Compressive Sensing for Target Detection and Tracking within Wireless Visual Sensor Networks-based Surveillance applications

Salema Fathy Fayed

A thesis submitted in partial fulfilment of the requirement of Staffordshire University for the degree of Doctor of Philosophy

April 2016

Acknowledgement

I feel indebted to many people, so many people I would like to thank, all those who shared either morally or practically in making this possible and an enjoyable experience for me.

I would like to express my deepest sense of gratitude and deep respect to my supervisors Prof Mansour Moniri and Dr. Mohammad Patwary, Staffordshire University, for their valuable support, patient guidance, immense knowledge and great advice. Above all and the most needed, they provided me unflinching encouragement, support in various ways and helped whenever I was in need throughout all the stages of this thesis. Their support also included teaching me the professional way of thinking, providing me with necessary references, as well as giving me extraordinary experiences throughout the work. In addition to their advice and their willingness to share their bright thoughts with me, those were very fruitful for shaping up my ideas and research. Furthermore, their honest and precise review were very helpful to trace points of weakness in the thesis and strengthen them. Without their help, this work would not be possible. Words cannot describe how grateful I am. I think their presence was the best thing that could happen. I am grateful in every possible way and hope to keep up this collaboration in the future.

I would like to pay my gratefulness and respect to my supervision team in Egypt Prof. Sherin Youssef and Dr. Amr EL-Helw for their continuous supports and help in various ways, patience and valuable suggestions throughout the years and during all stages of my PhD. Their guidance, motivation and knowledge helped me in all the time of research. Their advice, concern and patience are really appreciated. I would like to thank them for encouraging my research, they have been a tremendous mentor for me. Their astounding perspective and thorough criticisms were of great value.

I also wish to express my appreciation and thanks to Dr. Mohammad Patwary and Dr. Abdel Hamid Soliman for making my visits to Stafford really enjoyable, for keeping me going when times were tough, for being totally supportive in all times and above all making me feel home. They are my family in the UK.

The financial support provided by the Arab Academy for Science, Technology and Maritime Transport to carry out this research work is gratefully acknowledged.

Words fail me to express my appreciation and denote my sincere thanks to my family members who gave me spiritual continuous support and encouragement to complete this thesis successfully, their prayers for me was what sustained me thus far and their love and support has taken the load off my shoulder. Finally, I would like to thank everybody who was important to the realization of this thesis. ...

List of Publications

Salema Fayed, Sherin Youssef, Amr El-Helw, Mohammad Patwary, and Mansour Moniri. Adaptive compressive sensing for target tracking within wireless visual sensor networks-based surveillance applications. Multimedia tools and applications, Springer, pages 1-25, 2015.

Salema Fayed, Sherin Youssef, Amr El-Helw, Mohammad Patwary, and Mansour Moniri. A hybrid adaptive compressive sensing model for visual tracking in wireless visual sensor networks. International Journal of circuits, systems and signal processing, vol 8, 399-409, 2014.

Salema Fayed, Sherin Youssef, Amr El-Helw, Mohammad Patwary, and Mansour Moniri. Compressive sensing-based target tracking for wireless vi- sual sensor networks. In 18th international conference on Circuits, Systems, Communications and Computers (CSCC 2014), vol 1, pages 44-50, Santorini,Greece, July 2014.

Salema Fayed, Sherin Youssef, Amr El-Helw, Mohammad Patwary, and Mansour Moniri. Comparative analysis on the competitiveness of conventional and compressive sensing-based query processing. In 18th international conference on Circuits, Systems, Communications and Computers (CSCC2014), vol 1, pages 240-245, Santorini,Greece, July 2014.

Submitted

Salema Fayed, Sherin Youssef, Amr El-Helw, Mohammad Patwary, and Mansour Moniri. Analytical framework for Adaptive Compressive Sensing for Target Detection within Wireless Visual Sensor Networks. IEEE transactions on mobile computing

Abstract

Wireless Visual Sensor Networks (WVSNs) have gained significant importance in the last few years and have emerged in several distinctive applications. The main aim of this research is to investigate the use of adaptive Compressive Sensing (CS) for efficient target detection and tracking in WVSN-based surveillance applications. CS is expected to overcome the WVSN resource constraints such as memory limitation, communication bandwidth and battery constraints. In addition, adaptive CS dynamically chooses variable compression rates according to different data sets to represent captured images in an efficient way hence saving energy and memory space. In this work, a literature review on compressive sensing, target detection and tracking for WVSN is carried out to investigate existing techniques. Only single view target tracking is considered to keep minimum number of visual sensor nodes in a wake-up state to optimize the use of nodes and save battery life which is limited in WVSNs. To reduce the size of captured images an adaptive block CS technique is proposed and implemented to compress the high volume data images before being transmitted through the wireless channel. The proposed technique divides the image to blocks and adaptively chooses the compression rate for relative blocks containing the target according to the sparsity nature of images. At the receiver side, the compressed image is then reconstructed and target detection and tracking are performed to investigate the effect of CS on the tracking performance. Least mean square adaptive filter is used to predicts target's next location, an iterative quantized clipped LMS technique is proposed and compared with other variants of LMS and results have shown that it achieved lower error rates than other variants of lMS. The tracking is performed in both indoor and outdoor environments for single/multi targets. Results have shown that with adaptive block compressive sensing (CS) up to 31% measurements of data are required to be transmitted for less sparse images and 15% for more sparse, while preserving the $33dB$ image quality and the required detection and tracking performance. Adaptive CS resulted in 82% energy saving as compared to transmitting the required image with no CS.

Contents

List of Publications iii

Abstract iv

List of Figures viii

List of Tables x

Abbreviations xi

Symbols xii

1 Introduction 1
- 1.1 Context of investigations . . . 1
- 1.2 Visual Surveillance requirements and Applications . . . 5
- 1.3 Aim and objectives . . . 8
- 1.4 Thesis organization . . . 9

2 Literature review 11
- 2.1 Introduction . . . 11
- 2.2 WVSNs . . . 11
  - 2.2.1 Target detection . . . 13
  - 2.2.2 Target tracking . . . 17
- 2.3 Compressive Sensing . . . 22
- 2.4 Chapter summary . . . 26

3 Theoretical aspects of the proposed investigations 28
- 3.1 Introduction . . . 28
- 3.2 General WVSN model . . . 29
  - 3.2.1 Characteristics of WVSNs . . . 29
    - 3.2.1.1 Energy, storage and bandwidth constraint . . . 30
    - 3.2.1.2 Local processing . . . 31
    - 3.2.1.3 Scalability . . . 32
    - 3.2.1.4 Self-configuration . . . 32

3.3 Target detection . . . 33
3.3.1 Non-recursive techniques . . . 34
3.3.2 Recursive techniques . . . 36
3.4 Target tracking . . . 40
3.5 Compressive Sensing . . . 42
3.5.1 Introduction to CS . . . 43
3.5.2 Theoretical basis of CS . . . 44
3.5.2.1 Properties of random measurement matrix $\mathbf{\Phi}$ . . . 45
3.5.3 Image reconstruction . . . 47
3.6 Chapter summary . . . 48

4 Proposed detection and tracking model using CS 50
4.1 Introduction . . . 50
4.2 Proposed system model . . . 51
4.2.1 Energy model . . . 53
4.3 Proposed detection and tracking model . . . 54
4.3.1 Proposed detection technique . . . 55
4.3.1.1 Background subtraction . . . 55
4.3.1.2 Morphology operations and blob extraction . . . 56
4.3.2 Proposed adaptive block Compressive sensing . . . 59
4.3.2.1 Proposed Adaptive CS . . . 62
4.3.2.2 Proposed block CS . . . 64
4.3.3 Proposed tracking model . . . 67
4.3.3.1 Least mean square (LMS) . . . 67
4.3.3.2 Variants of LMS . . . 69
4.3.3.3 Proposed iterative quantized clipped LMS . . . 71
4.4 Chapter summary . . . 72

5 Experimental work and discussion of the proposed detection and tracking model 74
5.1 Introduction . . . 74
5.1.1 Experimental setup . . . 74
5.2 Adaptive block CS . . . 76
5.3 LMS tracking . . . 91
5.4 Computational complexity . . . 98
5.5 Chapter summary . . . 99

6 Analytical framework of the detection model 101
6.1 Introduction . . . 102
6.2 related work . . . 103
6.3 WVSN model . . . 105
6.4 Probability of missed detection . . . 106

6.4.1 Probability of missed detection as a function of mobility model of the target . . . 106
6.4.2 Probability of missed detection as a function of Compressive Sensing . . . 109
6.4.3 Probability of missed detection for multi-target detection scenario . . . 110
6.5 Analysis and discussion . . . 111
6.5.1 Probability of missed detection as a function of mobility model 111
6.5.2 Probability of missed detection as a function of Compressive sensing . . . 117
6.5.3 Probability of missed detection for multi-target detection scenario . . . 121
6.6 Chapter summary . . . 123

7 Conclusions and future work 124
7.1 Conclusions . . . 124
7.2 Future work . . . 128

Bibliography 131

List of Figures

3.1 Typical WVSN model 30
3.2 samples in frequency and time domain 44
3.3 Compressive sensing example 45

4.1 The proposed model for the visual sensor node 53
4.2 The proposed model for the sink side or base station 53
4.3 (a) and (b)Background frames before and after background modeling and (c) background subtracted frame 57
4.4 First row in (a)(b) and (c) shows test frames and background subtraction results in second row 60
4.5 Detected objects for dataset"5" 61
4.6 The reconstructed original image for "Walking men" 62
4.7 Flowchart for the training phase of the adaptive CS process 65
4.8 Background subtracted frame 66
4.9 Blocks containing the targets (non-zero pixels) 66
4.10 An N-tap LMS adaptive filter 68

5.1 Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset1 78
5.2 Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset2 79
5.3 Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset3 80
5.4 Relation between the percentage ratio of target size:frame size vs. M 82
5.5 Relation between the percentage ratio of target size:frame size and (a) reconstruction MSE, (b) average PSNR 83
5.6 Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for block CS 84
5.7 Correlation coefficient for different M 85
5.8 Comparing trajectory of multi-targets for CS using different M (dataset1) 86
5.9 Comparing trajectory of single target for CS using different M (dataset 2) 87

5.10 Comparing trajectory of single target for CS using different M (dataset 3) . . . 88
5.11 Probability of detection vs. (a) different values of M and (b) different values of background subtraction threshold γ . . . 89
5.12 Comparing reconstruction MSE and PSNR with and without considering channel impairments for "Shopping center 1" . . . 90
5.13 Comparing MSE for different variants of LMS for (a)dataset 1 (b) dataset 2 and (c) dataset3 . . . 92
5.14 Comparing MSE for different u for dataset 1 . . . 93
5.15 Comparing MSE for different filter length F for datasets 2 and 3 respectively . . . 94
5.16 Comparing trajectory tracking of moving targets for (a) dataset 1, (b) dataset 2 and (c) dataset 3 . . . 95
5.17 Comparing trajectory tracking of moving targets for dataset "5" . . . 97
5.18 Comparing predicted trajectory using LMS and Kalman filter . . . 98

6.1 Wireless visual sensor network . . . 106
6.2 Scheme for sensor's duty cycle . . . 106
6.3 Sensor model for (a) linear and (b) non-linear target trajectory . . . 107
6.4 Probability of missed detection vs. different duty cycles for (a) different r_s ($t_s = 15sec$, $N = 50$), (b)different t_s ($N = 50$, $r_s = 50$) and (c) different number of sensor nodes ($t_s = 15sec$, $r_s = 50$). In all cases the target enters with velocity $v = 15m/s$ . . . 112
6.5 Probability of missed detection vs. different duty cycles for (a) $N = 1$ and different r_s and (b) $N = 0$. In all cases, t_s=15sec and the target enters with velocity $v = 15m/s$ . . . 114
6.6 Probability of missed detection vs. different duty cycles for different target's velocity ($N = 50$, $r_s = 50$, $t_s = 15sec$) . . . 115
6.7 Probability of missed detection vs. different duty cycles for different r_s, v=100m/s, $ts = 15sec$, N=50 . . . 116
6.8 Probability of detecting a target after CS reconstruction vs. (a) M and (b) reconstruction PSNR . . . 118
6.9 Probability of detection using CS vs. M for different percentage of sparsity levels . . . 119
6.10 Probability of missed detection with and without CS vs. different duty cycles for different sparsity levels and M (a) $K = 30\%$ and (b) K=11% . . . 120
6.11 Probability of missed detection vs. different duty cycles for different number of targets (a) Probability of missing all targets and (b) probability of missing at least one target. In both cases different sparsity levels and M are considered for CS . . . 122

7.1 Proposed dynamic model for target detection and tracking . . . 130

List of Tables

2.1 Computational complexity for tracking algorithms 20

5.1 Transmission energy using CS, block CS and without CS for different k 99
5.2 Computational time for CS, block CS and LMS 99

Abbreviations

WVSN	**W**ireless **V**visual **S**sensor **N**etwork
WSN	**W**ireless **S**ensor **N**etworks
CS	**C**ompressive **S**ensing
MSE	**M**ean **S**quare **E**rror
PIR	**P**assive **I**nfra **R**ed
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
ARMA	**A**uto **R**egressive **M**oving **A**verage
P2P	**P**eer-to-**P**eer
LOTS	**L**ehigh **O**mnidirectional **T**racking **S**ystem
KDE	**K**ernel **D**ensity **E**stimation
HF	**H**ole **F**illing
AMF	**A**daptive **M**edian **F**iltering
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
LMS	**L**east **M**ean **S**quare
RLS	**R**ecursive **L**east **S**quare
IR	**I**nfra **R**ed
DMD	**D**igital **M**icro-**M**irror **D**evice
TMF	**T**emporal **M**edian **F**ilter
MoG	**M**ixture of **G**aussians
ARQ	**A**utomatic **R**epeat **Q**uery
NLMS	**N**ormalized **L**east **M**ean **S**quare
PDF	**P**robability **D**ensity **F**unction

Symbols

p	order of LMS polynomial
D_s	state vector dimension for Kalman filter
D_o	observation vector dimension for Kalman filter
$T_{inv}(D_o)$	time complexity of matrix inversion
$N \times N$	matrix (image) dimension
E	constant image brightness
$I(x, y, t)$	brightness value of pixel (x, y) at time t
I_x	partial derivatives of the brightness values with respect to x
I_y	partial derivatives of the brightness values with respect to y
I_t	partial derivatives of the brightness values with respect to t
(u_x, v_y)	velocity vectors of optical flow estimation
(u, v)	gradient operators of $I(x, y, t)$
$p_x, p_{xy}, p_y, p_{xt}, p_{yt}$	products of the partial derivatives I_x and I_y
F	number of video frames (size of buffer)
t	time instant
X_{b_t}	background frame at time instant t
X_{t-i}	set of previous image frames
X	image frame
X_t	current image frame
(x, y)	x^{th}, y^{th} (row column respectively) pixel in the frame
x_i	pixel value
K_h	kernel function

h	kernel scaling factor
μ	mean of the gaussian distribution
σ	variance of the gaussian distribution
α	constant update ranges between 0 and 1
G_k	number of Gaussian distributions
f	measurements or features vectors
$w_(g_i)$	gaussian mixture weights
μ_i	mean vector of G_k gaussian distributions
Σ_i	covariance matrix of G_k gaussian distribution
$g(f\|\mu_i, \Sigma_i)$	component gaussian mixtures
λ	notation where the Gaussian mixture model is parameterized
w	LMS filter weights
$x(t)$	LMS input vector
u	step size parameter
$y(t)$	LMS predicted output
$d(t)$	LMS reference signal
$e(t)$	LMS MSE between the predicted output
λ_{max}	maximum eigenvalue of autocorrelation matrix
c	Kalman state vector summarizes the past behavior
A	Kalman state transition model
υ_t	unknown zero mean white noise
Q_t	covariance of unknown zero mean white noise
z_t	Kalman measurement observation
H	observation model mapping the true state into the observed state
ω_t	unknown zero mean white measurement noise
R_t	covariance of unknown zero mean white measurement noise
C	Kalman state covariance
K_g	Kalman gain
K	number of non-zero pixels in an image
M	size of compressed measurements

$\mathbf{\Psi}$	sparsifying transform domain
S	matrix with sparse coefficients of X
$\mathbf{\Phi}$	random measurement matrix of size $(M \times N)$
Y	compressed measurements of an image
Y_d	compressed measurements of an difference background subtracted image
δ	smallest quantity to satisfy restricted isometry property
ϵ	error of restricted isometry property
X_d	background subtracted image
V	number of sensor nodes
B	image divided into B blocks
$X_{blk}(i)$	i_{th} block of an image
$N_b \times N_b$	dimension of each image block
Y_{blk}	compressed measurements of corresponding block in an image
d	distance in m between nodes
E_{tx}	transmission energy dissipated by a node over a distance d
k	size of transmitted data
E_{elec}	energy to run transmitter and receiver circuit
e_{amp}	energy for the transmitted amplifier
γ	given threshold to extract the foreground target by background subtraction
D_1, D_2	threshold values used to clip the input data
r_s	node's sensing radius
t_s	node's sensing time
β_s	node's duty cycle
P_d	Probability a target is detected
P_{md}	probability a target is missed detection
t_a	time at which target enters the sensing area
t_{cross}	time target crosses the sensing area
v	target's velocity
L	length of intersection between the target's trajectory and the sensing area
ξ_{target}	event where the sensor is ON when the target enter the sensing area

ξ_{det}	event where the sensor is ON when the target crosses the sensing area
ς	is the time interval where the target enters the sensing area
α_f	probability of false alarm

Chapter 1

Introduction

This chapter gives an overview introduction on wireless visual sensor network with the characteristics and constraints. It discusses the requirements and various areas in the field of surveillance applications. Aims and objectives of this work are then presented.

1.1 Context of investigations

Wireless Visual Sensor Networks (WVSNs) have gained significant importance in the last few years and have emerged in several distinctive applications [1, 2]. While traditional wireless sensor networks (WSNs) are limited to scalor sensors and 1D data sets such as measuring temperature, humidity or magnetic field, WVSNs deploy visual sensors to the network platform producing 2D and 3D data sets [1]. Hence targeting a number of application scenarios beyond the scope of traditional WVSNs ranging from civil and military applications, surveillance and security applications, tracking, environmental monitoring, detecting natural disasters, to modern health care applications such as monitoring elderly or patients.

Due to the evolvement of new technologies and techniques, there are immediate needs for automated energy-efficient surveillance systems. WVSN has targeted various surveillance applications in commercial, law enforcement and military purpose as well as traffic control, security in shopping malls and amusement parks. Systems have been developed for video surveillance including highway, subway and tunnel monitoring, in addition to remote surveillance of human activities such as elderly or patients care.

Visual sensor nodes are resource constraint devices bringing the special characteristics of WVSNs such as energy, storage and bandwidth constraints which introduced new challenges [2–5]. For instance, within WVSNs each sensor node is powered by an attached battery and embeds a visual sensor(can be integrated with other types of sensors), digital signal processing unit, memory and a wireless transceiver. Due to the limited battery power, optimum energy utilization is necessary where all processing must be done energy-efficiently in addition to energy-efficient transmission through the wireless channel to save power and maximize the lifetime of the network. Moreover, visual sensor nodes are required to capture large data sets such as videos, and still images from the environment requiring high storage and high bandwidth for transmission which are limited. As well as higher complexity of local data processing.

Visual sensor nodes have to perform some local processing (such as compression, filtering, object detection, tracking and recognition) with minimum processing power and complexity on the extracted data before transmitting it in the bandwidth-constrained wireless channel to the receiver side or sink node (base station). Hence, energy efficient processing and efficient compression techniques are required to overcome such constrains of WVSNs while efficiently transmitting data through the wireless channel. Among the many diverse application domains of WVSNs, Object detection and tracking (object can be a human being, a vehicle or any targeted object) are of the most important tasks. It is required to minimize the energy consumption during processing without compromising detection

reliability and tracking performance. At the same time, there is scope to achieve the same by minimizing the volume of data required to serve the purpose. An important issue that arises in WVSNs is the complex image processing algorithms, which are both memory and computational intensive. Data encoding/decoding involves significant processing, much more than the processing required for sensed data [6–8]. These requirements make image processing a challenging problem as sensor nodes are resource constrained devices. Memory limitations, local computational energy cost and processor speed are therefore to be considered in image processing in order to guarantee real-time response.

There exists several WSN standards [9], the most commonly used is the IEEE 802.15.4 Wireless personal area network, which supports a total of 16 channels in the 2.4-GHz band, 2MHz bandwidth each. It targets low power and data rate applications and supports a maximum of 250Kb/s data rate. IEEE 802.11b/g wireless local area network supports 14 channels in the 2.4-GHz band, each with a bandwidth of 22 MHz, it supports real time video streaming however at the expense of more power dissipation. The IEEE 802.15.1 Bluetooth standard operates in 79 channels in the 2.4-GHz band and a 1-MHz bandwidth each. The number of nodes in the networks controls the bandwidth available to each transmitter node (i.e., its average transmission rate). Therefore, there is a tradeoff between number of nodes and allocated bandwidth. Moreover, size of transmitted data leads to high bandwidth requirement per transmitter, which in turn decreases the number of sensors that can be accommodated by the WVSN. Hence, reducing the size of transmitted data via low power compression and processing leads to lower bandwidth requirement and as a result increasing the number of sensor nodes in the network [10].

As for high volume data sets are acquired for WVSN-based surveillance applications, data should be represented in such a way that it requires optimum storage, energy, and allow reliable transmission due to the constraint on the physical and

radio resources. Suppose for a surveillance application within WVSN, an image is captured and required to be sampled for storage as well as to be transmitted through wireless channel. According to Shannon-Nyquist sampling theory the minimum number of samples required to accurately reconstruct a set of data without losses is twice its maximum frequency [11]. It is always challenging to reduce this sampling rate as much as possible, hence reducing the computation energy and storage. Compressive Sensing (CS) is promising to overcome the above mentioned limitations. CS theory shows that a set of data is expected to be reconstructed from far fewer samples than required by Nyquist theory as it is always challenging to reduce the sampling rate as possible, provided that the set of data is sparse (where most of its energy is concentrated in few non-zero coefficients) or compressible in some basis domain [12]. CS is a simple and low energy consumption process which is suitable for power constraint sensor nodes where complex computations are just done at the receiver side.

The WVSNs have strict resource limitations hence make it desirable to design energy-efficient target deection and tracking techniques. In this work, CS-based single/multi target detection and tracking algorithm for energy efficient visual surveillance application within WVSN is proposed, with minimum energy-expenditure but efficient tracking reliability which will be represented as minimal mean square error (MSE) and accurate trajectory tracking. MSE measure is used to compare two signals/images by providing a quantitative score describing the degree of similarity or in other words, the level of error/distortion between them. Due to the battery constrain of sensor nodes, there is an immediate need to increase the WVSN's lifetime, energy consumption is reduced by periodically switching On and OFF the visual sensors. Each sensor node is assumed to be in 'wake-up' state according to a predefined duty cycle.

1.2 Visual Surveillance requirements and Applications

The purpose for visual surveillance is to provide constant operational information, in order to detect specific events or targeted objects to maintain security or intrusion detection at the monitored locations. Due the evolvement of new technologies and techniques, WVSN has targeted various Surveillance applications and can be applied in public places, such as banks, supermarkets, homes, department stores, amusement parks, parking lots, or football match attendance. Systems have been developed for visual surveillance including highway monitoring, subway monitoring, tunnel monitoring, intelligent video communication and indoor/quasi-outdoor monitoring, and in the remote surveillance of human activities such as elderly or patients care [13].

Each of the above-mentioned areas is potential to threats and requires situational awareness to the possible effective response of the relevant enforcement agencies, hence, there are immediate needs for automated surveillance systems. In addition to security applications, visual surveillance technology is used in transport applications, such as airports, sea environments, railways, underground, and motorways to observe traffic flow, detect accidents on highways and monitor congestion in public spaces. In addition to numerous military applications including patrolling national borders, measuring the flow of refugees in troubled areas, monitoring peace treaties, and providing secure perimeters around bases and embassies.

The followings are some of applications and situation specific visual surveillance

- One of the requirements for a WVSN-based surveillance applications is to monitor specific places such as stations, shopping areas and faculties for intrusion detection or abnormal behavior. A multimodal video surveillance for the safety and security of a faculty entrance has been proposed in [14]. The authors aim to design efficient and robust target detection and tracking

algorithm to detect intrusion or abnormal events. To save battery power of camera nodes, Passive Infra Red (PIR) sensors are integrated with multiple cameras, where when he PIR detects the presence of targets then triggers the corresponding cameras to get to wake up state. Moreover, for robust tracking PIR sensors resolve occlusion as they can detect changes in direction during occlusion.

- Another intrusion detection algorithm is introduced in [15], the requirement of this application is to reduce the energy consumed by each sensor node while ensure reliable detection and tracking. Multiple cameras are distributed over the monitored area and instead of wasting energy in continuous monitoring, each camera is given a probability of monitoring according to visibility regions.

- Various applications are present in the field of elderly care where there has been considerable recent interest in addressing the problem of object finding both in academia and industry such as in [16]. A video surveillance for detection, recognition and classification of objects (a small set of pre-selected common objects) has been proposed as a tool for assisting living for elderly. A technique that is less complex and consumes less energy but can still enable robust image recognition is the main aim. This ia achieved by composing the WVSN of two-tiered system which means a low power and a high power tier. The low power tier contains low power sensor nodes. While the high power tier contains high end processing sensor nodes. The field of view of both cameras is same because of being closely mounted. The system uses low power tier for still target detection and high power tier is being used for energy efficient accurate target recognition and classification.

- In [17] a smart home care system is designed to track elderly or patients and detect abnormal behavior such as a sudden fall. In these types of applications sensor nodes are to be sensitive to sudden changes for the safety of the

monitor person. In this paper, they integrated the video cameras with sensors to detect falls then triggering the most appropriate camera for detecting the target and classifying either the target is human or non-human.

- Compliance monitoring is useful in industries where standard operating procedures have to be strictly followed. Through video surveillance cameras, managers of restaurants or hotels can determine whether or not their staffs are following proper sanitation measures. Video surveillance cameras are also useful in cosmetics and pharmaceutical industries. They can monitor vital parts of the production process, such as processing and packing. For these applications high quality images is not a constraint as compared to the ability of sensors to continuously monitor the required area in an energy-efficient manner.

- Another field for surveillance applications is for environmental monitoring. A WSN can be installed in a forest to detect when a fire has started. The nodes can be equipped with sensors to measure temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the fire-fighters. In [18], it is required to provide the fire fighting community the ability to safely and easily measure and view fire and weather conditions to better predict fire behavior. Hence, the sensitivity of sensors is highly required in addition to increasing the lifetime of sensor nodes.

- Remote telepresence is another kind of visual surveillance applications that requires the cameras to be positioned in locations not accessible to humans, which includes sensing volcanoes, earthquakes, oceans, glaciers or forests [19]. Examples of these locations include the ocean, the bottom of the sea, desert landscapes, or the insides of a human body. Data from this highly specialized use of visual surveillance cameras are used in various practical applications, such as solving medical problems, investigating disputes over natural resources, habitat monitoring and saving endangered species where

there is great concern about the potential impacts of human presence in monitoring plants and animals in field conditions, as in [20] the authors designed a surveillance application to study the behavior of animals particularly zebras..

- Another important and critical WVSN application is military applications, they include several scenarios; in battlefields to position and track hostile targets, operations in urban environments such as patrolling national borders or measuring the flow of refugees in troubled areas, or other than war missions such as peacekeeping by monitoring peace treaties, and providing secure perimeters around bases and embassies. Military missions may last for months or years, hence requires energy aware schemes to maximize the network's lifetime as human interaction is not allowed.

1.3 Aim and objectives

The aim of this work is to investigate the impact of CS in designing efficient object detection and tracking techniques for WVSNs-based surveillance applications, without compromising the energy constraint which is one of the main characteristics of WVSNs. As mentioned previously, in WVSN-based surveillance applications large data sets such as video, and still images are to be retrieved from the environment requiring high storage and high bandwidth for transmission. As well as higher complexity of data processing and analysis for object detection and tracking which are all quite costly in terms of energy and memory requirements as visual sensor nodes are resource and bandwidth constrained. CS is expected to reduce the size of sampled data with a low power simple process hence saving space, energy of processing and transmission as well as channel bandwidth as the size of transmitted data is reduced.

The following objectives have been identified:

- To carry out a literature review on WVSNs, target detection, tracking and compressive sensing.
- To investigate existing target detection and tracking techniques with their strengths and weaknesses.
- To identify a WVSN model for a surveillance application, to meet some criteria such as energy efficiency and minimum communication overhead.
- To investigate existing work of CS in the context of target detection and tracking along with their strengths and weaknesses for a required performance.
- To identify and design an adaptive CS technique suitable for WVSNs constraints to overcome the shortcomings of high computational techniques present in the literature while achieving higher compression rates and least reconstruction mean square error.
- To design efficient target detection and tracking techniques for WVSN-based surveillance applications.
- To evaluate the performance of the proposed target detection and tracking schemes for surveillance applications in WVSN compared to other techniques in the context of resource requirements
- To derive an analytical framework to examine the impact of selecting node's duty cycles and dynamically choosing the appropriate compression rates for captured images and videos on the detection performance.

1.4 Thesis organization

The remainder of the thesis is organized as follows:

Chapter 2 gives a literature review on WNSNs and related work on different target detection and tracking techniques with their pros and cons. In addition, related work on compressive sensing is next presented with a discussion on various applications of compressive sensing in the context of WVSNs. Later, adaptive CS is discussed with related applications.

The theoretical aspects behind the proposed investigations are discussed in chapter 3 with respect to target detection and tracking, shortcomings of previous techniques, methods to overcome these shortcomings together with the motivation for the proposed work. A complete theoretical aspects of CS is then presented including the entire CS process and image reconstruction. In addition, an overview of a typical WVSN model is presented together with the main characteristics of WVSN.

Chapter 4 describes the proposed detection and tracking model based on compressive sensing. First, the detection algorithm with the chosen background modeling technique and morphological operations are presented, next the proposed adaptive block CS algorithm is introduced describing CS techniques, adaptive CS and block CS. Finally, the LMS algorithm for target tracking is proposed with different variants of LMS.

Chapter 5 presents the experimental results of the proposed model. The model handles single and multi-moving targets, indoor and outdoor monitoring. The performance is evaluated using the following performance indicators: mean square error (MSE), peak signal-to-noise ratio (PSNR), correlation coefficient, detection probability, trajectory tracking and energy dissipated.

Chapter 6 derives an analytical framework to inspect the impact of node's duty cycles and other network's parameters and the effect of dynamically choosing the appropriate compression rates for captured images and videos on the detection performance which is characterized in terms of the probability of missing a target.

Finally Conclusion and future work are discussed in chapter 7

Chapter 2

Literature review

2.1 Introduction

This chapter first presents literature review on surveillance WVSNs and related applications in the context of target detection and tracking. Next, literature review on target detection techniques and background modeling are discussed with their pros and cons. Related work and different target tracking algorithms are then illustrated. Finally, an overview on compressive sensing is presented, in addition to related work and applications of CS in the field of imaging, target detection and tracking within visual sensor networks.

2.2 WVSNs

Recently WVSNs has emerged as a powerful distributed systems, that has been widely used in several applications specially in target surveillance because of its outstanding performance in sensing and signal processing [21]. Much work is present in the literature for surveillance applications within WVSNs. In [21], the authors introduced a multiview visual-target-surveillance system in WVSN, which

implements target classification and tracking with collaborative online learning and localization. In [22], a practical target tracking WSN system is proposed based on the auto regressive moving average (ARMA) model in a distributed peer-to-peer (P2P) signal processing framework. Wireless sensor nodes act as peers that perform target detection, feature extraction, classification and tracking, whereas target localization requires the collaboration between wireless sensor nodes for improving the accuracy and robustness. A distributed multi-view tracking system using collaborative signal processing in distributed wireless sensor networks is proposed in [23]. In the tracking system, target detection and classification algorithms are based on single-node processing and target tracking is performed in sink node, whereas target localization algorithm is carried out by collaboration between multi-sensors. A progressive distributed data fusion is proposed to overcome the disadvantages of client/server based centralized data fusion. A multimodal video surveillance for the safety and security of a faculty entrance has been proposed in [14]. The authors aim to design efficient and robust target detection and tracking algorithm to detect intrusion or abnormal events. To save battery power of camera nodes, Passive Infra Red (PIR) sensors are integrated with multiple cameras, where when he PIR detects the presence of targets then triggers the corresponding cameras to get to wake up state. Moreover, for robust tracking PIR sensors resolve occlusion as they can detect changes in direction during occlusion. As battery energy is a crucial issue, in [24, 25] the authors distributed the processing on sensor nodes by sleep and wake up periodically, according to proper duty cycles, sensing and communication modules of wireless sensor nodes. Making these modules work in discontinuous fashion by random scheduling where it is probably the easiest to implement in sensor networks as it requires no coordination among nodes.

Several existing techniques for target detection and tracking present in the literature [21, 26–40] are presented next in Sec.2.2.1 and 2.2.2. However most techniques for target detection and tracking focus mainly on the reliability of the detection and tracking process without taking the energy-efficiency factor into consideration

as there is always a trade off between energy consumption and quality of service in terms of detection reliability and robustness. CS has been considered for different aspects of surveillance applications due to its energy efficient and low power processing as reported in [41, 42]. Brief description of these techniques for surveillance applications in the context of the proposed investigation is presented next in Sec.2.3

Target detection and tracking are of the most important tasks within WVSN-based surveillance applications. Although detecting isolated targets is relatively simple but the challenge remains in detecting and tracking multiple targets in complex backgrounds, especially in cases of deformation of objects (scale or pose variation), change in speed, occlusion and dynamic background. Most of the existing techniques [7, 30, 43, 44, 44] are not robust under certain conditions such as sudden illumination changes, heavy fog or moving background objects like trees. On the other hand, target tracking in surveillance applications with low power devices as in WVSN is challenging, as most of the target tracking techniques are computationally intensive and sometimes require real-time data processing [26–28]. Moreover, energy efficiency is another challenging factor as processing is performed by visual sensor nodes which have limited battery power and might be kept for long period of times without any human interactions. Energy consumption and reliability of the existing algorithms are therefore required to be investigated to obtain an energy efficient solutions for WVSN. Following subsections give a discussion on related work in the context of target detection and tracking within WVSNs.

2.2.1 Target detection

A target to be tracked within a WVSN is expected to be visually separable from its background In the context of surveillance applications, moving objects have made target detection more challenging. Generally, target detection techniques

can be categorized into three main approaches; optical flow, temporal difference and background subtraction based on their signal manipulation process [26, 45]. Optical flow-based target detection relies on flow vectors of moving objects over time to identify foreground in an image. Optical flow is time consuming and requires complex computations making it not an appropriate candidate for WVSNs and real time applications [46]. Temporal difference-based target detection consists of subtracting consecutive images followed by thresholding the difference. Frame differencing is simple and suitable for WVSNs, it can operate with low memory requirements, no floating point operations and relatively adaptive to dynamic changes. However, it is not very reliable, and fails to detect all interior object pixels. Hence, it is challenging to detect steady, slow moving targets or occluded as well as targets with uniform texture, as such techniques are solely dependent on pixel-based feature extraction [26, 29]. The most commonly used algorithms for target detection are based on basic background subtraction [30, 45]. Background subtraction is a simple approach to detect the presence of targets in a video sequences as each current frame is subtracted from the background frame and to classify each pixel as background or foreground is achieved by comparing this difference by a predefined threshold. Background subtraction schemes provide better detection results but are sensitive to dynamic changes in the scene such as lighting, rain, trees, etc.. This is considered as a binary hypothesis test problem with only two possible hypotheses; background and foreground. The background may need to adapt to several situations to get acceptable results. In recent years, many background modeling techniques have been proposed to adapt to changes and update the background.

Although this problem can be solved by updating the background by pre-processing and post-processing. There are different background modeling techniques ranging from simple ones to more complex and accurate ones that can handle variations in the scene but still its challenging to find a technique that is suitable for low power applications such as WVSN [30, 31].

In [31, 32] several background subtraction techniques for outdoor visual surveillance applications have been studied; Basic background subtraction, Lehigh Omnidirectional Tracking System (LOTS) proposed in [47], single gaussian model [48], mixture of Gaussian model (MoG) [49] and w^4 [34]. All are compared under the same lighting and background conditions. Basic background subtraction method achieved high probability of detection but also higher false detections. Authors have reported that LOTS, provides the best detection probability and with lower rate of false alarm. The detection performance of the Single Gaussian model is sensitive to illumination changes and moving background targets. However, MoG improves the detection probability with sharper illumination gradient in comparison to Single Gaussian model; but requires extensive computational complexity [6, 44]. Subsequently, it is expected that MoG is not suitable for low-power applications. On the other hand, W^4 a real time visual surveillance system for detecting and tracking multiple objects and monitoring produces the lowest probability of detection and higher probability of false alarm reaching up to 90% with respect to manually labeled ground truth. Another technique for background subtraction is Kernel Density Estimation (KDE) [50] where a probability density function is estimated for each pixel colour then the foreground detection is done according to a defined threshold. Though, KDE is slow and consumes higher memory space.

In [51], an enhanced version of MoG technique is proposed which is combined with Hole Filling Algorithm (HF) to alleviate the problems of MoG background modeling such as false classification due to noisy image. This situation may arise because several conditions in the video input such as, waving trees, rippling water, and illumination changes. The experimental result shows that the proposed method improved the accuracy up to 97.9%. However, it still requires extensive computational complexity hence still not suitable for low-power WVSN applications. Adaptive median filtering (AMF) is proposed in [52] to overcome the shortcomings of the Temporal Median Filter. AMF uses a simple recursive filter

to estimate the median where it simply increments the background model intensity by one, if the incoming intensity value (in the new input frame) is larger than the previous existing intensity in the background model. The reverse is also true, meaning that when the intensity of the new input is smaller than background model the corresponding intensity will be decreased by one. It has been proved by [52] that this trend will converge to the median of the observed intensities over time without requiring storing any frames in a buffer and tries to update the estimated background model online. Hence it is extremely fast and suitable for real time applications.

Running average background model in [53] dynamically update the background image to adapt to the scene changing by using the weighed sum of the current image and background image. Running average background needs to compute the weighted sum of two images, so it has low space and computational complexities satisfying the fundamental requirements for WVSN platforms such as in Cyclops [54]. Moreover, dynamically updating the background makes this model adaptive to very complex scenes.

Most VSN platforms implemented some type of background subtraction [45], such as Cyclops used a running average filter due to its simplicity to estimate the background [54]. For MicrelEye [45], the MicrelEye node is used for target detection where it implemented background subtraction assuming a fixed background. While in MeshEye to reduce the computations, it performed background subtraction and stereo matching on low resolution images (30×30) pixels first. Once detecting and matching the object, high-resolution images are triggered to take snapshots of regions of interest. When matching objects, simple features such as position, velocity, and bounding box were extracted [45].

One of the other major challenges that affects the detection performance of techniques based on background subtraction is to obtain an optimum threshold. To overcome the problems of thresholding, Absolute Difference Edge-Based detection

technique [8] that applies Sobel edge detector is proposed to get away from the problems of thresholding. Nevertheless, it is scene constrained where it does not perform well under all conditions such as in an outdoor scene with heavy fog.

Another direction for target detection is the codebook model [55], where for each pixel, a codebook is constructed consisting of codewords based on some predefined features. For each pixel in a new frame, a codeword match is performed to classify the pixel as either foreground or background. The codebook model has the advantage of being fast and unlike MoG model which compute probabilities using costly floating point operations, this method does not involve probabilistic estimation. Indeed, it simply computes the distance of the sample from the nearest cluster mean. It requires low memory space and works well on dynamic backgrounds. However, using normalized colors is undesirable because of their high variance at low brightness levels, one necessarily sacrifices sensitivity at high brightness.

2.2.2 Target tracking

Once moving targets have been identified, the next task of a surveillance system is to generate tracks of these targets over successive frames. Target tracking, by definition, is to track a target or multiple targets over a sequence of images. Target tracking is critical in many computer vision applications such as security and surveillance, perceptual user interfaces, augmented reality, smart rooms, target-based video compression, and driver assistance. Hence, target tracking is important as provide better sense of security using visual information, analyze shopping behavior of customers and to enhance building and environment design. In addition to video abstraction to obtain automatic annotation of videos and to generate target-based summaries, traffic management to analyze flow and to detect accidents [56]. Target tracking is a challenging tasks in terms of reliability and computational sensitivity. Difficulties can arise due to many factors, such as

non-static background and changes in targets appearance due to pose or scale variation, full or partial occlusion, different illumination conditions or abrupt motion [56].

As for target tracking, there is much literature for target tracking using visual sensor network which can provide higher accuracy [21, 27, 28, 33, 35–38, 57]. However, having the power constraint into consideration, there are few target tracking algorithms reported in the literature for WVSN-based surveillance applications that requires higher reliability [57]. Template correlation matching [33] is a method for tracking where a template is taken from previous frames and correlated with regions in next frame to find which region best accommodates the template. However, template correlation is not efficient in the presence of changes in the targets appearance (such as change in size, intensity, orientation). Classical active contour [38] for target tracking fails in tracking multiple targets when they partially or fully occlude, so occlusion problem is the main challenge. Hence, a modification to the active contour is proposed in [39]. It resolves the occlusion problems by performing merging and splitting without requiring preprocessing nor motion estimation, subsequently is suitable for real-time applications. Moreover, it supports the tracking of non-rigid targets in outdoor environments. However, there is probability that the target may get lost if the targets speed is high making the displacement of the target between two consecutive frames to be large.

Particle filtering [28, 36] which is known to be suitable for real time tracking and non-linear non-Gaussian processes, relies on motion parameter and other probabilistic parameter estimation. The number of particles that are used in the filter mainly determines the tracking precision where the tracking error decreases by using a larger number of particles and vice versa [58]. However, increasing the number of particles is restricted in WVSNs due to memory constraints. Subsequently, the performance of the particle filter in terms of tracking reliability decreases and false detection increases with low resolution frames. Tracking of deformable targets exploiting particle filters is challenging as it may require much

more particles to obtain the targeted tracking reliability [37].
Kalman filtering [27] is relatively the best linear estimator based on the accuracy of target tracking. Kalman filters are robust in terms of tracking accuracy under optimal conditions, otherwise adaptive approaches are needed to solve these problems and non-linearity which can be either computational expensive or not always applicable in real time tracking. Although, Adaptive Kalman filters can be used for non-linear systems and has been successfully used in various applications [35, 37]. However, this kind of nonlinear methods can lead to problems in stability and convergence. Moreover, a practical difficulty faced in the implementation of such adaptive filters is the requirement of prior information from previous images. This apriori information may not be available in certain cases. In [59], a visual surveillance system for moving object detection and tracking has been presented. The object is first detected using simple background subtraction then it is tracked along its path by estimating the object's location using kalman filter. In [60] an algorithm of feature-based using Kalman filter motion to handle multiple objects tracking has been proposed. The system is fully automatic and requires no manual input of any kind for initialization of tracking. The authors in [61] proposed a multiple object tracking algorithm that seeks the optimal state sequence which maximizes the joint state-observation probability. The algorithm is capable of tracking multiple objects whose number is unknown and varies during tracking. In [62], objects randomly chosen by a user are tracked using scale invariant feature transform descriptor (SIFT) and a Kalman filter. After sufficient information about the objects are accumulated, exploiting the learning to successfully track objects even when the objects come into the view after it had been disappeared for a few frames. However, these kinds of filtering require lots of matrix manipulation and is usually avoided despite its robustness [45].

Adaptive filters such as least mean square (LMS) and recursive least square (RLS) algorithms are the two fundamental adaptive filtering algorithms [63, 64] that work

TABLE 2.1: Computational complexity for tracking algorithms

Tracking technique	**Computational complexity**
LMS filter	$O(3p)$
Kalman filter	$O(3D_s^3 + 3D_s^2D_o + D_s(D_o^2 + D_o) + T_{inv}(D_o))$
Particle filter	$O(23D_s^3 + D_s^2 + 162D_s + 13)$

satisfactorily in the absence of apriori information, in contrast to Kalman filtering. They have been widely used in several applications such as motion estimation, tracking time variations in signals, vision applications in addition to system identification, inverse system modeling, 1D and 2D signal prediction and interference Cancelation. Among various estimation methods that have historically been used, LMS has received considerable attention in the past and has been used to solve the problem of estimation and tracking [65]. The reason behind this is that the LMS algorithm is relatively simple, has much lower computational complexity than the original Kalman filters and other adaptive algorithms as it does not require correlation function calculation nor does it require matrix inversions [66]. Moreover, suitable for real time images applications [64, 67, 68]. In [58], a performance analysis is carried out comparing the computational complexities of different tracking algorithms; LMS, Kalman and particles filters in terms of number of multiplication operations required. The analysis is illustrated in Table.2.1 , it shows that the complexity of the LMS algorithm is $O(3p)$, where p is the order of the polynomial used in the LMS filter. On the other hand, the complexity of Kalman filtering is $O(3D_s^3 + 3D_s^2D_o + D_s(D_o^2 + D_o) + T_{inv}(D_o))$, where D_s is the dimension of the state vector, D_o is the dimension of observation vector and $T_{inv}(D_o)$ is the time complexity of matrix inversion. Whereas, the complexity of particle filter is $O(23D_s^3 + D_s^2 + 162D_s + 13)$

Furthermore, the LMS filter has satisfied overall performance than the RLS filter for visual target tracking in noisy environments where the PSNR is low. Also the simplicity of implementation of the LMS filter causes new developments for this algorithm that enhance the capability and performance of this filter. Although for

some step sizes and filter lengths, the LMS has lower convergence rate, as compared to RLS. Nevertheless, LMS algorithm still has better tracking performance compared to RLS algorithm, as the fast convergence rates of RLS comes at the cost of high computational complexity, complicated implementation and sensitivity to noise that accumulates due to recursive computations which may results in instability [69–71]. Tracking potential of LMS has received considerable interest in the literature; in [72], the authors introduced an edge directional 2D LMS filter for small target detection in infrared (IR) images. Generally, the 2D LMS filter functions as a background prediction to apply to IR small target detection field. A new small moving target object (such as car, bikes, etc.) tracking algorithm is proposed in [73] which is based on a new clipping technique in the field of adaptive filter algorithms. The uncertainty and occlusion of such objects in noisy environment leads towards the requirement to introduce new clipping technique which can control noise in prediction. In [74], an analysis of the steady-state MSE convergence of the LMS algorithm was carried out when deterministic functions are used as reference inputs for the applications of biomedical signals. A new model for adaptive filter with LMS scheme is presented [75] to train the mask operation on low resolution images within an energy constrained surveillance system for moving object detection and tracking. In The work in [76] presented a novel method for local image registration based on adaptive filtering techniques, where a 1-D and 2-D LMS adaptive filters were utilized to estimate and track correspondences among multiple images containing overlapping views of common scene regions.

Based on the above literature, there is no generalized dynamic integrated algorithm which attains a trade off between computational complexity and detection and tracking accuracy in the context of energy constrained WVSN found in the literature. Yet, results can be obtained but with restrictions as pre-processing and post-processing that require high computational overhead. However, a suitable image processing scheme that can provide intended target detection and tracking

accuracy with optimal pre-processing and post-processing is expected to be efficient for the WVSN's energy constraint nature. Within the scope of the authors knowledge, recently proposed CS [11] is expected to be the strongest candidate to provide this. A brief overview of the existing work on CS in the context of WVSN based target detection and tracking applications is given below.

2.3 Compressive Sensing

Previous work [77, 78] have shown that CS can be a useful imaging tool when the underlying signal is compressible in a known basis or representation even under noisy conditions such sensing noise or channel noise which is generally modeled as additive white Gaussian noise under noise conditions such sensing noise or channel noise which is generally modeled as additive white Gaussian noise. CS is a new paradigm for data acquisition and processing. It was originally developed for the efficient storage and compression of digital images [79, 80], it has been widely used in several applications such as image processing, steganography and image watermarking[81, 82]. Moreover, CS offers an advantage especially if the signal is sparse or in high SNR imaging environments over conventional sampling and compression techniques such as the H.264 and MPEG4 video coding which are considered the most recent coding standard of video stream. H.264 and MPEG4 are based on complex encoders and simple decoders as the encoder performs intra-frame coding and exploits statistical dependence between frames in the source video signal to perform inter-frame coding. This configuration is suitable for many applications such as video broadcasting but for WVSNs it is different due to the limited energy and computational capabilities [83]. Moreover, although H.264 achieves high coding performance but at the expense of huge computational complexity as predictive encoding such as H.264 and MPEG4 requires complex processing algorithms, which lead to high energy consumption [84]. In

contrast to CS which is designed to aim simple encoders with very low computational complexity whereas complex computations are left at the decoder side which is not battery-powered. A comparative study between CS and traditional coding techniques is carried out in [83, 84], The H.264 quality decreases when the bit error rate increases above 10^{-4}. However, for CS the reconstruction quality will not be affected until the bit error rate increases above 5×10^{-3}. CS as a result can tolerate a fairly large amount of bit errors before the received video quality is affected. This is certainly not the case for predictive video encoding, and not even for transform-based image compression standards such as JPEG. This could result in significant transmission power savings or a significant decrease in the amount of forward error correction. Furthermore, the processor load is significantly lower for CS than for H.264 as predictive encoding requires complex processing algorithms, which lead to high energy consumption, in contrast to CS which is designed to aim simple encoders. This results in a reduction in the energy needed to encode the video. In addition, for H.264 and MPEG4, PSNR is affected by the frame type, whether I, P, or B frame due to inter-frame coding (Where I-frame is the key frame with highest quality and least compression, P-frames are predicted frames from previous I-frames or P-frames and B-frames are bi-directional predicted frames). However, for CS, size of measurement matrix (compression rates) satisfying a lower bound constraint guarantees more satisfactory quality.

Moreover, in a traditional signal processing system sampling is carried out according to Nyquist theory at a frequency which is at least twice the highest frequency component found in the signal. This is to guarantee signal recovery, where afterwards, the sampled signal can subsequently be subjected to further compression. This can be computationally intensive and unpractical in the case of battery-powered sensors. However, CS can sample at a lower than Nyquist rate while reducing computational complexity without threatening signal recovery. Furthermore, it is reported that CS has achieved higher PSNR and lower reconstruction error compared to traditional compression techniques such as JPEG and DCT

[85].

CS has been used within WVSNs for target detection and tracking but existing work did not focus on all parameters to be achieved, such as energy efficiency, accurate and reliable detection and tracking. In [41] compressive sensing for background subtraction is proposed where only the target in the difference image is recovered as a solution of a convex optimization known as basis pursuit or an orthogonal matching pursuit problem using the compressive measurements without any auxiliary image reconstruction. The difference image is always sparse (which is a fundamental requirement to apply CS) regardless of the sparsity nature of the original images. After the target has been detected, target tracking is performed and experiments have shown that the detection and tracking are not affected by CS. However, in this paper the focus has been in achieving target detection and tracking without focusing on the energy constraint.

Other work in compressive sensing for surveillance applications has been proposed in [86], where an image is projected on a set of random sensing basis yielding some measurements. In this paper, not only the background subtracted image is compressed but the whole captured image. At the receiving node or base station there is no interest to reconstruct the full image, however only specific targets of interest present in the image are reconstructed. Traditional reconstruction techniques reconstructs the compressed measurements yielding the original image, in contrary to reconstruct only specific parts in the image an adaptation to the reconstruction algorithm has been proposed in [86] through minimizing the weighted version of the l-2 norm to only reconstruct specific parts of the image. However, further research is required to address the selection of the weights and fully understand their impact on the target-specific reconstruction problem while taking into account the energy-efficiency parameter. In [87], compressive measurements are used for multi-view tracking where the measurements from the multiple cameras are sent to a central server or base station, where CS inversion needs to be performed at every time step.

In [42] a novel compressive particle filter for tracking one or more targets in video is presented using a reduced set of observations. It is shown that, by applying compressive sensing ideas in a multi-particle-filter framework, it is possible to preserve tracking performance while achieving considerable dimensionality reduction, avoiding costly feature extraction procedures. Additionally, the target locations are estimated directly, without the need to reconstruct each image. However, the proposed algorithm failed to provide acceptable performance for fast moving targets. In addition, it is not designed for WVSN applications thus constraints of WVSN such as energy and memory constraints were not taken into consideration.

Another promising direction is the adaptive CS, where CS dynamically chooses the compression rate according to the sparsity nature of frames which varies from one dataset to another. In contrast to static compression rates, different datasets have different sparsity levels, hence if the same dimension of the measurement matrix is used for more sparse images this will result in a waste of energy where more compression could have been applied. And for less sparse images, the quality after reconstruction will be affected which in returns degrades the detection performance. However, most existing adaptive CS techniques are computationally extensive making them not suitable for the constraints of WVSNs. In [88], energy efficient data collection in WSN using adaptive compressive sensing is proposed. An adaptive approach is proposed to select a routing path by choosing sensors required to transmit their data. However, in this approach adaptive CS is only applied for sensor nodes selection and no compression is performed on the transmitted data. A heuristic to solve the optimization problem (which is proven NP-hard) is proposed in [89] to find a measurement matrix that maximizes the information gain per energy expenditure. It was shown that under suitable conditions, one can reconstruct an $(N \times N)$ matrix from a small number of its sampled measurements. This is done by solving an optimization problem, provided that the number of measurements has a lower bound as a function of N, exact matrix recovery would be guaranteed with a reduced number of measurements. In

[90, 91], an adaptive approach to compressed sensing is proposed using a single pixel camera. Instead of using a representation (such as pseudo-random binary masks) that is incoherent with a conventional transform (as wavelets) to acquire the visual data. The image is sampled directly in the wavelet domain by tuning the Digital Micro-Mirror Device (DMD) of the single pixel camera to directly collect only the significant wavelet coefficients.

Adaptive CS is expected to outperform traditional CS as by choosing measurement matrices according to nature of datasets results in higher compression rates consequently saves energy. However, most adaptive techniques found in the literature use heuristics and np-hard techniques in choosing measurement matrices hence making them not suitable for WVSNs. To design adaptive techniques with simple and low power computations is therefore a great challenge in the context of data representation within WVSNs.

2.4 Chapter summary

Related work in the context of WVSNs has been presented, with various surveillance applications. Moreover, target detection and tracking applications in the context of WVSNs are discussed as they are considered the most important tasks within WVSNs. Next different target detection techniques are presented in addition to background modeling techniques. Among the presented background modeling, most of the techniques are robust but not suitable for WVSNs constraints due to either their high computational or high space complexities except for AMF and Running average can result in competitive performance as MoG with more simple implementation making them candidates to WVSNs constraints. Target tracking algorithms are then introduced with their advantages and disadvantages with respect to WVSNs. The LMS algorithm is chosen as the tracking technique due to its simplicity and its lower computational complexity compared to other tracking techniques such that Kalman filters. Finally, compressive sensing has

been presented in this chapter with some related applications. In addition, related work for adaptive CS has been identified showing its advantages over traditional compressive sensing. Based on the literature presented, there is no generalized dynamic integrated algorithm which attains a trade off between computational complexity and the accuracy of target detection and tracking in the context of energy constrained WVSN. CS has shown that it is expected to be a strong candidate to provide our aim in reducing the size of captured images with simple computations. Hence, CS is to be investigated in designing target detection and tracking techniques for an energy-efficient surveillance system. Most of the CS algorithms proposed in the literature are non-adaptive which means the random measurement matrix is not chosen according to information collected neither to nature of images. However, an important issue is to make the measurement matrix adaptive to achieve higher compression rates. Subsequently, most existing work in adaptive compressive sensing use heuristic techniques which are computationally expensive, hence taking only into consideration the accuracy of the approximate data field without considering the energy factor which is one of the main concerns for WVSNs. Designing adaptive CS techniques with simple computations is therefore an important issue to be considered to provide better performance energy efficiently. Considering the resource constraint within WVSN for surveillance applications, the feasibility of such feature specific adaptation of CS for target detection and tracking is the major focus of the proposed investigations, while providing the intended detection and tracking performance with minimum energy requirement to obtain optimal utilization of energy for wireless transmission with the cost of a set of nominal preprocessing. Before presenting the system model and the proposed detection and tracking techniques, the theoretical aspects behind the proposed investigations in target detection, tracking and compressive sensing are discussed in the next chapter. Besides, some shortcomings of the previous techniques, methods to overcome these shortcomings together with the motivation for the proposed work are presented.

Chapter 3

Theoretical aspects of the proposed investigations

3.1 Introduction

WVSN is considered to be the main focus of the proposed work. Resource constraints and radio constraints are the major characteristics of WVSNs, where sensor nodes have limited energy power and limited memory space, in addition the wireless channel has limited communication bandwidth. There has been a significant mount of literature in the field of WVSNs and image processing, however our main concern is to address the problems of resources and radio constraints over WVSNs without compromising the detection and tracking performance. As a result, designing detection and tracking algorithms for WVSN-based surveillance systems is the aim to provide intended probability of detection and minimizing the chances of false detection. In addition, to be able to perform this with optimized energy-expenditure and minimum memory space. For this reason, the theoretical background of the intended investigations for target detection, tracking and CS are provided in this chapter. Various techniques for target detection, background

modeling and tracking are investigated with their pros and cons in terms of computational complexity and memory space required. Moreover, complete theoretical aspects of CS is presented including the entire CS process and image reconstruction. First an overview of a typical WVSN model is presented in the next section with the main characteristics of WVSN that differentiate it from other traditional WSNs and are the key constraints in designing a surveillance WVSN application.

3.2 General WVSN model

A typical WVSN is composed of visual sensor nodes, where each sensor node is responsible of capturing the image frame, performs some preprocessing (which might involve several operations such as foreground extraction, noise removal methods, blob formation,etc) and image processing (such as compression, target detection, tracking, classification, etc). According to the applications some processing such as detection and tracking can be performed at the base station or sink node instead of at the battery-powered sensor nodes which may relax the power constraint. The processed data is then transmitted to the sink or base station through the wireless channel or through other sensor nodes for decompression and postprocessing if required. Hence visual sensor nodes act as transmitters while the base station receives the transmitted data. A general WVSN model is demonstrated in Fig.3.1

3.2.1 Characteristics of WVSNs

Wireless visual sensor networks differs from other types of sensor networks in the nature of how the image sensors perceive information from the environment. Most sensors provide sensed data as 1D data signals. However, visual sensors captures 2D data. The additional dimensionality of the data results in richer information content as well as in a higher complexity of data processing and analysis bringing unique characteristics to WVSNs which are described in following subsection:

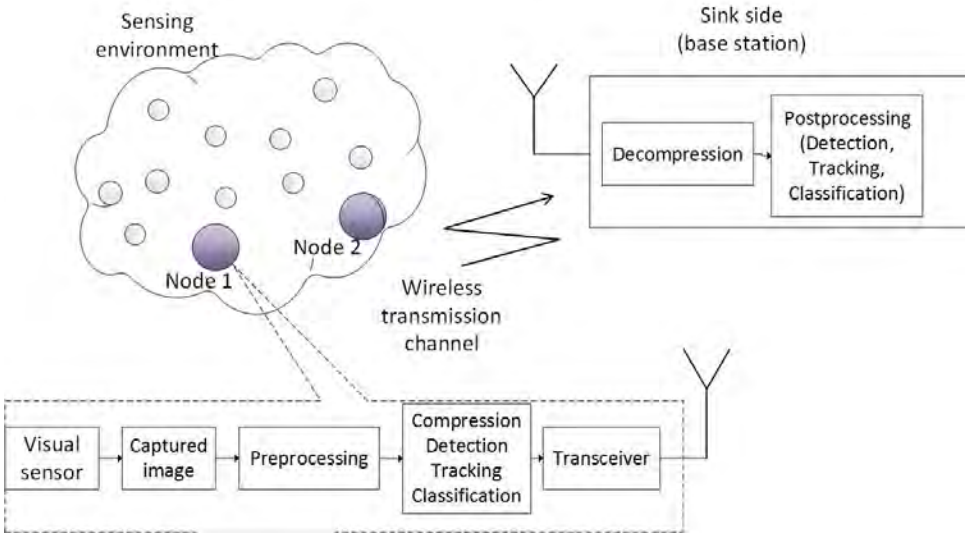


FIGURE 3.1: Typical WVSN model

3.2.1.1 Energy, storage and bandwidth constraint

Visual sensor nodes are resource constrained devices bringing the special characteristics of WVSNs such as energy, storage and bandwidth constraints which introduced new challenges [2–5]. For instance, within WVSNs each visual sensor node is powered by an attached battery and embeds a visual sensor(can be integrated with other types of sensors), digital signal processing unit, memory and a wireless transceiver. Energy is consumed by a sensor node due to any of the following tasks; image capturing, local data processing(such as target detection and tracking) and data transmission. Due to the limited battery power, the challenge of any WVSN design is to maximize the network's lifetime, hence optimum energy utilization is necessary.

Furthermore, visual sensor nodes are required to capture large data sets such as video, and still images from the environment requiring storage and large bandwidth for transmitting captured data which are quite costly in terms of energy, much more than for other types of sensor networks as visual sensor nodes are resource constraint. As well as higher complexity of data processing for object detection and tracking. To save storage and communication bandwidth, the size of captured

data can be reduced by performing some processing and compression to send only the vital data. Simple and energy-efficient compression techniques are required to reduce the size of data energy-efficiently.

Moreover, a great concern is the energy consumed during data transmission as in WVSNs most energy dissipated is during transmission. Since visual sensor nodes are required to capture large data sets such as video, and still images from the environment requiring high storage and high bandwidth for transmission which are also limited. Hence minimizing transmission energy can have more impact in energy saving and as a result maximizing the lifetime of the network [92, 93], where the energy consumed for processing is very low as compared with the transmission energy. The energy needed to transmit 1 KB over a 100m distance is approximately equivalent to the energy necessary to carry out 3 million instructions [94–96].

3.2.1.2 Local processing

Visual sensor nodes are fitted with onboard processing as local processing of the raw image helps reduce the the size of data to be transmitted through the bandwidth constrained communication channel. Instead of sending the whole captured image, the visual sensor node uses its processing ability to carry out simple processing and transmit only the required and locally processed data. Depending on the application, local processing can involve simple image processing algorithms such as background substraction (to send only the background subtracted image) or more complex image processing algorithms such as feature extraction or object classification. Moreover, for high volume datasets, compression is a very useful tool to reduce the size of raw images and transmit the compressed data.

3.2.1.3 Scalability

Most systems are configured in a variety of topologies to meet the requirements of specific applications. Configurations can be easily changed from independent networks satisfying small number of users to full infrastructure networks of thousands of users. Large number of sensor nodes may be required which can reach an extreme values of millions. Aside from the use of mobile sensors, sensors deployed on ocean surfaces and robotic sensor in military and other applications, most nodes in smart sensor networks are stationary, hence scalability is a major issue in the sense that the performance of the WVSN should not be affected by any changes in the network size. In several cases, recharging or replacing nodes' batteries are not applicable, hence adding new sensor nodes is the only way to maintain the functioning of the network. In such cases, the network should easily integrate any new sensor nodes, with minimal degradation of functionality.

3.2.1.4 Self-configuration

WVSNs consist of a large number of nodes and their potential placement is likely to be in hostile locations, it is necessary that the network be able to self-configure as manual configurations is not always feasible. Moreover, incase of sensor nodes failure which could be due to whether lack of energy or physical destruction, new nodes may join the network. To maintain a high degree of connectivity, the WVSN must be able to reconfigure itself to continue functioning.

Subsequent sections give an overview on the theoretical aspects of existing target detection and tracking techniques.

3.3 Target detection

The initial phase of visual data processing is object detection, hence if the detection is not achieved correctly, accurate tracking and higher level processing cannot be guaranteed.

Target detection using optical flow assumes a constant brightness denoted by E, where the gradient value of a pixel expected not to vary due to displacement of object[97], it is described as minimizes an energy cost function as follows:

$$E = \int\int [(I_x u_x + I_y v_y + I_t)^2] dxdy \tag{3.1}$$

Where, $I(x, y, t)$ represents the brightness value of pixel (x, y) at time t. I_x, I_y and I_t are the partial derivatives of the brightness values with respect to x, y, and t. (u_x, v_y) are the velocity vectors of optical flow estimation about $I(x, y, t)$, and (u, v) is the gradient operators of $I(x, y, t)$. The optical flow algorithm produces two equations for the velocity vectors (u_x, v_y) The brightness constancy assumes that the motion vectors are constant within small windows and that the image brightness values will not change significantly over a short period of time, this assumption is expressed as:

$$\triangle I(x, y, t) = \triangle I(x + u, y + v, t),\ |\triangle u|^2 + |\triangle v|^2 = 0 \tag{3.2}$$

$$p_x v_x + p_{xy} v_y + p_{xt} = 0 \tag{3.3}$$

$$p_{xy} v_x + p_y v_y + p_{yt} = 0 \tag{3.4}$$

Where, $p_x, p_{xy}, p_y, p_{xt}, p_{yt}$ are the products of the partial derivatives I_x and I_y. It is clear from the above equations describing the optical flow, all computations are based on partial derivatives that is computational expensive in terms of WVSNs.

As stated in previous chapter, background subtraction is a simple approach to detect the presence of targets in a video sequences as each current frame is subtracted from the background frame and to classify each pixel as background or foreground is achieved by comparing this difference by a predefined threshold. However, it is sensitive to dynamic changes in the scene such as lighting, rain, trees, etc.. The background may need to adapt to several situations to get acceptable results. In recent years, many background modeling techniques have been proposed to adapt to changes and update the background. Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Background modeling can be categorized into non-recursive and recursive techniques [98, 99] as follows:

3.3.1 Non-recursive techniques

A non-recursive technique uses a sliding-window approach for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. Non-recursive techniques are highly adaptive as they do not depend on the history beyond those frames stored in the buffer. On the other hand, the storage requirement can be significant if a large buffer is needed to cope with slow-moving traffic. Given a fixed-size buffer, this problem can be partially alleviated by storing the video frames at a lower frame-rate. Some of the commonly-used non-recursive techniques are described below:

1. **Frame Differencing:** Arguably the simplest background modeling technique [98], frame differencing uses the video frame at time $t-1$ as the background model for the frame at time t. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels

of a large, uniformly-colored moving object. This is commonly known as the aperture problem.

2. **Temporal Median filter (TMF):** TMF computes the median intensity for each pixel from all the stored frames in the buffer, where the estimate of the background is defined as the median at each pixels of all previous frames. The background X_{b_t} at time t is modeled as $X_{b_t}(x, y) = median(X_{t-i}(x, y))$, where, X_{t-i} are a set of previous frames. For each subsequent frame if $X_t(x, y) - B_t(x, y) > TH$ then a foreground object has been detected, where TH is a predefined threshold value to extract the foreground from the image. The disadvantage of the median filter is that a buffer is needed to store previous frames for modeling the background; as sensor nodes are memory constrained this technique is not a suitable candidate. Moreover, it does not perform well in noisy background situations[100].

   Considering the computation complexity and storage limitations it is not practical to store all the incoming video frames and make the decision accordingly. Hence the frames are stored in a limited size buffer. Admittedly the estimated background model will be closer to the real background scene as we grow the size of the buffer. However, speed of the process will reduce and also higher capacity storage devices will be required. In some cases the number of stored frames is not large enough (buffer limitations), therefore the basic assumption will be violated and the median will estimate a false value which has nothing to do with the real background model. This problem is partly due to the poor background estimation since the median is not correctly detected from the frames in the buffer and partly the incapability to handle the multi-modal scenes (shaking leaves are incorrectly detected as foreground).

3. **Kernel density function:** For the Kernel density function, a histogram of the L most recent pixel values (which could range from tens of frames up to hundreds according to the level of changes in the background, adaptation

rates and the available buffer size) x_i are used to represent the background density function, each smoothed with a kernel (specifically, a Gaussian kernel). An estimate of the density function is given by Eq.(3.5)

$$\hat{p}(x) = \frac{1}{L}\sum_{i=1}^{L} K_h(x - x_i) \tag{3.5}$$

K_h is the kernel function with a scaling factor h. Kernel density estimator is able to adapt quickly to the changes in the background process and able to detect targets with high sensitivity. Nevertheless, the main drawback of the kernel density estimator is its computational cost, the complexity is O(NF) evaluations of the kernel function, multiplications and additions, where F is the number of most recent pixel values(number of training images) and N is the number of image pixels. However, several pre-calculated lookup tables for the kernel function values can be used to reduce the burden of computations but still will consume high memory space [50].

3.3.2 Recursive techniques

Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update a single background model based on each input frame. As a result, input frames from distant past could have an effect on the current background model.

1. **Single Gaussian:** Among the background subtraction techniques, single Gaussian is one of the simplest background removal techniques where it calculates an average image of the scene, subtract each new frame from this image, and threshold the result to decide the presence of a target. The Gaussian distribution (or normal distribution) is a very common continuous probability distribution used in statistics to represent real-valued random variables whose distributions are not known. This basic Gaussian model

can adapt to slow changes in the scene (for example, gradual illumination changes). For each pixel in every frame $X_t(x, y)$ is represented according to normal distribution by its mean and standard deviation in the current color space. Then it is determined whether it is background pixel or not by comparing its probability of the current value with a predefined estimated threshold. Since single gaussian distribution is based only on means and variance, only two parameters for the background distribution are needed to be stored [6, 31]. The gaussian distribution is as defined in Eq.(3.6)

$$P(X_t(x,y)) = \frac{1}{\sqrt{(2Pi\sigma^2)}} exp \frac{(-X_t(x,y) - \mu)^2}{2\sigma^2} \tag{3.6}$$

Where, μ and σ are the mean and variance of the gaussian distribution respectively. The mean is the average of possible values in a given distribution and the variance measures the spread of a given distribution from the mean. For the background to adapt to gradual illumination changes, the single Gaussian model is updated by running average, the updated mean and variance μ_{t+1} and σ_{t+1} are updated as follows; $\mu_{t+1} = \alpha B_t + (1 - \alpha)\mu_t$ and $\sigma_{t+1} = \alpha(B_t - \mu_t)^2 + (1 - \alpha){\sigma_t}^2$. Here, μ_t and σ_t are the mean and variance of the current pixel in the t_0 image frame at the t_{th} time instant, X_{b_t} is the background frame, α is a constant update rate ranging between 0 and 1.

If each pixel intensity would result from specific lighting or from single mode background intensities then it would be feasible to represent the pixel value samples over time with a single distribution but unfortunately in real situation often multiple surfaces along with different illumination conditions appear in the pixel view. Hence, when a single gaussian is insufficient to model the distribution of pixel values as this model is sensitive to illumination changes and moving background targets. It is desired to model the background using Gaussian distributions, a finite mixture of Gaussians (MOG) may be used to model each pixel instead of a single one.

2. **Mixture of Gaussians (MoG):** MoG model is designed such that the foreground segmentation is done by modeling the background and subtracting it out of the current input frame, and not by any operations performed directly on the foreground objects (i.e. directly modeling the texture, color or edges). Second the processing is done pixel by pixel rather than by region based computations, and finally the background modeling decisions are made based on each frame itself instead of benefiting from tracking information or other feedbacks from previous steps. In the mixture model each pixel is modeled as a mixture of G_k-Normal distributions, where G_k Gaussian distributions are fitted to the intensities seen by each pixel up to the current time t. Having a mixture model containing G_k Gaussians, the parameters of this model have the same number of mean values, covariance matrices, and scaling factors to weight the relevance importance of each Gaussian. A weighted sum of g_k component Gaussian densities as given by Eq.(3.7)

$$p(f(\lambda)) = \sum_{i=1}^{k} w_{g_i} g(f|\mu_i, \Sigma_i) \tag{3.7}$$

Where, f is a measurements or features vector, w_{g_i} are the mixture weights, μ_i, Σ_i are the mean vector and covariance matrix, $g(f|\mu_i, \Sigma_i)$ is the component gaussian mixtures, λ is the notation where the Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities.

Although, multiple Gaussians model improves the detection probability with sharper illumination gradient in comparison to Single Gaussian model and other background modeling algorithms. However, it requires extensive computational complexity and usually the number of Gaussians needs to be carefully predefined [6, 44]. Furthermore, if a scene remains stationary for a long period of time, the variances of the background components may become very small. A sudden change in global illumination can then turn the

entire frame into foreground [98]. Subsequently, it is expected that MoG is not suitable for low-power applications

3. **Adaptive Median Filtering (AMF):**

   Like the TMF, both of these methods are based on the assumption that pixels related to the background scene would be present in more than half the frames of the entire video sequence. This is true in most of the situations unless in case of stationary foreground objects. AMF was first introduced by McFaralane and Schofield [52] which uses a simple recursive filter to estimate the median. This filter acts as a running estimate of the median of intensities coming to the view of each pixel. AMF applies the filtering procedure by simply incrementing the background model intensity by one, if the incoming intensity value (in the new input frame) is larger than the previous existing intensity in the background model. The reverse is also true, meaning that when the intensity of the new input is smaller than background model the corresponding intensity will be decreased by one. It has been proved by [52] that this trend will converge to the median of the observed intensities over time. Unlike TMF, this approach does not require storing any frames in a buffer and tries to update the estimated background model online. Hence it is extremely fast and suitable for real time applications.

   As concluded in [98], where the authors compared the performance of a number of popular background modeling techniques, AMF offers a simple alternative to MoG that produced the best results. AMF achieved competitive performance with an extremely simple implementation, the only drawback is that it adapts slowly toward a large change in background.

4. **Running average:** Running average background model [53] dynamically update the background image to adapt to the scene changing by using the

weighed sum of the current image and background image. The new background $X_{b_{t+1}}$ after background update is as follows:

$$X_{b_{t+1}}(x,y) = (1-\alpha)X_{b_t}(x,y) + \alpha X_t(x,y) \tag{3.8}$$

Where, X_t is the current frame, X_{b_t} is the current background frame and α is the updating rate, it reflects the speed of new changes in the scene updated to the background frame. However, it cannot be too large because it may cause artificial tails to be formed behind the moving objects. Because the running average background just needs to compute the weighted sum of two images, so it has low space and computational complexities satisfying the fundamental requirements for WVSN platforms such as in Cyclops [54]. Moreover, dynamically updating the background makes this model adaptive to very complex scenes.

To sum up, most Background modeling techniques are either computational intensive such as MoG or consumes high space such as non-recursive techniques to adapt to background dynamic changes, or can be simple in terms of implementation but not robust under all conditions such as sudden illumination change, fog, etc. However, it is clear from the literature that AMF and Running average can result in competitive performance as MoG with more simple implementation making them the strongest candidates to WVSNs constraints

3.4 Target tracking

Among the most of real time tracking algorithms, LMS is the simplest in terms of implementation as well as realization; which also has much lower computational complexity than the original Kalman filters and other adaptive algorithms [58]. LMS algorithm is referred to as adaptive filtering algorithm, can adapt to changes

since the statistics is estimated continuously. Adaptive filters constitute an important part of the statistical signal and image processing, it estimates a signal or next states from the received data, by minimizing the error between the reference input, which closely matches or has some extent of correlation with the desired output estimate. The LMS algorithm is initiated with an arbitrary $\mathbf{w}(0)$ at $t = 0$. The successive corrections of the weight vector eventually leads to the minimum value of the mean squared error. The weight update equations can be given as [64] by the following set of equations

$$\mathbf{w}(t+1) = \mathbf{w}(t) + u\mathbf{x}(t)\mathbf{e}(t) \tag{3.9}$$

where, $\mathbf{x}(t)$ is the input vector, u is the step size parameter, $\mathbf{e}(t)$ is the MSE between the predicted output $\mathbf{y}(t)$ and the reference signal $\mathbf{d}(t)$ which is given by

$$\mathbf{e}(t) = (\mathbf{d}(t) - \mathbf{y}(t))^2 \tag{3.10}$$

the output $\mathbf{y}(t)$ is calculated as follows

$$\mathbf{y}(t) = \mathbf{x}(t)\mathbf{w}(t) \tag{3.11}$$

On the other hand Kalman filter is an estimator predicting next states of a given process but with more complex computations [35, 60]. The state equation is given by:

$$\mathbf{c}_{t+1} = \mathbf{A}\mathbf{c}_t + \mathbf{x}_t + \upsilon_{\mathbf{t}} \ \mathbf{t} = \mathbf{0}, \mathbf{1}, ... \tag{3.12}$$

where, $\mathbf{c}_t$ is the t_{th} state vector that summarizes the past behavior, A is the state transition model, $\mathbf{x}_t$ is the known input vector, $\upsilon_{\mathbf{t}}$ is unknown zero mean white noise with covariance $\mathbf{Q}_t$

The measurement observation equation is

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{c}_t + \omega_\mathbf{t}\ \mathbf{t} = \mathbf{1}, ... \tag{3.13}$$

Where $\mathbf{z}_t$ is used later to update the unknown $\hat{\mathbf{c}}_t$, H is the observation model mapping the true state into the observed state and $\omega_\mathbf{t}$ is unknown zero mean white measurement noise with known covariance $\mathbf{R}_t$

Follows is the time update equations for the state vector $\hat{\mathbf{c}}_t^-$ and state covariance $\mathbf{C}_t^-$:

$$\hat{\mathbf{c}}_t^- = \mathbf{A}\hat{\mathbf{c}}_{t-1} + \mathbf{x}_t \tag{3.14}$$

$$\mathbf{C}_t^- = \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T + \mathbf{Q} \tag{3.15}$$

The objective is to estimate aposteriori estimating $\hat{\mathbf{c}}_t$ which is a linear combination of the apriori estimate and the new measurement $\mathbf{z}_t$ .These equations are given below

$$\mathbf{K}_g = \mathbf{C}_t^-\mathbf{H}^T(\mathbf{H}\mathbf{C}_t^-\mathbf{H}^T + \mathbf{R})^{-1} \tag{3.16}$$

$$\hat{\mathbf{c}}_t = \hat{\mathbf{c}}_t^- + \mathbf{K}_g(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{c}}_t^-) \tag{3.17}$$

$$\mathbf{C}_t = (1 - \mathbf{K}_g\mathbf{H})\mathbf{C}_t^- \tag{3.18}$$

$\mathbf{K}_g$ is the Kalman gain which is a function of the relative certainty between the measured and the current state estimates.

3.5 Compressive Sensing

As mentioned in previous chapter, CS is unlike traditional signal processing sampling where the sampling is carried out according to Nyquist theory at a frequency which is at least twice the highest frequency component found in the signal. In

contrast, CS can sample at a lower than Nyquist rate while reducing computational complexity without threatening signal recovery [78]. CS has achieved higher PSNR and lower reconstruction error compared to traditional compression techniques such as JPEG and DCT as reported in [85]. A detailed description of CS is presented next...

3.5.1 Introduction to CS

Suppose for a given set of data, sampling and compression are performed to keep only the important coefficients. According to Shannon-Nyquist sampling theory the minimum number of samples required to accurately reconstruct the image without losses is twice of its maximum frequency. It is always challenging to reduce this sampling rate as possible through undersampled measurements, hence reducing the computation power and storage with the cost of the reliability of the reconstructed data.

Recently in [12, 79, 101], authors have proposed a promising technique that results in significant amount of sample reduction while compressing sparse data named as compressive sampling or compressed sensing (CS). CS is a new paradigm for data acquisition and processing, which was originally developed for the efficient storage and compression of digital images. In their proposed theoretical model, it was reported that CS exploits the sparsity nature of images where most of the signal's energy is concentrated in few non-zero coefficients as represented in Fig.3.2 [102] . Furthermore, it is not necessary for the signal itself to be sparse but compressible or sparse in some known transform domain according to the nature of the signal, (for example. smooth signals are sparse in the Fourier basis, and piecewise smooth signals are sparse in a wavelet basis). Subsequently, CS compresses the signal during the sampling process using far fewer measurements to represent the whole set of signals such that it can be represented with far fewer samples. For example, within images/ image processing applications, this is done by projecting the image onto a set of random measurement projection matrix of

smaller size compared to that of the image without having any prior knowledge of the image. Moreover, CS is a simple process where it enables simple computations to be executed at the battery-powered encoder side (sensor nodes) whereas all the complex computations for recovery of images are left at the decoder side or receiver (not battery-powered).

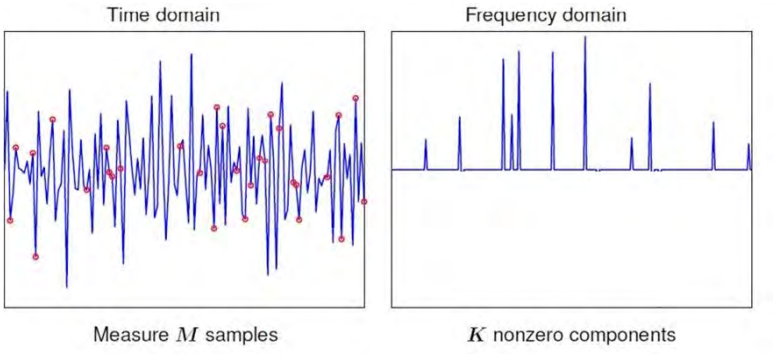


FIGURE 3.2: samples in frequency and time domain

3.5.2 Theoretical basis of CS

Suppose image $\mathbf{X}$ of size $(N \times N)$ is K-sparse where $K \ll N$, that is, only K coefficients of $\mathbf{X}$ are nonzero and the remaining are zero, thus the K-sparse image $\mathbf{X}$ is compressible. CS then guarantees acceptable reconstruction and recovery of the image from lower measurements compared to those required by shannon-Nyquist theory as long as the number of measurements satisfies a lower bound depending on how sparse the image is. Hence, $\mathbf{X}$ can be recovered from measurements of size M where $M \geq K \log N \ll N$.

Furthermore, it is not necessary for the image itself to be sparse but compressible or sparse in some known transform domain or basis, named $\mathbf{\Psi}$, according to the nature of the signal (smooth signals are sparse in the Fourier basis, and piecewise smooth signals are sparse in a wavelet basis [12, 101]). Suppose $\mathbf{X}$ is the image signal sparse in $\mathbf{\Psi}$ domain. $\mathbf{\Psi}$ is the basis invertible Orthonormal function of size $(N \times N)$ driven from a transform such as the DCT, Fourier, or Wavelet,, Eq.(3.19)

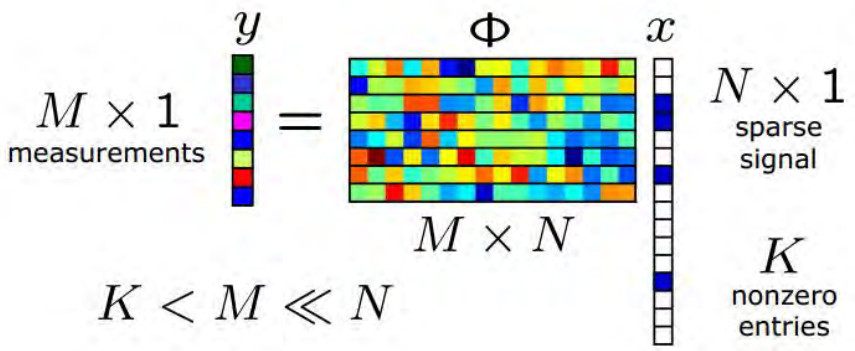


FIGURE 3.3: Compressive sensing example

shows the mathematical representation of **X**

$$\mathbf{X} = \mathbf{\Psi S} \tag{3.19}$$

Where, **S** is a matrix containing the sparse coefficients of **X** of size $(N \times N)$, $\mathbf{\Psi}$ is the transform basis to sparsify the image, $s_i =< \mathbf{X}, \psi_i^T >= \psi^T X$, $\mathbf{S} = \mathbf{\Psi^T X}$. The image is represented with fewer samples from **X** instead of all pixels by computing the inner product between **X** and $\mathbf{\Phi}$, namely through incoherent measurements **Y** in Eq.(3.20)

$$\mathbf{Y} = \mathbf{\Phi X} = \mathbf{\Phi \Psi S} = \mathbf{\Theta S} \tag{3.20}$$

Where $\mathbf{\Phi}$ is a random measurement matrix of size $(M \times N)$ where $K << M << N$. $\mathbf{y_1} =< \mathbf{x}, \phi_\mathbf{1} >$, $\mathbf{y_2} =< \mathbf{x}, \phi_\mathbf{2} >, \cdots, \mathbf{y_m} =< \mathbf{x}, \phi_\mathbf{m} >$. The compressive sensing process is illustrated in Fig.3.3 [11] showing compressed measurements **Y** produced by M different randomly weighted linear combinations of the elements of **X**.

3.5.2.1 Properties of random measurement matrix Φ

Selection of measurement matrix dimension plays significant role on the reconstruction of the image. Subsequently, for the selection of the required measurement matrix, the properties of such matrix is expected to play significant role in

the proposed investigation. Hence, the following key properties of measurement matrix are highlighted in this section.

- $\mathbf{\Phi}$ is unstructured and universally incoherent to $\mathbf{\Psi}$ which it can be paired with a variety of sparse basis $\mathbf{\Psi}$, and with every measurement one can pick up some partial information about the sparse coefficients The coherence between the sensing measurement basis $\mathbf{\Phi}$ and the transformation basis $\mathbf{\Psi}$ is measured as follows $\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \sqrt{\mathbf{n}}\mathbf{max}(< \phi, \psi >)$. Such as if $\mathbf{\Phi}$ is delta Dirac functions $\phi_{\mathbf{k}} = \delta(\mathbf{t} - \mathbf{k})$ and $\mathbf{\Psi}$ is the Fourier basis, then it can be shown that $\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \mathbf{1}$ that yields maximal incoherence. The interesting part of CS theory is that if even $\mathbf{\Phi}$ is selected uniformly at random, then with high probability, the coherence is about $\sqrt{2 \log n}$. Hence, $\mathbf{\Phi}$ can be generated by random gaussian, Bernoulli, random wavelet, or fourier measurements. Hence, it does not have to match any structure of the image but to looks more like random noise than any feature of the image[12].

- If $\mathbf{\Phi}$ is incoherent, then $\mathbf{\Psi\Phi}$ satisfies the restricted isometry property [RIP]. Let δ be the smallest quantity such that $\mathbf{X}$ obeys Eq.(3.21),

$$(1 - \delta_k)||\mathbf{X}||_{\mathbf{2}} \leq ||\mathbf{\Phi X}||_{\mathbf{2}} \leq (\mathbf{1} + \delta_{\mathbf{k}})||\mathbf{X}||_{\mathbf{2}} \tag{3.21}$$

  Where, $0 < \delta = \delta_k < 1$ for $k \leq c(\delta)M/\log(N/M)$. If Φ satisties RIP of order $2K$, then the difference between any two K sparse vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ must be preserved after the mapping from R^N into R^M

$$(1 - \epsilon)\text{dist}(\mathbf{x_1}, \mathbf{x_2}) \leq \text{dist}(\mathbf{\Phi}(\mathbf{x_1}), \mathbf{\Phi}(\mathbf{x_2})) \leq (\mathbf{1} + \epsilon)\text{dist}(\mathbf{x_1}, \mathbf{x_2}) \tag{3.22}$$

  Where, ϵ is the error.

- $\mathbf{\Phi}$ must obey uniform uncertainty principle 'UUP' and thats if $M \geq K \log N$. This can yield to optimal performance

- Should be fastly computable for both encoding image and recovery as the recovery involves repeated application of $\Psi\Phi$.
- $\mathbf{\Phi}$ should be easily implemented in hardware such as optical or analog system, or single-pixel camera [103].

3.5.3 Image reconstruction

It is required to obtain reconstructed image denoted as $\hat{\mathbf{S}}$ from the measurements $\mathbf{Y}$. Since the number of measurements are far less than the original size of the image, recovery of the image from the compressed measurements is under-determined. This is known as ill-conditioned system as any small variations of the input image can produce large variations of the output $\mathbf{Y}$, respectively. However, if the image is K-sparse, and $\mathbf{\Phi}$ obeys UUP, it has been shown in [11, 12] that the process can be inverted with high probability through the use of special convex optimization techniques such as exploiting ℓ_1-norm minimization to recover back $\mathbf{S}$ as in Eq.(3.23). To accomplish this, the receiver should know the transform domain/basis or transformation matrix $\mathbf{\Psi}$ that sparsifies $\mathbf{X}$.

$$\hat{\mathbf{S}} = argmin \parallel \mathbf{S'} \parallel_1 \text{ such that } \mathbf{\Theta S} = \mathbf{Y} \tag{3.23}$$

Hence, CS can exactly recover K-sparse images by ℓ_1 norm reconstruction, closely approximate compressible images with high probability using only $M \geq K \log N \ll N$ random measurements. Moreover, it does not assume any knowledge about the number of nonzero coordinates of $\mathbf{X}$, their locations, and their amplitudes which are assumed to be completely unknown apriori. Non linear recovery by Convex optimization problem can be reduced to linear programming known as basis pursuit or matching pursuit that has a polynomial computational complexity $O(N^3)$. While the CS literature has focused almost exclusively on problems in signal and

image reconstruction or approximation, reconstruction is frequently not the ultimate goal, as reported. The CS is a simple process, it enables simple computations at the encoder side or visual sensor nodes and all the complex computations for recovery and reconstruction of images(if recovery is needed) are left at the decoder side or receiver node [79, 104], where energy constraint may be relaxed.

Most of the compressive sensing algorithms proposed in [41, 42, 86, 87] are non-adaptive which means the random sensing measurement matrix is not chosen dynamically according to the intensity of the information collected so far. An important issue is to make $\mathbf{\Phi}$ adaptive not fixed to give better detection tracking results. In [88], energy efficient data collection in WSN using adaptive compressive sensing is proposed, it has proved that the choice of coefficients can affect both the information content and the energy expenses which is critical in WVSNs. However, target detection and tracking were not in the paper's scope. Recent efforts in adaptive compressive sensing show that by choosing the coefficients of the measurement matrix, the information content is maximized. In the proposed work, the detection reliability of targets can be improved. In [89] a heuristic to solve the optimization problem which is proven NP-hard is proposed to find a measurement matrix which maximizes the information gain per energy expenditure. It was shown that under suitable conditions, one can guarantee reconstruction of an $(N \times N)$ matrix from a smaller number of its sampled measurements by solving an optimization problem provided that this number has a lower bound on the degree of sparsity.

3.6 Chapter summary

After exploring the theoretical aspects and shortcomings of previous algorithms for target detection and tracking. It was shown that, most target detection techniques are either computational intensive to adapt to background dynamic changes, or

can be simple in terms of implementation but not robust under all conditions such as sudden illumination change, fog, etc. In addition, recursive and non-recursive background modeling techniques are investigated. Among the presented background modeling, most of the techniques are robust but not suitable for WVSNs constraints due to either their high computational high space complexities except for AMF and Running average can result in competitive performance as MoG with more simple implementation making them candidates to WVSNs constraints. On the other hand, among the existing tracking algorithms, LMS is the simplest in terms of implementation as well as realization; which also has much lower computational complexity than the original Kalman filters and other adaptive algorithms. Hence, it is a strong candidate for tracking within WVSN-based surveillance applications. The theoretical basics of CS shows it is a strong candidate in achieving our aims where it enables high compression rates through simple computations to be performed at the sensor nodes. While leaving complex computations of image reconstruction to be done at the receiver side that relax the power constraint and hence reduce the energy expenditure. To sum up, their is scope to design an adaptive compressive sensing based on degree of sparsity within the sensing environment. Besides this, the effect of compressive sensing is to be investigated in implementing target detection and tracking schemes for low-power surveillance WVSN applications to give better detection and tracking results.

In the next chapter, the system model for the WVSN-based surveillance application is introduced with all the phases involved in the proposed adaptive CS and the detection and tracking techniques.

Chapter 4

Proposed detection and tracking model using CS

4.1 Introduction

WVSNs deal with large data sets of videos and images resulting in high demand on memory space and higher complexity of data processing and analysis. As well as high bandwidth demand for transmitting the large image data. To represent the captured data in such a way to save storage due to memory constraint, an adaptive block CS technique is proposed to represent the data with just few number of measurements. Consequently, it is expected to save bandwidth requirement for transmission and processing power. As illustrated in previous chapter, the CS is a simple process where it enables simple computations to be executed at the encoder side (sensor nodes) and all the complex computations for recovery of images are left at the decoder side or receiver. In addition to power and memory constraints, CS should not affect quality of image (as denoted PSNR) for later detection and tracking.

In this chapter an outline for our proposed detection and tracking model using CS is presented with the notations used throughout the thesis. Furthermore, as energy is a critical constraint in WVSNs, an energy model is described to calculate energy dissipated which is used later as a performance indicator comparing the proposed block CS model with traditional CS and without CS. A detailed description of the proposed adaptive block CS algorithm together with the training/calibration phase and the phases involved in the detection and tracking techniques including target extraction, noise removal using morphological operation and LMS tracking are then described.

4.2 Proposed system model

Consider a WVSN-based surveillance application model, composed of V number of visual sensor nodes allocated to share a viewable range and cover the required sensing region, and one or more receiver/sink node (base station) at fusion center. Lets assume most features of the targets are known to the monitoring center and the existence and the location of targets are required for monitoring. The receiver also has prior explicit information of the background (the background is initially assumed to be static for simplicity). Each sensor node is assumed to be in 'wake-up' state only with the presence of a target within its area of coverage and required to capture images to form a video sequence. We assume that only single view multi-target tracking is achieved to keep minimum number of visual sensor nodes in a wake-up state to optimize the use of nodes and save battery life which is limited in WVSNs. Hence, each visual sensor node is responsible of capturing the image, preprocessing, compressing and transmitting the compressed captured frame.

For sensor V_i at the time where the sensor node enters a 'wake-up' state, the time reference for the frame count is assumed to be $t = 0$. Hence, a single snapshot at $t = 0$ is expected to be stored within the memory allocated at the sensor node; that

is assumed to be the background for the intended target tracking; will be denoted as $\mathbf{X_b}$. Following frames $\mathbf{X_t}$ with $t > 0$ are subsequent captured frames in the video sequence, where $\mathbf{X_b}$ and $\mathbf{X_t}$ are of size $(N \times N)$. As in the general WVSN model (Fig.3.1) described in previous chapter, some preprocessing might be required. In our case, to assure sparsity within the image frame, the foreground target is extracted first by background subtraction by subtracting $\mathbf{X_t}$ from $\mathbf{X_b}$ resulting in the difference frame $\mathbf{X_d}$. Hence, instead of producing the compressed measurements for $\mathbf{X_b}$ and $\mathbf{X_t}$ separately, the compressed measurements are produced directly for $\mathbf{X_d}$, as the difference frame is always sparse regardless the sparsity nature of real frames.

CS adaptively chooses the compression rate according to the sparsity nature of difference frames which varies from one dataset to another. The training/calibration phase is pre the CS phase and is discussed later in Sec.4.3.2.1. Afterwards, the block CS divides $\mathbf{X_d}$ into B blocks $\mathbf{X_{blk}}$ each of size $(N_b \times N_b)$ as will be presented in sec.4.3.2.2. CS is then performed by multiplying blocks $\mathbf{X_{blk}}$ containing the target by random projection measurement matrices $\mathbf{\Phi}$ producing the compressed measurements for each block denoted later as $\mathbf{Y_{blk}}$ to be ready for transmission through the wireless channel. At the receiver side, the received compressed data is to be decompressed for the reconstruction and recovery of the estimate data $\hat{\mathbf{X}}_\mathbf{d}$. As mentioned, $\mathbf{X_b}$ is known to the receiver making it possible to estimate and reconstruct the original test frame denoted as $\hat{\mathbf{X}}_\mathbf{t}$ by adding $\mathbf{X_b}$ to $\hat{\mathbf{X}}_\mathbf{d}$. Finally, the system detects and tracks the moving targets. The system model of the proposed processing for the WVSN-based surveillance application at individual visual sensor nodes and at the sink or base station are shown in Fig.4.1 and 4.2, respectively.

After describing the proposed system model for target and tracking within WVSNs, an energy model is presented next.

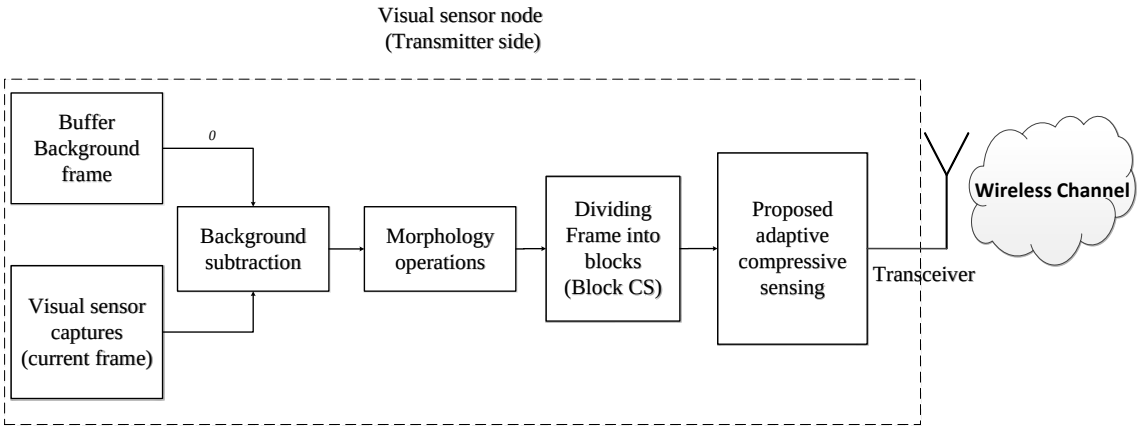


FIGURE 4.1: The proposed model for the visual sensor node

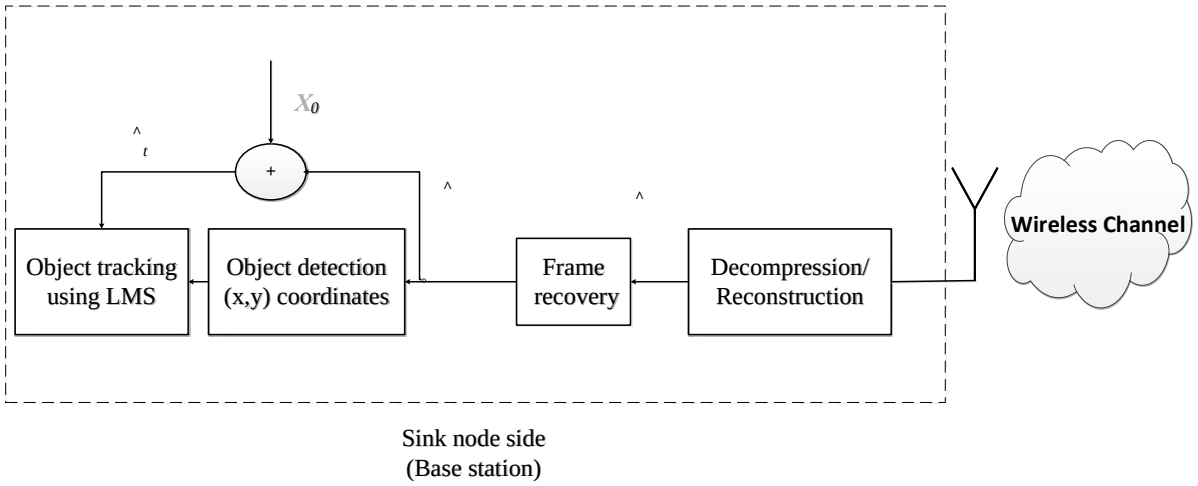


FIGURE 4.2: The proposed model for the sink side or base station

4.2.1 Energy model

Currently, there is a great research in the area of low-energy radios. In our work, the same energy model as in [105] is used, where energy cost dissipated by a node over a distance d is denoted by E_{tx} as shown in (4.1).

$$E_{tx} = E_{elec} * k + e_{amp} * k * d^2 \tag{4.1}$$

Where, k is size of data (samples) transmitted, $E_{elec} = 50nJ/bit$, is the energy being used to run transmitter and receiver circuit, $e_{amp} = 100pJ/bit$ for the transmitted amplifier.

In WVSNs, most energy dissipated is during the transmission and reception, in our case the reception is the base station node which is assumed not to be battery-powered. Hence minimizing transmission energy can have more impact in energy saving [92, 93] where the energy consumed for processing is very low as compared with the transmission energy. The energy needed to transmit 1 KB over a 100m distance is approximately equivalent to the energy necessary to carry out 3 million instructions [94–96].

4.3 Proposed detection and tracking model

This work proposes an adaptive block compressive sensing model which is expected to reduce energy consumption, space requirements and communication overhead. Each sensor node is set to capture a target when entering its monitoring area either by periodic monitoring or if being notified by a neighbor sensor node that a target is to enter its monitoring area. The sensor node segments the target by background subtraction followed by morphological operations. The sensor node afterwards applies the proposed adaptive block CS and transmits the compressed measurements to the receiver for reconstruction and tracking the target using the modified LMS technique. Next in subsequent sections, all the procedures undertaken during the entire surveillance process are illustrated.

4.3.1 Proposed detection technique

4.3.1.1 Background subtraction

Due to the limitation of resources in WVSN, CS is applied to the background subtracted frame instead of the whole frame. Hence, background subtraction needs to be performed first before applying CS to increase the sparsity of images, as the difference image always provides higher degree of sparsity regardless of the sparse nature of the original images. As a result, reducing the required size of transmitted measurements to the base station. Assuming the visual sensor node has captured an image denoted as $\boldsymbol{X}_t$. The target afterwards is detected based on thresholding the absolute difference between current frame $\boldsymbol{X}_t$ and background frame $\mathbf{X_b}$, $\boldsymbol{X_d} = |\boldsymbol{X_t} - \boldsymbol{X_b}| > \boldsymbol{\gamma}$, where γ is a given threshold to extract the foreground target as in Eq.(4.2),

$$X_d(i) = \begin{cases} |X_t(i) - X_b(i)| \text{ (foreground pixel)} & |X_t(i) - X_b(i)| > \gamma \\ 0 \text{ (background pixel)} & otherwise \end{cases} \tag{4.2}$$

The value for γ is chosen in order to reduce scattered noise that could exist in the background due to many factors such as rain, dust, illumination changes, trees movements, etc. Its value is determined as in [106], where the image is divided into blocks (as will be illustrated in Sec.4.3.2.2 for block CS) and each block should be identified by a unique value that allows to properly choose the threshold. Statistic functions are commonly used such as mean, median, mean of minimum and maximum values of the local intensity distribution as they largely depends on the input images. In this work, mean value is used as threshold to describe each image block. The threshold value helps reduce unwanted background subtraction noise and at the same time without causing disconnected targets as possible. Where, lower values of γ will result in more noise and higher values will

result in disconnected targets. In both cases, resulting in either lower probability to detect the target or more preprocessing to overcome this problem.

To adapt to changes in the background, running average is applied to continuously update the background with changes such as moving trees, rain, or unwanted moving objects. As stated previously in Sec.2.2.1 Running average is a simple background modeling technique with low space and computational complexities. Background modeling uses the new video frame and previous background frame to calculate and update the background model to provide a statistical description of the entire background scene after adapting to changes in the background. The background update is as follows:

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha X_t(x, y) \tag{4.3}$$

Where, B_{t+1} is the updated background model, B_t is the previous background frame, X_t is the current captured frame and α is the updating rate, it reflects the speed of new changes in the scene updated to the background frame. However, it cannot be too large because it may cause artificial tails to be formed behind the moving objects. Fig.4.3(a) and 4.3(b) show the background frame before and after background modeling with an update rate $\alpha = 0.7$, the background subtracted frame after updating the background is shown in Fig.4.3(c)

4.3.1.2 Morphology operations and blob extraction

Once the foreground is detected, morphology operations as image postprocessing are then applied for the removal of the remaining noise (if present) after the background modeling. Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations probe an image with a small shape or template known as a structuring element. The structuring element is a small binary matrix of pixels, each with a value of zero or one specifying the shape of the structuring element (such as a disc, rectangle, square,etc). The

(a) Current background frame

(b) Background frame after running average

(c) Background subtracted frame

FIGURE 4.3: (a) and (b)Background frames before and after background modeling and (c) background subtracted frame

structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels, afterwards, according to the operation applied some pixels are converted from ones to zeros and vice versa [107]. The most basic morphological operations are dilation and erosion. Dilation adds pixels (convert zeros to ones) to the boundaries of targets in an image to help fill holes and forms connected blobs, while erosion removes excess pixels (noise) to eliminate small unwanted details by setting these pixels to zeros according to the structuring element specified. The number of pixels added or removed from an image containing potential target depends on the size and shape of the structuring element used to process the image[107]. Other morphological operations are a compound of the basic operations erosion and dilation. The most commonly used compound operations are the opening and closing. Opening is an erosion followed by a dilation, it is so called because any regions that have survived the erosion during noise removal are restored to their original size by the dilation. While closing is a dilation followed by an erosion where it can fill holes in the regions while keeping the initial region sizes. The simplicity of Morphological operations made it one of the strongest candidates for processing within WVSN based applications [108]

In the context of our work, after background subtraction an opening or closing operations are then applied depending on the nature of images. Sometimes an opening operation is performed where erosion is first applied as a noise removal method by applying the specified structuring element to remove unwanted pixels, followed by dilation to fill the holes within target objects forming a connected object blob by linking the unconnected parts of the target. Hence, any regions that have survived the erosion are restored to their original size by the dilation. Or a closing operation, obtained by dilation of the image using the specified structuring element to form a connected object blob, followed by erosion of the resulting image to restore the original size of objects. It can fill holes in the regions while keeping the initial region sizes[107]. Fig.4.4 shows the blob formation after background

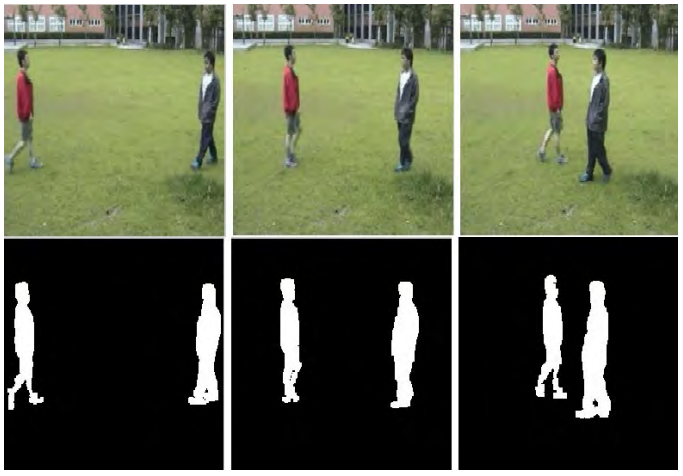
subtraction and morphological operations. A bounding box containing the target is then formed to calculate the target's coordinates.

In some surveillance scenes where targets are to be tracked in the streets, vehicles might be present in the scene such as in Fig.4.5(a). Hence, after all objects are detected, if vehicles are present in the frame, their bounding boxes are ignored as in Fig.4.5(b) and afterwards unwanted objects are eliminated from the background subtracted frame by checking the dimensions of the bounding boxes as in Fig.4.5(c).

4.3.2 Proposed adaptive block Compressive sensing

As mentioned in the previous chapter, instead of producing the compressed measurements $\boldsymbol{Y}_b$ and $\boldsymbol{Y}_t$ for $\boldsymbol{X}_b$ and $\boldsymbol{X}_t$, respectively. For lowering processing requirement, $\mathbf{X_t}$ is subtracted from $\mathbf{X_b}$ first as CS exploits the fact that difference frame $\mathbf{X_d}$ are always more sparse regardless the nature of the real frames. Hence, it is proposed to perform CS on $\boldsymbol{X}_d$ at the visual sensor node by being projected onto random sensing measurement matrix $\mathbf{\Phi}$ producing the compressed measurements $\boldsymbol{Y}_d$. These compressed measurements are then transmitted through the wireless channel to the receiver side for decompression and reconstruction of the real frame for further processing, where a specified target is to be detected and tracked. Furthermore, it is required to analyze the relation between the number of non-zero pixels and the total number of pixels in a given frame to find the optimal number of compressed measurements required to result in accurate tracking. Below are the steps undertaken during the entire process of CS applied in the proposed work, next subsections present the proposed adaptive block CS.

- **Step 1:** $\mathbf{\Phi}$ is a randomly chosen sensing matrix of size $M \times N$, where $M \ll N$
- **Step 2:** produce the compressed measurements $\mathbf{Y_d} = \mathbf{\Phi X_d}$
- **Step 3:** sensor nodes transmits $\mathbf{Y_d}$ through the wireless channel



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

FIGURE 4.4: First row in (a)(b) and (c) shows test frames and background subtraction results in second row

(a) Bounding boxes for all objects detected

(b) Bounding boxes for only human targets

(c) Background subtracted frame eliminating vehicles

FIGURE 4.5: Detected objects for dataset"5"

FIGURE 4.6: The reconstructed original image for "Walking men"

- **Step 4:** at the receiver side, $\mathbf{\Phi}$ must be known for the decompression of $\mathbf{Y_d}$. $\hat{\mathbf{X}}_\mathbf{d}$ is reconstructed from the compressed measurements $\mathbf{Y_d}$, resulting in a frame with only the foreground target present.

- **Step 5:** the original image $\hat{\mathbf{X}}_\mathbf{t}$ is then obtained by adding $\hat{\mathbf{X}}_\mathbf{d}$ (the reconstructed background subtracted image) to the masked background frame (masking the targets locations), the background frame is also assumed to be known to the receiver side apriori as in Fig.4.6.

- **Step 6:** the targets locations are obtained after reconstructing the real frame producing a trajectory for the complete path of each moving target

4.3.2.1 Proposed Adaptive CS

For any given dataset, different M and $\mathbf{\Phi}$ are needed, as stated earlier the value of M is inversely proportional to the degree of sparsity of an image. If the same

value of M is used for all different datasets, it is expected that the reliability of target detection will be different as the degree of sparsity varies from one image to another. For this reason there is a great challenge for adaptive CS by making M variable depending on how sparse the image is. Adaptive CS dynamically chooses the compression rate according to the sparsity nature of frames which varies from one dataset to another. In contrast to static compression rates, different datasets have different sparsity levels, hence if the same dimension of the measurement matrix is used for more sparse images this will result in a waste of energy where more compression could have been applied. And for less sparse images, the quality after reconstruction will be affected which in returns degrades the detection performance. As a result, dynamic size of measurement matrices results in saving energy, space requirements, as well as channel bandwidth. For the adaptive CS, the aforementioned CS process is preceded by a calibration phase. During that phase an Automatic Repeat Query (ARQ) transmission protocol is used between sensor nodes and the receiver side, as a feedback is needed for the adaptation phase. A dictionary is constructed for different values of M and corresponding sensing matrices Φ. For each dataset the sparsity level is calculated by finding the ratio between the number of non-zero pixels and the total number of pixels in a frame. At the end of each adaptation/calibration phase, the dictionary is updated with the chosen M and Φ for the equivalent sparsity level that can be used later for other datasets with the same sparsity levels. Initially, an arbitrary value of M is chosen according to a sparsity measure and is used to obtain the compressed measurements $\mathbf{Y_d}$. The sensor node is then set to transmit $\mathbf{Y_d}$ to the receiver side where the image is to be reconstructed, and based on the reconstruction error a decision is made whether the reconstruction is satisfactory or not. In case the reconstruction results are satisfactory, the receiver node sends a 'zero' flag through the feedback channel ending the calibration phase; otherwise a 'one' flag is to be sent. While the sensor node receives a 'one' flag, it is expected to change the value of M and change $\mathbf{\Phi}$ accordingly, the sensor node repeats the search for an optimum value of M at the CS adaptation process till it receives a zero feedback

from the receiver. At this point, the optimum values for M and $\mathbf{\Phi}$ obtained are used next in the CS process. Fig.4.7 shows a flow chart summarizing the entire adaptive CS process.

4.3.2.2 Proposed block CS

To exploit the fact that the difference frame is always sparse, instead of compressing the whole frame, the image is divided into blocks and only blocks with non-zero pixels (containing the target) are expected to be compressed and transmitted. This strategy is expected to help reduce the required value of M, subsequently, save the communication bandwidth and preserve the energy at the transmitter side. To illustrate this process, Fig.4.8 shows a (256×256) background subtracted frame which is then divided into 16 blocks (64×64). It can be found that only 7 blocks have non-zero pixels as shown in Fig.4.9 and the rest of the blocks are all zeros, hence do not required to be processed. An index for each block is embedded to the compressed measurements before transmission such that the receiver side can reconstruct the whole image correctly again in the correct order. It has to be noted that the missing blocks (which has not been transmitted) are to be considered as pixels sets with all zero values

The same procedures as in sec.4.3.2 are undertaken with few additional steps to divide each frame into blocks, below is a summary of the proposed block CS:

- **Step 1:** A frame with dimension $(N \times N)$ denoted as $\mathbf{X_d}$ is divided into B blocks of size $(N_b \times N_b)$ each, where $\frac{N^2}{N_b^2} = B$, each block is denoted as $\mathbf{X_{blk}}$
- **Step 2:** for each block $\mathbf{X_{blk}}$ with non-zero pixels, perform the following steps
  - **Step 3:** $\mathbf{\Phi}$ is a randomly chosen sensing matrix of size $M_b \times N_b$, where $M_b \ll M$ and $M_b \ll N_b \ll N$
  - **Step 4:** produce the compressed measurements $\mathbf{Y_{blk}} = \mathbf{\Phi X_{blk}}$

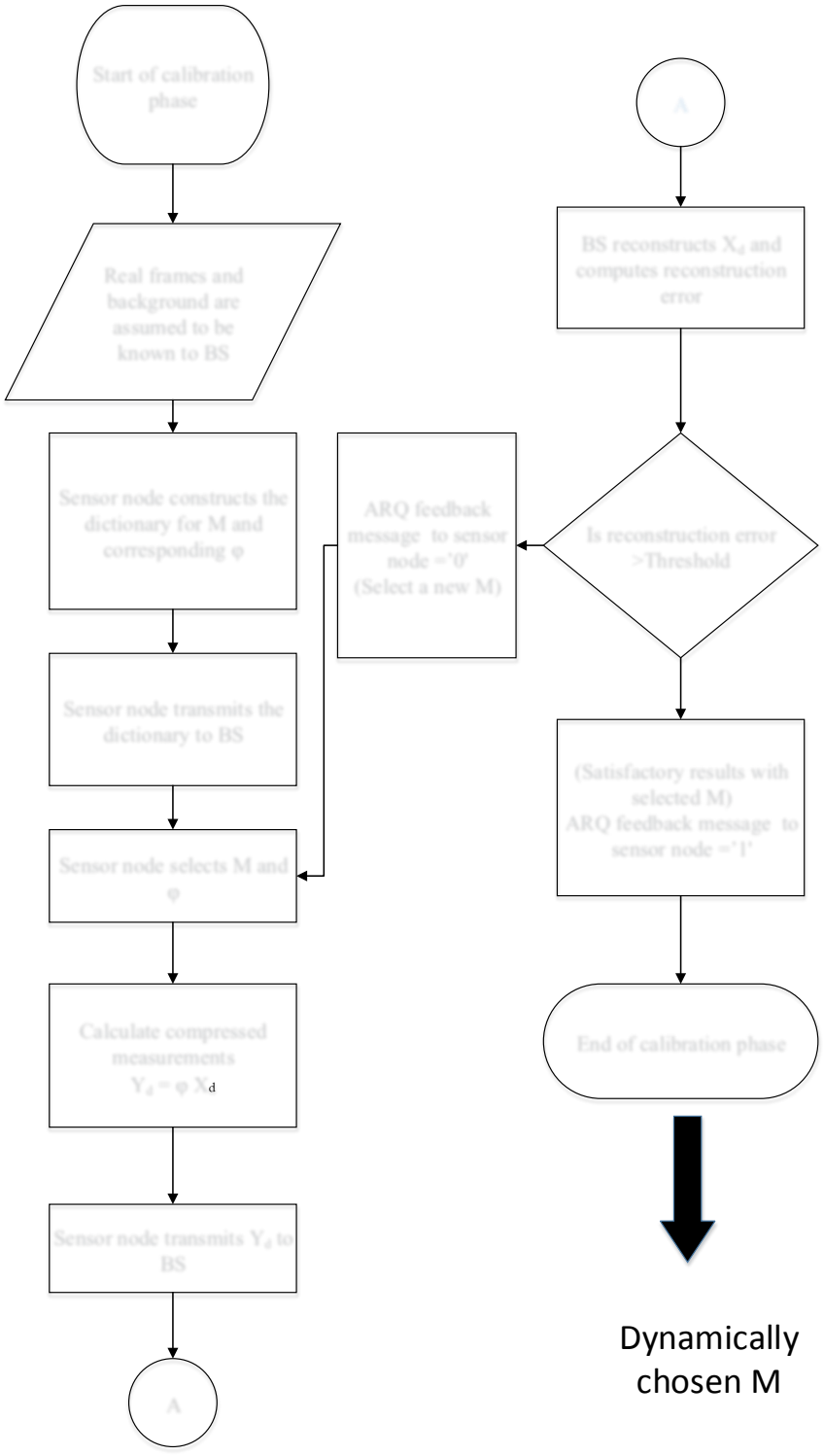
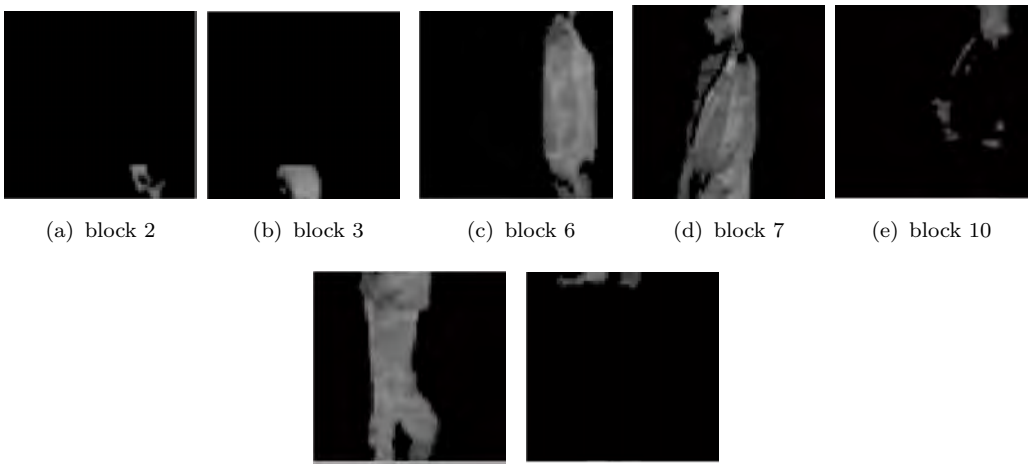


FIGURE 4.7: Flowchart for the training phase of the adaptive CS process

FIGURE 4.8: Background subtracted frame



(a) block 2 (b) block 3 (c) block 6 (d) block 7 (e) block 10

(f) block 11 (g) block 15

FIGURE 4.9: Blocks containing the targets (non-zero pixels)

  - **Step 5:** sensor nodes transmits the compressed block $\mathbf{Y_{blk}}$ together with an index of the block number through the wireless channel
  - **Step 6:** at the receiver, $\mathbf{\Phi}$ is assumed to be known for the decompression of $\mathbf{Y_{blk}}$. $\mathbf{\hat{X}_{blk}}$ is reconstructed from the compressed measurements $\mathbf{Y_{blk}}$.

- **Step 7:** using the index number transmitted with every block all received blocks are placed together in the correct order resulting in a frame $\mathbf{\hat{X}_d}$ with only the foreground target present.
- **Step 8:** the real frame $\mathbf{\hat{X}_t}$ is then obtained by adding $\mathbf{\hat{X}_d}$ to the background frame $\mathbf{X_b}$ which is also assumed to be known to the receiver side apriori.
- **Step 9:** the targets locations are obtained after reconstructing the real frame producing a trajectory for the complete path of each moving target

4.3.3 Proposed tracking model

4.3.3.1 Least mean square (LMS)

The LMS algorithm, introduced by Widrow and Hoff in 1959 is an adaptive algorithm, which uses the same principles as the method of the Steepest descent, but where the statistics are estimated continuously. LMS algorithm is referred to as adaptive filtering algorithm since the statistics are estimated continuously, hence it can adapt to changes. LMS incorporates an iterative procedure during the training phase where it estimates the required coefficients to minimize the MSE. This is accomplished through successive corrections to the expected set of coefficients which eventually leads to the minimum MSE. LMS algorithm is relatively simple, has much lower computational complexity than the original Kalman filters and other adaptive algorithms [58]; it does not require correlation function calculation nor does it require matrix inversions [66]. Moreover, suitable for real

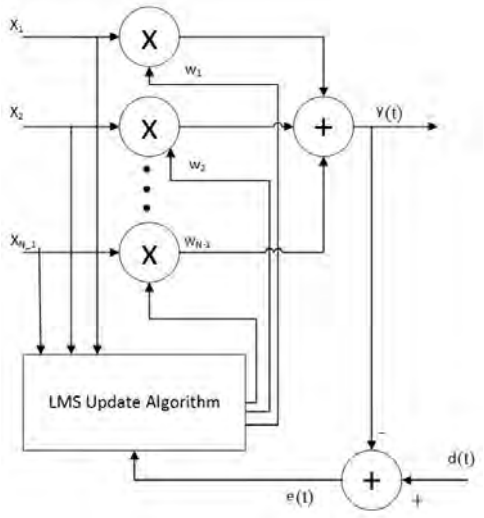


FIGURE 4.10: An N-tap LMS adaptive filter

time images applications such as motion estimation and target tracking , where it showed robustness on fast moving targets and non-linear moving targets even in noisy environments as reported by the authors in [65, 72, 73, 75].

As shown in Fig.4.10 [68], the outputs are linearly combined after being scaled using corresponding weights. The weights are computed using LMS algorithm based on MSE criterion hence the spatial filtering problem involves estimation of a signal from the received signal, by minimizing the error between the reference signal, which closely matches or has some extent of correlation with the desired signal estimate and the output. The LMS algorithm is initiated with an arbitrary value $\mathbf{w}(0)$ for the weight vector at $t = 0$. The successive corrections of the weight vector eventually leads to the minimum value of the mean squared error. The weight is updated as in Eq.(3.9), where $\mathbf{x}(t)$ is the input signal, u is the step size parameter, $\mathbf{e}(t)$ is the MSE (as in Eq.(3.10)) between the predicted output $\mathbf{y}(t)$ from Eq.(3.11) and the reference signal $\mathbf{d}(t)$.

u is selected by the autocorrelation matrix of the filter inputs. In other words, the tap-weights can converge to an optimum result if and only if the step-size parameter u is selected as $0 < u < 1/\lambda_{max}$ where, λ_{max} is the maximum eigenvalue of the autocorrelation matrix which has a relationship of the input signal $\mathbf{x}(t)$. If u

is chosen to be very small then the algorithm converges very slowly. A large value of u may lead to a faster convergence but may be less stable around the minimum value. The smallest the eigen value spread the faster the convergence rate. Eigen value spread is defined as the ratio between the maximum and minimum eigen values.

4.3.3.2 Variants of LMS

The simplicity of implementation of the LMS filter causes new developments for this algorithm that enhance the capability and performance of this filter. Although the LMS adaptive filter is popular for its simplicity but reduction of the complexity of the LMS filter has received attention in the area of adaptive filters and even simpler approaches are required for many real-time applications[73]. There are several variants of the LMS algorithm present in the literature that deal with the shortcoming of its basic form and aim for lower computational complexity and faster adaptation processes as it is required for high speed communication as well as to be applicable in real time applications where the time is critical. A simple modification of LMS is called the Sign LMS algorithm [109], it uses the **sgn** function to update the weights as follows:

$$sgn(\mathbf{x}(t)) = \begin{cases} 1 & \mathbf{x}(t) > 0 \\ 0 & \mathbf{x}(t) = 0 \\ -1 & \mathbf{x}(t) < 0 \end{cases} \tag{4.4}$$

Signed LMS uses the **sgn** of the error to update the weights as in Eq.(4.5)

$$\mathbf{w}(t+1) = \mathbf{w}(t) + u\mathbf{x}(t)\ \mathrm{sgn}(\mathbf{e}(t)) \tag{4.5}$$

In clipped LMS [73, 110], clipped input is used to update the weight instead of the input itself, as in Eq.(4.6)

$$\mathbf{w}(t+1) = \mathbf{w}(t) + u\ \mathrm{sgn}(\mathbf{x}(t))\mathbf{e}(t) \tag{4.6}$$

In [73], a new version of the clipped LMS known as quantized LMS is proposed where a quantization scheme is used to represent the input according to a modified sign function **msgn** function where instead of representing the input by a two level signals, it is quantized into a three level signals as defined below:

$$msgn(\mathbf{x}(t)) = \begin{cases} 1 & \mathbf{x}(t) > \varrho \\ 0 & -\varrho < \mathbf{x}(t) < \varrho \\ -1 & \mathbf{x}(t) < -\varrho \end{cases}$$

Where ϱ is a specified threshold. The implementation of such a modified adaptive filter is fast and has a reduced computational complexity as for those times when the input is less than the specified threshold $\mathbf{msgn}(\mathbf{x})$ is to be equal to zero and no coefficient adaptation for the corresponding weight needs to be performed. This means that some of the time-consuming operations in the weight update can be neglected.

Another version of the LMS is the Normalized LMS (NLMS) [110], it forces the input samples to have a constant norm. Hence, it improves the convergence speed in a non-static environment by introducing a variable adaptation rate.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \frac{u\mathbf{x}(t)\mathbf{e}(t)}{\mathbf{x}^T(t)\mathbf{x}(t)}$$

In the Newton LMS [64, 68], the weight update equation includes whitening in order to achieve a single mode of convergence. For long adaptation processes the Block LMS [64, 68]is used to make the LMS faster where, the input signal is divided into blocks and weights are updated block wise.

4.3.3.3 Proposed iterative quantized clipped LMS

For the proposed model, LMS is used to predict target's locations, a quantized clipped LMS technique is used with threshold values chosen to use the proposed tracking model [73]. To guarantee least MSE an iterative method is proposed with a defined threshold of acceptable MSE, in addition to a threshold on the maximum number of iterations to maintain the algorithm's applicability for real time applications which is one of the main WVSNs properties.

$$mqsgn(\mathbf{x}(t)) = \begin{cases} 1 & \mathbf{x}(t) > D_1 \\ 0 & -D_2 < \mathbf{x}(t) < D_1 \\ -1 & \mathbf{x}(t) < -D_2 \end{cases}$$

Where, D_1 and D_2 are threshold values used to clip the input data. The modified iterative quantized clipped LMS algorithm consists of two main phases;

- **Learning Phase:** The LMS algorithm learns the targets locations to estimate new updates for the filter's weights till minimizing the MSE.
- **Prediction Phase:** The updated weights from the previous phase are then used to predict the target's next locations. The MSE will start rising again if the target changes its direction or speed, in that case the LMS needs to undergo the learning phase for further weight updates before next predictions.

The summary of the application of the LMS algorithm in the proposed tracking model:

- Prepare the input data and set the filter length
- Learning phase

  While MSE < error's threshold and number of iterations < iteration's threshold

-Determine the output data using modified iterative quantized clipped LMS algorithm

-Calculate the MSE

-Update the filters weights according to the MSE

- Repeat the above steps till finishing the learning phase
- Predict the next locations using the updated weights
- If the MSE fell below some defined threshold repeat the learning phase

4.4 Chapter summary

In this chapter the proposed target detection and tracking model using adaptive block CS for WVSN-based surveillance application has been introduced with all the phases the algorithm passes through. First an overview on how a new frame is being captured by the visual sensor node and a complete model for the WVSN-surveillance application are explained. Next, background subtraction is first applied to increase the sparsity of image frames and as a result reach higher compression rates. Background modeling using running average is then presented to update the background with changes followed by morphology operations for noise removal and connected blob formation. Running average is chosen as it is a simple background modeling technique with both low space and computational complexities, hence applicable for WVSN's constraints. The proposed adaptive block CS phase is then described where the images are divided into blocks and adaptively applying CS to relative blocks containing the target by choosing the number of compressed measurements according to the sparsity nature of each dataset. The proposed adaptive block CS is expected to reduce the size of transmitted data as only blocks containing the target are transmitted instead of transmitting the entire image and as a result saves communication bandwidth and transmission

energy. Moreover, adaptive CS is expected to save energy as appropriate compression rates are chosen according to sparsity level of datasets. Finally, the iterative quantized LMS is introduced due to its simplicity and low power computations as a tracking technique to predict target's next locations and to test the effect of CS on the tracking performance, after the compressed image is transmitted through the wireless channel and reconstructed at the receiver side.

In the following chapter, experiments and results are conducted to illustrate the performance of the adaptive block CS, the effect of different sensing matrices is tested on different performance indicators. The performance of the quantized LMS tracking techniques is then illustrated in terms of MSE and trajectory tracking.

Chapter 5

Experimental work and discussion of the proposed detection and tracking model

5.1 Introduction

Based on the system model proposed in previous chapter, simulations and experiments are conducted to evaluate the performance of the adaptive block CS and the detection and tracking algorithm. Simulations are performed for the WVSN-based surveillance application in both outdoor and indoor scenes for single and multi-target tracking.

5.1.1 Experimental setup

Background and target's appearance are assumed to be static to investigate the effect of the proposed adaptive CS on the detection and tracking algorithms, accordingly datasets are chosen to reflect these assumptions except for some background

variations such as non-stationary objects(later for the future work simulations will be performed on dynamic background and target appearance models). Moreover, to illustrate the relation between the number of compressed measurements required for CS to guarantee reconstruction and how sparse the image is, simulations are performed on different schemes resembling both indoor and outdoor schemes from standard datasets chosen with different sparsity levels to investigate the effect of sparsity on the compression rates and how dynamically compression rates are selected; dataset"1":'WalkingMen' is chosen to resemble an outdoor scenes for multi target tracking captured by [111]. While dataset"2":'OneStopNoEnter2front', dataset"3":'Walk1' and dataset"4": 'OneStopMoveNoEnter1cor' filmed for the Context Aware Vision using Image-based Active Recognition (CAVIAR) data set. CAVIAR is a project of the European Commission's Information Society Technology program found in [112] for indoor scenes tracking a single target from two different views; top view with a wide angle lens camera and a corridor side view, for dataset"2" and dataset"3" respectively. dataset"5":'Walking' resembles an outdoor scene with a street view for cars and targets tracking from PETS surveillance datasets [113].

Simulations were carried out on an Intel dual Core $i5$ 2.40GHz CPU with 3M cache and 4G RAM, code is written using Matlab v.7.6.0 and experiments were conducted by averaging total number of frame sequences of each dataset. Since each image frame is divided into blocks and hence each block has different sparsity nature, constant threshold values for background subtraction will not be applicable. To properly choose the threshold, statistic functions such as mean, median, mean of minimum and maximum values of the local intensity distribution are commonly used as they largely depends on the input images. In the simulation, mean value is used as threshold to describe each image block. The α value for the running average background update is set to 0.7

As stated in previous chapters, adaptive CS is expected to overcome the WVSN resource constraints such as memory limitations, communication bandwidth and

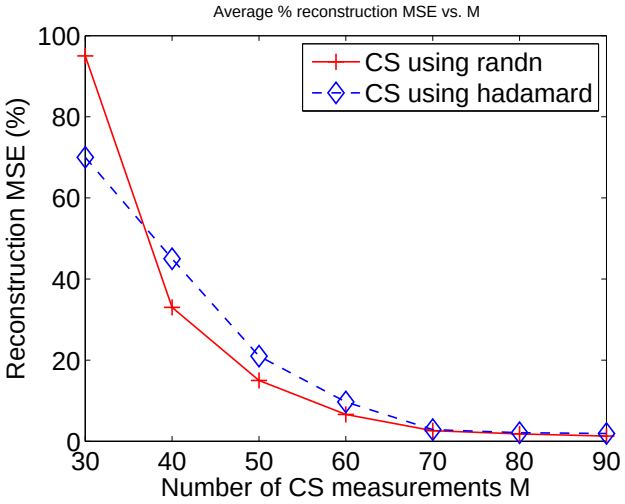
battery constraints. This is illustrated as the reduction in the size of captured images, which as a result saves memory space and communication bandwidth as the size of transmitted data is reduced. Furthermore, this reduction results in energy saving as most energy is dissipated during transmission. First adaptive CS results are presented comparing MSE, PSNR and correlation coefficient on different sensing matrices and compressed measurements, results of examining various compression rates on the trajectory tracking are shown. The effect of block CS on increasing compression rates and reducing MSE are then discussed. Next section illustrates the results of comparing several variants of LMS and different convergence rates with respect to MSE. In addition, comparing the proposed iterative LMS technique with the ground truth. Next, the LMS tracking is compared with Kalman filter. Finally, a section on computational complexities summarizes the energy dissipated for block CS versus traditional CS and without CS. Moreover, it summarizes the computation time for the proposed block CS and LMS algorithm.

5.2 Adaptive block CS

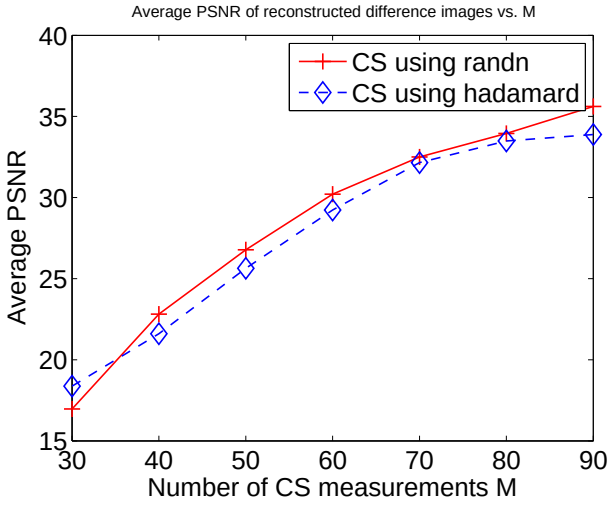
Percentage MSE and PSNR are used as performance indicators to test the reliability of the proposed adaptive block CS. MSE and PSNR are compared for different number of CS measurements M, where the percentage MSE is the percentage reconstruction error measured between real and reconstructed frames and PSNR is measured after frames recovery to reflect the quality of image reconstruction which will later on reflects the ability of reliable tracking. The background frame and $\mathbf{\Phi}$ are known to the receiver node and during the calibration phase, it is assumed that real frames are also known by the receiver. As stated in sec.3.5.2.1 $\mathbf{\Phi}$ is unstructured and random, two candidate sensing matrices have been compared; normally distributed random numbers using Matlab function "randn" and a walsh-hadamard matrices which are shown to have RIP, and have been successfully used as measurement matrices in compressive video sensing. Although the

measurements are defined by a matrix multiplication, the operation of matrix-by-vector multiplication is seldom used in practice, because it has a complexity of O(MN) which may be too expensive for real time applications for less sparse images. When a randomly permutated Walsh-Hadamard matrix is used as the sensing matrix, the measurements may be computed by using a fast transform which has complexity of O(K log(N)), as Hadamard matrices have the orthogonal property which is one of the main properties of the random sensing matrices in contrast to "randn" which does not have the orthogonal property [114]. The Hadamard matrix, is an $(N \times N)$ square matrix whose entries are either +1 or -1 and whose rows are mutually orthogonal, the matrix is first randomly reordered then, M samples are randomly chosen to construct the $(M \times N)$ random sensing matrix $\mathbf{\Phi}$.

The ability of reliable tracking depends on acceptable recovery of images. In other words, if CS fails in image reconstruction the targets location can not be detected. Hence, for adaptive CS, M is adaptively chosen depending on the sparsity nature of images, an initial value of M is selected by the sensor node according to image sparsity. As long as the image is reconstructed with quality below some defined threshold; where the image quality is measured in terms of reconstruction PSNR (denoted by image standard PSNR$< 33dB$), the receiver requests the sensor node through a feedback channel retransmission using a different value of M. This adaptation process is repeated during the calibration phase until reaching an optimum value of M. It is clear from the results in Fig.5.1,5.2 and 5.3 for datasets 1, 2 and 3 respectively that for different sparsity levels different values of M and compression rates are required. When reaching optimum value of M, least MSE and $33dB$ PSNR are successfully achieved. For illustration, MSE decreases as M increases till reaching the optimum value, it has been shown that the lower bound on M is depending on how sparse the difference frame $\mathbf{X_d}$ is or in other words proportional to the ratio between the number of non-zero coefficients and the total number of pixels in a frame. For dataset 1, adaptive CS set M to 90 in Fig.5.1(a) to achieve



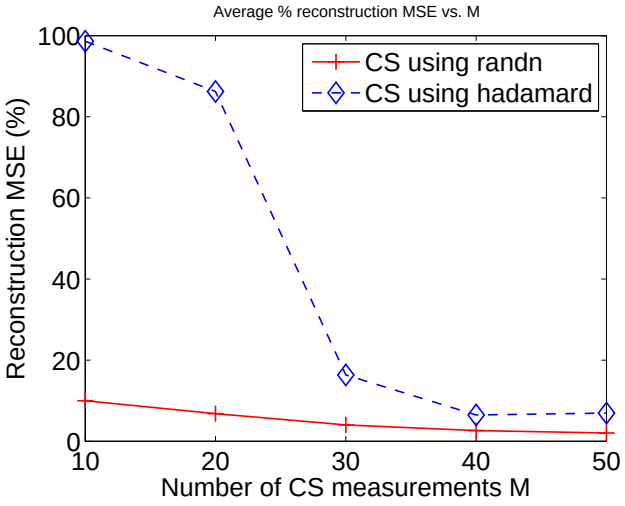
(a) Reconstruction MSE



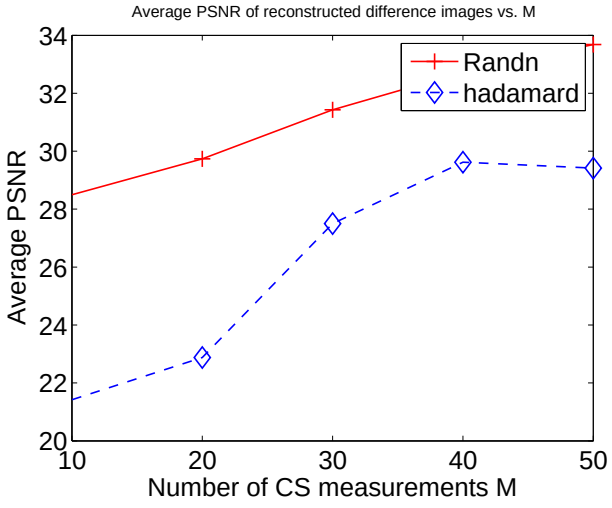
(b) PSNR

FIGURE 5.1: Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset1

satisfactory results. While for datasets 2 and 3, it is obvious from Fig.5.2(a) and Fig.5.3(a) respectively that for single-target tracking (where there is lower number of non-zero coefficients), better MSE is achieved with lower M, reduced to 50 for dataset 2 and 60 for dataset 3 compared to multi-target tracking (dataset 1) while maintaining least MSE and $33dB$ PSNR as in Fig.5.2 and 5.3. As a result, making CS adaptive helps in increasing the compression rate and avoiding the waste of using a higher value of M at the times where the image is sparse allowing for



(a) Reconstruction MSE

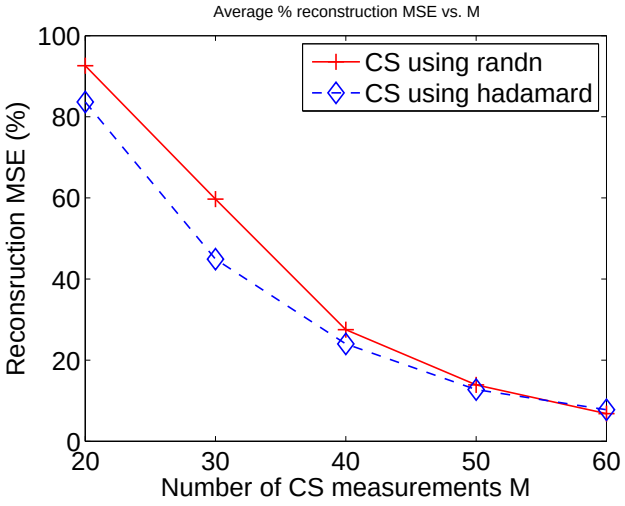


(b) PSNR

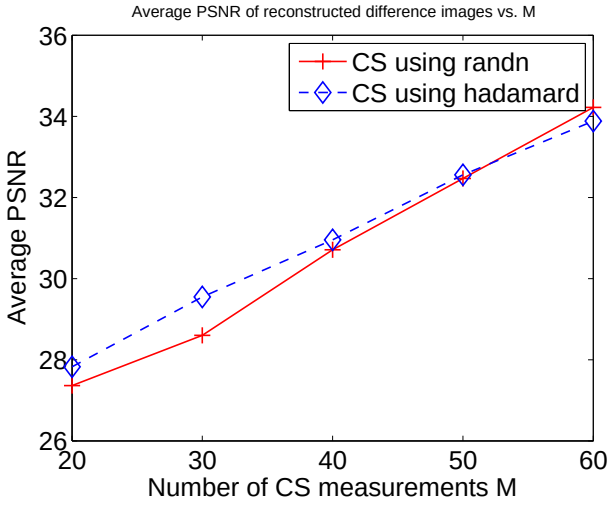
FIGURE 5.2: Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset2

lower M. The above discussion reflects the reduction in channel bandwidth using adaptive CS by 72%, (for ex. the least sparse dataset "Walking men") instead of transmitting the whole (256×256) image, the compressed measurements of size (70×256) are transmitted. Whereas for more sparse images the reduction reaches 82% of the total image size.

As for MSE, Fig.5.1(b), 5.2(b) and 5.3(b) show the effect of M on PSNR for the 3 datasets. For each dataset, according to the number of targets and the size of



(a) Reconstruction MSE



(b) PSNR

FIGURE 5.3: Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for dataset3

targets in each set of frames, the number of measurements M required will differ to obtain guaranteed reconstruction which is defined here in terms of PSNR. For low values of M it is hard to achieve a good PSNR, to reach the acceptable value, M should increase till reaching its optimum value as discussed earlier. To illustrate this for dataset 2, to achieve a PSNR of $\approx 33dB$ M reached 50, while for dataset 1 if the same M is used, we could not attain a PSNR higher than $25dB$ unless M adaptively increases till reaching the lower bound to attain the $33dB$ PSNR.

The above simulation were carried out using two different sensing matrices, Randn and walsh-Hadamard. They are compared with respect to MSE and PSNR as in Fig.5.1, 5.2 and 5.3. It is clear from the results that when reaching the optimum value of M both sensing matrices perform nearly the same except in some cases in Fig.5.2 shows that Randn gives slightly a better performance than Hadamarad. But this can be negligible when compared to the reduction in complexity and time gained by using Hadamard matrix due to its orthogonal property which helps in accomplishing the main objective to save sensor nodes power and as a result maximizes their lifetime.

Fig.5.4 and 5.5 summarize and demonstrate the effect of the target size ratio on the number of measurements M needed in terms of reconstruction MSE and PSNR (the target size ratio is expressed as a ratio between non-zero pixels representing the target and the total size of the image frame, which reveals how much space the target acquires and how sparse the image is). It is clear from Fig.5.4 that for smaller target sizes, lower values of M are used while at the same time achieving the least MSE and PSNR of $\approx 33dB$ as in Fig.5.5(a) and 5.5(b), respectively. While for larger target sizes, a higher M is required to achieve the same performance achieved for frames with smaller targets. Experiments were carried out using the same M set to 50 for the 3 datasets (different sparsity levels). For example, frames with small size targets gave better reconstruction results in terms of least MSE and a $33dB$ PSNR as in Fig.5.5(a) and 5.5(b). Whereas, if the targets size grew bigger such as acquiring 60% space of the total frame size, with M set constant reconstruction results in high MSE and only $18dB$ PSNR. In that case M should be set to 90 or higher based on the adaptive phase to reach a low MSE and a PSNR of $\approx 30dB$ that was attained by lower M (M=50) when compressing frames with targets of size $< 10\%$ of the frame size. These results reflect the constraint of the lower bound of M discussed in sec.3.5.2 and give a key to the problem when M is required to be kept as small as possible. Where in that case the size of targets is controlled by zooming or changing the location of sensor nodes during

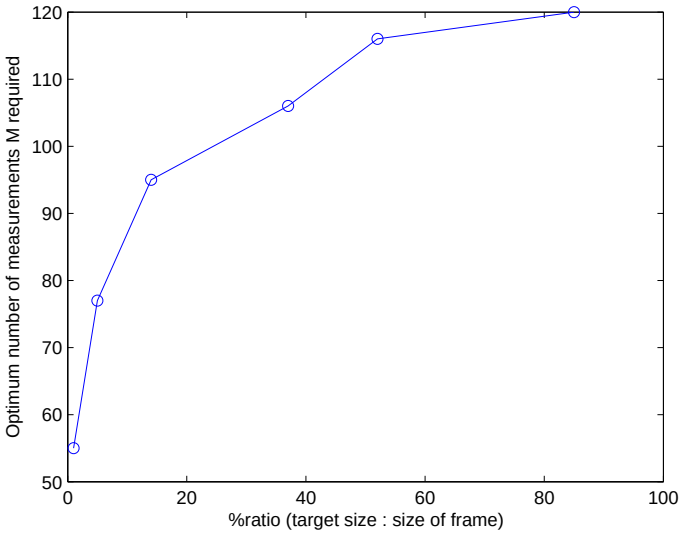
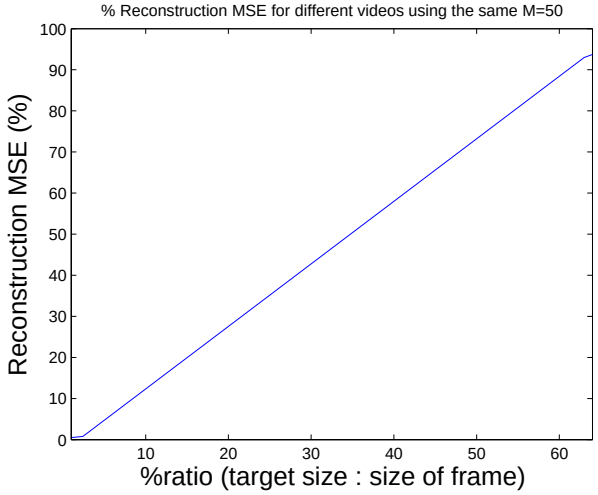


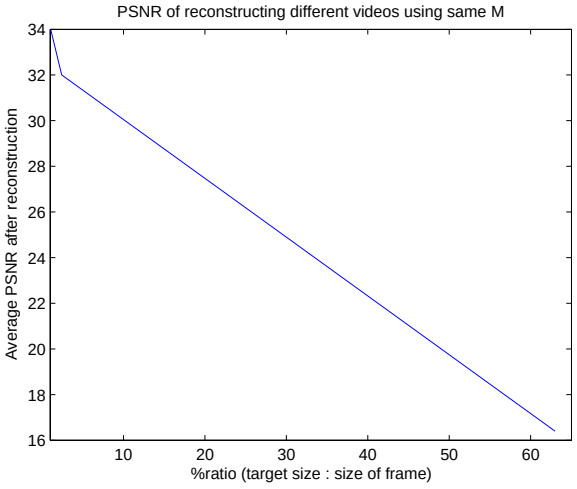
FIGURE 5.4: Relation between the percentage ratio of target size:frame size vs. M

the calibration phase while bearing in mind to keep the scene of interest in the camera's field of view. By taking snapshots from a further location the total space acquired by the target is hence reduced and as a result M can be reduced, and the goal of reducing the size of transmitted data is met .

Fig.5.6(a) illustrates the reduction in MSE and the number of measurements M required when dividing each frame into 16 blocks (64×64) each, and compressing only those with non-zero pixels. Compared to Fig.5.1(a) (compressing the whole frame), $\approx 70\%$ reduction in MSE is achieved without compromising an adequate PSNR of $\approx 33dB$ attained in Fig.5.1(b), PSNR versus the number of measurements M for block CS is shown in Fig.5.6(b). Demonstrating the reduction of the number of measurements needed, as seen in figures, for the normal scenario, M is set to 90 yielding $(90 \times N)$ measurements. Whereas, for the blocks scenario $\approx (35 \times 64)$ measurements are required per block which yields an extra communication bandwidth reduction by 40% for the total blocks transmitted compared to the normal CS-scenario which yields a total compression rate of 82%. This saves the communication bandwidth and resulting in faster transmission while saving energy at sensor nodes.



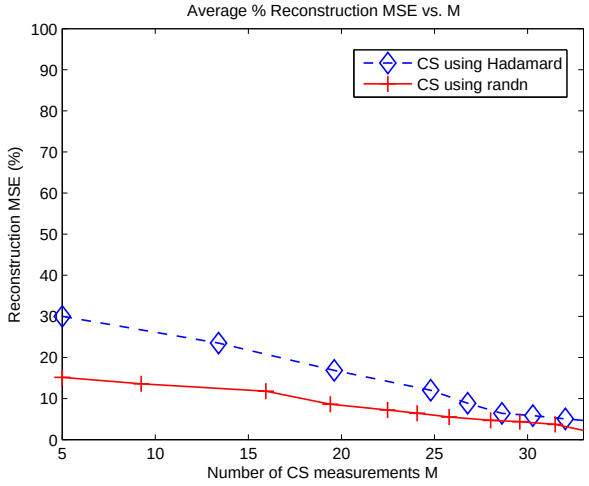
(a)



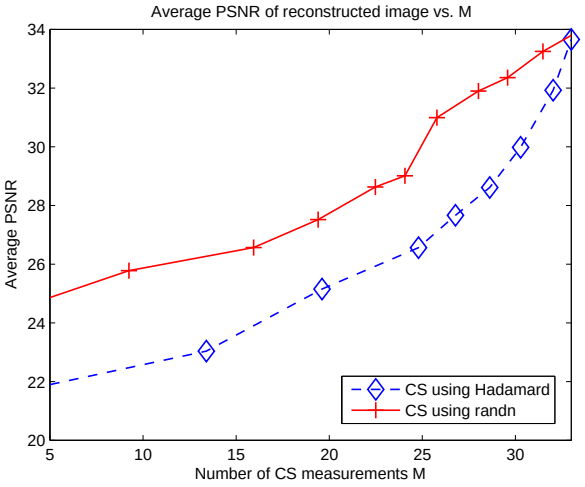
(b)

FIGURE 5.5: Relation between the percentage ratio of target size:frame size and (a) reconstruction MSE, (b) average PSNR

Another performance indicator is the correlation coefficient. After reconstructing all blocks and putting them back as a single frame, the correlation coefficient indicates how likely the reconstructed frame correlates with the original one before dividing it into blocks for compression. Fig.5.7 shows by increasing M till reaching its optimum values the correlation coefficients is nearly 100%, this implies that compressing each block separately did not affect the image quality after recovery, moreover less number of measurements were required reducing the size



(a) Reconstruction MSE



(b) PSNR

FIGURE 5.6: Comparing reconstruction MSE and PSNR using randn and walsh sensing matrices for block CS

of transmitted data. CS states that when enough measurements are used for compression, the reconstruction is done with high accuracy depending on a lower bound of M. Trajectory tracking of moving targets is considered to reflects the degree of reconstruction accuracy. Fig.5.8, 5.9 and 5.10 show the (x,y) position plots of the path tracked for the targets in the camera's scene. Fig.5.8(a) and 5.8(b) show that for lower values of M <optimum value (40 and 70 respectively),

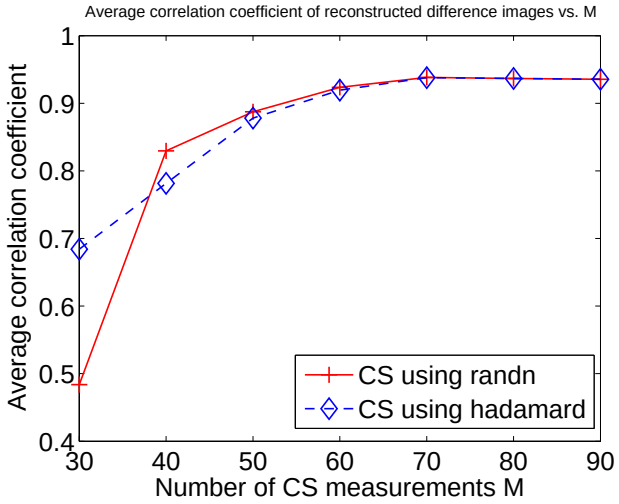
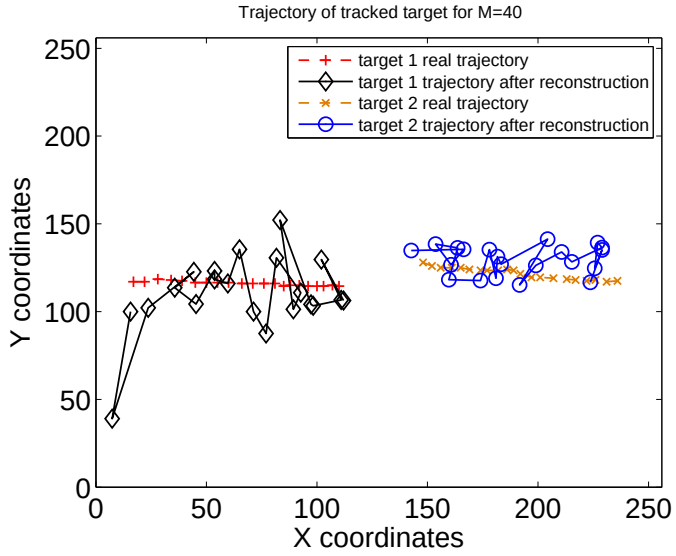


FIGURE 5.7: Correlation coefficient for different M

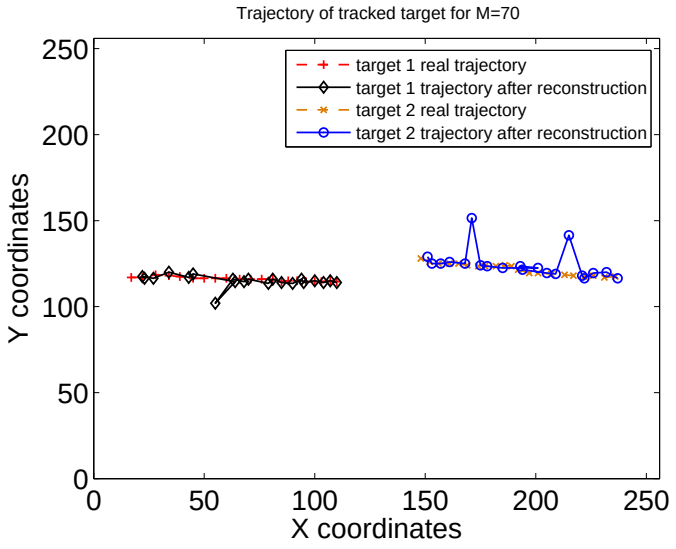
frames can not be reconstructed properly and due to noise there is no unique solution and as a result the targets tracks are not matching their real trajectories as wrong values for the targets positions are produced, whereas for M reaching 90 (the value selected based on adaptive CS), the trajectory of the tracked targets after the reconstruction matches those of the real frame before compression. Fig.5.9 and 5.10 illustrates the same for dataset 2 and dataset 3 respectively.

Fig.5.11 shows the probability of detection under different parameters, different values of measurements M and background subtraction threshold γ. It is clear from Fig.5.11(a) that for lower values of M the target is misdetected. This reflects the fact that the reconstruction can not be guaranteed with lower values of M. The probability of detection increases till reaching 100% as M increases to its optimum value selected during the adaptive CS process. Fig.5.11(b) demonstrates the effect of background subtraction threshold γ on the detection problem, where for lower values of γ, low probability of detection is achieved as the target may be misdetected due to unwanted noise.

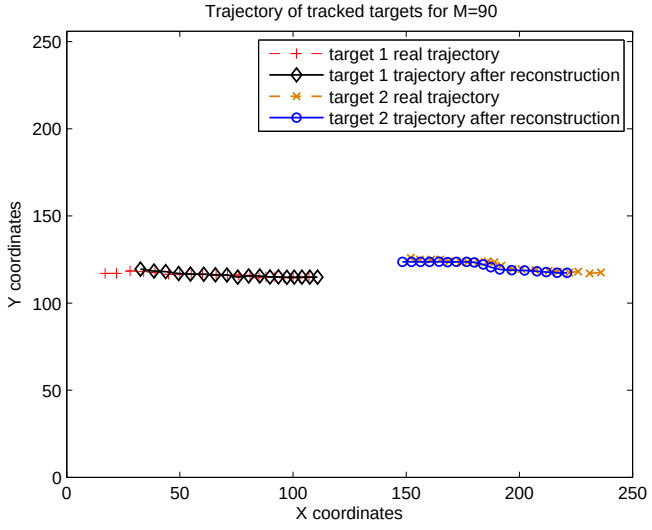
In the simulation results presented, the performance degradation due to channel impairments was not considered as we were examining the impact of CS on the tracking performance and MSE and PSNR are used as performance indicators



(a) M=40

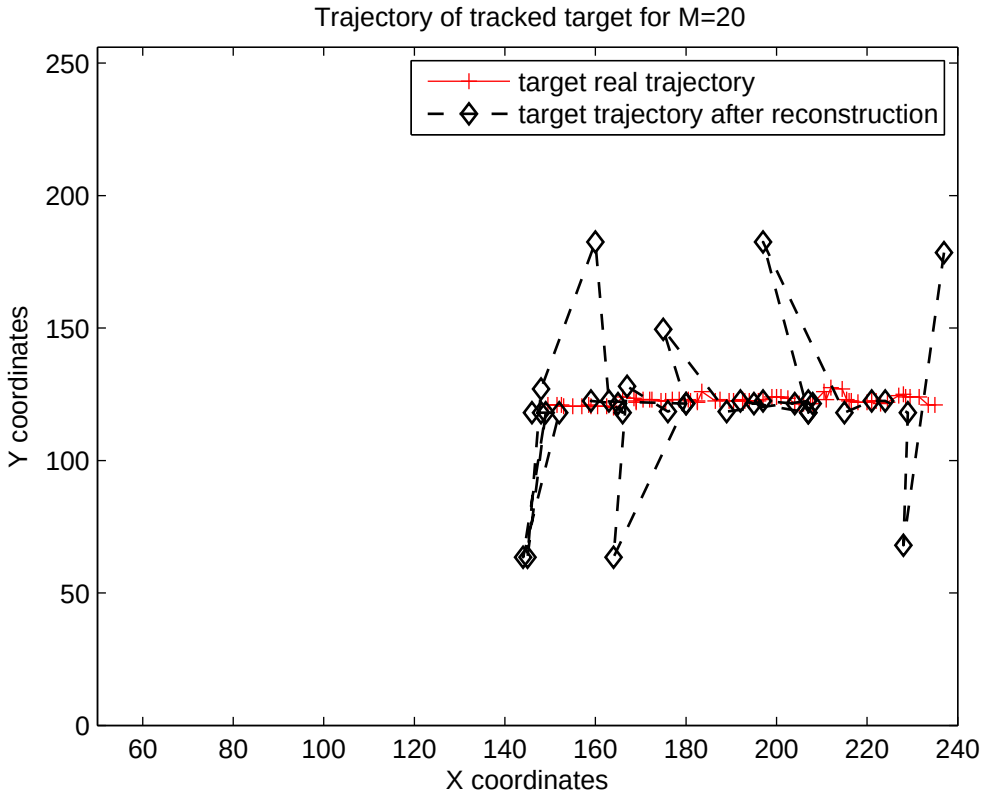


(b) M=70

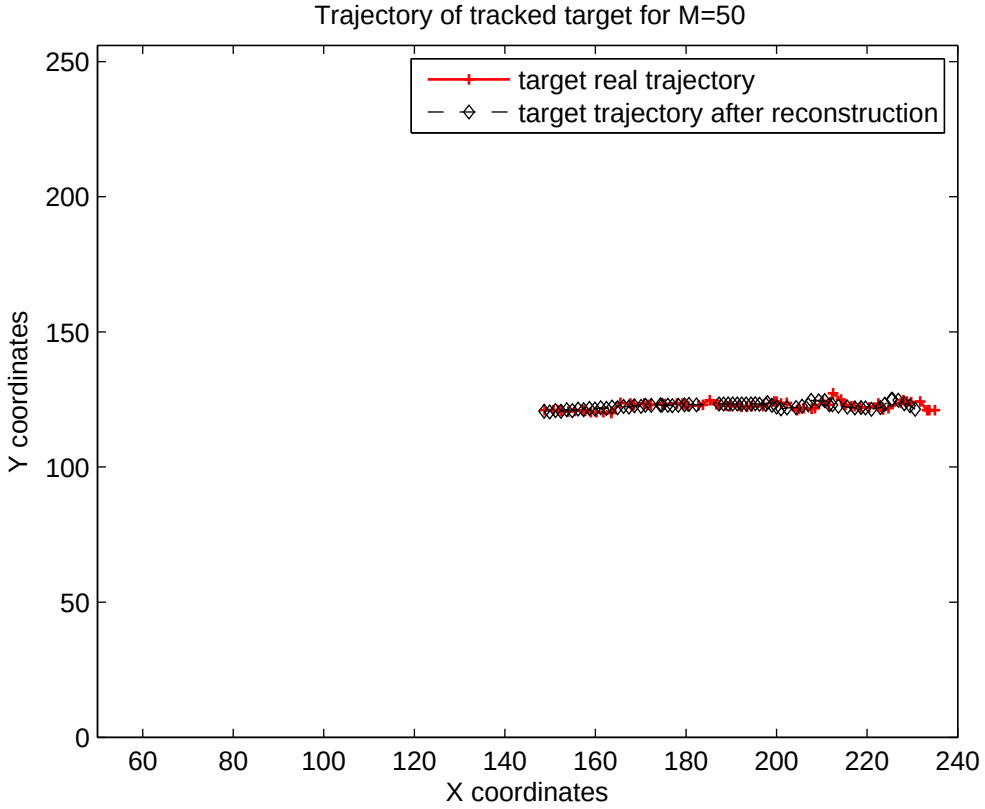


(c) M=90

FIGURE 5.8: Comparing trajectory of multi-targets for CS using different M (dataset1)

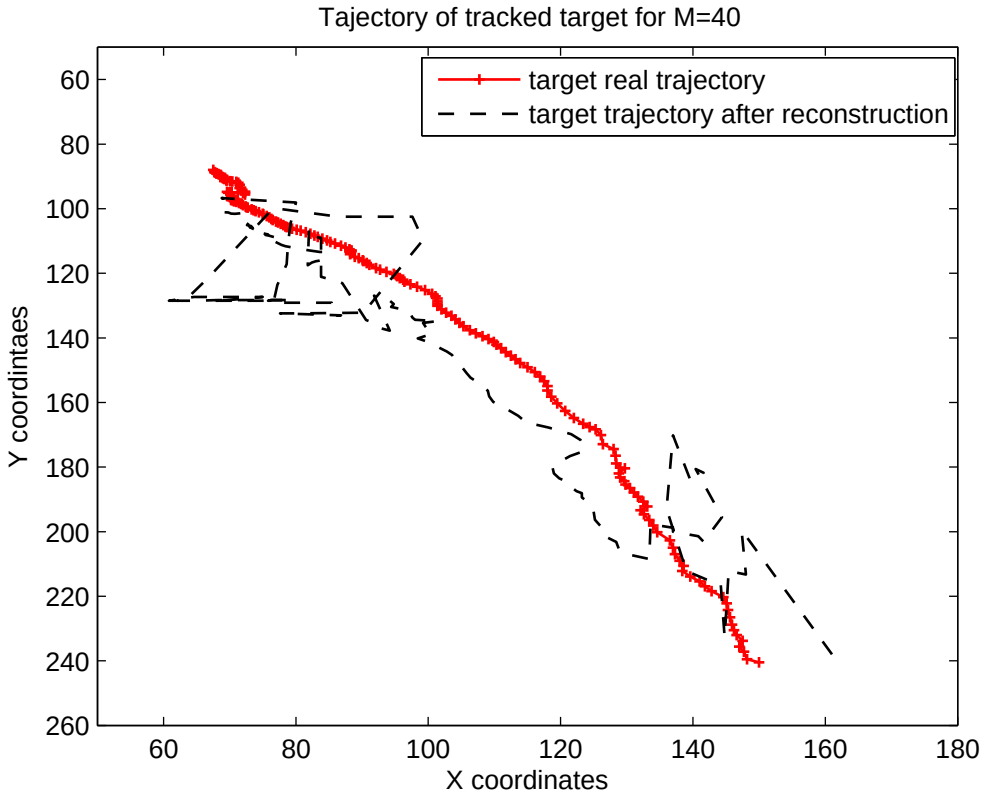


(a) M=20

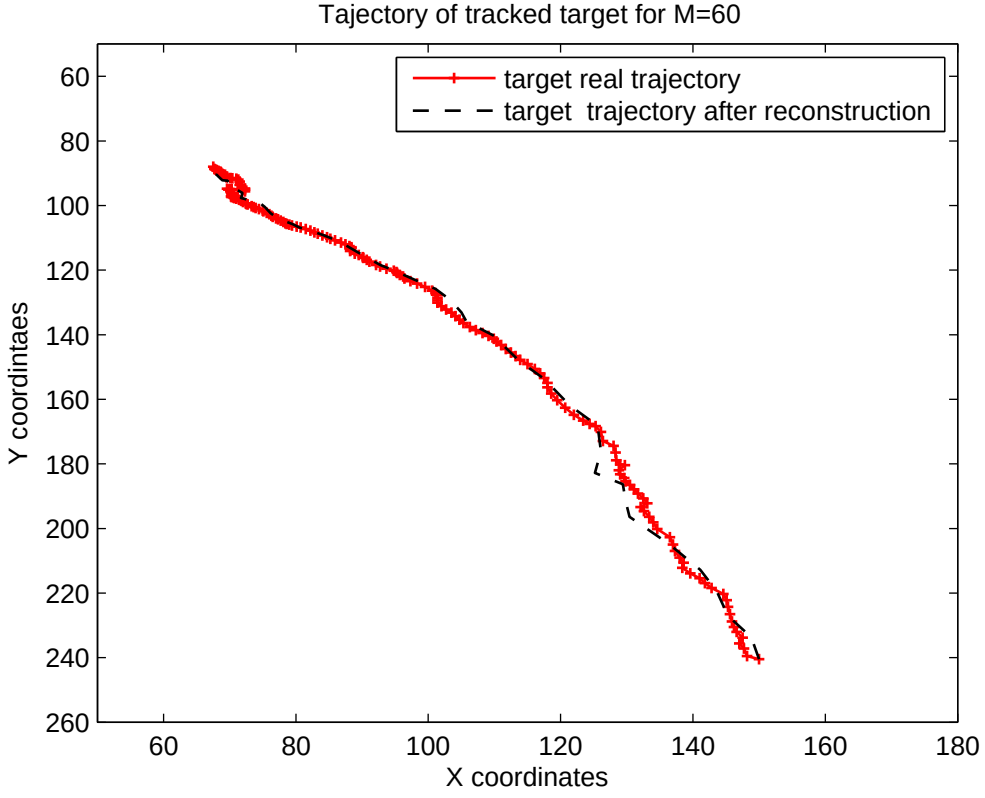


(b) M=50

FIGURE 5.9: Comparing trajectory of single target for CS using different M (dataset 2)

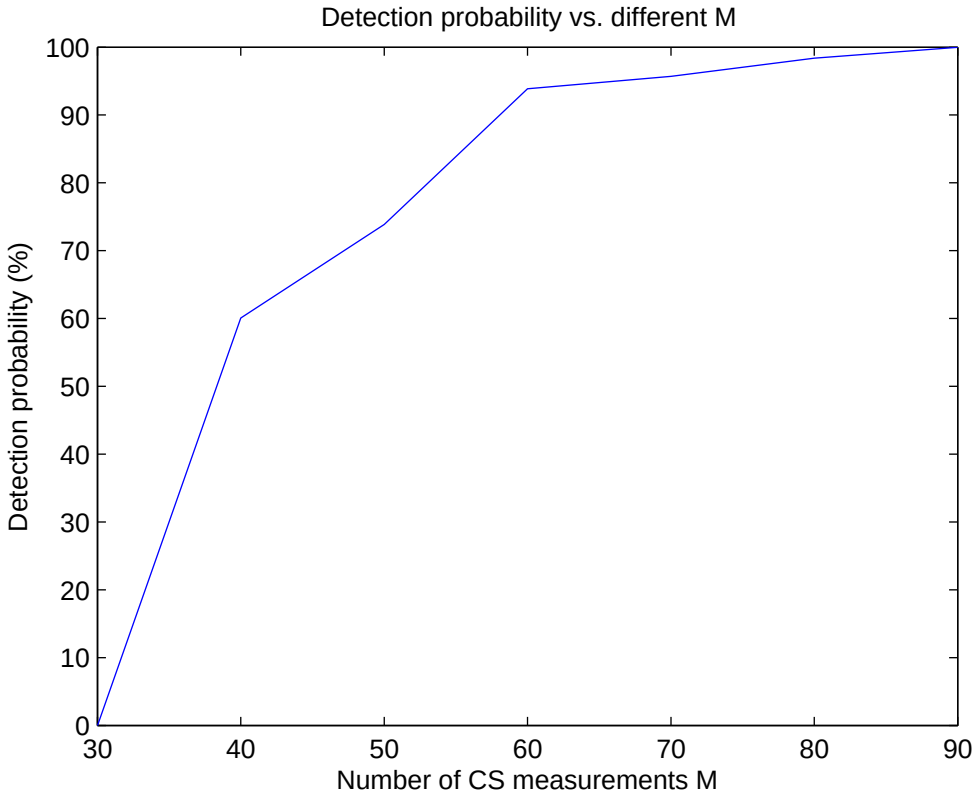


(a) M=40

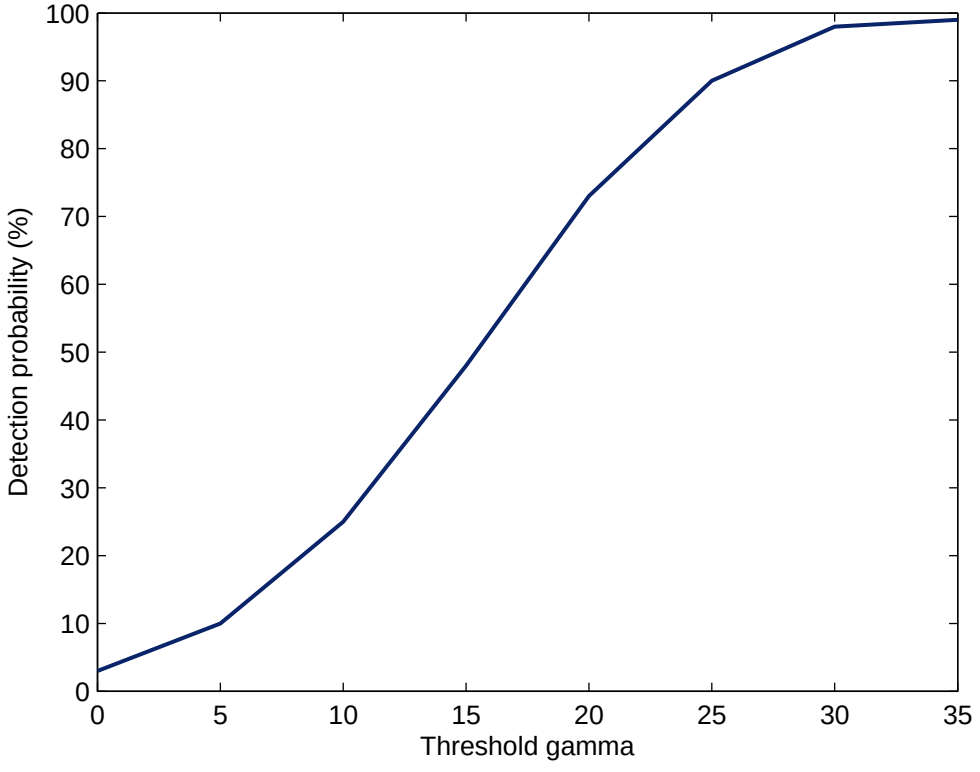


(b) M=60

FIGURE 5.10: Comparing trajectory of single target for CS using different M (dataset 3)

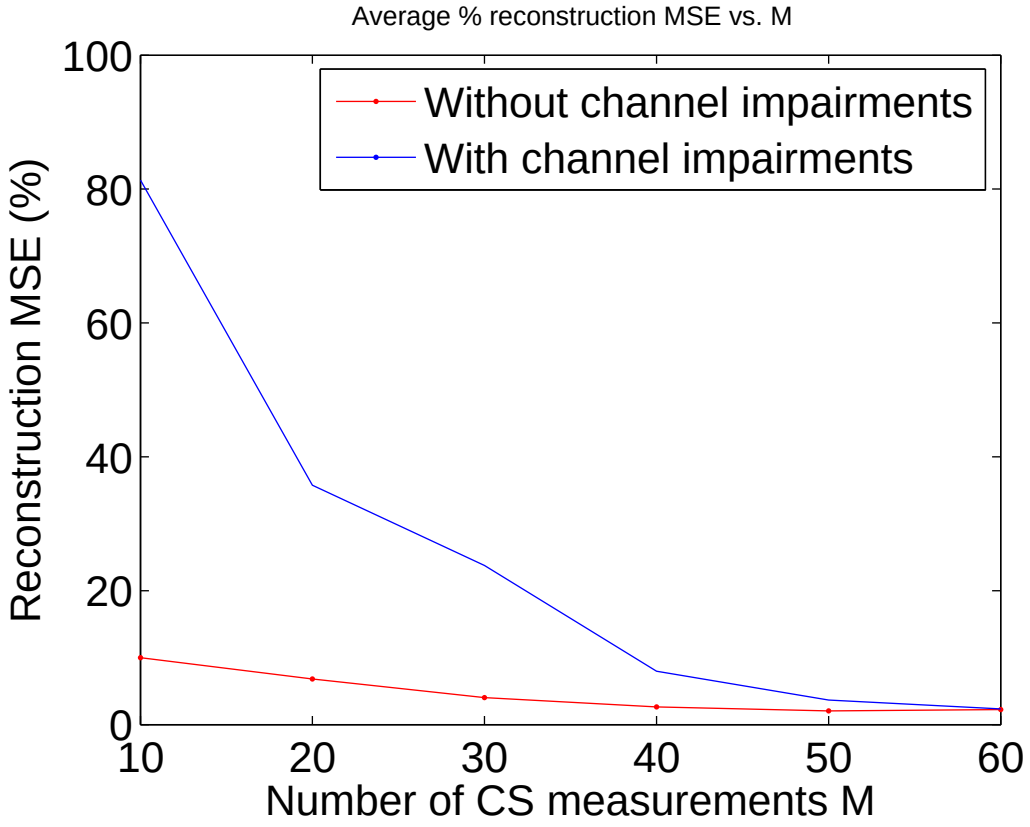


(a)

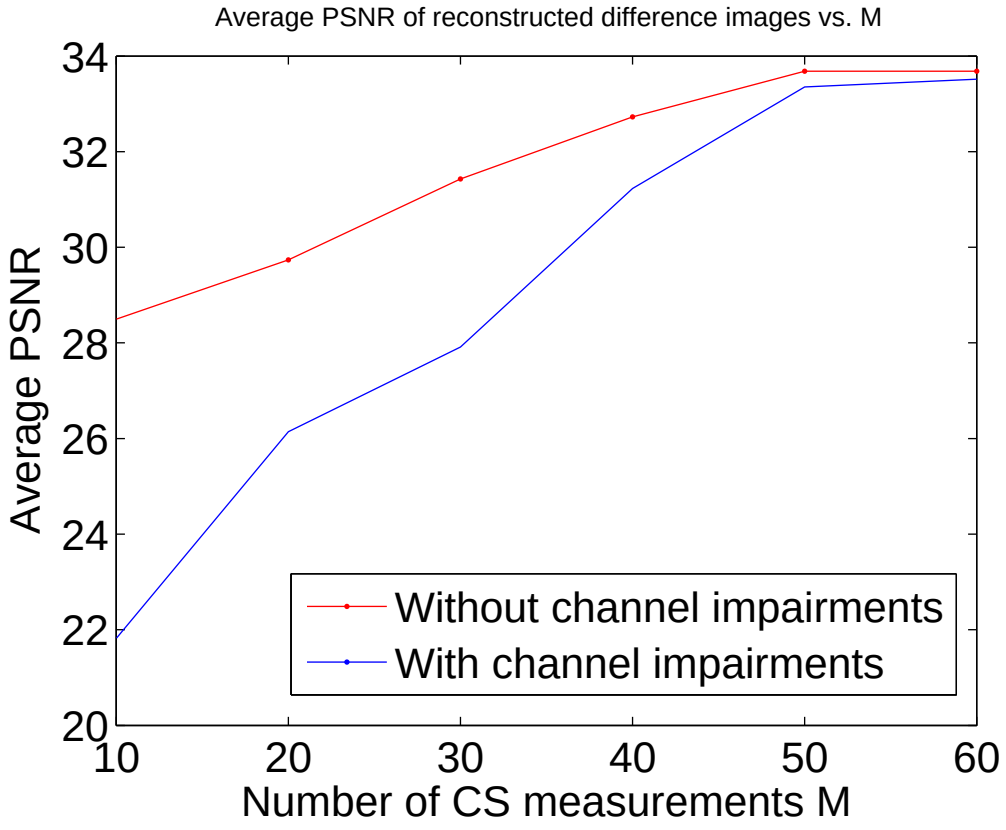


(b)

FIGURE 5.11: Probability of detection vs. (a) different values of M and (b) different values of background subtraction threshold γ



(a) Reconstruction MSE with and without channel impairments



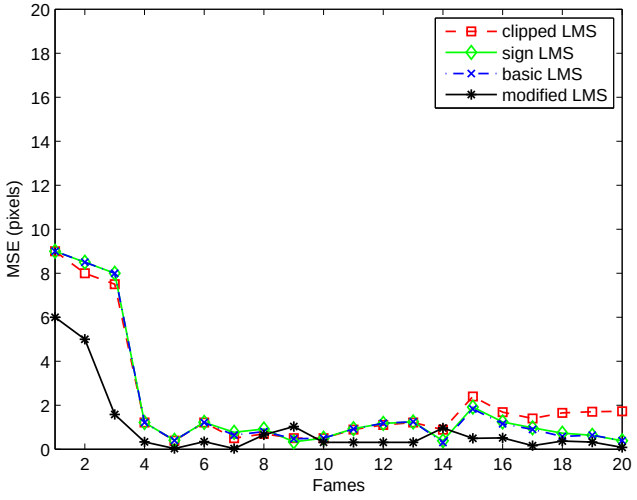
(b) Average PSNR after reconstruction with and without channel impairments

FIGURE 5.12: Comparing reconstruction MSE and PSNR with and without considering channel impairments for "Shopping center 1"

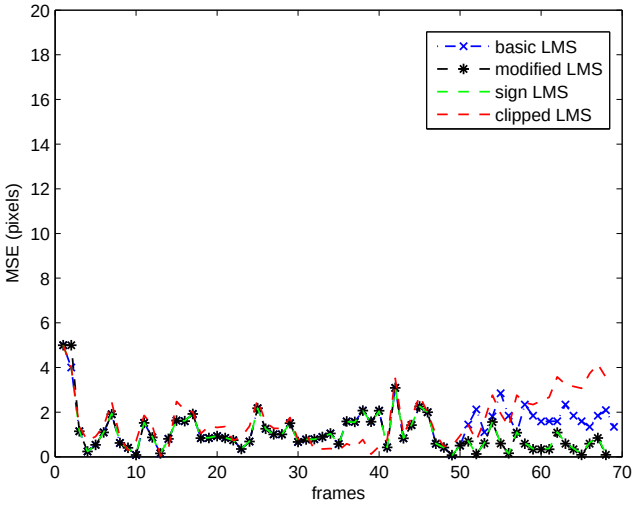
(which are widely used quality measures of Image reconstruction). However, considering channel impairments, we have generated the simulation again for one "Shopping center1" using a 30dB PSNR during wireless transmission of 2Mbps bitrates and tested the performance of CS reconstruction along with additive white Gaussian noise in the channel. As seen in Fig.5.12, reconstruction MSE reaches the same level as that without considering channel noise with the same PSNR by a slight increase in M (starting from 55) compared to the previous results without channel noise M reaches 50 for a guaranteed reconstruction.

5.3 LMS tracking

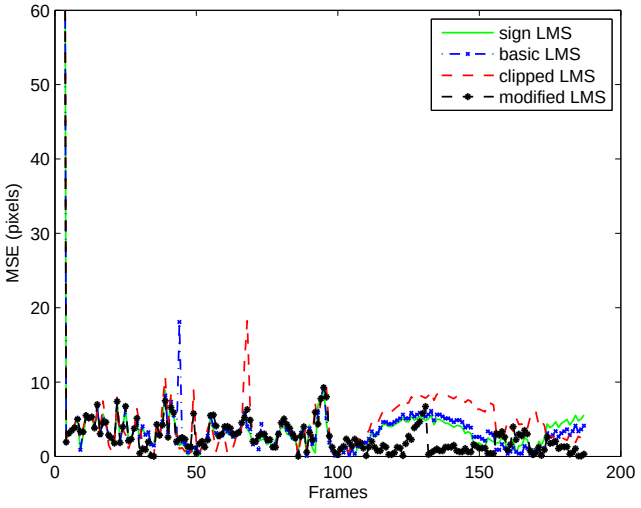
Several variants of LMS are implemented and compared, the basic LMS, clipped LMS, sign lMS, and the iterative modified quantized clipped LMS. MSE is also used as an indicator to test tracking reliability which is the error of prediction in terms of target's location (pixels) between the real target locations and the estimated target locations predicted by the different LMS algorithms for consecutive frames after the image recovery. The LMS algorithm is initiated with an arbitrary value $\mathbf{w}(0)$ for the weight vector at $t = 0$. The thresholds are chosen for the modified LMS based on experiments and targets locations in each dataset. They are set as follows; for the first dataset $D_1 = 200$ and $D_2 = 120$, for the second dataset $D_1 = 160$ and $D_2 = 140$, and for the third dataset $D_1 = 240$ and $D_2 = 200$. Fig.5.13 show the MSE for the different variants of LMS for the 3 datasets. Experiments have shown that the clipped LMS did not perform better than the basic LMS as all input data were clipped to value one. The iterative modified LMS have the least MSE due to the lower bound constraint on the MSE, at the times when the MSE rises, the algorithm goes into the learning phase again before estimating new locations. In Fig.5.13(b), the signed LMS gave the same performance as the modified LMS , whereas the MSE for some datasets as in Fig.5.13(a) and 5.13(c) is the same as the basic LMS.



(a)



(b)



(c)

FIGURE 5.13: Comparing MSE for different variants of LMS for (a)dataset 1 (b) dataset 2 and (c) dataset3

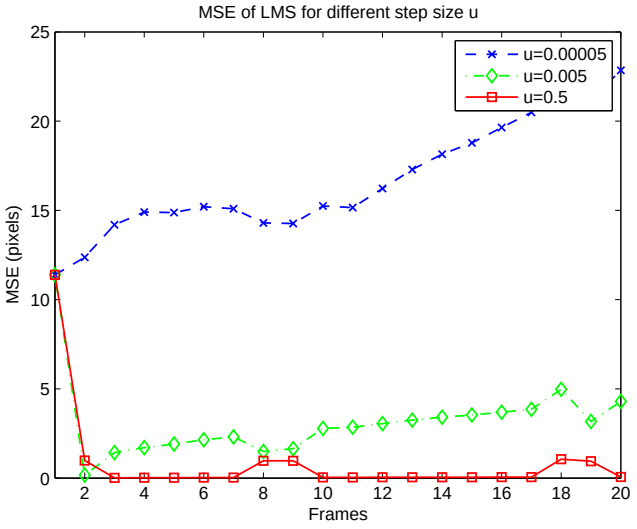
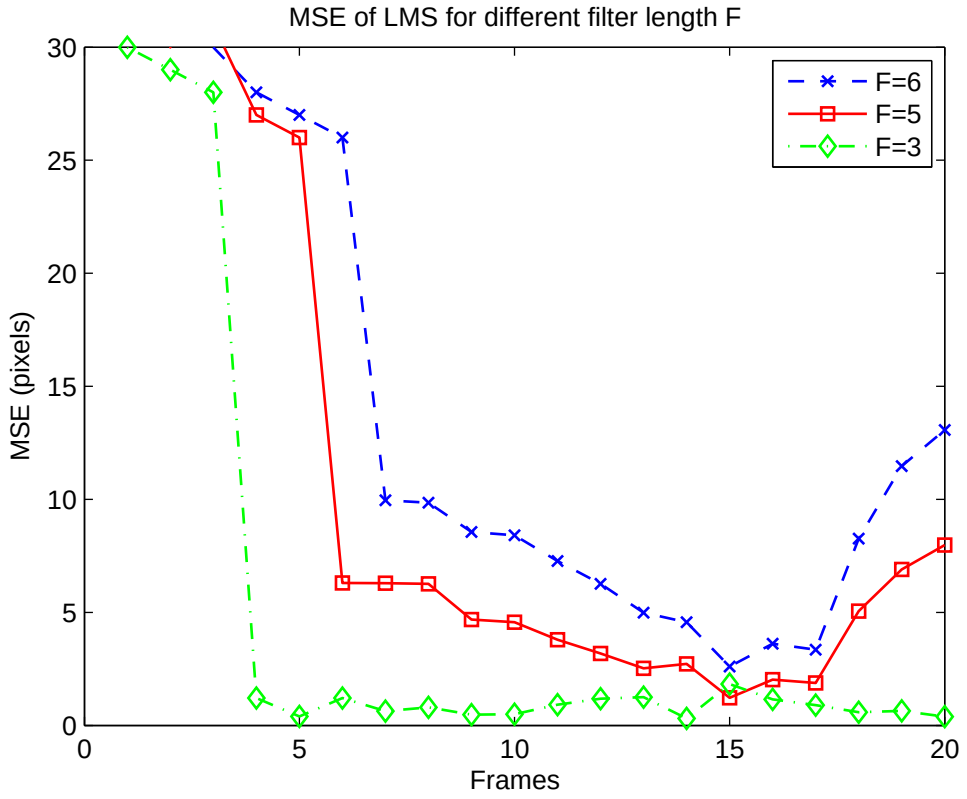


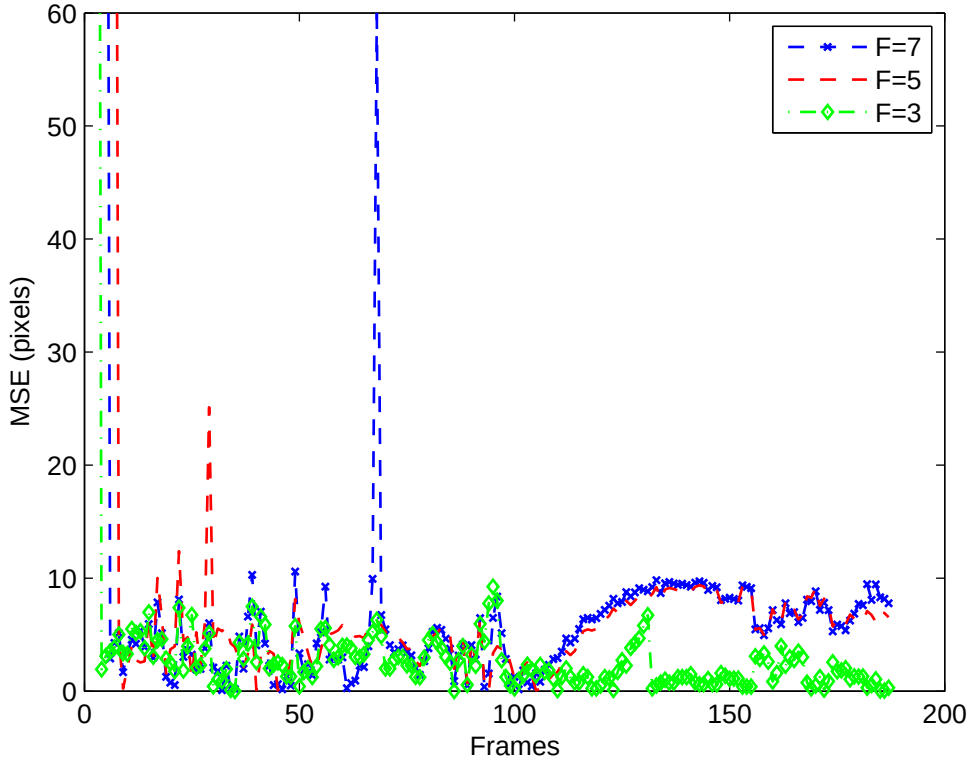
FIGURE 5.14: Comparing MSE for different u for dataset 1

Fig.5.14 shows the effect of different values of u (0.00005,0.005,0.5)on the tracking convergence for "Dataset1", for large values of u the algorithm converges faster with the least MSE as shown in figure. Moreover, the effect of the filter length is illustrated in Fig.5.15, it is clear that lower MSE is achieved for lower filter length. However, as the filter length is increased, the speed of convergence of the LMS adaptive filter decreases, and the MSE increases, as changes in the targets trajectory are not detected causing the prediction to deviate away from the real path as reflected in figure where a sudden rise occurred in MSE. Therefore, the filter length should be chosen as short as possible but long enough to adequately model the unknown system, as too short a filter model leads to poor modeling and prediction performance. This problem could be alleviated by more frequent training to update the filter's weights to sustain the convergence of minimum MSE.

Tracking reliability is also tested by comparing the moving target's real and predicted trajectories using the iterative quantized clipped LMS. As shown in Fig.5.16, the target's locations are accurately predicted and the results are closely matching the real target trajectory.

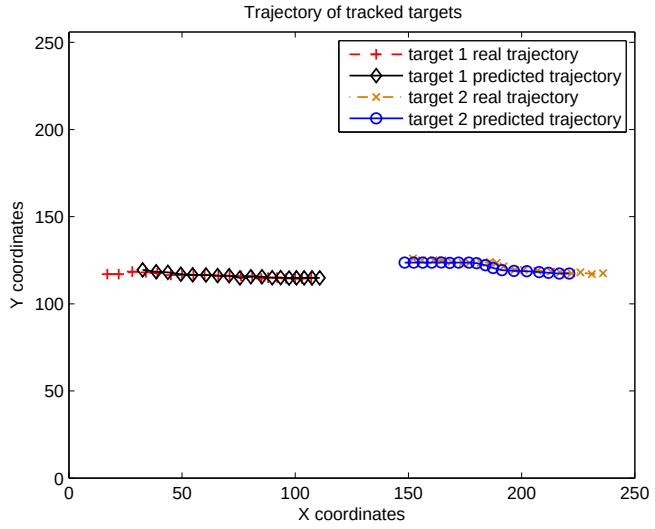


(a)

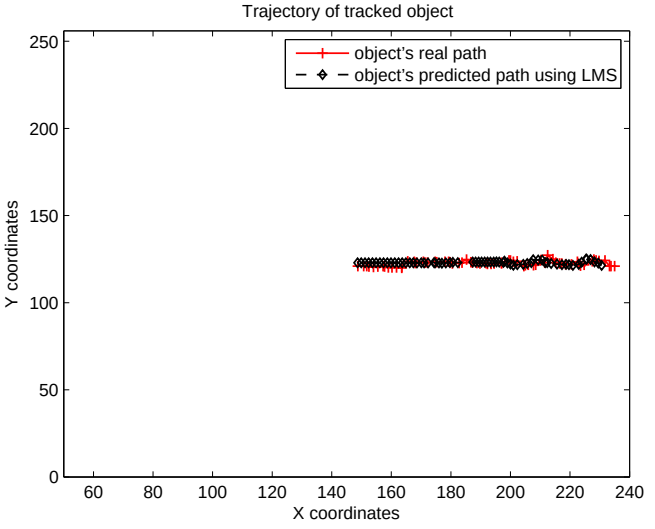


(b)

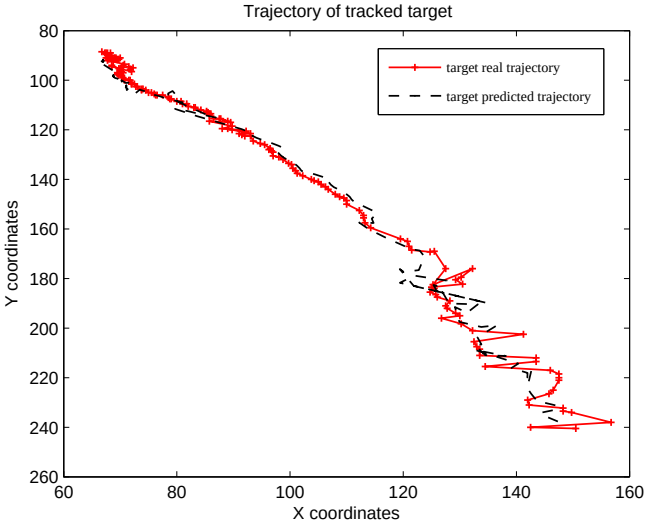
FIGURE 5.15: Comparing MSE for different filter length F for datasets 2 and 3 respectively



(a) Predicted trajectory tracking for multi-targets using LMS



(b) Predicted trajectory tracking for single targets using LMS

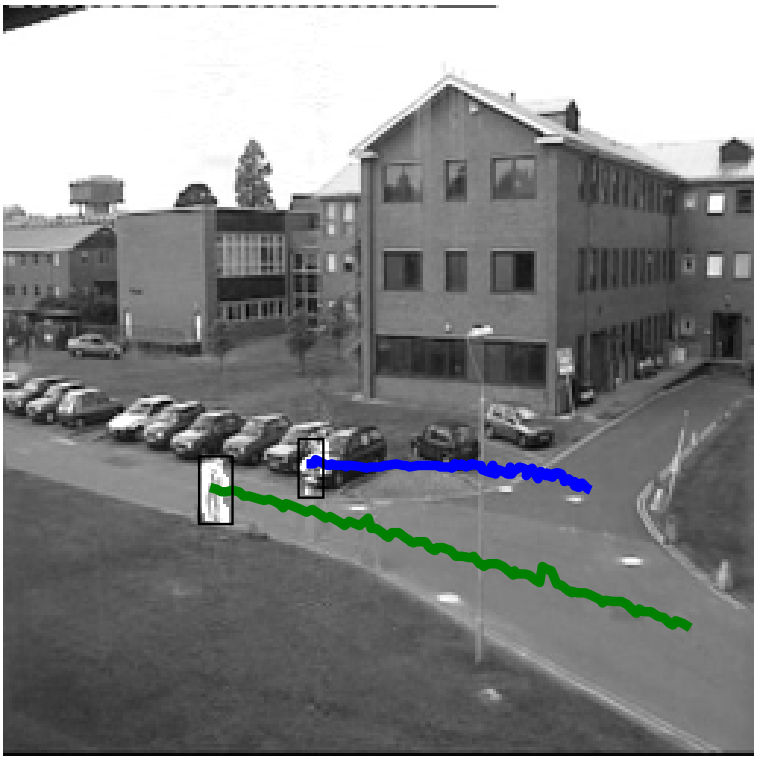


(c) Predicted trajectory tracking for single targets using LMS

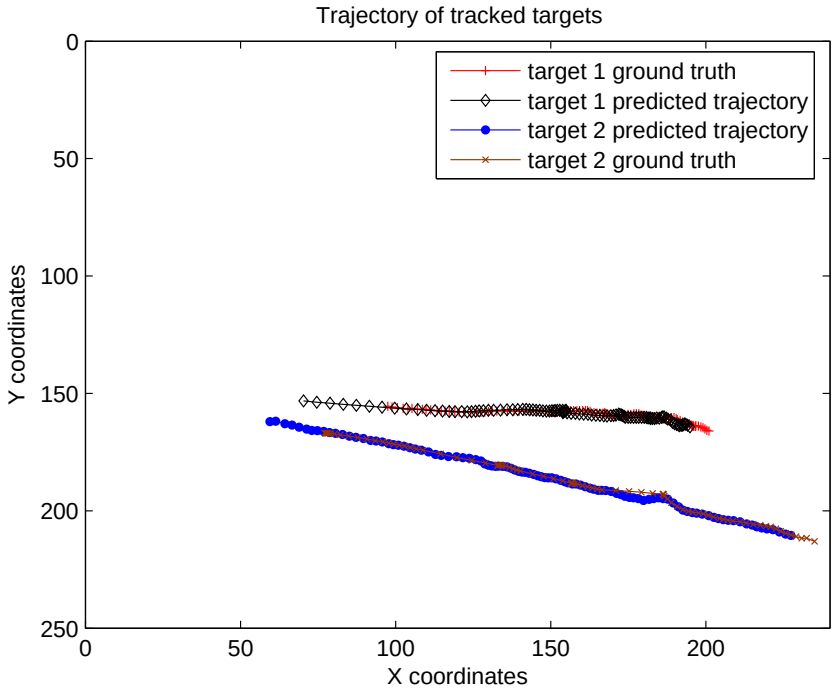
FIGURE 5.16: Comparing trajectory tracking of moving targets for (a) dataset 1, (b) dataset 2 and (c) dataset 3

Fig.5.17(a) shows trajectory tracking for multi targets for dataset"5" after image reconstruction. The predicted trajectory tracks using LMS is illustrated in Fig.5.17(b) and compared with the ground truth, the results show that LMS prediction matches ground truth values.

The performance of the proposed LMS algorithm is compared with state-of-the-art Kalman filter [58], both algorithms are applied on a standard surveillance video 'OneStopMoveNoEnter1cor' from CAVIAR [112](same dataset used by the authors in [58]). In the video a man is selected for tracking and tracked in subsequent frames using both LMS algorithm and Kalman filter. Fig.5.18 shows output frame number 997 with trajectories of the target moving in the corridor since its appearance in the video, the figure shows that both LMS and Kalman filter matches the real trajectory of the target.



(a) Predicted trajectory tracking for multi-targets using LMS



(b) Predicted trajectory tracking for single targets using LMS

FIGURE 5.17: Comparing trajectory tracking of moving targets for dataset "5"

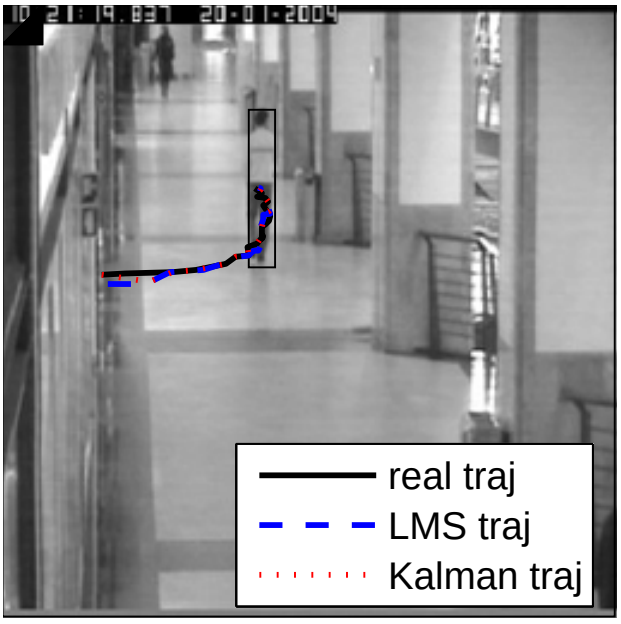


FIGURE 5.18: Comparing predicted trajectory using LMS and Kalman filter

5.4 Computational complexity

Assuming all sensor nodes have the same unit distance d from the receiver side, Table.5.1 shows the energy dissipated during transmission for different k (number of data samples transmitted). As illustrated, according to different k (which varies depending on compression rates due to sparsity levels), there is an 82% energy saving as compared to transmitting the captured image without CS. In addition, using block CS will result in 20% more energy saving compared to traditional CS.

Table.5.2 summarizes the computational time for the traditional CS process, block CS and the LMS tracking technique. As stated in sec.4.3.3.1 the LMS algorithm is relatively simple, has much lower computational complexity than the original Kalman filters and other adaptive algorithms and suitable for real time applications due to its fast convergence as demonstrated.

TABLE 5.1: Transmission energy using CS, block CS and without CS for different k

Dataset with/without CS		Size of transmitted data k	Transmission Energy Etx
"Walking men" Without CS		64K	3.3mJ
CS	"Walking men"	17K	0.85mJ
	"Shopping center 1"	15K	0.7mJ
	"Shopping center 2"	12K	0.6mJ
"Walking men" Block CS		11K	0.5mJ

TABLE 5.2: Computational time for CS, block CS and LMS

	Computational time
CS process	0.03s
Block CS	0.002s/block
LMS	0.002s

5.5 Chapter summary

In this chapter, experiments were carried out to evaluate the performance of the adaptive CS and its effect on target detection and tracking. Results have shown that using adaptive CS, the reconstruction MSE decreases till reaching the lower bound on the number of compressed measurements while preserving the acceptable PSNR. In addition, for different datasets where the sparsity nature of each image differs, CS adaptively chooses the compression rates accordingly. Moreover, block CS achieved higher compression rates with lower reconstruction MSE saving the communication bandwidth and resulting in faster transmission. After image reconstruction, the impact of adaptive CS on target tracking is investigated where the proposed iterative quantized LMS is performed for target tracking and is compared with other variants of LMS. Results have demonstrated that the proposed LMS technique achieved the least MSE. Target's trajectory tracking has been used as another performance indicator for the LMS algorithm, it is shown that the predicted path closely matches the target's real path which illustrates the accuracy of LMS and that CS has reduced energy consumption and at the same time has not affected the performance of target detection and tracking .

The next chapter derives an analytical framework for the selection of node's duty cycles and its effect on detection performance. Moreover, the impact of CS versus adaptive CS to reduce the size of transmitted data on the object detection problem for WVSNs is analyzed after integrating node's duty cycles.

Chapter 6

Analytical framework of the detection model

This chapter addresses the target detection problem within WVSNs where visual sensor nodes are left unattended for long-term deployment. As battery energy is a critical issue it is always challenging to maximize the network's lifetime. In order to reduce energy consumption, nodes undergo cycles of active-sleep periods that saves their battery energy by switching sensor nodes ON and OFF, according to a predefined duty cycles. Moreover, as proven in previous chapter adaptive compressive sensing dynamically reduces the size of transmitted data through the wireless channel saving communication bandwidth hence saving energy. The aim is to derive an analytical framework for the integration of selecting node's duty cycles and dynamically choosing the appropriate compression rates for captured images and videos which is expected to reduce energy waste by reaching the maximum compression rate for each dataset without compromising the probability of detection.

6.1 Introduction

Due to the advancement of new technologies, there are immediate requirements for automated energy-efficient WVSNs applications. WVSNs have addressed various applications such as environmental monitoring to study environments conditions and animal behavior, surveillance applications, law enforcement, industrial automation and military purposes. Visual sensor nodes are resource constraint devices bringing the special characteristics of WVSNs such as energy, storage and bandwidth constraints which introduced new challenges [2–5, 115, 116]. Within WVSNs, each visual sensor node is powered by an attached battery and embeds a visual sensor(can be integrated with other types of sensors such as vibration and acoustic sensors), digital signal processing unit, limited memory and a wireless transceiver. Hence, due to the limited battery power and communication bandwidth, energy utilization is necessary to maximize the network's lifetime.

The target detection problem within WVSNs where visual sensor nodes are left unattended for long-term deployment. Among the many diverse application domains of WSNs, object detection (object can be a human being, a vehicle or any targeted object) is of the most important tasks in image processing applications. As battery energy is a critical issue it is always challenging to maximize the network's lifetime by minimize the energy consumption due to sensing, processing and transmission without compromising the detection performance. In order to reduce energy consumption, nodes undergo cycles of active-sleep periods that saves their battery energy by switching sensor nodes ON and OFF, according to a predefined duty cycles.

At the same time, there is a scope to achieve the same energy saving by minimizing the volume of data required for target detection. The adaptive CS technique proposed in 4.3.2.1 to represent the data with just small number of measurements depending on the sparsity of different images is integrated to the target detection

problem. Consequently, it is expected to save bandwidth requirement for transmission and processing power. In addition to energy, memory constraints and communication bandwidth, CS should not affect quality of image (as denoted by PSNR) for later target detection.

The main goal is to maximize the sensor node's lifetime by setting predefined duty cycles to visual sensor nodes to switch On and OFF while ensuring the detection of targets. Due to many factors such as node deployment, number of nodes, velocity and position of targets, the performance of detection may degrade. Moreover, the impact of CS versus adaptive CS to reduce the size of transmitted data on the object detection problem for WVSNs is analyzed. Due to the integration of adaptive CS to the detection problem, the performance may features further degradation than the desired and acceptable level due to other factors such as image sparsity, loss of information in compression. Hence, there is always a tradeoff between energy consumption (network lifetime) and detection performance. As a result, we focus on deriving an analytical framework for selecting these duty cycles and dynamically choosing the appropriate compression rate for different images and videos which is expected to reduce energy waste by reaching the maximum compression rate for each dataset without compromising the probability of detection.

6.2 related work

As battery energy is a crucial issue, in [25] the authors addressed the target detection problem for long lasting surveillance application using unattended WSNs. In this context, the authors distributed the processing on sensor nodes by switching ON and OFF according to proper duty cycles the sensing and communication modules of wireless sensor nodes. Making these modules work in discontinuous fashion by random scheduling saves energy however it has an impact on the detection problem. In order to maintain a given performance objectives, the authors

derived an analytical framework to evaluate the probability of missed target detection. In [24], the authors adopted a model of unsynchronized duty-cycle scheduling for individual nodes. Where, nodes sleep and wake-up periodically, according to duty cycles by setting the length of the duty cycle period and the percentage of time nodes are awake within each duty cycle. However, the wake-up times are not synchronized among nodes as random scheduling is probably the easiest to implement in sensor networks since it requires no coordination among nodes. Moreover, coordination among nodes requires additional energy as it involves some message exchange. In contrast, random scheduling does not require communication, each node simply sets its own duty-cycle schedule according to the agreed-upon wakeup ratio.

A node selection scheme is presented in [117] which gives full consideration to both the information utility for the quality of tracking and the remaining energy of nodes to determine the longetivity of nodes. Each sensor node is responsible of computing the detection probability, whereas the optimal set of sensors performs target tracking by integrating partial estimations. The node selection is formalized as an optimization problem and solved by genetic algorithms to optimize the tradeoff between the accuracy of tracking and the energy cost of nodes. While in [118] energy conservation in target tracking is achieved using different methods. Prediction-based scheme coupled with selective activation of nodes is one of such methods where nodes are wakes-up on-demand following the target path. Previous active nodes collaborate between each other to generate an accurate estimation of the target.

In [119], the authors integrated reactive mobility of sensor nodes to improve the target detection performance of WSNs. Sparsely deployed mobile sensors collaborate with static sensors and move in a reactive manner to achieve required detection performance. Specifically, mobile sensors remain stationary until a possible target is detected.

To summarize, in the context of target detection within WVSNs, it is always a challenge to maximize the WVSN's lifetime without degrading the detection performance. Hence, the aim is to provide the intended detection performance with minimum energy requirement to obtain optimal utilization of energy. Moreover, to achieve energy-efficient wireless transmission with nominal preprocessing, an adaptive CS technique is proposed to reduce the size of transmitted data and compared with the traditional CS in the context of energy saving and detection reliability. Deriving an analytical framework considering the resource constraints within WVSNs for target detection to evaluate the impact of energy saving due to visual sensor nodes' duty cycles and the integration of adaptive CS is hence the major focus of the proposed investigations. Next sections describe a general WVSN model with the integration of node's duty cycles and an analytical framework for the detection problem.

6.3 WVSN model

Consider a WVSN-based surveillance application model, composed of N visual sensor nodes randomly distributed over a specific geographical region of $(100m \times 100m)$as in Fig.6.1. Each sensor node is assumed to have a sensing radius r_s allocated to share a viewable range and fully cover the required geographical region, and one or more receiver/sink node at fusion center. The geographical region is assumed to be a square area with each side of size d_s. It is required to increase the WVSN's lifetime by reducing the energy consumption this is accomplished by periodically switching On and OFF the visual sensors. Each sensor node is assumed to be in 'wake-up' state according to a duty cycle $\beta_s \in [0, 1]$ over a period t_s, hence each sensor is awake for an interval of length $\beta_s t_s$ and sleep for an interval $(1 - \beta_s)t_s$ as shown in Fig,6.2.

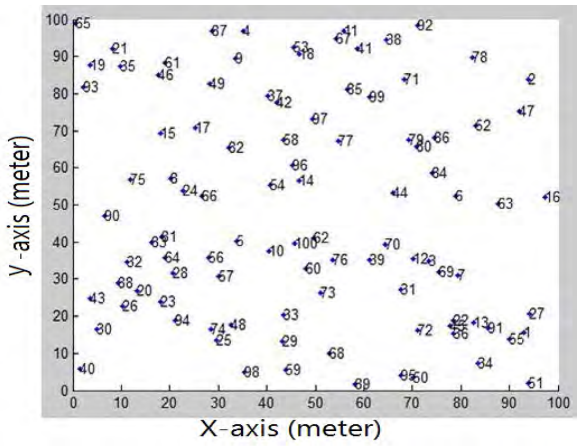


FIGURE 6.1: Wireless visual sensor network

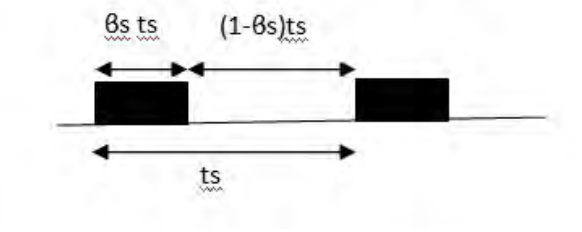


FIGURE 6.2: Scheme for sensor's duty cycle

6.4 Probability of missed detection

In subsequent sections an analytical framework is derived, first to evaluate the probability of missed detection as a function of the target's mobility model due to the predefined duty cycles. Second, the probability of missed detection is derived after the integration of adaptive CS to compare the performance of detection with and without CS.

6.4.1 Probability of missed detection as a function of mobility model of the target

In order to detect a target in a squared geographical area, N sensors are randomly deployed and set to periodically switch ON and OFF according to a predefined duty cycle β_s. To evaluate the probability of missed detection P_{md}, it is required to

integrate the sensor's duty cycle. Assume that the target enters the sensing area at time t_a with angle θ and velocity vm/s crossing the sensing area in $T_{cross} = L/v$ where L is the length of intersection between the target's trajectory and the sensing radius r_s as shown below in Fig.6.3

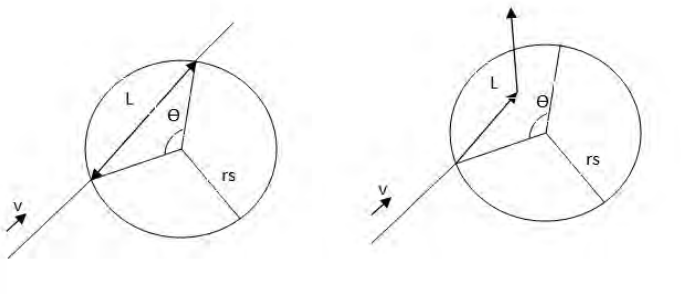


FIGURE 6.3: Sensor model for (a) linear and (b) non-linear target trajectory

to find the probability of missed detection, assume that ξ_{target} is the event where the sensor is ON when the target enter the sensing area and ξ_{det} is the event where the sensor is ON when the target crosses the sensing area. During a single time interval t_s, any incoming target entering a sensor's sensing area at time T_a during the interval $\beta_s t_s$ (i.e. the sensor is ON) will be detected. However, in the case where T_a is during the interval $(1-\beta_s)t_s$ (i.e. during the sensor's sleep interval), in order to successfully detect the target, the target must remain in the sensing area till the sensor's next duty cycle where it turns ON again. Hence, the Probability of detection P_d is defined as in [24] by the total probability theorem as:

$$P_d = P\{\xi_{det}|\xi_{target}\}P\{\xi_{target}\} + P\{\xi_{det}|\overline{\xi}_{target}\}P\{\overline{\xi}_{target}\} \tag{6.1}$$

Where, $P\{\xi_{det}|\xi_{target}\} = 1$, $P\{\xi_{target}\} = \beta_s$, and $P\{\overline{\xi}_{target}\} = (1 - \beta_s)$, the last term $P\{\xi_{det}|\overline{\xi}_{target}\}$ in (6.1) denotes the case where the target is detected given it enters the sensing area during the sensor's sleep interval $(1 - \beta_s)t_s$. This suggests that either the target's crossing time $T_{cross} > (1 - \beta_s)t_s$, as a result the target is detected. Or the case where $T_{cross} < (1 - \beta_s)t_s$, in this case the target will only be detected if it enters the sensing area during the last part of the sleep interval,

such that the target remains in the sensing area till the sensor turns ON in the next duty cycle. Hence, as in [24] $P\{\xi_{det}|\overline{\xi}_{target}\}$ is calculated in terms of the joint probability density function (pdf) as follows:

$$P\{\xi_{det}|\overline{\xi}_{target}\} = \int\int_{D} fT_aT_{cross}(t,\tau)dtd\tau \tag{6.2}$$

where D is the integration domain described in [24, 120], $fT_aT_{cross}(t,\tau)$ is the probability density function expressed as:

$$fT_aT_{cross}(t,\tau) = \begin{cases} \frac{v}{\Pi\varsigma\sqrt{r_s^2-(\frac{v\tau}{2})^2}} & \text{if } 0<\tau<2r_s/v,\ 0<t<\varsigma \\ 0 & \text{else} \end{cases} \tag{6.3}$$

where $\varsigma = (1-\beta_s)t_s$ is the time interval where the target enters the sensing area, hence:

$$P\{\xi_{det}|\overline{\xi}_{target}\} = \begin{cases} \frac{4r_s}{\pi\varsigma v} & \text{if } 2r_s/v<\varsigma \\ \frac{4r_s-2\sqrt{4r_s^2-\varsigma^2v^2}}{\pi\varsigma v}+1-\frac{2\arcsin(\frac{\varsigma v}{2r_s})}{\pi} & \text{else} \end{cases} \tag{6.4}$$

Finally, the P_d is written as:

$$P_d = \beta_s + (1-\beta_s)P\{\xi_{det}|\overline{\xi}_{target}\} \tag{6.5}$$

Taking into consideration the independence of the N randomly deployed sensor nodes, the probability of missed detection can then be evaluated as:

$$P_{md} = (1-P_d)^N \tag{6.6}$$

$$P_{md} = (1-[\beta_s + (1-\beta_s)P\{\xi_{det}|\overline{\xi}_{target}\}])^N \tag{6.7}$$

After describing an analytical framework for the detection problem as a function of node's duty cycles. Next subsection update the analytical framework after integrating CS to the detection problem.

6.4.2 Probability of missed detection as a function of Compressive Sensing

Integrating CS to reduce the size of transmitted information to the target detection problem might lower the detection performance, as M the size of compressed measurements must be $M \geq Klog(N/K)$, where, the captured image is $(N \times N)$ and K is the number of non-zero pixels (which defines the sparsity level of the image). Hence, if M is chosen according to this bound the target is detected with high probability. Moreover the performance of the detection problem is directly proportional to the PSNR of the image after reconstruction. First, probability of detection using CS $P_{d_{cs}}$ is calculated subject to the constraint that the probability of false alarm $P_{FA} \leq \alpha_f$ as in [121, 122].

$$P_{d_{cs}} = Q(Q^{-1}(\alpha_f) - \sqrt{M/N}\sqrt{PSNR}/\sqrt{K/N}) \tag{6.8}$$

Where, $Q(x) \triangleq \int_x^\infty e^{-t^2/2}dt$ is the complementary error function of x. This gives a way to measure how much information is lost after the reconstruction, not in terms of the reconstruction error of the image, but in terms of the ability to detect the target. To reach an acceptable P_{dcs}, Φ is dynamically chosen according the sparsity nature of the image but without relaxing the randomness property of the projection measurement matrix. Thus the size of Φ (MxN) will be adaptively changing with respect to K.

The total probability of detection will then be evaluated by integrating adaptive CS to the detection problem. Hence, the total probability of detection P_{d_t} becomes

as follows:

$$P_{d_t} = (\beta_s + (1 - \beta_s)P\{\xi_{det}|\overline{\xi}_{target}\})P_{d_{cs}} \tag{6.9}$$

Resulting in a total probability of missed detection $P_{md_{CS}}$:

$$P_{md_{CS}} = (1 - [(\beta_s + (1 - \beta_s)P\{\xi_{det}|\overline{\xi}_{target}\})P_{d_{cs}}])^N \tag{6.10}$$

To maintain a high probability of detection P_{d_t} and a required PSNR while given the target's velocity, sparsity level of the image and sensing radius. One can dynamically find the best value for M that suits these requirements as in (6.12) by solving the following:

$$\hat{P}_{d_{cs}} = \begin{cases} \frac{P_{d_t} 4 d_s \pi \varsigma v}{(2\pi r_s)(\beta_s \pi \varsigma v) + 4 r_s - 4 r_s \beta_s} & \text{if } 2r_s/v < \varsigma \\ \frac{P_{d_t} 4 d_s}{2\pi r_s Z} & \text{else} \end{cases} \tag{6.11}$$

Where $Z = \beta_s + (1 - \beta_s)(\frac{4r_s - 2\sqrt{4r_s^2 - \varsigma^2 v^2}}{\pi \varsigma v} + 1 - \frac{2\arcsin(\frac{\varsigma v}{2r_s})}{\pi})$

$$M = \frac{(Q^{-1}(\alpha_f) - Q^{-1}(\hat{P}_{d_{cs}}))^2 k}{PSNR} \tag{6.12}$$

6.4.3 Probability of missed detection for multi-target detection scenario

The analysis of the detection problem is extended to consider the case of multiple targets entering the monitoring area. In this case it will be useful to evaluate the probability of missing all targets or missing at least one of the incoming targets. Assume N_T is the number of incoming targets and the probability of missing all incoming targets is denoted as P_{ma} and since the incoming targets are independent, then it can be evaluated as follows:

$$P_{ma} = (P_{md_{CS}})^{N_T} \tag{6.13}$$

Where $P_{md_{CS}}$ is the probability of missed detection in the case of single target detection after the integration of CS. The probability of missing at least one of the N_T, denoted as P_{mo} is expressed as

$$P_{mo} = 1 - (1 - P_{md_{CS}})^{N_T} \tag{6.14}$$

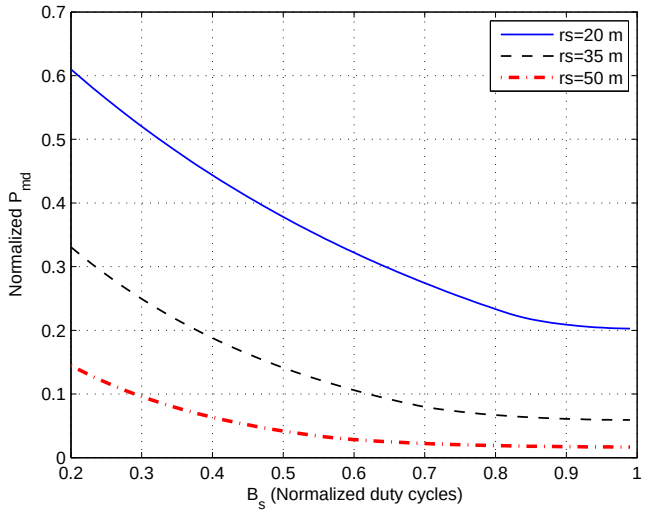
6.5 Analysis and discussion

After deriving an analytical framework for integrating node's duty cycles and CS to the detection problem, the performance of the duty-cycled WVSN is characterized in terms of probability of missed target detection. In subsequent sections, the detection performance has been tested under several parameters; different values of sensing times t_s, duty cycles β_s, sensing areas and number of sensor nodes N. All sensors are assumed to have the same sensing area r_s, and targets enter the monitored area with the same velocity v.

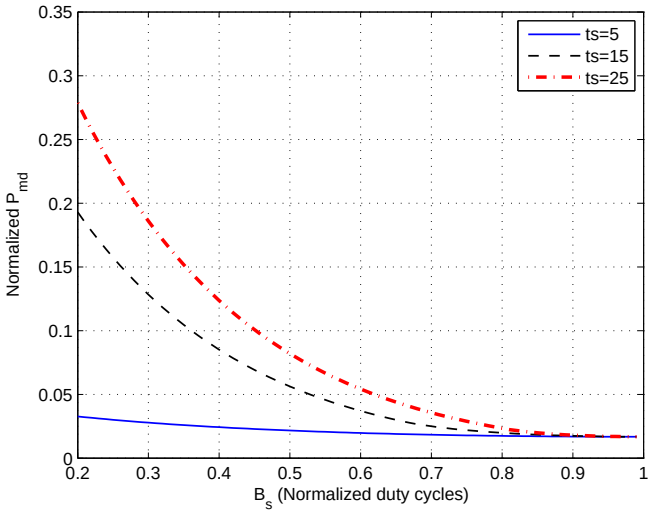
Subsequent sections illustrates the analytical results conducted after testing the mentioned parameters.

6.5.1 Probability of missed detection as a function of mobility model

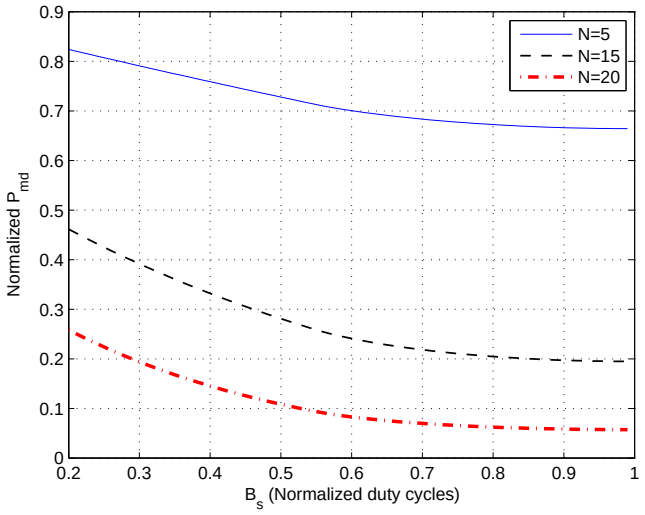
Fig.6.4 shows the P_{md} as a function of normalized duty cycles time β_s for various values of r_s, t_s and sensor nodes N, in all cases the target's velocity is $15m/s$. As illustrated, for lower values of β_s, there is a high chance the target enters



(a)



(b)



(c)

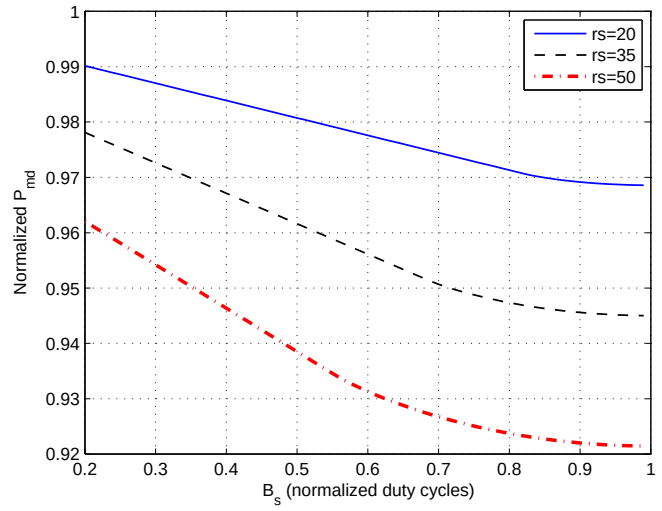
FIGURE 6.4: Probability of missed detection vs. different duty cycles for (a) different r_s ($t_s = 15sec$, $N = 50$), (b)different t_s ($N = 50$, $r_s = 50$) and (c) different number of sensor nodes ($t_s = 15sec$, $r_s = 50$). In all cases the target

and crosses the sensed area during the sleeping interval of the sensor, resulting in higher probability of missed detection. As β_s increases, the sensor node stays on for a longer time, decreasing the probability of missing a target. In Fig.6.4(a), the P_{md} is evaluated for different r_s, while t_s is set to $15sec$ and N to 50 nodes. As shown, for larger sensing areas, the higher the probability of detecting the incoming target.

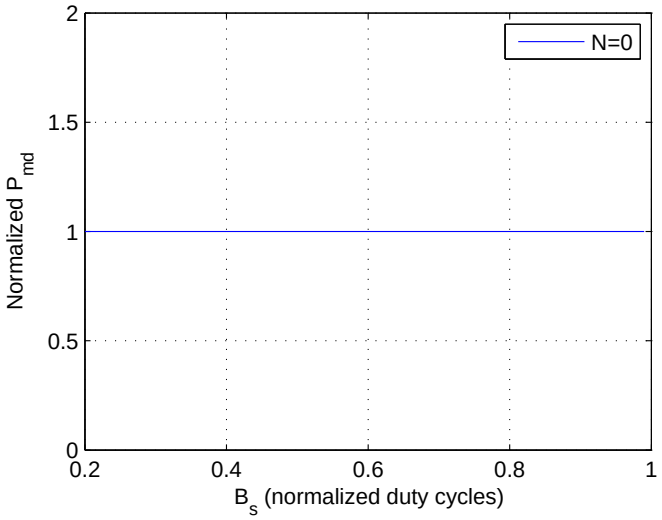
In Fig.6.4(b), P_{md} is shown for different values of t_s, while r_s and N are set to 50. It is clear from the figure that for lower values of t_s, the lower the probability of missed detection and the lower the impact of β_s on the detection problem where the total sensing period t_s is short and sensors switch to the ON state more often. While for longer t_s, β_s has a direct impact on the probability of missing a target, as β_s becomes small, the probability a target crosses the sensing area while the sensor is in the OFF state gets higher leading to a higher P_{md}. In contrast, when β_s approaches 1 (sensors remain ON), the P_{md} converges for different values of t_s as the effect of t_s on the probability of detecting a target becomes negligible.

The impact of different numbers of sensor nodes on P_{md} is illustrated in Fig.6.4(c), where t_s is set to $15sec$ and r_s to 50. As shown, as N increases, the P_{md} decreases which explains that by deploying more sensors in the monitoring geographical area the higher the chance to guarantee more sensing coverage hence reducing the probability a target is missed. On the other hand, for fewer sensors deployed, the probability a target enters a non-coverage area is high as a result the probability the target is missed is higher. Fig.6.5 shows the P_{md} as a function of β_s for $N = 1$ and different values of r_s, there is significant increase in $P_{md} > 90\%$ even the effect of increasing the sensing area on the target detection problem becomes significantly low.

Another important parameter that has a direct impact on P_{md} is the target's velocity when crossing the sensing area which in return also affects the time required by the target to cross a given sensing area T_{cross}. In Fig.6.6, P_{md} is analyzed as



(a)



(b)

FIGURE 6.5: Probability of missed detection vs. different duty cycles for (a) $N = 1$ and different r_s and (b) $N = 0$. In all cases, t_s=15sec and the target enters with velocity $v = 15m/s$

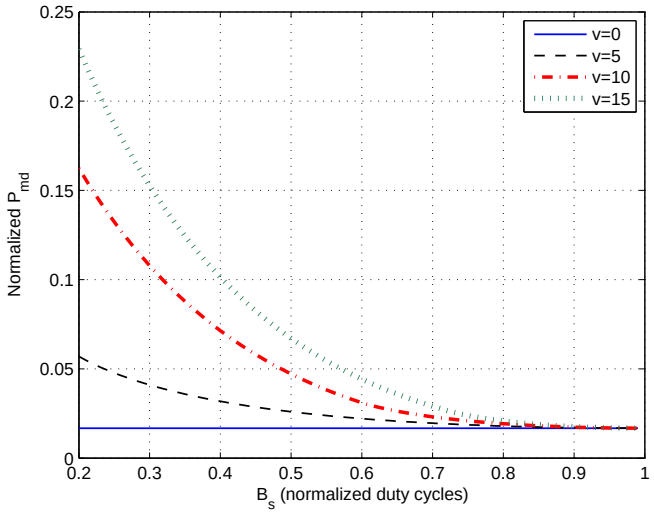


FIGURE 6.6: Probability of missed detection vs. different duty cycles for different target's velocity ($N = 50$, $r_s = 50$, $t_s = 15sec$)

a function of β_s for different values of v while other parameters are kept constant ($N = 50$, $t_s = 15sec$, $r_s = 50$). For small values of β_s (the sensor is ON for short intervals) there is a high impact of v on P_{md}, where P_{md} increases as the target's velocity increases. The higher the velocity of which the target crosses the sensing area, the shorter the time where the target crosses the sensing area T_cross and as a result the target might cross the sensing area during a sensor's sleeping interval hence resulting in higher P_{md}. On the other hand, for lower velocities, the target crosses the sensing area for T_{cross} long enough so that any sensor on the target's trajectory will detect it even if the sensor is in sleeping mode when the target enters its sensing area, there will be a high probability the sensor turns ON before the target leaves its sensing area. As β_s approaches 1, target's velocity v has a limited impact on the detection performance and P_{md} for different v values converges to reach a lower bound. It is shown that for $v = 0$ (the target stopped), the impact of β_s on P_{md} is again negligible and P_{md} becomes constant regardless the value of β_s, in such case, the probability to detect the stationary target is solely dependent on whether the target is in an area of coverage or not.

Fig.6.7 shows the P_{md} as a function of β_s for various values of r_s and v is set

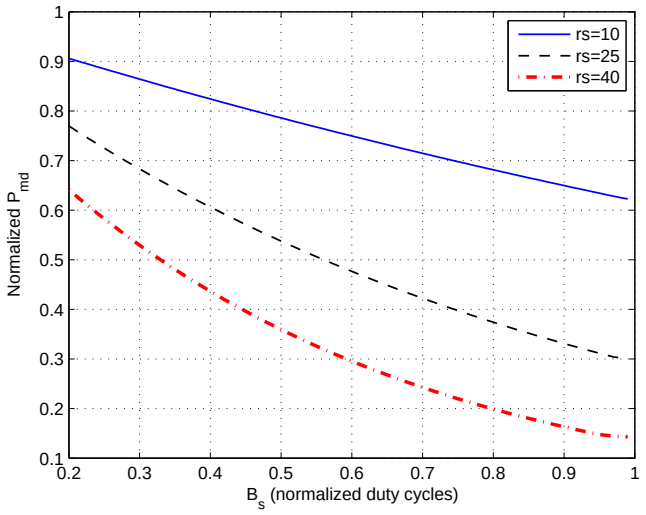


FIGURE 6.7: Probability of missed detection vs. different duty cycles for different r_s, v=100m/s, $ts = 15sec$, N=50

to $100m/s$, N to 50 and t_s to $15sec$. As stated above, higher velocities results in higher probability of missing a target specially during short β_s duty cycles as the target crosses the sensing area in a short interval of time (the case where the sensor is OFF when target enters a sensing area and target leaves the sensing area before the sensor turns On again). However, the impact of high velocities could be eliminated by longer β_s and larger r_s, to increase the probability the target remains in the sensing area for a longer interval of time till detected. This is reflected in the figure, where for a $100m/s$ velocity the P_{md} decreases as r_s and β_s increase.

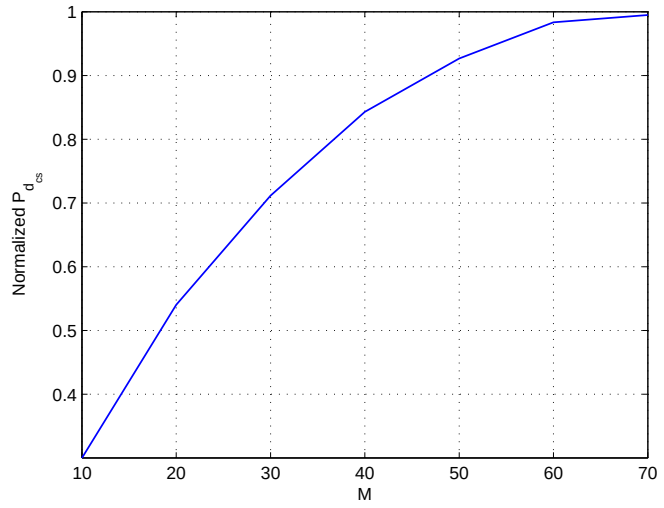
All analysis previously presented were carried out to address the target detection problem after applying sensor's duty cycles and to evaluate the probability of missing a target under various parameters. Next, the probability of missing a target is reevaluated after the integration of adaptive CS to the detection problem.

6.5.2 Probability of missed detection as a function of Compressive sensing

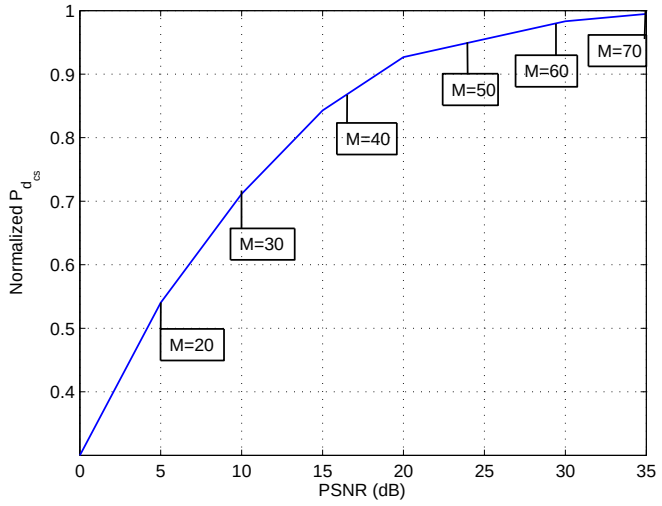
The advantages of adaptive CS to conventional CS will be illustrated in terms of energy saving and detection performance. First, we show the probability of detection by applying CS for various compression rates. Then, the impact of CS on the total probability of missed detection is illustrated.

Fig.6.8 shows the probability of detecting a target $P_{d_{cs}}$ after image reconstruction using CS for $K = 30\%$ (non-zero coefficients) with respect to various number of measurements M and reconstruction PSNR in dB. The detection is performed on the reconstructed images of different compression rates where various sizes of measurements are produced with different M till reaching $M = 70$ satisfying (6.12) as shown in Fig.6.8(a). It is clear from the figures there is a direct relation between reconstruction PSNR and the size of measurement matrix M (compression rates), which is reflected in Fig.6.8(b) where lower values of M results in low PSNR reconstructed images and as a result low $P_{d_{cs}}$. On the other hand, as M increases the $P_{d_{cs}}$ increases till nearly reaching $\approx 100\%$ and a $35dB$ PSNR.

Fig.6.9 shows the $P_{d_{cs}}$ using CS for different sizes of measurement matrices and different sparsity levels. As shown, for more sparse images the probability of detection reaches $\approx 100\%$ requiring lower values of M. For instance, $P_{d_{cs}}$ in images with $K = 3\%$ (which is 97% sparse where only 3% of the total coefficients of an image are non-zeros, the rest are zeros) reaches $\approx 100\%$ with $M = 40$. Whereas, with less sparse images $K = 30\%$, the value of M is increased till reaching 70. This illustrates the save in energy by dynamically choosing the size of measurement matrices according to the sparsity of images, where it will be a waste for a 97% sparse image to be compressed by projecting a measurement matrix with $M = 70$ at the time it could be compressed with a measurement matrix with $M = 40$ without compromising the detection probability. Furthermore, if lower values of M



(a)



(b)

FIGURE 6.8: Probability of detecting a target after CS reconstruction vs. (a) M and (b) reconstruction PSNR

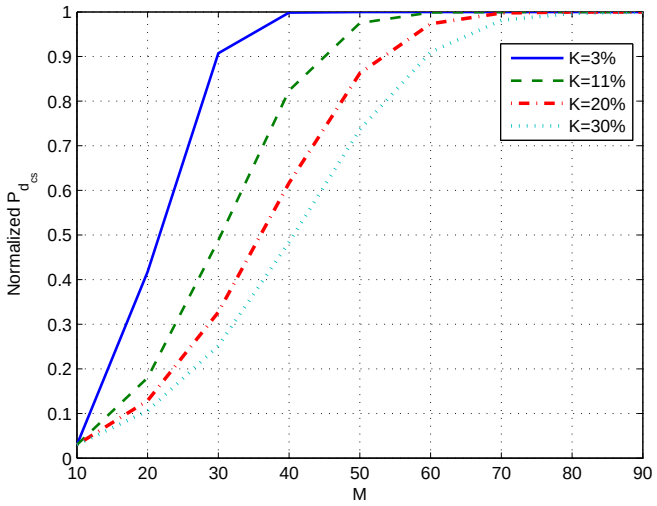
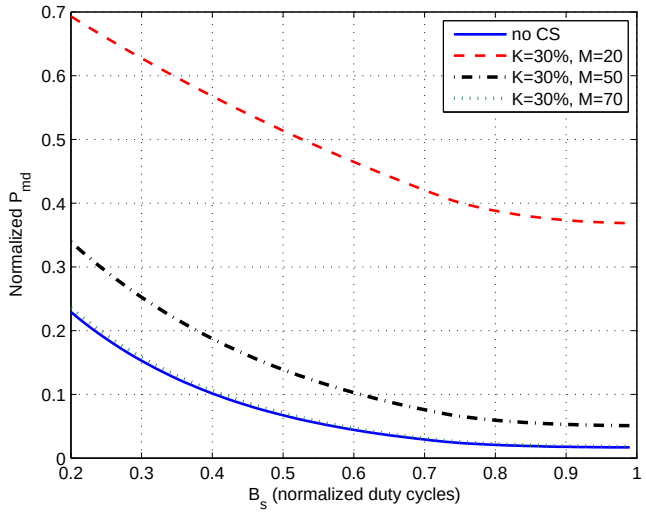


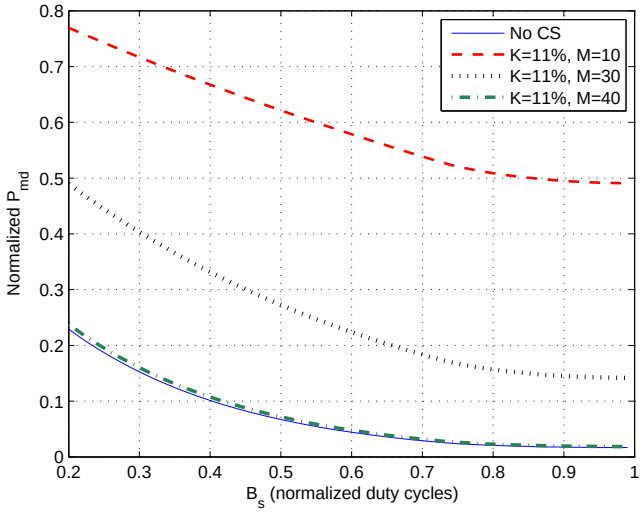
FIGURE 6.9: Probability of detection using CS vs. M for different percentage of sparsity levels

are used with less sparse images, CS fails to achieve a high PSNR of reconstructed images, and as a result the probability of detection is affected.

Integrating the concept of CS to the detection problem might degrade the probability of detection by increasing the probability of missed detection if choosing wrong values of M as shown in Fig.6.10. As previously mentioned, the value of M should be dynamically altered according to the sparsity nature of images. For illustration, Fig.6.10(a) and 6.10(b) consider different levels of sparse images $K = 30\%$ and 11%, respectively. If values of M are lower than required, the compressed image cannot be reconstructed properly, hence the probability of missing a given target increases compared to previous analysis without the integration of CS. To maintain the same probability of detection as without incorporating CS to the detection problem, CS adaptively chooses the optimum values of M according to sparsity levels. For instance, Fig.6.10(a) shows that for a $K = 30\%$ image, 70 measurements are required to achieve the same P_{md}, while to achieve the same performance of detection for a more sparse image ($K = 11\%$) without wasting energy of the communication channel bandwidth, Fig.6.10(b) shows that M is reduced to 40 measurements. If the value of M is kept constant regardless



(a)



(b)

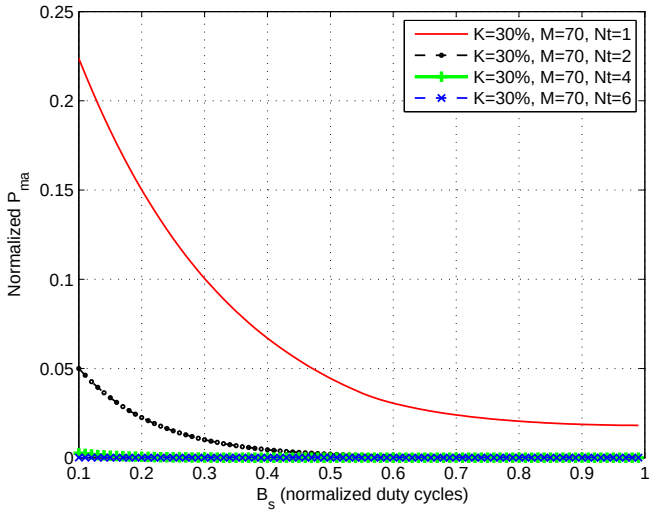
FIGURE 6.10: Probability of missed detection with and without CS vs. different duty cycles for different sparsity levels and M (a) $K = 30\%$ and (b) K=11%

the sparsity nature of different images two cases might occur; (i) if the value of M is lower than required, the probability of missed detection increases due to low PSNR reconstructed images, as a result affecting the performance of the detection problem, or (ii) if the value of M is higher than required, more measurements are produced whereas more compression could be applied, hence wasting communication bandwidth.

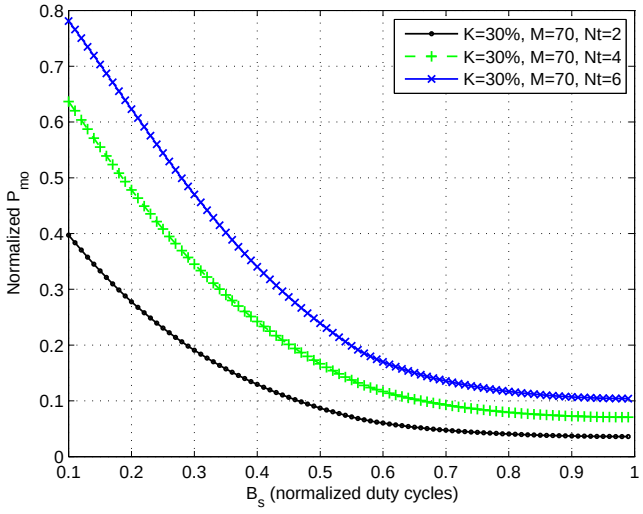
Previous results presented so far refer to the cases where a single target enters the monitored area. However, analyzing the impact of multi-targets entering the monitoring area at the same time on the probability of missed detection is challenging as illustrated in nest section.

6.5.3 Probability of missed detection for multi-target detection scenario

Analysis are carried out on the CS-integrated target detection scenario to evaluate the impact of multi-targets on the probability of missed detection. It is assumed that a single sensor can detect and take a snapshot of multiple targets crossing its sensing area. Fig.6.11 shows the effect of various number of targets entering the sensing area at the same time in the CS scenario for a given sparsity level image and the corresponding adaptively chosen value of CS measurements M. The targets enter with a velocity $v = 15m/s$, r_s is set to 50, $N = 50$ sensor nodes are deployed and t_s is set to $50sec$. Fig.6.11(a) shows the probability of missing all incoming targets (2,4 and 6) as a function of β_s, by increasing the number of incoming targets the probability of missing all targets becomes lower than P_{md} of a single-target (solid line). While in Fig.6.11(b), P_{mo} is shown as a function of β_s for various number of incoming targets (2,4 and 6), it is clear that by increasing the number of monitored targets, there is a probability that at least one of the targets is not detected.



(a)



(b)

FIGURE 6.11: Probability of missed detection vs. different duty cycles for different number of targets (a) Probability of missing all targets and (b) probability of missing at least one target. In both cases different sparsity levels and M are considered for CS

6.6 Chapter summary

In this chapter, results of the analytical framework are evaluated where probability of missed detection is used as a performance indicator. It is tested as a function of visual sensor's duty cycles under different parameters characterizing the WVSN surveillance application such as visual node's sensing radius, nodes sensing times, number of visual sensing nodes and target's velocity while crossing the sensing region. Evaluation shows the tradeoff of integrating node's duty cycles to reduce energy consumption and the detection reliability expressed in terms of probability of missed detection. Results demonstrated that integrating CS to the detection problem has not degraded the detection performance given that CS dynamically chooses appropriate compression rates depending on different sparsity levels of each video scene. Hence, saving energy while achieving same detection performance. Finally the probability of missed detection is compared for the detection of multi targets with respect to single target detection.

Due to many factors such as node deployment, number of nodes, velocity and position of targets, the performance of detection may degrade. Moreover, by integrating CS to the detection problem, the performance may features further degradation than the desired and acceptable level due to other factors such as image sparsity, loss of information in compression. Hence, there is always a tradeoff between energy consumption (network lifetime) and detection performance. As a result, we derived an analytical framework for selecting these duty cycles and dynamically choosing the appropriate compression rate for different images and videos which is expected to reduce energy waste by reaching the maximum compression rate for each dataset without compromising the probability of detection.

The next chapter briefly summarizes the contribution of the thesis, it starts with a summary of the work completed, followed by the conclusion. The chapter ends with a discussion for the future work.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this work characteristics and constraints of WVSNs are presented, together with various requirements and applications of WVSNs for surveillance applications. WVSNs are characterized as resource constraints due to limited battery power, memory space and communication bandwidth. These constraints brought new implementation challenges to investigate adaptive CS in designing efficient target detection and tracking techniques for for multi-tracking surveillance WVSN applications without compromising the tracking performance as well as energy constraint.

Existing work in the context of WVSNs surveillance applications has been presented, in addition to target detection and tracking applications together with their theoretical background as they are considered the most important tasks within WVSNs. Furthermore, different target detection techniques has been explored with their strengths and weaknesses. In addition, recursive and non-recursive background modeling techniques. Among the presented background modeling, most of the techniques are robust but not suitable for WVSNs constraints due

to either their high computational high space complexities except for AMF and Running average can result in competitive performance as MoG with more simple implementation making them candidates to WVSNs constraints. Target tracking algorithms are then introduced with their advantages and disadvantages with respect to WVSNs. Most robust techniques present in the literature are either computational extensive or high memory consuming, hence the LMS algorithm is chosen as the tracking technique due to its simplicity and its lower computational complexity compared to other tracking techniques such that Kalman filters.

Compressive sensing is investigated with some related applications in the context of target detection and tracking along with their strengths and weaknesses. CS has shown that it is expected to be a strong candidate to provide our aim in reducing the size of captured images with simple computations. After exploring related work present in the literature in the context of adaptive CS, it has shown advantages over traditional compressive sensing as compression rates are chosen according to the sparsity nature of images. However, most existing work in adaptive compressive sensing use heuristic techniques which are computationally expensive, which is not suitable for WVSN's constraints. Designing adaptive CS techniques with simple computations is therefore an important issue to be considered to provide intended performance energy efficiently while considering the resource constraint within WVSN for surveillance applications.

Next, the proposed target detection and tracking model using adaptive block CS for WVSN-based surveillance application is introduced with all the phases the algorithm passes through. CS exploited the fact that background subtraction always results in sparse frames despite the nature or how sparse the real frames are. Where after a target is being detected, CS is applied on the difference frame for efficient storage and transmission through the limited bandwidth wireless channel, afterwards reconstruction and tracking are performed at the receiver side. First, starting with how a new frame is captured by the visual sensor node and a complete

model for the WVSN-surveillance application are explained. Next, background subtraction is applied to increase the sparsity of image frames and as a result reach higher compression rates. Background modeling using running average is then applied to update the background with changes followed by morphology operations for noise removal and connected blob formation. The proposed adaptive block CS phase is later described where the images are divided into blocks and adaptively applying CS to relative blocks containing the target by choosing the number of compressed measurements according to the sparsity nature of each dataset. The proposed adaptive block CS is expected to reduce the size of transmitted data as only blocks containing the target are transmitted instead of transmitting the entire image and as a result saves communication bandwidth and transmission energy. Moreover, adaptive CS is expected to save energy as appropriate compression rates are chosen according to sparsity level of datasets. Finally, the iterative quantized LMS is introduced due to its simplicity and low power computations as a tracking technique to predict target's next locations and to test the effect of CS on the tracking performance, after the compressed image is transmitted through the wireless channel and reconstructed at the receiver side.

Experiments were carried out to evaluate the performance of the adaptive CS and its effect on target detection and tracking. MSE is used as an indicator to measure the accuracy and reliability of tracking, results yields minimum MSE where it decreases till reaches the lower bound of the number of compressed measurements while preserving the acceptable PSNR, while addressing the problems of WVSN, such as energy, memory and bandwidth constraints. Results also proved that for single target tracking fewer number of measurements are needed as the difference frame is more sparse compared to multi-target tracking, reaching a relation between the the number of compressed measurements and ratio of non-zero pixels to the total number of pixels. As a result, when higher compression rates are required, one control the targets size by zooming out or changing the location of sensor nodes during the CS calibration phase while bearing in mind to keep

the scene of interest in the camera's field of view. Moreover, to guarantee CS reconstruction, trajectory tracking paths of detected moving targets are compared with real targets locations and results proved the efficiency and feasibility of the tracking technique.

The adaptive block CS technique has has achieved higher compression rates with minimum reconstruction error, hence saves memory and communication bandwidth, as well as consuming low processing power at sensor nodes. Moreover, since the compression rates differ according to different datasets, as it depends on the degree of sparsity from one image to another. For this reason adaptive CS has proven to achieve relatively high compression rates for each dataset depending on how sparse an image is. In addition, block CS is energy efficient where it divides the image frame into blocks and only blocks containing the target is compressed and transmitted in contrast to basic CS where the whole frame is compressed, hence saving power of processing. Moreover, the reconstruction has produced acceptable results which is illustrated by testing the ability of the proposed target detection and tracking technique to correctly locate and track the desired target extracted from the reconstructed images. The proposed CS method as a result did not degrade the performance of detection and tracking. However, the background and target's appearances are initially assumed to be static for simplicity. Hence, there is a direction for future work to handle the cases of dynamic background and target's appearance.

Due to many factors such as node deployment, number of nodes, velocity and position of targets, background movements, the performance of detection may degrade. Moreover, by integrating CS to the detection problem, the performance may features further degradation than the desired and acceptable level due to other factors such as image sparsity, loss of information in compression. Hence, there is always a tradeoff between energy consumption (network lifetime) and detection performance. An analytical framework is derived for selecting these duty cycles and dynamically choosing the appropriate compression rate for different

images and videos which is expected to reduce energy waste by reaching the maximum compression rate for each dataset without compromising the probability of detection. The results showed that by choosing the appropriate compression rates according to the sparsity levels of different datasets, the probability of detection is maintained without further degradation.

7.2 Future work

Since the background is not always static due to several factors such as illumination changes, rain, fog, or moving background objects as trees, one of the objectives for future work is to design a robust and reliable object detection technique, to handle dynamic background, Moreover, the visual sensing node itself might not be fixed as it could rotate, zoom or tilt. The background as a result should be modeled in such a way to continuously adapt to any changes or motion in the background and prevent false detection of any background object as a target.

Visual tracking has always been a challenging problem as the appearance of targets often changes as a result of different factors as pose, scale variations, full or partial occlusion, illumination conditions, abrupt motion, or different viewpoints. Hence, a robust target tracking algorithm is to be established for the future work. The algorithm is expected to handle the appearance changes of the target by designing an energy efficient robust target appearance model that is to be updated online over time to adapt to any changes. In some literature, some model only the target appearance while others model both the target and the background. The latter approach has shown that it achieved better results as separating the target from the background is modeled as a binary discriminative classification problem, this is related to tracking by detection.

Much work has been proposed in the literature for the problems of appearance changes in visual tracking. In [123], the authors proposed a feature selection theme

named Active Feature Selection to select the most discriminative features for separating the target from the background. This is done like the multiple instance learning (MIL) tracker [124]; by constructing image patches known as positive and negative bags but instead of maximizing the log likelihood of the bags to train the classifier, a bag fisher information function is optimized. Finally a boosting function is used to update the classifier. In [125], an online tracking algorithm based on sparse coding is proposed. The authors encode the appearance of both the target and the background through an online discriminative dictionary. The appearance model is then trained using the sparse codes. The authors in [126] proposed a robust tracking algorithm with a local sparse appearance model composed of a static dictionary for limiting the drifts and a dynamic dictionary represented by a sparse coding histogram and updated online. Moreover, K-selection scheme is introduced as a sparse dictionary learning method. Finally, matching the target and candidate models through the coding histogram and using a voting map to track the target object.In [127], a linear coding strategy is proposed to represent the targets appearance while to effectively integrate multiple features by introducing a multi-cue fusion strategy. A learning approach is developed to update the dictionary to adapt to changes in targets appearance. This update is done according to the tracking results.

For the future work, a robust appearance model for the target is to be constructed after the reconstruction of the compressed measurements (frame). A typical way of building appearance models is by first computing local descriptors of an image; where the appearance of a target object can be modeled by various features (such as color, brightness histograms, edge histogram, shape, texture, etc.). During the training phase, an initial dictionary is to be constructed from features collected to represent extracted image patches around the target region (which is manually detected), the target location is denoted as the tracker location. The reason for collecting image patches around the target region (that might be overlapping), is to model different variations of the target appearance to help in tracking in

complex environment (such as occlusion). Subsequently, after the training phase, these features are to be updated through consecutive image sequences. Each image patch is associated with a label, that is computed and set as a 1 or -1 during the training of the dictionary to indicate whether the image patch contains a target or just the background.

Later, for each new captured frame a new set of image patches are collected around the previous tracker location of the target forming a set denoted as a feature bag. Using the dictionary, the new target location is detected by matching the target model to candidate targets, afterwards the new tracker location is updated together with the appearance model dictionary to adapt to changes. Fig.7.1 shows a block diagram that summarizes the future model.

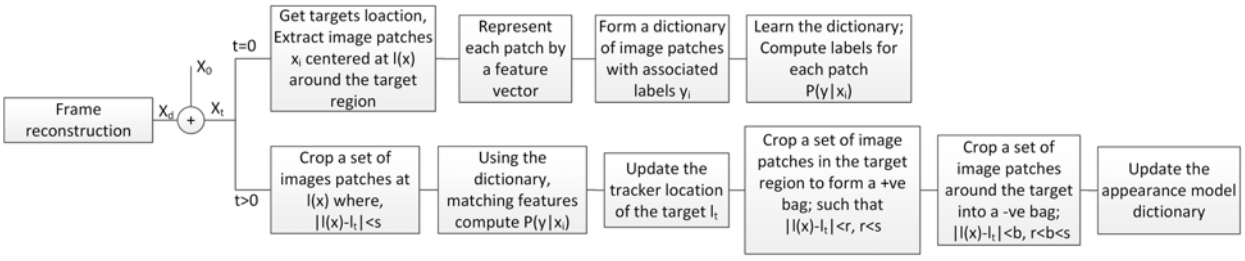


FIGURE 7.1: Proposed dynamic model for target detection and tracking

Furthermore, the proposed compressive sensing technique achieved good results in terms of least MSE, PSNR, and targets trajectory tracking. But for future work the performance of the proposed integrated target detection and tracking schemes using CS for WVSN-based surveillance applications is to be compared to other techniques in the context of resource requirements, robustness, and reliability.

Bibliography

[1] A.Sharif, V.Potdar, and E.Chang. Wireless multimedia sensor network technology: A survey. In *Proceedings of Industrial Informatics, 7th IEEE International Conference, 2009. INDIN 2009.*, pages 606 –613, June 2009.

[2] F.Wang and J. Liu. Networked wireless sensor data collection:issues, challenges, and approaches. *Communications Surveys and Tutorials, IEEE*, 99: 1–15, 2010.

[3] S.Soro and W.Heinzelman. A survey on visual sensor networks. *Hindawi publishing corporation, Advances in Multimedia*, 2009(640386):1–21, May 2009.

[4] Y.Charfi, B.Canada, N.Wakamiya, and M.Murata. Challenges issues in visual sensor networks. In *IEEE on wireless Communications*, pages 44–49, April 2009.

[5] I.F.Akyildiz, T.Melodia, and K.R.Chowdhury. A survey on wireless multimedia sensor networks. *computer Networks*, 51:921–960, March 2007.

[6] S.Y. Elhabian, K.M. El-Sayed, and S.H. Ahmed. Moving object detection in spatial domain using background removal techniques - state-of-art. *Recent Patents on Computer Science*, 1(1):32–54, 2008.

[7] Z.Tanf, Z.Miao, and Y.Wan. Background subtraction using running gaussian average and frame difference. In *Entertainment Computing ICEC 2007*, volume 4740 of *Lecture Notes in Computer Science*, pages 411–414, 2007.

[8] M.Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, October 2004.

[9] M.A.Najjar, M.Ghantous, and M.Bayoumi. *Video Surveillance for Sensor Platforms: Algorithms and Architectures*. Springer, 2014. ISBN 978-1-4614-1856-6.

[10] A. Redondi, D. Buranapanichkit, M. Cesana, M. Tagliasacchi, and Y. Andreopoulos. Energy consumption of visual sensor networks: Impact of spatiotemporal coverage. *IEEE Transaction on Circuits System and Video Technology*, 24(12):2117–2131, Dec 2014.

[11] R.G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, pages 118–124, July 2007.

[12] J.Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, pages 14–20, March 2008.

[13] M.A.Patricio, J.Carbo, O.Perez, J.Garcia, and J.M.Molina. Multi-agent framework in visual sensor networks. *Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing*, 2007(98639), August 2006.

[14] A.Prat, R.Vezzani, L.Benini, E.Farella, and P.Zappi. An integrated multimodal sensor network for video surveillance. In *Proceedings The ACM International Workshop on Video Surveillance and Sensor Networks*, pages 95–102, November 2005.

[15] P.Skraba and L.Guibas. Energy efficient intrusion detection in camera sensor networks. In *Proceedings of the International Conference on Distributed Sensor Systems DCOSS 07*, pages 309–323, June 2007.

[16] D.Xie, T.Yan, D.Ganesan, and A.Hanson. Design and implementation of a dual-camerawireless sensor network for object retrieval. In *IEEE 2008*

International Conference on Information Processing in Sensor Networks, 2008.

[17] C.R.Baker, K.Armijo, S.Belka, M.Benhabib, V.Bhargava, N.Burkhart, A.D.Minassians, Gunes Dervisoglu, Lilia Gutnik, M. Brent Haick, Christine Ho, Mike Koplow, Jennifer Mangold, Stefanie Robinson, Matt Rosa, Miclas Schwartz, Christo Sims, Hanns Stoffregen, Andrew Waterbury, Eli S. Leland, Trevor Pering, and Paul K. Wright. Wireless sensor networks for home health care. In *IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 2007.

[18] C.Hartung, R.Han, C.Seielstad, and S.Holbrook. Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. *Proceedings of the 4th international conference on Mobile systems, applications and services, Uppsala, Sweden*, 2006.

[19] P.Rajeswari, S.Pratheeba, and S.R.Karthika. A comprehensive overview on different applications of wireless sensor network. *International Journal of Engineering and Advanced Technology (IJEAT)*, 3(4):80–84, April 2014.

[20] P.Zhang, C.M.Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems Sensys 04*, November 2004.

[21] X.Wang, S.Wang, and D.Bi. Distributed visual-target-surveillance system in wireless sensor networks. *IEEE Transactions on Systems, MAN, and Cybernetics*, 39(5):1134–1146, October 2009.

[22] X.Wang, S.Wang, D.W.Bi, and J.J.Ma. Distributed peer-to-peer target tracking in wireless sensor networks. *MDPI, open access journal on the science and technology of sensors and biosensors*, 7:1001–1027, 2007.

[23] X.Wang and S.Wang. Collaborative signal processing for target tracking in distributed wireless sensor networks. *Elsevier journal on Parallel and distributed computing*, 67:501 515, 2007.

[24] P.Medagliani, J.Leguay, G.Ferrari, V.Gay, and M.Lopez-Ramos. Energy-efficient mobile target detection in wireless sensor networks with random node deployment and partial coverage. *Pervasive and Mobile Computing*, 8 (3):429447, 2012.

[25] Q.Cao, T.Yan, J.Stankovic, and T.Abdelzaher. Analysis of target detection performance for wireless sensor networks. In *In DCOSS05*, pages 276–292, 2005.

[26] B.Stojkoska, D.Davcev, and V.Trajkovik. N-queens-based algorithm for moving object detection in distributed wireless sensor networks. *Journal of Computing and Information Technology CIT*, 4:325332, 2008.

[27] J. M.Gomez, A.J.Picazo, and I. G. Varea. A particle-filter-based self-localization method using invariant features as visual information. In *Robocup*, 2010.

[28] A.C.Sankaranarayanan, A.Veeraraghavan, and R.Chellappa. Object detection, tracking and recognition for multiple smart cameras. In *Proceedings of the IEEE*, volume 96, pages 1606–1624, October 2008.

[29] Y.Wang, S.Velipasalar, and M.Casares. Cooperative object tracking and composite event detection with wireless embedded smart cameras. *IEEE Transactions on Image Processing*, 19(10):2614 –2633, October 2010.

[30] J. Nascimento and J. Marques. Performance evaluation of object detection for video surveillance. *IEEE Transaction on Multimedia*, 8(4):761–774, 2006.

[31] F. ElBaf, T. Bouwmans, and B. Vachon. Comparison of background subtraction methods for a multimedia application. In *14th International Workshop*

on Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, pages 385 –388, June 2007.

[32] D.Hall, J.Nascimento, P.Ribeiro, E.Andrade, and P.Moreno. Comparison of target detection algorithms using adaptive background models. *In Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2005.

[33] H.S.Parry, A.D.Marshall, and K.C.Markham. Region template correlation for the flir target tracking. In *British Machine Vision Conference*.

[34] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809 –830, April 2000.

[35] D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *The Institution of Engineering and Technology, Control theory applications*, 4(8):1303–1318, 2010.

[36] D.H.ying, C.Bin, and Y.Y.ping. Application of particle filter for target tracking in wireless sensor networks. In *International Conference on Communications and Mobile Computing (CMC)*, volume 3, pages 504 –508, April 2010.

[37] J.C.Noyer, P.Lanvin, and M.Benjelloun. Non-linear matched filtering for object detection and tracking. *Elsevier Pattern Recognition Letters*, 25:655–668, 2004.

[38] S. Lefevre and N. Vincent. Real time multiple object tracking based on active contours. September 2004.

[39] J.Malcolm, Y.Rathi, and A.Tannenbaum. Multi-object tracking through clutter using graph cuts. In *The International Conference on Computer Vision (ICCV)*, 2007.

[40] Q.Chen, Q.S.Sun, P.A.Heng, and D.Xia. Two-stage object tracking method based on kernel and active contour. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(4):605 –609, April 2010.

[41] V.Cevher, A.Sankaranarayanan, M.F. Duarte, D.Reddy, R.G. Baraniuk, and R.Chellappa. Compressive sensing for background subtraction. 2008.

[42] E. Wang, J. Silva, and L. Carin. Compressive particle filtering for target tracking. In *IEEE/SP 15th Workshop on Statistical Signal Processing, SSP*, pages 233 –236, September 2009.

[43] L.D.Stefano, F.Tombari, and S.Mattoccia. Robust and accurate change detection under sudden illumination variations. In *ACCV Workshop on Multidimensional and Multi-view Image Processing*, November 2007.

[44] P.W.Power and J. A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings of IVCNZ*, pages 267–271, November 2002.

[45] M.AlNajjar, M.Ghantous, and M.Bayoumi. *Video Surveillance for Sensor Platforms: Algorithms and Architectures, Lecture Notes in Electrical Engineering 114*, chapter Visual Sensor Nodes, pages 17–35. Springer, 2014. doi: 10.1007/978-1-4614-1857-3_2.

[46] N.Lu, J.Wang, Q. H.Wu, and L.Yang. An improved motion detection method for real-time surveillance. *IAENG International Journal of Computer Science*, 2008.

[47] T. Boult, R. Micheals, X. Gao, and M. Eckmann. Into the woods: Visual surveillance of non-cooperative camouflaged targets in complex outdoor settings. In *Proceedings of the IEEE*, pages 1382–1402, October 2001.

[48] C.Wren, A.Azarbayejani, T.Darrell, and A.Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19 of *7*, pages 780–785, July 1997.

[49] N.Friedman and S.Russell. Image segmentation in video sequences: a probabilistic approach. In *International Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[50] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. In *Proceedings of the IEEE*, July 2002.

[51] A.Nurhadiyatna, W.Jatmiko, B.Hardjono, A.Wibisono, I.Sina, and P.Mursanto. Background subtraction using gaussian mixture model enhanced by hole filling algorithm (gmmhf). In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4006–4011, Oct 2013.

[52] N.McFarlane and C.Schofield. Segmentation and tracking of piglets in images. *Machine Vision Application*, 83:187–193, 1995.

[53] Z.Yi and F.Liangzhong. Moving object detection based on running average background and temporal difference. In *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 270–272, November 2010.

[54] M.Rahimi, R.Baer, O.I.Iroezi, J.C.Garcia, J.Warrior D.Estrin, and M.Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. *In SenSys 2005*, 2005.

[55] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:172–185, 2005.

[56] A.Yilmaz, O.Javed, and M.Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006.

[57] D.Gao, L.Liang, X.Wang, and S.Zhang. Image based target tracking for a moving objectin camera sensor networks. In *The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 4, pages 627 –632, February 2010.

[58] R.Cai, Q.Wu, P.Wang, X.Zhang, and S.Hu. Performance analysis of object tracking algorithm. In *2011 International Conference on Image Analysis and Signal Processing (IASP)*, pages 463–467, October 2011.

[59] H.A.Patel and D.G.Thakore. Moving object tracking using kalman filter. *International Journal of Computer Science and Mobile Computing, IJCSMC*, 2(4):326–332, April 2013.

[60] X.Li, K.Wang, W.Wang, and Y.Li. A multiple object tracking method using kalman filter. In *IEEE International Conference on Information and Automation (ICIA)*, pages 1862–1866, 2010. doi: 10.1109/ICINFA.2010.5512258.

[61] M.Han, W.Xu, H.Tao, and Y.Gong. An algorithm for multiple object trajectory tracking. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, volume 1, pages 864–871, 2004.

[62] Y.M.Kim. Object tracking in a video sequence. Technical report, Stanford, final year project report, 2007.

[63] M. Elad and A. Feuer. Recursive optical flow estimation-adaptive filtering approach. *Journal of Visual Communication and Image Representation*, 9 (2):119 – 138, 1998.

[64] S.Haykin. *Adaptive Filter Theory*, volume 0-13-048434-2, chapter Least mean square adaptive filters, pages 231–247. Prentice Hall, 2002.

[65] Y.Zheng, H.Wang, and Q.Guo. A novel mean shift algorithm combined with least square approach and its application in target tracking. In *IEEE*

11th International Conference on Signal Processing (ICSP), volume 2, pages 1102–1105, Oct 2012.

[66] G.H.Costa and J.C.M.Bermudez. Statistical analysis of the lms algorithm applied to super-resolution image reconstruction. *IEEE Transactions on Signal Processing*, 55(5):2084–2095, May 2007.

[67] Z.Q.Zhoo, X.Liu, J.Chen, and H.J.Wu. Intelligent pid control with adaptive filter for the target tracking sytem. *International Conference on Mechanical Design, Manufacture and Automation Engineering*, pages 212–216, 2014.

[68] P.S.R.Diniz. *Adaptive Filtering*, volume 399, chapter The Least-Mean-Square (LMS) Algorithm, pages 79–135. The Springer International Series in Engineering and Computer Science, January 1997.

[69] *Appendix E: Orthogonalizing Adaptive Algorithms: RLS, DFT/LMS, and DCT/LMS*, pages 383–395. John Wiley & Sons, Inc., 2007. ISBN 9780470231616. URL `http://dx.doi.org/10.1002/9780470231616.app5`.

[70] Y.Xia, L.Jianchang, and L.Hongru, editors. *Performance Analysis of Adaptive Filters for Time-Varying Systems*, July 2013. Proceedings of the 32nd Chinese Control Conference.

[71] D.V.A.N. Kumar, S.KoteswaraRao, and K.P.Raju. Under water active target tracking using kalman filter. *International Journal of Engineering Research and Technology (IJERT)*, 2(10):3982–3988, October 2013.

[72] T.W.Bae, F.Zhang, and I.S.Kweon. Edge directional 2d lms filter for infrared small target detection. *Infrared Physics and Technology, Sciencedirect*, 55 (1):137–145, 2012.

[73] H.S.Yazdi, M. Lotfizad, and M.Fathy. Car tracking by quantised input lms, qx-lms algorithm in traffic scenes. *Vision, Image and Signal Processing, IEE Proceedings*, 153(1):37–45, 2006.

[74] S.Olmos and P. Laguna. Steady-state mse convergence of lms adaptive filters with deterministic reference inputs with applications to biomedical signals. *Signal Processing, IEEE Transactions on*, 48(8):2229–2241, 2000.

[75] Chih-Hsien Hsia, Yi-Ping Yeh, Tsung-Cheng Wu, Jen-Shiun Chiang, and Yun-Jung Liou. Low resolution method using adaptive lms scheme for moving objects detection and tracking. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on*, pages 1–4, 2010.

[76] G.Caner, A.M.Tekalp, G.Sharma, and W.Heinzelman. An adaptive filtering framework for image registration. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP '05)*, volume 2, pages 885–888, 2005.

[77] J.Haupt and R.Nowak. Compressive sampling vs. conventional imaging. In *IEEE International Conference on Image Processing*, pages 1269–1272, October 2006.

[78] A.M.Abdulghani and E.R.Villegas. Compressive sensing: From "compressing while sampling" to "compressing and securing while sampling". In *Annual International Conference of the IEEE EMBS Buenos Aires*, volume 32, pages 1127–1130, 2010.

[79] E.J.Candes and M.B.Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pages 21–30, March 2008.

[80] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[81] M.Zhao, A.Wang, B.Zeng, L.Liu, and H.Bai. Depth coding based on compressed sensing with optimized measurement and quantization. *Ubiquitous International Journal of Information Hiding and Multimedia Signal Processing*, 5(3):475–484, July 2014.

[82] C.Patsakis and N.G.Aroukatos. Lsb and dct steganographic detection using compressive sensing. *Ubiquitous International Journal of Information Hiding and Multimedia Signal Processing*, 5(4):20–32, January 2014.

[83] H.R.ALZoubi. Video coding and routing in wireless video sensor networks. *AASRI Conference on Parallel and Distributed Computing and Systems*, 5 (0):48 – 53, 2013.

[84] S.Pudlewski, A.Prasanna, and T.Melodia. Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. *IEEE Transactions on Mobile Computing*, 11(6):1060–1072, June 2012.

[85] M.S.Asif, F.Fernandes, and J.Romberg. Low-complexity video compression and compressive sensing. In *In Asilomar Conference on Signals, Systems, and Computers*, 2013.

[86] A.Mahalanobis and R.Muise. Object specific image reconstruction using a compressive sensing architecture for application in surveillance systems. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1167–1180, July 2009.

[87] D. Reddy, A.C. Sankaranarayanan, V. Cevher, and R. Chellappa. Compressed sensing for multi-view tracking and 3-d voxel reconstruction. In *Proceedings of the IEEE International Conference on Image Processing (ICIP*, 2008.

[88] C.T.Chou, R.Rana, and W.Hu. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In *IEEE 34th Conference on Local Computer Networks (LCN)*, pages 443–450, Zrich, Switzerland, October 2009.

[89] E.J.Candes and B.Recht. Exact matrix completion via convex optimization. *CoRR*, abs/0805.4471, 2008.

[90] S. Deutsch, A. Averbuch, and S. Dekel. Adaptive compressed image sensing based on wavelet modeling and direct sampling. In *Proceedings of the 8th International Conference on Sampling Theory and Applications, Marseille, France*, 2009.

[91] S. Dekel. Adaptive compressed image sensing based on wavelet-trees. online. Available: http://www.dsp.ece.rice.edu/cs/, 2008.

[92] A.Redondi, D.Buranapanichkit, M.Cesana, M.Tagliasacchi, and Y.Andreopoulos. Energy consumption of visual sensor networks: Impact of spatio-temporal coverage. *IEEE tranaction on Circuits and Systems for Video Technology*, 24(12):2117–2131, December 2014.

[93] P.Mohanty, M.R.Kabat, and M.K.Patel. Energy efficient transmission control protocol in wireless sensor networks. In *Wireless Networks and Computational Intelligence*, volume 292 of *Communications in Computer and Information Science*, pages 56–65. Springer Berlin Heidelberg, 2012.

[94] G.J.Pottie and W.J.Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, may 2000.

[95] T.Melodia, D.Pompili, and I.F.Akyildiz. A communication architecture for mobile wireless sensor and actor networks. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON*, volume 1, pages 109–118, Sept 2006.

[96] F.Shebli, I.Dayoub, A.O.M'foubat, A.Rivenq, and J.M.Rouvaen. Minimizing energy consumption within wireless sensors networks using optimal transmission range between nodes. In *IEEE International Conference on Signal Processing and Communications, ICSPC*, pages 105–108, Nov 2007.

[97] P.Han, J.Du, J.Zhou, and S.Zhu. An object detection method using wavelet optical flow and hybrid linear-nonlinear classifier. *Mathematical Problems in Engineering, Hindawi publishing corporation*, 2013, 2013.

[98] Sen-C.S.Cheung and C.Kamath. Robust techniques for background subtraction in urban traffic video. *Video Communications and Image Processing, SPIE Electronic Imaging, San Jose*, 2004.

[99] Nima Seif Naraghi. A comparative study of background estimation algorithms. Master's thesis, Eastern Mediterranean University, Gazimausa, North Cyprus, 2009.

[100] D.Reynolds. Gaussian mixture models. Technical report, Department of Defense, under Air Force Contract,MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA.

[101] E.J.Candes. Compressive sampling. In *Proc. of the International Congress of Mathematicians*, 2006.

[102] J.Romberg and M.wakin. Compressed sensing: A tutorial ieee statistical signal processing workshop madison. *IEEE Statistical Signal Processing Workshop, georgia tech university of michigan*, 2007.

[103] M.F.Duarte, M.A.Davenport, D.Takhar, J.N.Laska, T.Sun, K.F. Kelly, and R.G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, pages 83–91, March 2008.

[104] A.Hormati, O.Roy, Y.M.Lu, and M.Vetterli. Distributed sampling of signals linked by sparse filtering: theory and applications. *IEEE Transactions on Signal Processing*, 58(3):1095–1109, March 2010.

[105] Ms.V.MuthuLakshmi. Advanced leach protocol in large scale wireless sensor networks. *International Journal of Scientific and Engineering Research*, 4 (5):248–254, May 2013.

[106] E.M.Martn and .P.Pobil. *Robust Motion Detection in Real-Life Scenarios*, chapter Ch.2 Motion Detection in Static Background, pages 5–41. Springer, 2012.

[107] N.Efford. *Digital Image Processing: A Practical Introduction Using JavaTM*, chapter Morphological image processing. Pearson Education, 2000.

[108] P.Kupidura. *Application of mathematical morphology operations for the improvement of identification of linear objects preliminarily extracted from classification of VHR satellite images*. 2006. ISBN 978-90-5966-053-.

[109] S.B.Jebara and H.Besbes. A variable step size filtered sign algorithm for acoustic echo cancellation. In *IEEE electronic letters*, volume 39, pages 936–93, 2003.

[110] S.Dhull, S.Arya, and O.P. Sahu. Performance variation of lms and its different variants. *International Journal of Computer Science and Security, (IJCSS)*, 4, 2010.

[111] F. Cheng and Y. Chen. Real time multiple objects tracking and identification based on discrete wavelet transform. *Elsevier Pattern Recognition Journal*, 39:1126 1139, 2006.

[112] Caviar datasets. Dataset: EC Funded CAVIAR project/IST 2001 37540, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, 2001.

[113] Y.Wu, J.Lim, and M.Yang. Visual tracker bemchmark. https://sites.google.com/site/trackerbenchmark/benchmarks/v10, 2013.

[114] Hong Jiang, Wei Deng, and Zuowei Shen. Surveillance video processing using compressive sensing. *arXiv preprint arXiv:1302.1942*, 2013.

[115] F.G.H.Yap and H.H.Yen. A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks. *Sensors*, 14:3506–3527, February 2014.

[116] T.Winkler and B.Rinner. Security and privacy protection in visual sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 47(1), July 2014.

[117] Y.Wang and D.Wang. Energy-efficient node selection for target tracking in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[118] O.Demigha, W.K.Hidouci, and T.Ahmed. On energy efficiency in collaborative target tracking in wireless sensor network: A review. *IEEE Communications Surveys Tutorials*, 15(3), 2013.

[119] R.Tan, G.Xing, J.Wang, and H.C.So. Collaborative target detection in wireless sensor networks with reactive mobility. In *16th International Workshop on Quality of Service IWQoS*, pages 150–159, June 2008.

[120] P.Medagliani, J.Leguay, V.Gay, M.Lopez-Ramos, and G.Ferrari. Engineering energy-efficient target detection applications in wireless sensor networks. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 31–39, March 2010.

[121] Z.Wang, G.R.Arce, B.M.Sadler, J.L.Paredes, and Xu Ma. Compressed detection for pilot assisted ultra-wideband impulse radio. In *IEEE International Conference on Ultra-Wideband ICUWB*, pages 393–398, September 2007.

[122] M.A.Davenport, M.B.Wakin, and R.G.Baraniuk. Detection and estimation with compressive measurements. Technical report, Rice University, Department of ECE, Technical Report, 2006.

[123] K.Zhang, L.Zhang, M.H.Yang, and Q.Hu. Robust object tracking via active feature selection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(11):1957–1967, 2013.

[124] B.Babenko, M-H. Yang, and S.Belongie. Visual tracking with online multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009.*, pages 983–990, 2009.

[125] Y.Xie, W.Zhang, C.Li, S.Lin, and Y.Qu Y.and Zhang. Discriminative object tracking via sparse representation and online dictionary learning. *Cybernetics, IEEE Transactions on*, PP, 2013.

[126] L.Baiyang, H.Junzhou, C.Kulikowski, and Y.Lin. Robust visual tracking using local sparse appearance model and k-selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2968–2981, 2013.

[127] H.Liu, M.Yuan, F.Sun, and J.Zhang. Spatial neighborhood-constrained linear coding for visual object tracking. *IEEE Transactions on Industrial Informatics*, PP, 2013.