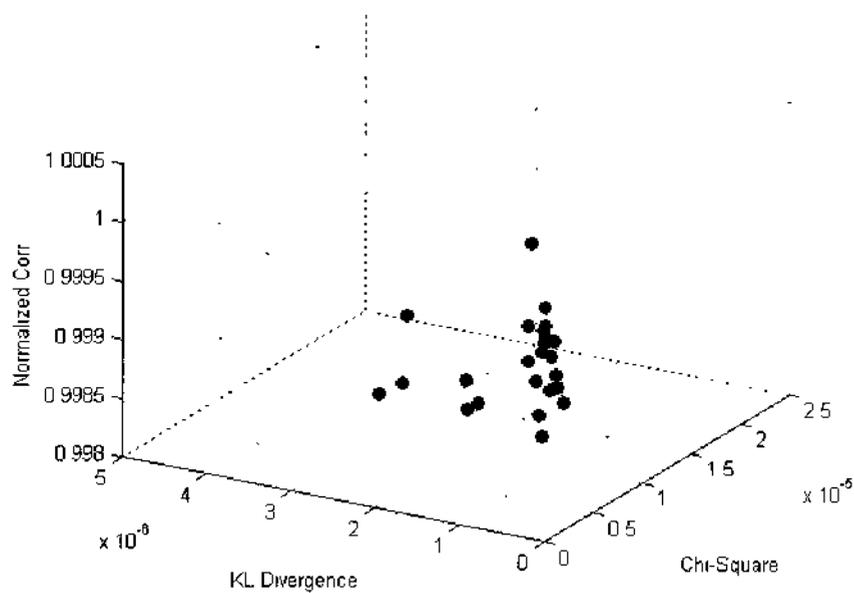
FIG. 59: The Results of the Steganalysis Measures for Magnitude and Phase Quantized VO Stego Frames.

quantization, the per frame spectral difference between the original frames and the perturbed frames were computed. The results illustrated in Fig.59b also prove that the degree of distortion in the DFT domain is almost negligible.

From the above results it can be concluded that both magnitude only and magnitude-phase joint quantization scheme could be used for perturbation based data hiding. The only constraint is the smoothness of the motion trajectory as there could be cases for which even small perturbations to the motion direction (motion angle) may result in noticeable distortions. Therefore, it will be valid to say that the generalization of joint phase-magnitude quantization depends upon the VO motion trajectory. For cases where the VO has smooth linear motion, only magnitude quantization might be feasible from the perceptual invisibility perspective.



(a) Per Frame MSE versus MAE



(b) Per Frame Histogram Measures

FIG. 60: The Results of the Steganalysis Measures for Magnitude and Phase Quantized VO Stego Frames.

In another experiment the proposed algorithm was tested by using the “Soccer Ball” video sequence, which consists of smooth linear VO motion. The “Soccer Ball” sequence has 80 frames with the frame size of 360x480 pixels. As it was done for the “Table Tennis” sequence, a binary string, representing random data as the output of a cryptological algorithm, was generated and used as the message to be sent to the decoder side.

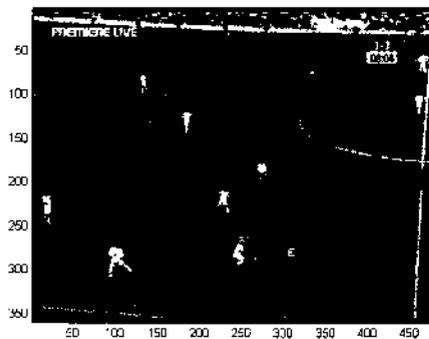
The first step for setting up the quantization scheme parameters is the analysis of the motion distribution. For this purpose, the object motion magnitude distribution, illustrated in Fig.61, was computed. Smoothness of the Ball VO motion can easily be recognized from Fig.61c where the ball has linear motion in both axes.

After histogram analysis of the motion magnitude (see Fig.61b) the dynamic range (r_{max}, r_{min}) of the ball VO motion magnitude was determined as $r_{min} = 0$ and $r_{max} = 8$. Linear motion of the object and the comparatively small motion magnitude (limited dynamic range) between the frames have set limitations on the quantization step size and the number of the levels. Since the magnitude quantization step size could not be set arbitrarily large due to highly likely perceptual distortions and loss of the semantic meaning of the object, the quantizer parameters were empirically set to be $L=2$ resulting in two disjoint quantization rings (or cells). As done previously for the Table Tennis video sequence example, the dynamic range was divided into disjoint quantization rings as $r_1 = [0, 4], r_2 = (4, 8]$ with step size $\Delta = 2$ (Refer to the Chapter V Section V.2 for explanation of how motion parameters are quantized.). For each quantization cell one bit quantizer resulting in two levels was used.

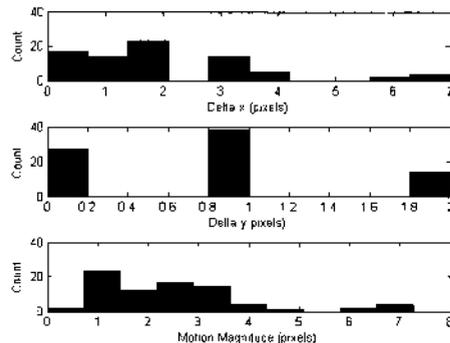
Reiterating the reason behind using disjoint rings as opposed to a one scalar quantizer for the whole dynamic range is to decrease the distortion and preserve the semantic meaning of the VO motion which would otherwise result in big motion magnitude differences between the actual and the quantized values.

For a total of 80 frames and the magnitude only quantization case where 1 bits could be embedded per frame, the total capacity came out to be 80 bits. The motion compensation was done via inpainting and recomposition of the video sequence was done by creating an avi file and saving the modified (stego) frames.

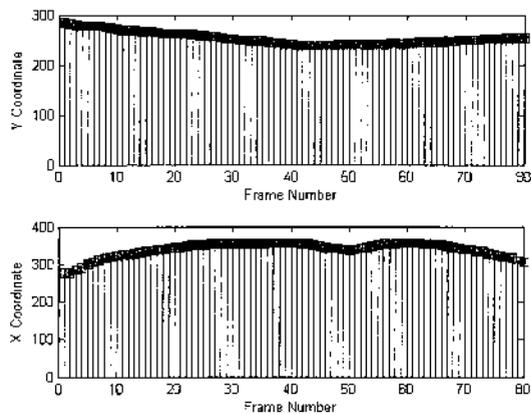
BER performance tests were conducted for varying noise variances. Noisy first frames of the “Soccer Ball” sequence for $\sigma_n^2 = [0.001, 0.01, 0.05 \text{ and } 0.1]$ are illustrated in Fig.62. As can be seen from the figure, when $\sigma_n^2 = 0.001$ visual quality of the frame is acceptable but when the noise variances is increased, it is almost impossible to



(a) An Example of Soccer Video Sequence and the “Ball VO”



(b) Soccer Ball VO Motion Magnitude Histogram



(c) X and Y Direction Motion Magnitude Plots

FIG. 61: Soccer Ball VO Motion Magnitude.

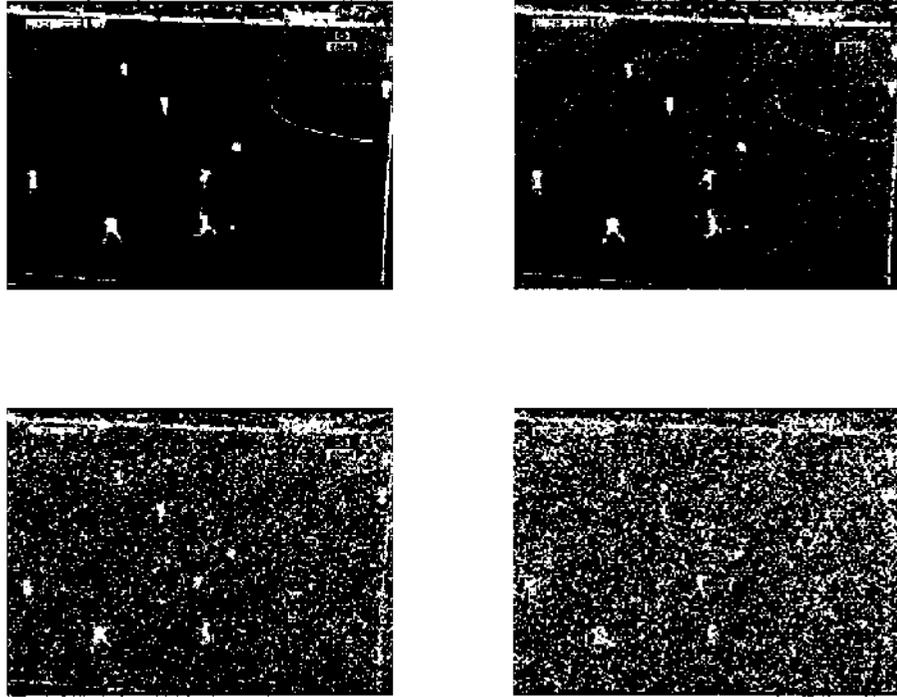


FIG. 62: Noise Added 1st Frame. Top-Left, Top-Right, Bottom-Left and Bottom-Right represent the noise variance in order $\sigma_n^2 = [0.001, 0.01, 0.05, 0.1]$.

perceive the ball. One observation that can be made is that the small size of the ball when the scene is in the global view may cause problems in the tracking phase under AWGN. The BER results are tabulated in Table 3. Since the tracker could not track the “Ball VO” in noisy frames for $\sigma_n^2 = 0.05$ and 0.1 , BER results are not included. Clearly from the results one can observe the effects of object size. Compared to the table tennis, the soccer ball has smaller size in pixels which in return affects the BER performance adversely due to the increased tracking errors under AWGN.

In the second experiment, both the magnitude and phase were quantized to embed two bits, one bit by magnitude and one bit by phase quantization. This quantization strategy has resulted in an embedding capacity of 160 bits for a total of 80 frames. Error performance of the magnitude-phase joint quantization scheme was tested for AWGN with varying noise variance. Note that, as in the case of magnitude-only quantization error performance for AWGN, the tracker could not track VO for the

TABLE 3: BER Performance of “Soccer Ball” Sequence for Magnitude Quantization Based Embedding under AWGN with Varying Variance.

σ_n^2	BER
0.001	14%
0.01	23%
0.05	-
0.1	-

cases where noise variance was either 0.05 or 0.1. The BER performance results for the other two cases are tabulated in Table 4. From the results, it can be deduced that the bit error increases due to the fact that the magnitude errors yields errors in the phase angle of the motion.

TABLE 4: BER Performance of “Soccer Ball” for both Magnitude and Phase Quantization Based Embedding under AWGN with Varying Variance.

σ_n^2	BER
0.001	25%
0.01	35%
0.05	-
0.1	-

Discussion on Perceptual and Statistical Invisibility

Steganalysis of the “Soccer Ball” VO was done using the measures given in Appendix B. The results are illustrated in Fig.63 and Fig.64. From the subjective quality assessments, it was observed that the perceptual visual quality of the stego frames were almost identical to those of the original ones. This observation was also supported by high PSNR results of the steganalysis. Neither local (e.g.blockiness) nor global artifacts, such as the change in the semantic meaning of the motion, was observed in the stego frames. Pixel wise and histogram based steganalysis metrics were used to compute the deviations from the original frames.

Both the results of the pixel wise and histogram difference based metrics prove that the modifications made to the trajectory of the ball result in small localized changes that are almost negligible. From this perspective, it is proven one more time that the proposed algorithm provides steganographic stealth against an outside observer.

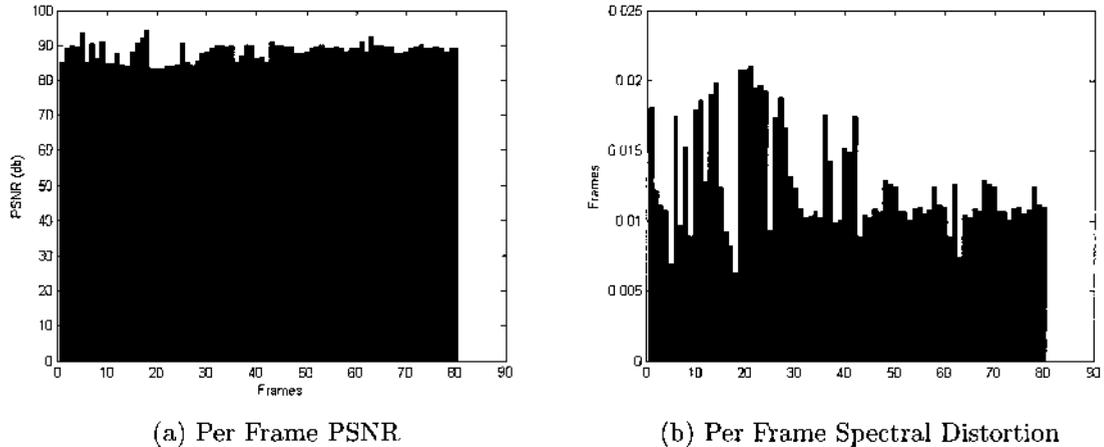


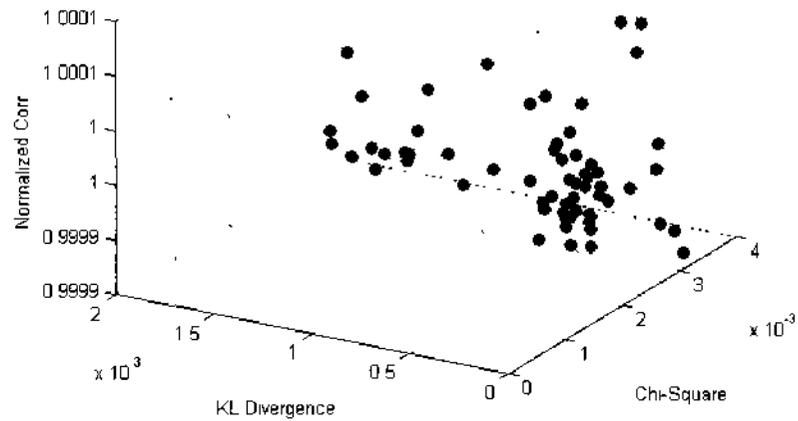
FIG. 63: PSNR and Spectral Distortion Results.

VI.2 COMPARISON OF THE PROPOSED METHOD WITH OTHER METHODS

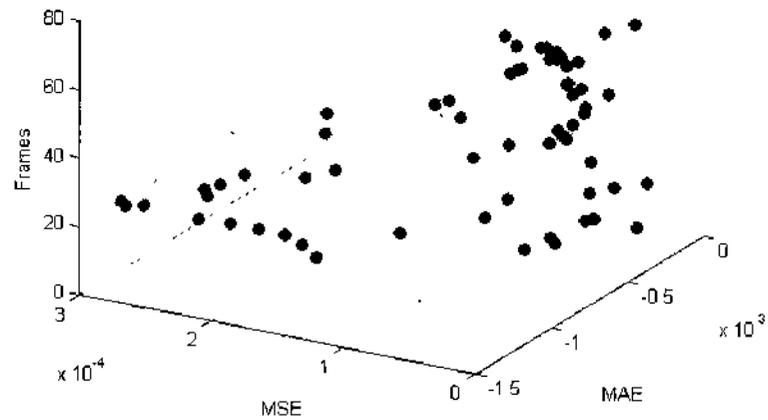
To the best of author’s knowledge there is no prior published work on a trajectory perturbation based data hiding method. Moreover, perceptual and especially statistical invisibility are not considered as constraints for most of the prior motion vector based data hiding algorithms. Instead, those methods are focused primarily on capacity and robustness design criteria. Because of this, it is difficult to do a fair comparison of the performance of the proposed method and other methods with respect to perceptual and statistical invisibility.

As emphasized earlier, the focus in this work is on designing an algorithm satisfying invisibility constraints while at the same time providing moderate rate data hiding capacity.

The most relevant existing application is the watermarking of motion vectors. The proposed method is different than motion vector based watermarking algorithms in several aspects. Despite the motion vector based digital video watermarking algorithms, the proposed method is not a block based one that utilizes motion vectors computed via “Motion Estimation” algorithms. Instead, arbitrary VO centroid coordinates, obtained by using a tracking algorithm, are used to convey the message. More importantly, prior motion vector based watermarking algorithms have not taken



(a) Pixel Wise Distortion Based Steganalysis Results



(b) Histogram Based Steganalysis Results

FIG. 64: Pixel Wise and Histogram Based Steganalysis Results of “Soccer Ball” VO Motion Magnitude and Angle Perturbation Based Data Hiding.

into consideration the semantic meaning of the object motion which result in degradation of the perceptual invisibility.

In a more recent paper by Aly [81], a method which employs motion vector LSB value modification was proposed. In this method, adaptive prediction error thresholds are used as the mechanism to select and control the motion vector modification where only one bit of information is sent per motion vector by changing the LSBs of the selected motion vectors. Although PSNR values were provided to prove that the perceptual quality of the frames have been preserved after modification, nothing has been included on whether the method actually preserves the semantic meaning in the video frame or not. In that respect, it is arguable that the designer can modify the motion vectors arbitrarily but rather the motion semantics should play a role in the selection of a subset of motion vectors for modification. Also, the author did not discuss the performance of the method against noise, illumination change and/or filtering, all of which could impact the performance of a motion estimation algorithm.

In another work by Fang et al. [82], a motion vector modification based data hiding method was proposed. In this method, the phase of the motion is quantized by using a circular quantizer. The drawback of this method is again not taking into consideration the semantic meaning object motion. Arbitrary modification of the motion phase using a multi-level circular quantization scheme will result in abrupt changes in the motion trajectory of an object e.g., VO having a linear motion will have sudden and meaningless motion direction change because of codeword-motion vector assignments.

VI.3 CHAPTER SUMMARY

This chapter presents the experimental results of the proposed algorithm used to embed binary data into two types of sports videos. To set up the polar quantizer parameters, first the motion magnitude distribution is analyzed. Based on the analysis, the quantization cell parameters are determined. Future work could be well invested on automatic selection of these parameters obtained by a classifier trained on different motion models commonly encountered in sports videos. Results of the experiments for varying noise variance have shown that as the noise variance increases BER also increases. For large noise variances, cases in which the perceptual visual quality of the video frames are completely deteriorated, the tracking error results are so large that it is not possible obtain reliable results.

CHAPTER VII

CONCLUSIONS AND FUTURE PERSPECTIVES

This thesis presents a novel data hiding method based on a simple idea of sending the information to the receiver through perturbations of a VO trajectory. To the best of author's knowledge there is no similar work in existence as upon completion of this thesis.

Implementation of the proposed algorithm covers multiple areas from different disciplines such as digital image (morphological operations and inpainting), video (tracking, video frame processing basics), signal processing (quantization) and digital communications (modulation and signal constellation analysis).

From the subjective assessment of the experimental results, it has been assessed that the perceptual qualities of the original cover and stego frames are almost identical with almost no visual artifacts. The statistical visibility tests have also been conducted by employing some commonly used statistical measures on the data embedded frames and found that the proposed algorithm did not leave any statistical signature after the embedding process. Hence, from the statistical invisibility and imperceptibility perspectives, the proposed method showed promising results.

Overall Contributions :

The main contributions of this thesis are the following.

1. The proposed method is the first in the literature in terms of VO trajectory perturbation based data hiding.
2. Existing spatial, temporal and transform domain data embedding schemes embed data into most of the cover media features to maximize the capacity. This approach causes the inherent problem of the trade-off between steganographic security and data hiding capacity since most of the steganalysis techniques use statistical measures in detecting the existence of hidden data. Based on the findings, it is concluded that the proposed method could provide both statistical and perceptual invisibility.
3. Kundur et al. have shown in their work on motion coherent watermarking [55] that the capacity of a video data hiding scheme is actually smaller than the general approach that considers the video frames as still images. If the

individual frames are considered as still images, the visual artifacts may occur in areas of static background. Therefore, they suggested that motion information should be considered to decrease the artifacts and increase statistical invisibility. Based on their research it can be concluded that both statistical and perceptual invisibility must be the first priority of the designer. In that respect, the strength of this method is its feature of providing both perceptual and statistical invisibility.

4. From the embedding capacity perspective, this method could be used for low and moderate rate steganography. It can also be used with other existing data hiding methods to increase the overall embedding data rate.
5. A new expression of the symbol error probability of 2-D non-uniform signal constellations is provided by using two approximations.

VII.1 FUTURE DIRECTIONS

The method in this thesis is provided as a proof-of-concept. It is the first implementation of a semantic data hiding method in the literature which aims at establishing a covert channel between the sender and the decoder. Improvement of the proposed method is possible in different areas as listed below.

VII.1.1 Near Term Focus

This algorithm may be implemented using more sophisticated trajectory estimation methods which take into consideration the ball motion phases (i.e. rolling, flying, in possession etc.) resulting in fully automatic methods. This is a promising scope for building more complex algorithms for data hiding in sports videos.

The proposed method is demonstrated for table tennis and soccer videos. This method could be investigated for extension to other sports videos such as basketball, golf etc. for which the tracking algorithm requirements are different.

VII.1.2 Long Term Focus

As the technology move towards 3D video standards, a 3D implementation of the proposed method could also be considered as a possible research topic.

Robustness of a tracking algorithm against illumination changes in the video frames is still an active research field. Considering the volumetric attacks aiming at changing the illumination, the proposed method could be improved by incorporation of a tracking algorithm such as the one proposed in [83, 84] to mitigate effects of such attacks.

The accuracy of a tracking algorithm is essential to increasing the performance of this method. More robust methods that can deal with segmentation and tracking of non-rigid objects such as [85] could well be investigated to improve the performance of the proposed algorithm.

Incorporation of error control coding could be another area that needs further investigation.

In order to incorporate the proposed method into an object based video coding standard, such as MPEG-4, a motion vector based object tracking algorithm could be implemented. In this case, after motion estimation, a global common motion vector should be computed for the pixels belonging to the same segmented object. This approach would give the designer the opportunity to shift the pixel locations, which will be computed by embedding quantizer, to non-integer pixel locations i.e. $1/2, 1/4$ and $1/8$. This could be a very promising future work in the field of motion based data hiding which will help increase the capacity.

Another solution worth investigating is to frame-by-frame coordinate quantization while preserving the semantic meaning of the object motion. In this case, individual x and y coordinates of the centroid coordinates are quantized using QIM techniques or a reversible transform such as Difference Expansion.

And finally the proposed method operates on consecutive frames to embed the data through the perturbations of video object trajectories. As discussed earlier under different chapters, what happens when intentional or unintentional frame dropping occurs? One solution is to resend the data based upon an ACK (Acknowledgement) signal from the decoder.

REFERENCES

- [1] Petitcolas, F., “LSB hiding examples,” Online Resource www.petitcolas.net/fabien.
- [2] Fridrich, J., “Applications of data hiding in digital images,” Online Resource www.ws.binghamton.edu/fridrich.
- [3] Moore, B., Takahara, G., and Alajaji, F., “Pairwise optimization of modulation constellations,” in [*Communications, Computers and Signal Processing, 2009. PacRim 2009. IEEE Pacific Rim Conference on*], 181–186 (Aug. 2009).
- [4] Yilmaz, A., Javed, O., and Shah, M., “Object tracking: A survey,” *ACM Comput. Surv.* **38**(4), 13 (2006).
- [5] Maggio, E., Smerladi, F., and Cavallaro, A., “Adaptive multifeature tracking in a particle filtering framework,” *Circuits and Systems for Video Technology, IEEE Transactions on* **17**(10), 1348–1359 (2007).
- [6] Ren, J., Xu, M., Orwell, J., and Jones, G., “Real-time modeling of 3-d soccer ball trajectories from multiple fixed cameras,” *Circuits and Systems for Video Technology, IEEE Transactions on* **18**, 350–362 (March 2008).
- [7] Bertalmio, M., Vese, L., Sapiro, G., and Osher, S., “Simultaneous structure and texture image inpainting,” in [*Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*], **2**, II 707–12 vol.2 (June 2003).
- [8] Ourique, F., Licks, V., Jordan, R., and Perez-Gonzalez, F., “Angle qim: a novel watermark embedding scheme robust against amplitude scaling distortions,” in [*Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*], **2**, ii/797–ii/800 Vol. 2 (March 2005).
- [9] Cay, A., R. Z. and Popescu, D., “Video object trajectory perturbation based data hiding satisfying statistical and perceptual invisibility,” in [*45th Annual Conference on Information Sciences and Systems (CISS 2011)*], (March 2011).
- [10] Moulin, P., “The role of information theory in watermarking and its application to image watermarking,” *Signal Processing* **81**(6), 1121–1139 (2001).

- [11] Wu, M., *Multimedia Data Hiding*, PhD Thesis, Princeton University, Princeton, NJ (June 2001).
- [12] Cox, I., Kilian, J., Leighton, F., and Shamoon, T., “Secure spread spectrum watermarking for multimedia,” *Image Processing, IEEE Transactions on* **6**, 1673–1687 (December 1997).
- [13] Pérez-Freire, L. and Pérez-González, F., “Spread-spectrum watermarking security,” *Information Forensics and Security, IEEE Transactions on* **4**, 2–24 (March 2009).
- [14] Ramkumar, M. and Akansu, A., “Signaling methods for multimedia steganography,” *Signal Processing, IEEE Transactions on* **52**, 1100–1111 (Apr. 2004).
- [15] Chen, B. and Wornell, G., “Quantization index modulation: a class of provably good methods for digital watermarking and information embedding,” *Information Theory, IEEE Transactions on* **47**, 1423–1443 (May 2001).
- [16] Wu, H.-C., Wu, N.-I., Tsai, C.-S., and Hwang, M.-S., “Image steganographic scheme based on pixel-value differencing and lsb replacement methods,” *Vision, Image and Signal Processing, IEE Proceedings -* **152**, 611–615 (October September).
- [17] Yang, C.-H., Weng, C.-Y., Wang, S.-J., and Sun, H.-M., “Adaptive data hiding in edge areas of images with spatial lsb domain systems,” *Information Forensics and Security, IEEE Transactions on* **3**, 488–497 (September 2008).
- [18] Mielikainen, J., “LSB matching revisited,” *Signal Processing Letters, IEEE* **13**, 285–287 (May 2006).
- [19] Luo, W., Huang, F., and Huang, J., “Edge adaptive image steganography based on LSB matching revisited,” *Information Forensics and Security, IEEE Transactions on* **5**, 1–1 (Jun 2010).
- [20] Sencar, H. T., Ramkumar, M., and Akansu, A. N., “An overview of scalar quantization based data hiding methods,” *Signal Process.* **86**(5), 893–914 (2006).
- [21] Barni, M., Bartolini, F., and Checcacci, N., “Watermarking of mpeg-4 video objects,” *Multimedia, IEEE Transactions on* **7**, 23–32 (Feb. 2005).

- [22] Koz, A. and Alatan, A., “Oblivious spatio-temporal watermarking of digital video by exploiting the human visual system,” *Circuits and Systems for Video Technology, IEEE Transactions on* **18**, 326 –337 (March 2008).
- [23] Steven, S., “Motion sensitive video watermarking,” Tech. Rep. 825, NAT.LAB. (Aug. 2001).
- [24] Mihcak, M., *Information Hiding Codes and Their Applications to Images and Audio*, PhD Thesis, Univesity of Illinois at Urbana-Champaign, Urbana, Illinois (2002).
- [25] Moulin, P. and Mihak, M. K., “The parallel-gaussian watermarking game,” *IEEE Trans.on Information Theory* **50**, 272-289 (2000).
- [26] Cohen, A. and Lapidoth, A., “On the gaussian watermarking game,” *IEEE Trans. Inform. Theory* **48**, 1639 -1667 (2000).
- [27] Jeng-Shyang Pan, Hsiang-Cheh Huang, L. C., [*Intelligent Watermarking Techniques*], World Scientific Publishing, Tuck Link, Singapore, 1st ed. (2004).
- [28] Cox, I. J., Miller, M. L., and Bloom, J. A., “Watermarking applications and their properties,” in [*International Conference on Information Technology: Coding and Computing*], 6–10 (2000).
- [29] Anderson, R., ed., [*Information Hiding:First International Workshop*], Springer-Verlag, Berlin, Germany (1996).
- [30] Petitcolas, F. A. P., Anderson, R. J., and Kuhn, M. G., “Information hiding – a survey,” *Proceedings of the IEEE* **87**, 1062–1078 (July 1999).
- [31] Langelaar, G., Setyawan, I., and Lagendijk, R., “Watermarking digital image and video data. a state-of-the-art overview,” *Signal Processing Magazine, IEEE* **17**, 20 – 46 (Sep. 2000).
- [32] Alattar, A., Lin, E., and Celik, M., “Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video,” *Circuits and Systems for Video Technology, IEEE Transactions on* **13**, 787 – 800 (Aug. 2003).
- [33] Hartung, F. and Girod, B., “Watermarking of uncompressed and compressed video,” *Signal Processing* **66**(3), 283 – 301 (1998).

- [34] Langelaar, G. and Lagendijk, “Optimal differential energy watermarking of dct encoded images and video,” *Image Processing, IEEE Transactions on* **10**, 148–158 (Jan. 2001).
- [35] Chen, S. and Leung, H., “A temporal approach for improving intra-frame concealment performance in H.264/AVC,” *Circuits and Systems for Video Technology, IEEE Transactions on* **19**, 422–426 (March 2009).
- [36] Wong, K. S., Tanaka, K., Takagi, K., and Nakajima, Y., “Complete video quality-preserving data hiding,” *Circuits and Systems for Video Technology, IEEE Transactions on* **19**, 1499–1512 (Oct. 2009).
- [37] Swanson, M., Zhu, B., and Tewfik, A., “Multiresolution scene-based video watermarking using perceptual models,” *Selected Areas in Communications, IEEE Journal on* **16**, 540–550 (May 1998).
- [38] Mukherjee, D., Chae, J. J., Mitra, S. K., and Manjuneath, B. S., “A source and channel coding framework for vector based data hiding in video,” *Circuits and Systems for Video Technology, IEEE Transactions on* **10**, 630–645 (June 2000).
- [39] Wang, P., Zheng, Z., and Ying, J., “A novel video watermark technique in motion vectors,” in [*Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*], 1555–1559 (July 2008).
- [40] Kezheng, L., Wei, Y., and Pie, L., “Video watermarking temporal synchronization on motion vector,” in [*Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*], **1**, 1105–1110 (Nov. 2008).
- [41] Liu, Z., Liang, H., Niu, X., and YixianYang, “A robust video watermarking in motion vectors,” in [*Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*], **3**, 2358–2361 vol.3 (Aug. 2004).
- [42] Bodo, Y., Laurent, N., and Dugelay, J.-L., “Watermarking video, hierarchical embedding in motion vectors,” in [*Image Processing, 2003. ICIIP 2003. Proceedings. 2003 International Conference on*], **2**, II 739–42 vol.3 (Sept. 2003).
- [43] Zhang, C. and Su, Y., “Video steganalysis based on aliasing detection,” *Electronics Letters* **44**, 801–803 (June 2008).

- [44] Mohaghegh, N. and Fatemi, O., "H.264 copyright protection with motion vector watermarking," in [*Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*], 1384 –1389 (July 2008).
- [45] Deguillaume, F., Csurka, G., O’Ruanaidh, J. J., and Pun, T., "Robust 3D DFT video watermarking," *Security and Watermarking of Multimedia Contents* **3657**(1), SPIE (1999).
- [46] Liu, Y. and Zhao, J., "Rst invariant video watermarking based on 1d dft and radon transform," in [*Visual Information Engineering, 2008. VIE 2008. 5th International Conference on*], 443 –448 (Aug. 2008).
- [47] Avcibas, I., Memon, N., and Sankur, B., "Steganalysis using image quality metrics," *Image Processing, IEEE Transactions on* **12**, 221 – 229 (Feb. 2003).
- [48] Gul, G. and Kurugollu, F., "Svd-based universal spatial domain image steganalysis," *Information Forensics and Security, IEEE Transactions on* **5**, 349 –353 (June 2010).
- [49] Pevny, T., Bas, P., and Fridrich, J., "Steganalysis by subtractive pixel adjacency matrix," *Information Forensics and Security, IEEE Transactions on* **5**, 215 –224 (June 2010).
- [50] Yang, C., Liu, F., Luo, X., and Liu, B., "Steganalysis frameworks of embedding in multiple least-significant bits," *Information Forensics and Security, IEEE Transactions on* **3**, 662 –672 (Dec. 2008).
- [51] Dumitrescu, S. and Wu, X., "A new framework of lsb steganalysis of digital media," *Signal Processing, IEEE Transactions on* **53**, 3936 – 3947 (Oct. 2005).
- [52] Ker, A., "Steganalysis of lsb matching in grayscale images," *Signal Processing Letters, IEEE* **12**, 441 – 444 (June 2005).
- [53] Gul, G., Dirik, A., and Avcibas, I., "Steganalytic features for jpeg compression-based perturbed quantization," *Signal Processing Letters* **14**, 205 - 208 (March 2007).
- [54] Li, B., Huang, J., and Shi, Y. Q., "Steganalysis of yass," *Information Forensics and Security, IEEE Transactions on* **4**, 369 –382 (Sept. 2009).

- [55] Budhia, U., Kundur, D., and Zourntos, T., "Digital video steganalysis exploiting statistical visibility in the temporal domain," *Information Forensics and Security, IEEE Transactions on* **1**, 502 –516 (Dec. 2006).
- [56] Pankajakshan, V., Doerr, G., and Bora, P., "Detection of motion-incoherent components in video streams," *Information Forensics and Security, IEEE Transactions on* **4**, 49 – 58 (March 2009).
- [57] Doerr, G. and Dugelay, J.-L., "Security pitfalls of frame-by-frame approaches to video watermarking," *Signal Processing, IEEE Transactions on* **52**, 2955 – 2964 (Oct. 2004).
- [58] Su, K., Kundur, D., and Hatzinakos, D., "Statistical invisibility for collusion-resistant digital video watermarking," *Multimedia, IEEE Transactions on* **7**, 43 – 51 (Feb. 2005).
- [59] Voran, S. and Scharf, L., "Polar coordinate quantizers that minimize mean-squared error," *Signal Processing, IEEE Transactions on* **42(6)**, 1559 –1563 (1994).
- [60] Peric, Z., Djordjevic, I., Bogosavljevic, S., and Stefanovic, M., "Design of signal constellations for gaussian channel by using iterative polar quantization," in [*Electrotechnical Conference, 1998. MELECON 98., 9th Mediterranean*], **2**, 866 –869 (May 1998).
- [61] Ibnkahla, M., [*Signal Processing for mobile communications handbook*], CRC Press, Boca Raton,FL, 1st ed. (2005).
- [62] Babu, R., Ramakrishnan, K., and Srinivasan, S., "Video object segmentation: a compressed domain approach," *Circuits and Systems for Video Technology, IEEE Transactions on* **14**, 462 – 474 (April 2004).
- [63] Erdem, C., Sankur, B., and Tekalp, A., "Performance measures for video object segmentation and tracking," *Image Processing, IEEE Transactions on* **13**, 937 –951 (July 2004).
- [64] Meier, T. and Ngan, K., "Segmentation and tracking of moving objects for content-based video coding," *Vision, Image and Signal Processing, IEE Proceedings -* **146**, 144 –150 (June 1999).

- [65] Tsai, Y.-P., Lai, C.-C., Hung, Y.-P., and Shih, Z.-C., “A bayesian approach to video object segmentation via merging 3-d watershed volumes,” *Circuits and Systems for Video Technology, IEEE Transactions on* **15**, 175 – 180 (jan. 2005).
- [66] Comaniciu, D. and Ramesh, V., “Mean shift and optimal prediction for efficient object tracking,” in [*Image Processing, 2000. Proceedings. 2000 International Conference on*], (2000).
- [67] Collins, R., “Mean-shift blob tracking through scale space,” in [*Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*], **2**, II – 234–40 (2003).
- [68] Mihaylova, L. and Boel, R., “A particle filter for freeway traffic estimation,” in [*Decision and Control, 2004. CDC. 43rd IEEE Conference on*], **2**, 2106 – 2111 (Dec. 2004).
- [69] Zhou, S. K., Chellappa, R., and Moghaddam, B., “Visual tracking and recognition using appearance-adaptive models in particle filters,” *Image Processing, IEEE Transactions on* **13**(11), 1491 –1506 (2004).
- [70] Rathi, Y., Vaswani, N., Tannenbaum, A., and Yezzi, A., “Tracking deforming objects using particle filtering for geometric active contours,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **29**, 1470 –1475 (Aug. 2007).
- [71] Maggio, E., Smerladi, F., and Cavallaro, A., “Adaptive multifeature tracking in a particle filtering framework,” *Circuits and Systems for Video Technology, IEEE Transactions on* **17**(10), 1348 –1359 (2007).
- [72] Huang, C.-L., Shih, H.-C., and Chen, C.-L., “Shot and scoring events identification of basketball videos,” in [*Multimedia and Expo, 2006 IEEE International Conference on*], 1885 –1888 (2006).
- [73] Chu, W.-T., Wang, C.-W., and Wu, J.-L., “Extraction of baseball trajectory and physics-based validation for single-view baseball video sequences,” in [*Multimedia and Expo, 2006 IEEE International Conference on*], 1813 –1816 (2006).
- [74] Yu, X., Leong, H., Xu, C., and Tian, Q., “Trajectory-based ball detection and tracking in broadcast soccer video,” *Multimedia, IEEE Transactions on* **8**, 1164 –1178 (Dec. 2006).

- [75] Kim, J.-Y. and Kim, T.-Y., “Soccer ball tracking using dynamic kalman filter with velocity control,” in [*Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on*], 367 –374 (Aug. 2009).
- [76] Wong, K. and Dooley, L., “High-motion table tennis ball tracking for umpiring applications,” in [*Signal Processing (ICSP), 2010 IEEE 10th International Conference on*], 2460 –2463 (2010).
- [77] Sarkar, A., Nataraj, L., ManJuneath, B., and Madhow, U., “Estimation of optimum coding redundancy and frequency domain analysis of attacks for yass - a randomized block based hiding scheme,” in [*Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*], 1292 –1295 (2008).
- [78] Bertalmio, M., Vese, L., Sapiro, G., and Osher, S., “Simultaneous structure and texture image inpainting,” *Image Processing, IEEE Transactions on* **12**, 882 – 889 (Aug. 2003).
- [79] Bhat, S., “Object removal by exemplar-based inpainting,” Online Resource www.cc.gatech.edu/~sooraj/inpainting.
- [80] Porikli, F., T. O. and Meer, P., “Covariance tracking using model update based on means on riemannian manifolds,” in [*Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Computer Society Conference*], **1**, 728 –735 (2006).
- [81] Aly, H., “Data hiding in motion vectors of compressed video based on their associated prediction error,” *Information Forensics and Security, IEEE Transactions on* **6**, 14 – 18 (March 2011).
- [82] Fang, D.-Y. and Chang, L.-W., “Data hiding for digital video with phase of motion vector,” in [*Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*], 1422 –1425 (2006).
- [83] Cogun, F. and Cetin, A., “Object tracking under illumination variations using 2d-cepstrum characteristics of the target,” in [*Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*], 521 –526 (2010).
- [84] Porikli, F., “Achieving real-time object detection and tracking under extreme conditions,” in [*Real Time Image Processing, Journal of*], **1**(1), 33 –40 (2006).

- [85] Erdem, C., Tekalp, A., and Sankur, B., "Video object tracking with feedback of performance measures," *Circuits and Systems for Video Technology, IEEE Transactions on* (April 2003).

APPENDIX A

Q FUNCTION

The Q function is the tail probability of the standard normal distribution. Fig.65 shows the plot of the Q function whose definition is given as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{u^2}{2}} du \quad (68)$$

$Q(x) = 1 - Q(-x) = 1 - \Phi(x)$ where $\Phi(x)$ is the cumulative distribution of the Normal distribution.

The Q function can also be expressed in terms of the error function as

$$Q(x) = \frac{1}{2} - \frac{1}{2} \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$$

where complementary error function erfc is defined as

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \quad (69)$$

It is also easy to show that $\operatorname{erf}(x) = 1 - 2Q(\sqrt{2}x)$ and $\operatorname{erfc}(x) = 2Q(\sqrt{2}x)$

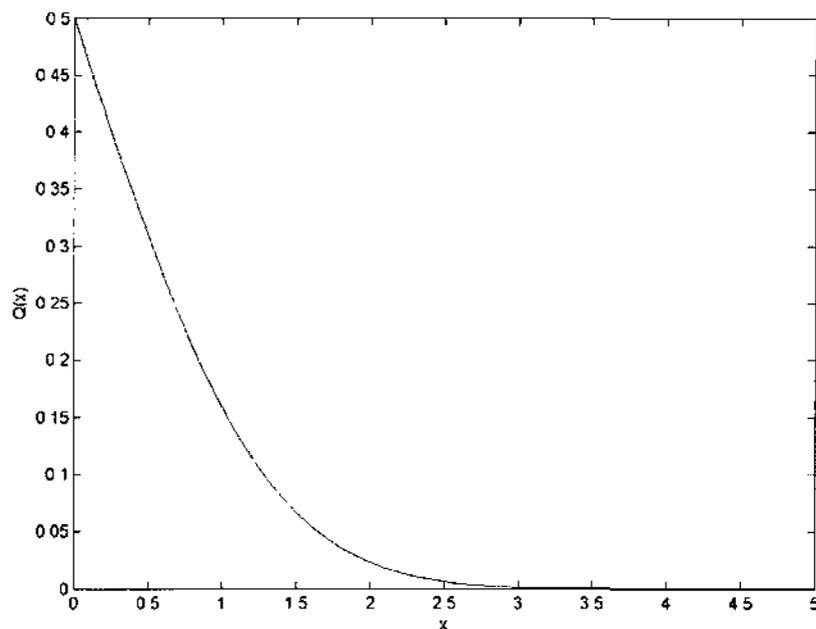


FIG. 65: Q function.

APPENDIX B

STEGANALYSIS

Information hiding in digital media results in modifications of the cover signal properties, introducing degradations which yield information leakage about the hidden data. Therefore, steganalysis of the suspected data aims at detecting and/or estimating potentially hidden information from the observed stego data with little or no knowledge about the underlying steganographic algorithm. The first step of steganalysis is feature extraction where a set of distinguishing statistics are obtained from the stego and cover data. The second step is the classification of the input data by a trained classifier (e.g., Support Vector Machine (SVM) classifier) as either being clear or carrying a hidden message.

Avcibas et al. in [47] propose a set of image quality metrics as the feature set for steganalyzer design. Brief descriptions of the metrics that are used for steganalysis are provided next.

B.1 MEASURES BASED ON PIXEL DIFFERENCES

These measures calculate the total distortion between cover and stego frames based on pixel-wise differences between the two or certain moments of the difference frame.

Let $C_k(i, j)$ represents the pixel value at (i, j) in band k^{th} of the frame. Note when RGB color space is used $k = 1, 2, 3$ and the pixel values can be in the range $\{0, \dots, 255\}$ at each band. Let $S_k(i, j)$ represent the k band stego frame, $M_i, i = 1, 2, \dots, 10$ represent the features used in the steganalyzer and finally ϵ_k denotes the error over all pixels. Specifically $\epsilon_k = C_k(i, j) - S_k(i, j) = \sum_{k=1}^K [C_k(i, j) - S_k(i, j)]^2$ will denote the sum of errors in each band at pixel (i, j) .

B.1.1 Minkowsky Measures:

The L_γ norm of the dissimilarity of two frames (which can be considered as still images) can be calculated by taking the Minkowsky average of the pixel differences spatially and then over the bands as

$$M_\gamma = \frac{1}{K} \sum_{k=1}^K \left\{ \frac{1}{N^2} \sum_{i,j=1}^N [C_k(i, j) - S_k(i, j)]^\gamma \right\}^{\frac{1}{\gamma}} \quad (70)$$

$\gamma = 1$ corresponds to mean absolute error M_1 and $\gamma = 2$ is the mean square error (MSE), M_2 respectively.

B.1.2 Correlation Based Measures:

The closeness between two frames can also be measured in terms of correlation function. The Image Fidelity and Normalized Cross-Correlation measures the similarity between two frames. If the difference between the two frames approaches zero, the correlation based measures tend to go to one. Some commonly used correlation based measures are image fidelity and normalized-cross correlation described below.

$$\text{Image Fidelity} = M_3 = 1 - \left[\frac{1}{K} \sum_{k=1}^K \frac{\sum_{i,j=0}^{N-1} [C_k(i,j) - S_k(i,j)]^2}{\sum_{i,j=0}^{N-1} C_k(i,j)^2} \right] \quad (71)$$

$$\text{Normalized Cross - Correlation} = M_4 = \frac{1}{K} \frac{\sum_{i,j=0}^{N-1} C_k(i,j) S_k(i,j)}{\sum_{i,j=0}^{N-1} C_k(i,j)^2} \quad (72)$$

B.2 SPECTRAL MEASURES

In this category we consider the distortion penalty functions obtained from the complex Fourier spectrum of the frames [47]. Let Discrete Fourier Transforms (DFT) of the original and data embedded (stego) frames be denoted by $\Gamma_k(u, v)$ and $\hat{\Gamma}_k(u, v)$ respectively. The 2D-DFT is defined as:

$$\Gamma_k(u, v) = \sum_{m,n=0}^{N-1} C_k(m, n) \exp \left[-2\pi i m \frac{u}{N} \right] \exp \left[-2\pi i n \frac{v}{N} \right] \quad (73)$$

where $k = 1, 2, \dots, K$. The phase and magnitude of the DFT are defined by $\phi(u, v) = \arctan(\Gamma_k(u, v))$ and $F(u, v) = |\Gamma(u, v)|$ respectively. The spectral magnitude distortion is given by

$$M_5 = \frac{1}{N^2} \sum_{m,n=0}^{N-1} \left| F(u, v) - \hat{F}(u, v) \right|^2 \quad (74)$$

B.3 PERCEPTUAL MEASURES

B.3.1 Peak Signal-to-Noise Ratio (PSNR):

$$M_6 = 10 \log_{10} \left(\frac{[\max(C(i, j))]^2}{MSE} \right) \quad (75)$$

where $\max(C(i, j))$ is the maximum pixel value and MSE is the mean squared error that can be found using metric M_2 .

B.4 HISTOGRAM MEASURES

B.4.1 K-L Divergence:

$$M_7 = \sum_i P_i \log \frac{P_i}{\hat{P}_i} = \sum_i \frac{H_o(i)}{N_{total}} \log \frac{H_o(i)}{H_m(i)} \quad (76)$$

where H_m and H_o represents the modified (stego) and original frame histogram respectively.

B.4.2 χ^2 (Chi-Square) Metric:

$$M_8 = 0 \leq \chi^2(H_m, H_o) = \frac{\sum_{j=1}^B \frac{[H_m(j) - H_o(j)]^2}{H_m(j) + H_o(j)}}{N_{H_o} + N_{H_m}} \leq 1 \quad (77)$$

where H_m , H_o and B represent the modified (stego), original frame histograms and number of bins respectively. Also $N_{H_o} = \sum_{j=1}^B H_o(j)$ and $N_{H_m} = \sum_{j=1}^B H_m(j)$.

VITA

Abdullah Cay

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

Abdullah Cay was born in Manisa, Turkey on November 4, 1969. He received his B.S. degree in Electrical Engineering from Turkish Army Academy in 1992 and M.S. degree in Electrical and Computer engineering from US Naval Postgraduate School, Monterey, CA in 1999 respectively. His thesis was focused on Connection Utilization Masking in ATM networks. His current research interests include information hiding, radar signal processing, image and video signal processing.