A Low-Complexity Algorithm for Intrusion Detection in a PIR-Based Wireless Sensor Network

A Thesis

Submitted For the Degree of

Master of Science (Engineering)

in the Faculty of Engineering

by

Ramanathan Subramanian



Department of Computer Science and Automation

Indian Institute of Science

BANGALORE – 560 012

May 2010

©Ramanathan Subramanian

May 2010

All rights reserved

Acknowledgements

I thank my advisor, Prof. P. Vijay Kumar, for his guidance and support throughout the duration of my masters program, without whom this work would not have materialized. I would like to thank my collaborator in research, Abu Sajana, whose ideas are also contained in this thesis. I would also like to thank Prof. Bharadwaj Amrutur and Syam Krishnan for having helped us in addressing the problem of PIR sensor modeling. I would like to thank Jeena Sebastian, Dr. Malati Hegde, S. V. R. Anand for having been instrumental in implementing our algorithm in real-time. I would like to thank our faculty colleagues: Prof. V. K. Aatre, Prof. Anurag Kumar, Prof. Joy Kuri and Prof. Rajesh Sundaresan. I would also like to thank CAIR, DRDO, Bangalore, which is the funding agency for this project. I would like to express my thanks to the WSN team at IISc, in particular, Shalini Keshavamurthy, Sundeep Patil, Anurag Ranjan, Sharath Kumar, V. L. Poornima, R. Santhosh, Aditya Prasad. I would like to thank Dr. A. Vijay Krishna for introducing the SVM to us. I thank my parents and my brother for having been a constant source of motivation and support. Thanks are also to my seniors Venkatesan, Naveen, Gagan Garg, Sreeram, Vinodh, Krishnakumar and Birenjith and my other friends at IISc for their kind support and making my stay at IISc a memorable one. I would like to thank Prof. M. Narasimha Murthy, Chairman, Department of CSA, IISc, for all his kindness and support at difficult times. Finally, sincere thanks to my examiner Prof. D. Manjunath, IIT Bombay for reviewing my thesis and providing me his valuable comments.

Publications based on this Thesis

Conference

• Abu Sajana R., Ramanathan Subramanian, P. Vijay Kumar, Syam Krishnan, Bharadwaj Amrutur, Jeena Sebastian, Malati Hegde, S. V. R. Anand, "A Low-Complexity Algorithm for Intrusion Detection in a PIR-Based Wireless Sensor Network," accepted for publication in *Proc. of the Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2009)*, Melbourne (first two authors are in alphabetical order).

• The SmartDetect WSN Team, "SmartDetect: An Efficient WSN Implementation for Intrusion Detection," won the best research demo award in *WISARD 2010*, a workshop in *The Second International Conference on Communication Systems and Networks (COMSNETS 2010)*, Bangalore, India.

Abstract

This thesis investigates the problem of detecting an intruder in the presence of clutter in a Passive Infra-Red (PIR) based Wireless Sensor Network (WSN). As one of the major objectives in a WSN is to maximize battery life, data transmission and local computations must be kept to a minimum as they are expensive in terms of energy. But, as intrusion being a rare event and cannot be missed, local computations expend more energy than data transmission. Hence, the need for a low-complexity algorithm for intrusion detection is inevitable.

When comparing two algorithms with the same detection performance, the algorithm with lower complexity can run on the microcontroller with a lower clock rate, for the same delay constraint. Operating at lower clock rate helps us take advantage of very low power modes of operation available in present day microcontrollers.

A low-complexity algorithm for intrusion detection in the presence of clutter arising from wind-blown vegetation, using PIR sensors is presented. The algorithm is based on a combination of Haar Transform (HT) and Support Vector Machine (SVM) based training. The amplitude and frequency of the intruder signature is used to differentiate it from the clutter signal. The HT was preferred to Discrete Fourier Transform (DFT) in computing the spectral signature because of its computational simplicity - just additions and subtractions suffice (scaling coefficients taken care appropriately). Intruder data collected in a laboratory and clutter data collected from various types of vegetation were fed into SVM for training. The optimal decision rule returned by SVM was then used to

separate intruder from clutter. Simulation results along with some representative samples in which intrusions were detected and the clutter being rejected by the algorithm is presented. The algorithm was found to exhibit good performance.

The implementation of the proposed intruder-detection algorithm in a network setting comprising of 20 sensing nodes is discussed. The field testing performance of the algorithm is then discussed. The limitations of the algorithm is also discussed.

A closed-form analytical expression for the signature generated by a human moving along a straight line in the vicinity of the PIR sensor at constant velocity is provided. It is shown to be a good approximation by showing a close match with the real intruder waveforms. It is then shown how this expression can be exploited to track the intruder from the signatures of three well-positioned sensing nodes.

Contents

Acknowledgements **ii**

Publications based on this Thesis **iii**

Abstract **iv**

1 Introduction **1**
1.1 Overview of Wireless Sensor Networks 1
1.2 Contributions of the thesis 3

2 The Intruder Detection Algorithm **5**
2.1 Literature Review 5
2.2 Problem Description 6
2.3 Goals 6
2.4 Challenges 7
2.5 Choice of Sensing Modality 7
2.6 PIR Sensor Operation 8
2.7 Sensing Model for the Analog Panasonic Motion Sensor AMN24111 . . . 11
2.8 Choice of the Sensor 14
2.9 The Haar Transform and Frequency Binning 16
2.10 Support Vector Machine 18
2.11 Intrusion Detection Algorithm Description 20
2.12 Computational Complexity 22
2.13 Intruder and Clutter Data Collection 23
2.14 Training and Testing Performance 24
2.15 Field Testing 26
2.16 Wireless Trip-Wire 29
2.17 Limitations of the Algorithm 30

3 Analytical Model For Intruder Signature **33**
3.1 Idealized Intruder Waveform Analysis 33
3.2 Comparison of Actual and Analytical Intruder Signatures 36
3.3 Observations made from the Model 37

4 Optimal Positioning of the Sensing Nodes for Tracking the Intruder 38
4.1 Problem Description 38
4.2 Tracking 38
4.3 Optimal Positioning of the 3 Sensing Nodes 40

5 Conclusion and Future Work 42

A Source Code for the Intrusion Detection Algorithm 44

B Field Data 48

References 52

List of Figures

2.1 Pyroelectricity [7]. 9
2.2 Time dependence of pyroelectric current response [18]. 9
2.3 Basic sensing model of a dual-pixel PIR sensor [36]. 10
2.4 Analog Panasonic Motion Sensor $AMN24111$ 11
2.5 Arrangement of the quad pixels in $AMN24111$. 12
2.6 Polarity markings of the quad pixels in $AMN24111$. 12
2.7 Proper orientation of the quad pixel sensor. 12
2.8 Golf ball lens [10]. 13
2.9 Top view of the beams [6]. 13
2.10 Virtual Pixel Array [6]. 14
2.11 Virtual pixel array rotated by 45° 14
2.12 Intruder signature for $3m$ slow walk. 15
2.13 Intruder signature for $(50°, 1.5m)$ slow walk. 15
2.14 Block diagram of analog PIR sensor's circuit [6]. 15
2.15 Block diagram of digital PIR sensor's circuit [6]. 15
2.16 Intruder signature in the presence of clutter from analog and digital PIR sensor. 16
2.17 256 point DFT of intruder and clutter at the output of the digital PIR sensor averaged over 112 data. 17
2.18 256 point DFT of intruder and clutter at the output of the analog PIR sensor averaged over 112 data. 17
2.19 Frequency bins corresponding to the 8-point Haar matrix (without the normalization constants). 17
2.20 8-sample fast Haar transform. 18
2.21 Illustration of a $2D$ linear SVM. 19
2.22 Illustration of over-fitted $2D$ data with a high degree SVM. 20
2.23 Functional block diagram of the algorithm. 21
2.24 Computational complexity of the algorithm. 23
2.25 The indoor location used for accumulating intruder data. 24
2.26 A location in IISc with a variety of vegetation where a part of clutter data was accumulated. 24
2.27 Experimental floor layout. 25
2.28 A location in ECE Dept. lawn where a part of clutter data was accumulated. 25
2.29 Linear SVM: Intrusion detected. 25

2.30 Linear SVM: Clutter rejected. 25
2.31 Three sensor platform. 26
2.32 Wireless trip-wire. 27
2.33 The graphical user interface (GUI) at the base station. 28
2.34 Line geometry of the wireless trip-wire. 29
2.35 Quadratic SVM on summer clutter data. 31
2.36 Overall training performance. 31

3.1 Geometry used for modeling intruder signature. 34
3.2 Illustrating t_0 and d_{min}. 34
3.3 Intruder signature for $(v, d, \phi) = (0.7, 3, 90°)$. 37
3.4 Intruder signature for $(v, d, \phi) = (0.3, 2, 50°)$. 37

4.1 Equilateral triangle node arrangement. 41

B.1 Sensor facing vegetation and the corresponding data at an ambient temperature of $42°C$. This experiment was performed at IIT Madras. 48
B.2 Sensor facing an open ground and the corresponding data at an ambient temperature of $42°C$. This experiment was performed at IIT Madras. . . 48
B.3 Sensor data recorded with the sensor facing an open ground with moving clouds at an ambient temperature of $34°C$. This experiment was performed at the air-strip in IISc Bangalore. 49
B.4 Sensor data recorded with the sensor facing an open ground at an ambient temperature of $34°C$. This experiment was performed at the air-strip in IISc Bangalore. 49
B.5 Sensor placed amidst dense grass. 50
B.6 Sensor placed at a distance of 1 foot from the grass. 50
B.7 Data corresponding to Fig. B.5. 50
B.8 Data corresponding to Fig. B.6. 50
B.9 Sensor data recorded when a dog moved in the vicinity of the sensor. . . 51

Chapter 1

Introduction

1.1 Overview of Wireless Sensor Networks

Technological advancements in both hardware and software have given rise to densely distributed networked systems that can sense and control the environment, perform local computations and communicate the results. Such sensor actuator networks enable us to interact with the physical world to a degree previously imagined only in science fiction [4].

Sensing nodes are small, low-power devices with sensing, computational and communication capabilities. In addition to one or more sensors, each sensing node is equipped with a radio transceiver, a microcontroller, external memory and an energy source, usually a battery. The microcontroller is usually thought of as having limited computational power. This is not the case with the present generation nodes.

There are specific programming languages such as nesC to program the sensor nodes. A sensing node may be equipped with one or more of the sensing modalities such as vibration, humidity, visible-light, infrared, acoustic, magnetic, temperature, pressure sensors etc.

A sensor network is a dense wireless network of resource-constrained sensing nodes that

cooperatively monitor physical or environmental conditions and efficiently convey meaningful information to the base-station which will take suitable action. The nodes in a Wireless Sensor Network (WSN) can be deployed randomly in various environments, including remote and hostile terrains. Depending on the application, the nodes can either be hand placed or be dropped from an aeroplane. The nodes self-configure and form a network after deployment. The nodes may be inaccessible once deployed making battery replacement impossible. Hence, energy consumption of the nodes becomes an important issue. The issues in WSN are quite different as compared to conventional networks. For instance, data-rate improvement will not be the focus in WSN whereas in conventional networks it is an important issue.

WSN finds numerous applications. In military applications, WSN could be used for identifying breach along a perimeter and monitor high risk regions where soldiers cannot be positioned. WSN finds new classes of environmental applications like predicting the onset of flash snow storms [5] and earthquakes. It can be used to monitor the pollution of a large geographical area. WSN can also be used for efficient lighting in a large building by monitoring occupancy pattern. In agriculture, WSN helps in automating the spraying of water and pesticides in precise quantities by monitoring the soil conditions.

The issues and challenges in WSN are different across applications. For example, in a military application where the sensing nodes are strewn over an area, the issues include network formation, node localization, time synchronization, event detection, target tracking etc. Information routing, querying, MAC protocols, node deployment, data security are other issues. The challenges include developing customized hardware and programming software for the sensing nodes.

In recent times there are attempts to integrate sensing, processing and wireless networking capabilities in a minute (ultimately one cubic millimeter) semiconductor grain

called a speck. This emerging technology, Speckled Computing [32], envisions data sensing and information processing capabilities to disappear into everyday objects and the living environment. This would pose bigger challenges as compared to the conventional WSN.

There have been very few large-scale practical experiments. Most notable of them is the ExScal project [3] conducted at the Ohio State University led by Prof. Anish Arora. Another large-scale demonstration of real-time surveillance system was performed at University of California, Berkeley [1]. These expose the practical problems that need to be considered while developing algorithms, firmware and hardware for WSN.

The motivation for this work is border surveillance. When we think of an unattended surveillance application, surveillance cameras like the ones we find in malls are a natural choice. But, such cameras do not operate in the night. In that case, infrared cameras that are capable of operating at night is the next option. But, surveillance cameras are expensive and power hungry. They are power hungry as they are required to do a lot of image processing and have to power up a lot of electronics. Being expensive is not a serious limitation but being power hungry is. Power outlets are not going to be available in the terrains of interest. Terrains of interest could be the mountain slopes, ravines, dense forests etc. Such terrains are obviously prone to infiltration and it is impossible to position a standing army. So we require a system that is power efficient, reliable and is not tied down by wires. WSN is well suited for this application.

1.2 Contributions of the thesis

In this thesis, the challenging problem of intrusion detection using Passive Infra-Red (PIR) sensors in the presence of clutter arising from wind-blown vegetation is addressed [8]. Also, the key challenge in WSN of being energy-efficient [2] is reasonably met. A field-tested low-complexity Support Vector Machine (SVM) training based algorithm

that uses the Haar Transform (HT) to separate intruder from clutter is presented. This algorithm does achieve one of the major objectives in a WSN as it significantly reduces the energy expenditure, thereby maximizing the network lifetime.

A closed-form analytical expression for the signal generated by a human moving along a straight line at constant velocity in front of the PIR sensor in a clutter-free environment is provided. It is shown how this expression can be exploited to track the intruder from the signals of three well-positioned sensors. Many of our observations and analytical results presented have a natural extension to other PIR sensors as well.

Chapter 2

The Intruder Detection Algorithm

Intrusion detection using Passive Infra-Red (PIR) sensors in the presence of clutter arising from wind-blown vegetation is a challenging problem [8]. Manufacturers recommend careful placement of their PIR Detectors to prevent false alarms resulting from vegetation, air currents, etc [9]. In these systems however, false alarms are of a lesser concern as typically motion is used to trigger a light which is then switched off after a pre-determined interval. Also, these algorithms employed in these devices are designed for operation on the mains and not on batteries. A low-complexity Support Vector Machine-training based algorithm that uses the HT to separate intruder from clutter is presented.

2.1 Literature Review

Zhiqiang et al. [26] presents an adaptive threshold algorithm with constant false alarm rate to detect a moving target. The PIR signal is first high-pass filtered to remove the low frequency components resulting from slow environment changes. Then the signal energy in a window is computed. A detection is confirmed when the computed energy exceeds a threshold. The threshold is dynamically updated based on the noise level.

L. Gu et al. [27] presents an algorithm developed by the University of Virginia which consists of high-pass filtering, adaptive thresholding and windowing. In particular they use

an unsupervised adaptation technique to adjust the energy threshold for target detection.

Most other works either use simple thresholding on the PIR signal or thresholding on the energy computed in a window in declaring a detection. In case of a digital PIR sensor, the runs of ones together with the number of transitions in a window are used in declaring a detection.

2.2 Problem Description

Given a placement of sensors, the problem is to detect an intruder, who is assumed to be moving during the period of detection, in the presence of clutter with low false alarm rate. The intruder is a human traveling in the vicinity of the sensor. The term clutter is used to describe the waveform generated at the output of the sensor as a result of the alternate source of disturbances like animal or vegetative movement or natural phenomenon such as wind, rain etc. The problem is challenging because intrusion is a random, rare and ephemeral [21] event while clutter is always present. Moreover, frequent false alarms would effectively render the system useless.

The nodes usually run on two AA-size batteries and the network is expected to last at least one year. Battery replacement is impossible as the nodes are not going to be attended after deployment. Moreover, the nodes could be inaccessible because of their risky sensing environments. Hence, energy consumption in the nodes is another equally important issue.

2.3 Goals

The goal is to come up with a robust intruder detection algorithm - high detection rate and low false alarm rate. The algorithm must have very good intrusion detection and clutter rejection capability. We try to minimize the energy spent in detection by

minimizing the number of local computations.

2.4 Challenges

There are a number of challenges to be overcome in order to achieve the goals. The more important ones are as follows:

- Handle various speeds of the intruder.
  We roughly classify the intruder's speed into three classes: slow walk, fast walk and running. The sensor that we use has a reliable sensing range of $6m$ and a horizontal angular coverage of $110°$. As a result, the duration of the intruder's signature would vary widely from $3s$ to $18s$. A crawl results in a large amplitude, low frequency and a long duration signature while a sprint results in a small amplitude, high frequency and short duration signature.

- Reject clutter from various forms of vegetation.
  Performance of the algorithm should not be terrain dependent. Types of vegetation in terrains of interest is manifold and hence the algorithm must be able to reject clutter from such varied sources.

- Low-complexity algorithm.
  The number of computations performed in declaring a detection reflects in the energy spent for the same. Hence, in general, the number of computations must be minimized. Also, a low latency in detection necessitates a low-complexity algorithm.

2.5 Choice of Sensing Modality

Every sensing modality comes with its own weaknesses and limitations together with its ease of signal processing. A brief comparison of PIR, acoustic and magnetic sensors is given in [3] and [21]. Among the various types of sensors available, we felt that

the PIR sensor is best suited for this application. The passive nature of these sensors allows it to operate at very low power. PIR sensors currently available consume less power than cameras by up to two orders of magnitude. PIR sensors possess highly deterministic operation. PIR sensors are also cheap and easily available. PIR sensors are very popular in smart lighting applications. In fact, PIR sensors can be used as a low-power wake-up mechanism for cameras. PIR sensors are triggered by blowing debris, birds, warm-blooded animals, vegetation, hot air currents, cloud cover movement etc. Clutter resulting from the wind-blown vegetation is the serious issue here as inordinate amount of time is spent in rejecting the same. Hence, our focus is to detect the intruder rejecting clutter arising due to wind-blown vegetation.

2.6 PIR Sensor Operation

A basic understanding of the working of the PIR sensor is inevitable to come up with a model for the intruder. This understanding helps in abstracting out certain dependencies between the intruder motion and the sensor parameters.

We know from Planck's law that every blackbody at a temperature T radiates at all wavelengths and the peak emission occurs at a wavelength governed by Wien's displacement law (i.e.)

$$\lambda_{max} T = 2.89 \times 10^{-3}\ mK \tag{2.1}$$

Humans with a normal body temperature of $37°C$ emit peak radiation at $9.4\mu m$ (far Infra-Red). Hence, PIR sensors along with the optical filters are tuned to detect wavelengths in the range of $8 - 14\ \mu m$. A PIR sensor converts the spatial and temporal variations of intensity of IR falling onto its sensitive element(s), into an electrical signal. Wind-blown vegetation also causes variations in the ambient IR intensity perceived by the sensor, which leads to clutter. This is primarily due to varying occlusions of background IR emissions caused by wind-blown vegetation.

A PIR sensor works on the principle of pyroelectricity. In certain dielectric crystals, such as a tourmaline crystal, a homogenous change in temperature produces a change in the net electric dipole moment. This physical process is called pyroelectricity and such crystals are called pyroelectric crystals. When such a thin, parallel-sided crystal, cut appropriately with respect to its crystallographic symmetry axis, is sandwiched between electrodes, heat sinked and connected to an ammeter as shown in Fig. 2.1 and subjected to a change in temperature, say for instance, a human appears in front of the crystal, a current flow is observed. This setup can be viewed as a single pixel PIR sensor.

A current pulse is observed from the single pixel sensor in response to temperature

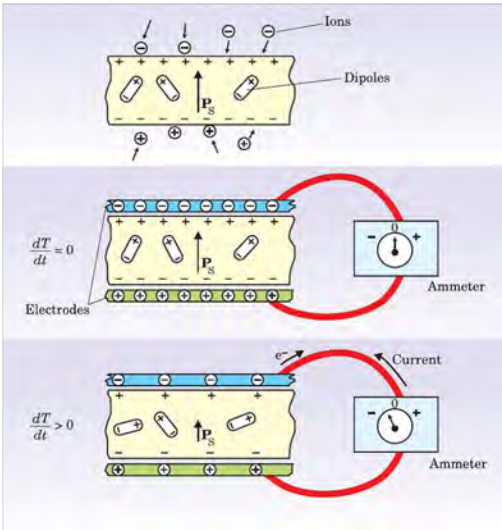


Figure 2.1: Pyroelectricity [7].

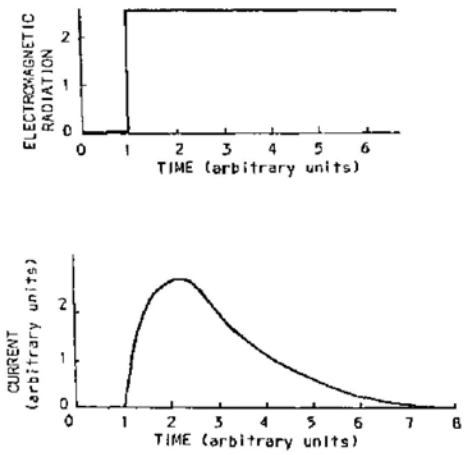


Figure 2.2: Time dependence of pyroelectric current response [18].

variation in the form of a step input. This current response is shown in Fig. 2.2. When the temperature of the crystal is steady, like when a human continues to stand in front of the sensor, there is no current flow. Thus, any stationary source of IR does not trigger the sensor. But, this sensor is triggered for even a slight change in ambient temperature. This can be easily overcome by using two such pixels wired in differential mode (which

makes the pixels have opposing polarity), referred to as a dual pixel sensor (See Fig. 2.3). Any change in temperature that is common to both the pixels are canceled out. As a result, the sensor is triggered only by moving objects in its vicinity. Thus, the dual pixel sensor behaves as a true motion detector. This sensor in conjunction with a focussing lens system creates sensitive regions in the field of view of the sensor. In case of a plano-convex lens, the sensitive regions can equivalently be viewed as two virtual beams projecting outward from the sensor. When a human moves into the two sensitive regions one after the other, which can equivalently be viewed as cutting the beams, an alternating signal resembling a sine-like wave is observed (See Fig. 2.3).

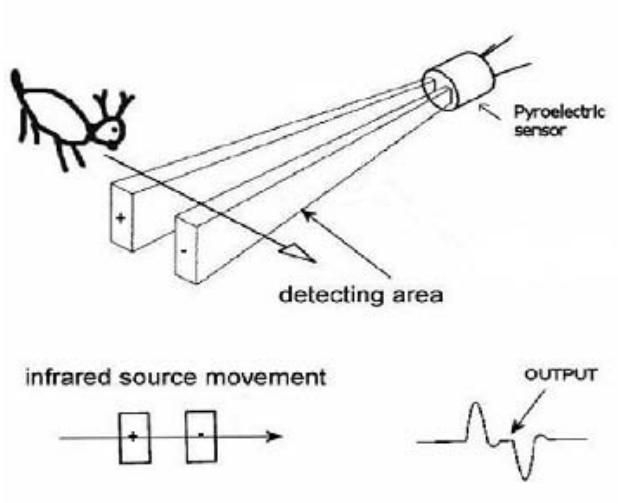


Figure 2.3: Basic sensing model of a dual-pixel PIR sensor [36].

The orientation of the dual pixel sensor must be such that the plane containing the two beams must be along the horizontal plane in which the human walks. If, for instance, the sensor is rotated by 90°, the human will cut both the beams more or less simultaneously. This implies that the sensor must be properly oriented.

It is interesting to note that direction of the motion is readily identified by looking into the signature. We can say whether the human moved from left to right or from

right to left with respect to the sensor by just looking at the polarity of the first half-cycle (Note that the human will cut both the beams more or less simultaneously, if the human walks very close to the sensor, resulting in a poor signature). But, when the human walks towards the sensor in between the beams, (which we refer to as the radial case), the intruder is unlikely to cut any of the beams, resulting in no signature ideally.

2.7 Sensing Model for the Analog Panasonic Motion Sensor AMN24111



Figure 2.4: Analog Panasonic Motion Sensor $AMN24111$

The sensor under study is the Panasonic PIR sensor AMN24111 (See Fig. 2.4). The sensor has a quad array of active elements (See Fig. 2.5) and a multilens made of multiple contiguous plano-convex lens. In the quad, the pixels along the diagonal have the same polarity (See Fig. 2.6). The sensor produces an electrical signal proportional to the differences in the rate of intensity variations across the two diagonals. This quad pixel sensor is intended for ceiling mount application. In our application, a dual pixel sensor would have sufficed since vertical motion is not anticipated.

The characteristics of the PIR sensor are largely dictated by the characteristics of the multilens [11] and [12]. The multilens modulates the thermal signature incident on it and



Figure 2.5: Arrangement of the quad pixels in $AMN24111$.

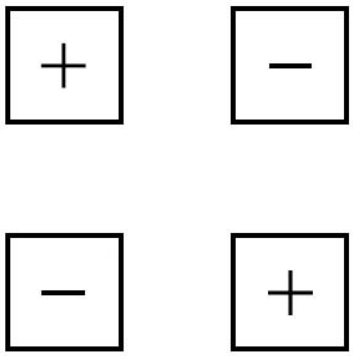


Figure 2.6: Polarity markings of the quad pixels in $AMN24111$.

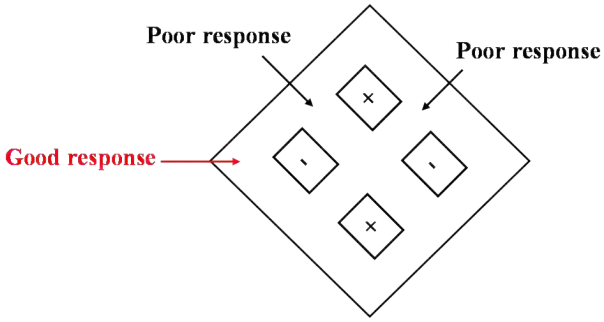


Figure 2.7: Proper orientation of the quad pixel sensor.

this signature is fed into the sensing elements. Radiation received by each plano-convex lens from a zone in the field of view is focused in the sensing region for sensing by the infrared detector. The multilens can be imagined to generate an array of virtual quad beams, one for each plano-convex lens, in the sensor's field of vision (See Fig. 2.8). The top-view of the beams is shown in Fig. 2.9. The cross-section of the beams at a distance of 5m is shown in Fig. 2.10. We refer to this array of virtual pixels created by multilens as the Virtual Pixel Array (VPA). In practice, this sensor must be rotated 45° about the sensor axis from the orientation as shown in the Fig. 2.6 (See Fig. 2.7). As a result, the VPA as shown in Fig. 2.10 is rotated by 45° and appears as in Fig. 2.11. The Fig. 2.11

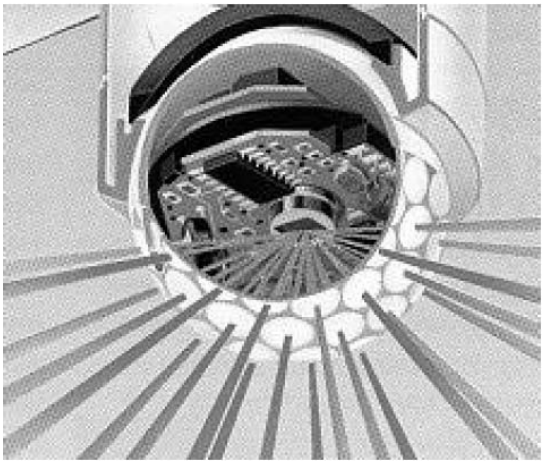


Figure 2.8: Golf ball lens [10].

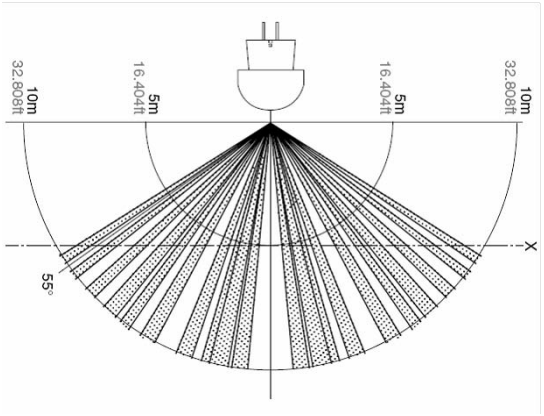


Figure 2.9: Top view of the beams [6].

explains how the intruder signature is generated when a human walks in the vicinity of the sensor along a straight line perpendicular to the axis of the sensor.

Since the vertical displacement is negligible compared to the horizontal displacement when a human walks, the intruder signature is largely characterized by the beams cut along the horizontal plane. When a human moves at a uniform speed along a circular path around the sensor, the beams are cut at a uniform rate. Thus, the sensor's output resembles a sine-like wave in which the number of half-cycles is same as the number of beams in the horizontal plane (roughly 15 in case of this sensor) and the frequency observed is directly proportional to the angular velocity at which the human walks. Fig. 2.12 shows the signature when a human was made to walk slow along a circular path at a radius of $3m$ around the sensor. Fig. 2.13 shows the signature when the intruder was made to walk slow starting close to the sensor and walking away from it along a straight line at a distance of $1.5m$ and making an angle of $50°$ with the sensor axis (See Fig. 2.27)). The instantaneous frequency of the signature can be seen to progressively decrease with time. This figure further strengthens the fact that the instantaneous frequency of the intruder signature is proportional to the instantaneous angular velocity of the intruder motion.

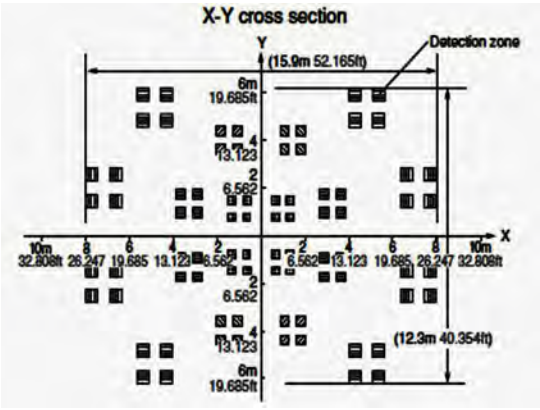


Figure 2.10: Virtual Pixel Array [6].

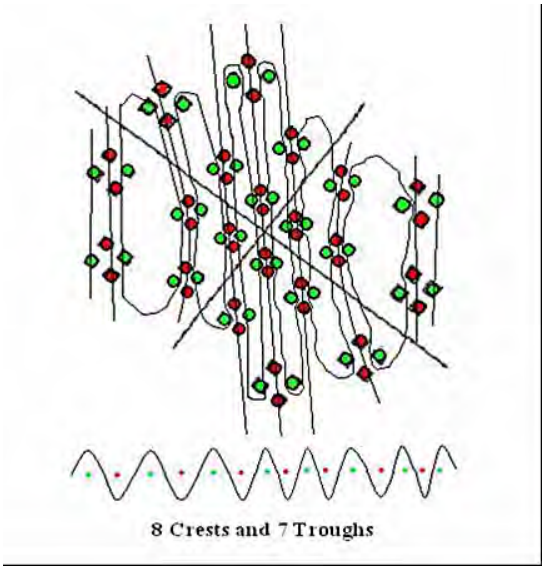


Figure 2.11: Virtual pixel array rotated by 45°

2.8 Choice of the Sensor

We were initially faced with a choice between the Digital Panasonic Motion Sensor $AMN14121$ and the Analog Panasonic Motion Sensor $AMN24111$ (See Fig. 2.14 and Fig. 2.15). The analog PIR sensor consists of a multilens, an optic filter, the active elements, followed by an in-built amplifier. The digital sensor has the same components as the analog sensor but with a comparator circuit at the output.

The digital PIR sensor offers no control over the threshold values which are fixed and are chosen rather close to the DC level. This is not surprising as these sensors are meant for ceiling mount applications perhaps to detect even slight motion in the indoors (clutter-free environments) of a large building for automated lighting. But, in the outdoors any small variation in IR intensity due to movement of vegetation results in long runs of zeros and ones. Hence, it is not good to base the detection on the number of ones and/or the transitions. Also, by thresholding we lose a lot of information. For instance, when both the above mentioned sensors are juxtaposed to monitor a common region of space, it was

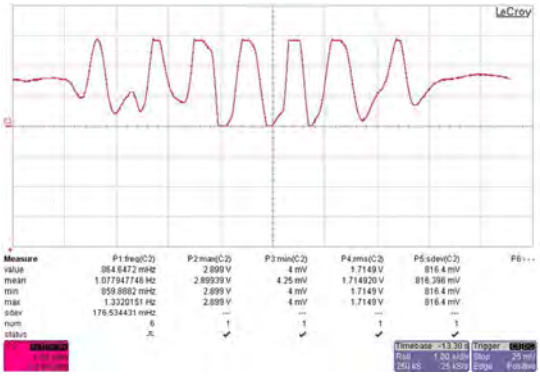


Figure 2.12: Intruder signature for $3m$ slow walk.

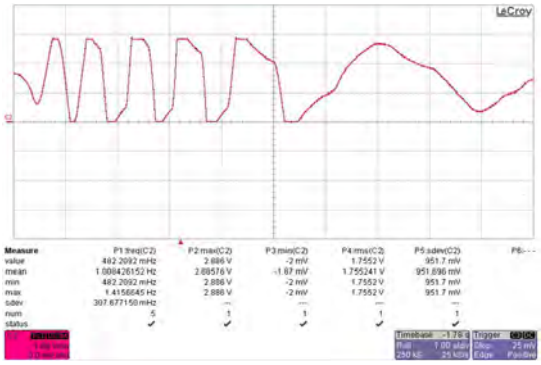


Figure 2.13: Intruder signature for $(50°, 1.5m)$ slow walk.

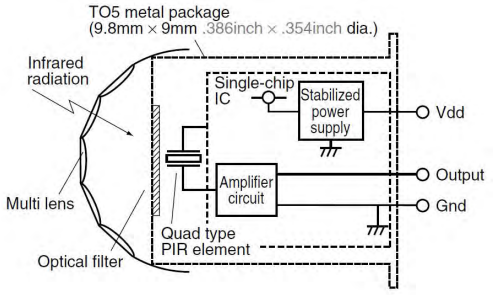


Figure 2.14: Block diagram of analog PIR sensor's circuit [6].

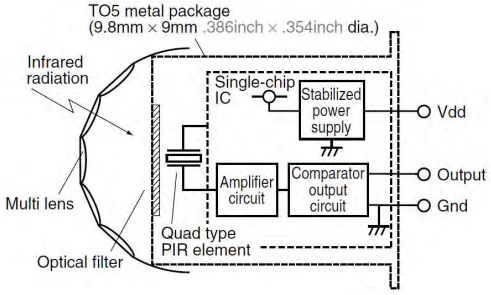


Figure 2.15: Block diagram of digital PIR sensor's circuit [6].

easy to recognize visually from the analog sensor's output that there are intrusions in the time interval between $15s - 20s$ and $105s - 115s$ and nowhere else (which was actually the case. The two intrusions were slow walk case. See Fig. 2.16). While we can claim the same for the digital PIR sensor, it is easy to see that there are no intrusions in the time interval between $40s - 70s$ which would incorrectly be identified as intrusions. Just by looking into the analog PIR sensor's output, it is easy to discriminate whether the intruder is moving slowly or rapidly and also distinguish gusty and non-gusty conditions.

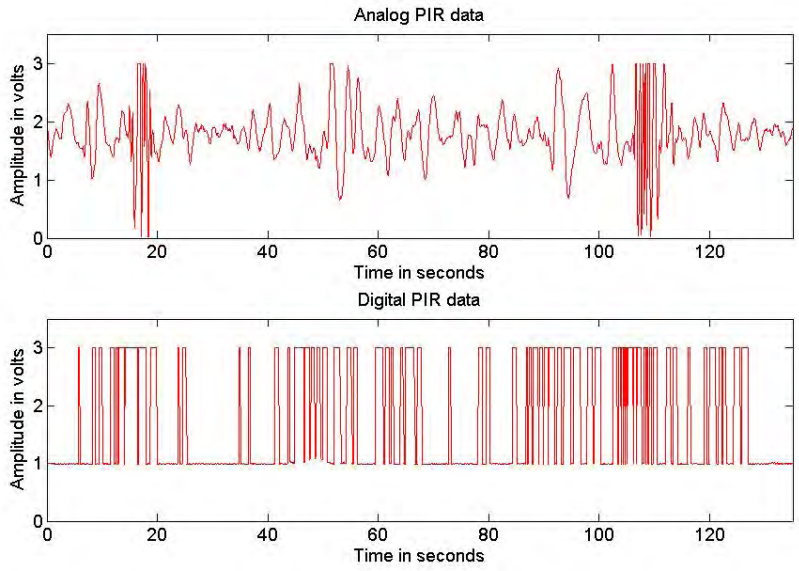


Figure 2.16: Intruder signature in the presence of clutter from analog and digital PIR sensor.

This fact is further corroborated when we look into the Discrete Fourier Transform (DFT) spectrum of intruder and clutter signatures from the output of the analog and digital sensor (See Fig. 2.17 & Fig. 2.18). In case of the digital PIR sensor, there is practically no frequency separation between intruder and clutter signatures. This is because of the fixed thresholds not under our control. We preferred the analog PIR sensor to the digital PIR sensor as it was easier to distinguish between the spectrum of intruder and clutter from the output of the analog sensor (Note that intruder has significantly higher frequency content). We believe that some amount of training will be largely helpful since the clutter can be caused by variety of vegetation.

2.9 The Haar Transform and Frequency Binning

In DFT, each coefficient is associated with a different frequency. But in case of HT, which is wavelet based, the coefficients are designed to provide both frequency and time localization information [35]. As a result, the breakdown of N Haar coefficients is as

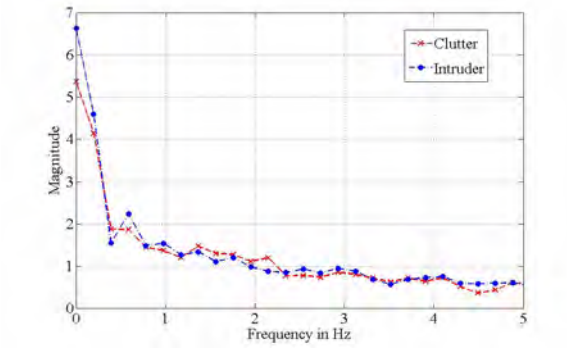


Figure 2.17: 256 point DFT of intruder and clutter at the output of the digital PIR sensor averaged over 112 data.

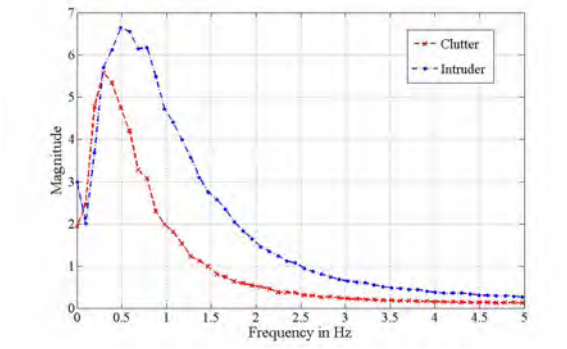


Figure 2.18: 256 point DFT of intruder and clutter at the output of the analog PIR sensor averaged over 112 data.

follows: there is one coefficient assigned to frequency 0 (the DC component) and 2^k coefficients attached to signals of frequency 2^k, $0 \leq k \leq [\log(N) - 1]$. Thus, there are a total of $[\log(N) + 1]$ frequencies or frequency 'bins'. To compute the spectral signature, the energy in the same frequency at different time intervals is combined into one. This process that we refer to as frequency binning results in a $[\log(N) + 1]$ dimensional frequency binned vector from an N dimensional Haar transformed vector. The Haar signals associated with an example 8-sample transform are shown in Fig. 2.19. The Fig. 2.20

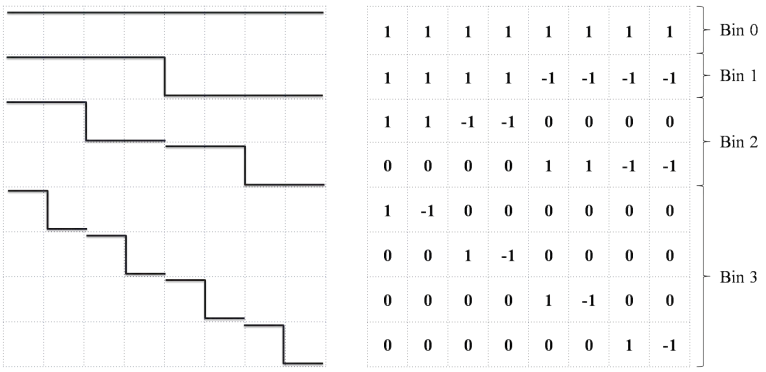


Figure 2.19: Frequency bins corresponding to the 8-point Haar matrix (without the normalization constants).

depicts the butterfly diagram of an 8-point fast Haar transform. Another advantage in

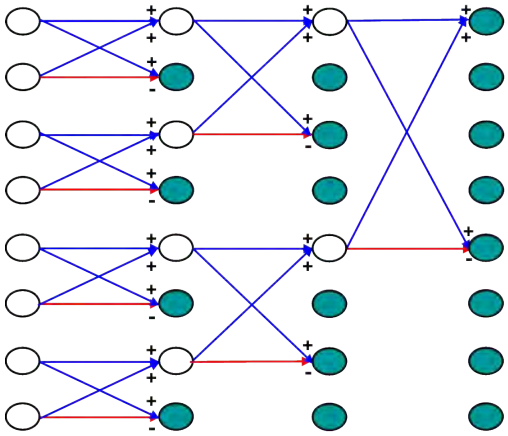


Figure 2.20: 8-sample fast Haar transform.

HT is that most of the components of the previously transformed vector can be reused for the current computation. The time localization information allows reuse of most of the components of the last transformed vector for the current window if there is overlap of the current window with the previous window. If the current computation is done on a window having L new samples together with the past $(N-L)$ samples from the previous window, on which HT has already been computed, then the components corresponding to the rows in the HT matrix that acted on only the last $(N-L)$ samples can be reused for the computation of HT on current window. It can be proven that the number of components that can be reused for a slide of L in a window of size N is $(N-L)\left(1-\frac{1}{L}\right)$. The number of reusable components is maximum when $L=\sqrt{N}$.

2.10 Support Vector Machine

Support Vector Machine (SVM) is a machine-learning technique used for classification which when input with two labeled sets of data (here the binned vectors) returns a decision surface which tends to maximize the margin between the two data sets. For a formal treatment on SVM refer [24]. We consider linear SVM classifiers initially and quadratic SVM classifiers subsequently. Under linear SVM, the decision surface is chosen by SVM

to be the hyperplane (for eg., See Fig. 2.21) in the input space that maximizes the margin, i.e., that maximizes the distance between the hyperplane and the input data set. In our case, the input space is 8 dimensional. The optimal hyperplane is typically found in practice by reformulating it as a quadratic programming (QP) optimization problem which can be efficiently solved. If the input data in linear SVM are not separable by a hyperplane, we allow training errors to occur. The tradeoff between the margin and the training errors can be controlled by a parameter C in SVM. The larger the value of C, the lesser the number of training errors leading to a smaller margin. An alternative means of handling training errors is to use a nonlinear separating surface. In quadratic

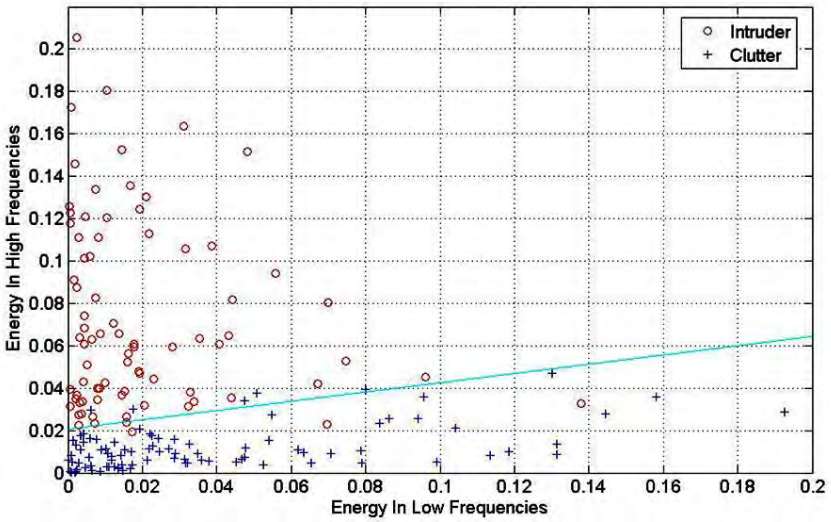


Figure 2.21: Illustration of a $2D$ linear SVM.

SVM, the optimal decision surface is chosen to be the hyperplane in a larger dimension space which maximizes the margin from points in the larger dimension space to which points in the data set are mapped. Each coordinate in the larger dimension space is associated to a unique monomial of degree 1 or 2 in the variables attached to the input space. Thus, the larger space is of dimension $\binom{8}{2}$+8+8=44.

Computing on a high degree separating surface is computationally complex and as a consequence, latency involved in detection is high. Also, it is to be kept in mind that

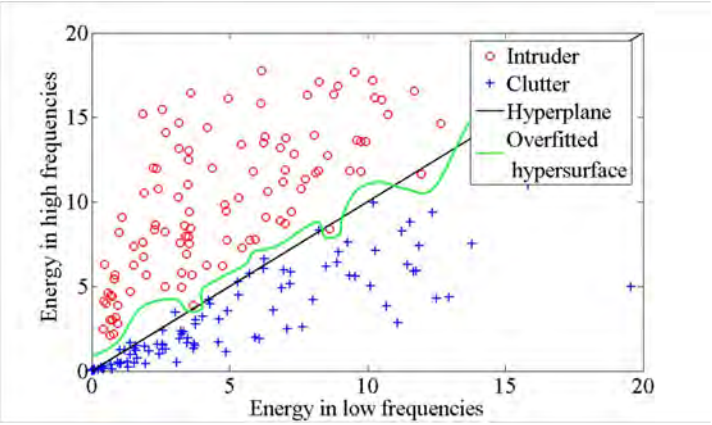


Figure 2.22: Illustration of over-fitted $2D$ data with a high degree SVM.

there is a risk of over-fitting the data with a high degree separating surface that will perfectly separate the training data and is likely to fail on any new data (See Fig. 2.22).

It is our belief that the training of the form, inherent in SVM, should form an essential part of any solution, given the large variations in intruder and clutter signatures possible. Also, SVM makes possible a more fine-grain frequency-domain approach to separating intruder from clutter.

2.11 Intrusion Detection Algorithm Description

The algorithm comprises of two phases: offline and online phase. During the offline phase, a variety of intruder and clutter data is collected. Further, the data is transformed by HT, then frequency binned and fed into an SVM for training. The SVM returns a hyperplane $(\mathbf{w}, b)$ which will be used to identify intrusions in the online phase. Not surprisingly, the weights computed by the SVM for the higher frequency bins are relatively high when compared to the lower frequency bins. The online phase corresponds to the real-time working of the algorithm on the sensing nodes.

Keeping in mind the major objective of maximizing battery life in a Wireless Sensor Network (WSN), it was decided to use HT for computing the spectrum of intruder and clutter signals in preference to the computationally more complex DFT as only additions and subtractions suffice to compute the HT (scaling coefficients taken care appropriately). An alternative would have been to use Walsh Hadamard Transform (WHT) but HT was preferred as it is less complex compared to WHT, it has ability to reuse past computed HT coefficients for the next window and can potentially be used to yield time-frequency localization information. Though HT gives only a coarser form of frequency spectrum as compared to DFT, there wasn't significant difference between the two in terms of the detection performance. This could possibly be because the coarse spectral resolution provided by HT is enough to separate intruder from clutter. Sampling frequency (f_s) was chosen to be $12.5Hz$ as the signature frequencies were less than $5Hz$ (both intruder and clutter, see Fig. 2.18). A functional block diagram of the algorithm appears in Fig. 2.23.

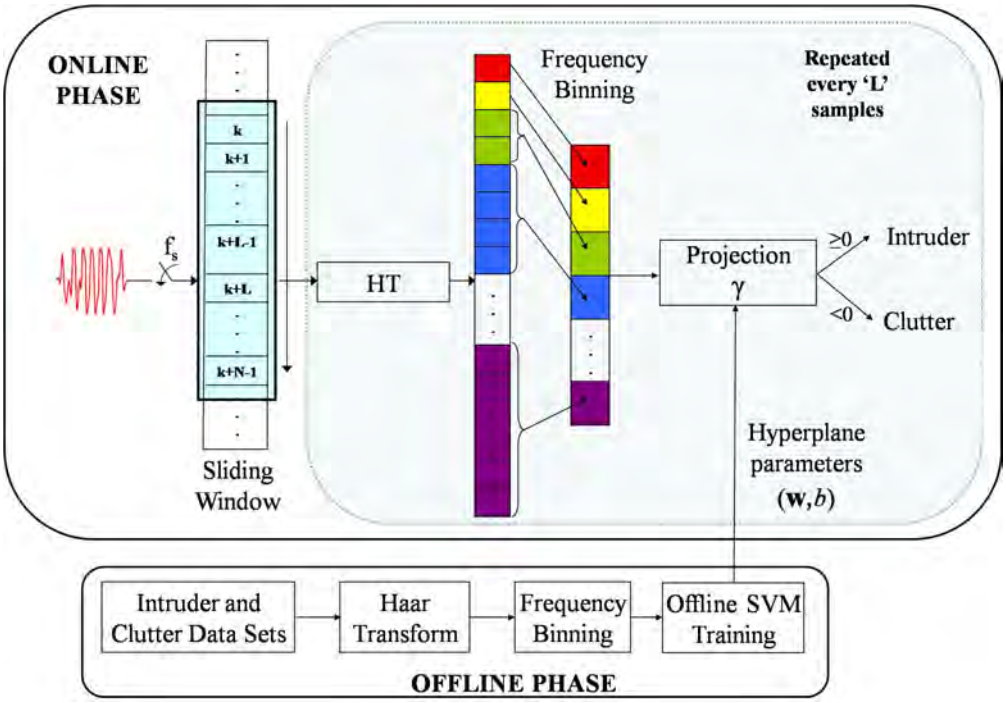


Figure 2.23: Functional block diagram of the algorithm.

At a sampling rate of $12.5Hz$, the intruder signature roughly varies from 40 samples to 225 samples (intruder signature varies widely from $3s$ to $18s$). This suggests that a window size 'N' of 256 should be appropriate. Further, to cut down on the number of computations involved, we moved to a window size of 128. We found no significant decrease in performance. During the online phase, a block of 128 (N) consecutive samples is transformed by HT. The energy in each of these transformed components are binned into 8 frequency bins ($[\log(128) + 1] = 8$). The resultant binned vector is projected onto the hyperplane $(\mathbf{w}, b)$ returned by the SVM classifier. Depending on the sign of the projected value, γ (γ=$\mathbf{w}^T\mathbf{x} + b$ where $\mathbf{w}$ is the normal to the hyperplane, b represents the affine shift of the hyperplane and $\mathbf{x}$ is the binned vector [γ is proportional to the distance of $\mathbf{x}$ to the hyperplane]), the algorithm declares it as intruder or clutter. This process will only be repeated on the arrival of the new $16(L)$ samples. L was chosen as 16 (16 samples roughly corresponds to $1s$) as this guarantees a significant change in subsequent γ values. If L is chosen to be high, it is likely that the algorithm might miss detections that correspond to sprinting as these cases register only few samples.

2.12 Computational Complexity

We consider only the computations carried out on incoming new data (online phase) in estimating the computational complexity, since training of the SVM was conducted offline. Computations involve Fast Haar Transform, Frequency binning and projection onto the hyperplane. There are no memory accesses for data or the hyperplane coefficients. This is a clear advantage as memory accesses consume quite a bit of power. The number of computations required for the online part of the algorithm for both linear and quadratic SVM appears in Fig. 2.24. It is to be noted that the computations are in the order of input size 'N'.

Computations	Additions/ Subtractions	Multiplications	Additions/ Subtractions	Multiplications
FHT	2(N-1)	-	2(N-1)	-
Binning	N-logN	N	N-logN	N
SVM	Linear SVM		Quadratic SVM	
	logN+1	logN+1	0.5(1+logN)(4+logN)	(1+logN)(3+logN)
Total Computations	3N-1	N+logN+1	3N+0.5(1+logN)(2+logN)-1	N+(1+logN)(3+logN)

Figure 2.24: Computational complexity of the algorithm.

2.13 Intruder and Clutter Data Collection

The data used for training SVM was collected in a laboratory (i.e., clutter-free) environment (See Fig. 2.25 and Fig. 2.27) in case of intruder data and across 20 outdoor locations (which resembled the wild that included grass shoots, shrubs, big ferns, trees etc.) on the forested campus of the Indian Institute of Science (IISc) in the case of clutter data (See Fig. 2.26 and Fig. 2.28).

The approach of *in vitro* training for intruder data and *in situ* training for clutter data has been chosen for its obvious advantages when it comes to deployment in a variety of environments. The training data set for intruder includes data collected from making a human walk along different straight lines oriented in a variety of ways with respect to the sensor. Clutter data was accumulated over the 6-month period from October 2008 to March 2009. Clutter data collection would simply involve setting down the sensing nodes on the ground in selected clutter-prone areas and letting them observe and record data for hours.



Figure 2.25: The indoor location used for accumulating intruder data.



Figure 2.26: A location in IISc with a variety of vegetation where a part of clutter data was accumulated.

2.14 Training and Testing Performance

The data streams collected were partitioned into two sets, one used for training, the other for testing. The data streams from some of the more challenging scenarios (for example, a sensor placed in the close proximity of large fern, or an intruder moving with high velocity) was used for training. From the data streams, a total of 224 blocks were extracted, 112 each representing intruder and clutter, to be used for training. Each block is a vector containing 128 consecutive time samples spaced 80 ms apart. The offline training of SVM was done using LIBSVM [25] interfaced to MATLAB. Linear kernel, with C set a high value, say 50000, was used. A high value of C would try to force the SVM to make the number of training errors negligible. The training performance recorded $7/112 = 6.3\%$ misses and $4/112 = 3.6\%$ false alarms. Testing performance was significantly better than training performance because the SVM was intentionally fed with data from both the classes that was difficult to separate. The testing performance recorded $3/500 = 0.5\%$ misses and $2000/160000 = 1.25\%$ false alarms. Some representative samples appear in Fig. 2.29 and Fig. 2.30. Fig. 2.29 shows four intrusions when a human was made to sprint in front of the sensor at a distance of $3m$ over a period of 70 seconds, all of which are successfully detected by the algorithm. The algorithm

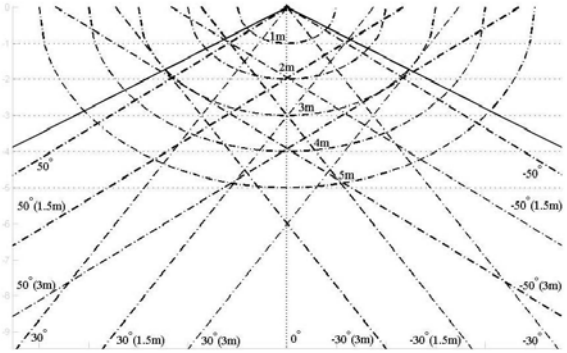


Figure 2.27: Experimental floor layout.



Figure 2.28: A location in ECE Dept. lawn where a part of clutter data was accumulated.

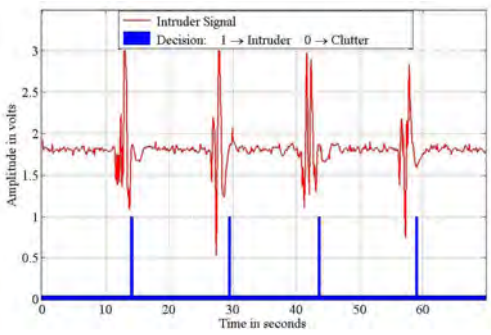


Figure 2.29: Linear SVM: Intrusion detected.

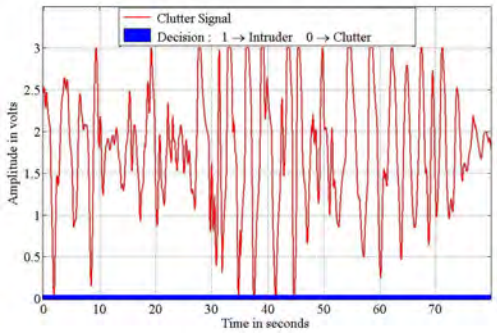


Figure 2.30: Linear SVM: Clutter rejected.

managed to detect the intrusions despite the small amplitude as it made use of the frequency information. Fig. 2.30 shows clutter data collected over a period of 80 seconds, when the sensor was placed in front of a $10ft$ tall fern at a distance of $3m$. It can be seen that the clutter is successfully rejected by the algorithm. The algorithm managed to reject the clutter despite the large amplitude as it made use of the frequency information.

We found that in most cases, the intruder is detected no later than $3s$ from the time the intruder enters the field of view of the sensor. Note that the intruder is likely to be in

the field of view of the sensor for $3s$ to $18s$.

A slow moving intruder is likely to result in a burst of positive γ values. This, which we call hysteresis, can be suppressed by ignoring the subsequent γ values until the gamma values become negative.

2.15 Field Testing

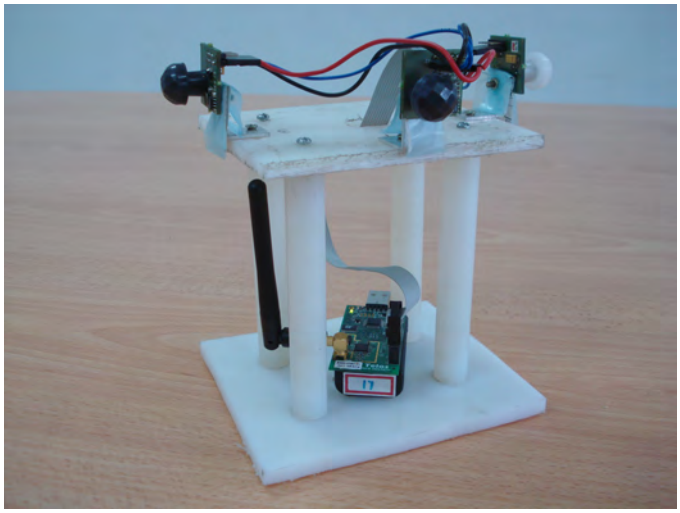


Figure 2.31: Three sensor platform.

The field testing was conducted in the Electrical Communication Engineering (ECE) Department lawn in Indian Institute of Science (IISc) campus. In the field tests, three sensors were mounted onto a single node ($6\ inch \times 6\ inch \times 6\ inch$ cubes) with an angular spacing of 120°. It is cost effective to have multiple sensors on the same sensing node. When combined with the approximately 110° angular field of view of each sensor, this essentially gave each node an omni-directional sensing range of $6m$ (See Fig. 2.31).

Data emanating from the 3 sensors were fed to the 3 ADC channels of the TelosB motes (See Fig. 2.31) [19] and [20]. Initially, the nodes were simply used to collect



Figure 2.32: Wireless trip-wire.

and retrieve data from the field for subsequent analysis in the lab. Subsequently, the intrusion-detection algorithm was itself loaded onto the TelosB mote.

As one of our themes was border surveillance, the initial decision was to deploy the sensing nodes in the form of a linear array, with inter-node spacing chosen to maximize the area covered by a single node, while ensuring that every point in the sensing range was covered by at least 3 nodes. The idea here was that the sensing nodes would serve as a wireless trip wire (See Fig. 2.32 and Section 2.16). However, it was found that this linear array would on occasion, fail to detect high-speed motion, possibly because at high speeds, the intruder was in the field of view for only a very short duration. With this in mind, the decision was made to create a double array comprising of two identical, linear and parallel arrays spaced apart by 5m (which is just 1m under the maximum sensing radius of 6m).

Decisions were then made locally as follows. If a node detected an intruder in its vicinity using the Haar transform-cum-SVM algorithm outlined earlier, it would broadcast its local detection (via the Zigbee protocol available on TelosB motes) to all of its neighbors.

A node was made permitted to declare a *confirmed* detection if in addition to making a local detection, it also received news of local detection from any node within a distance of twice the sensing range of each sensor. It is to be noted that the algorithm is scalable with growing number of nodes as the detection of an intruder results from the consensus of a few neighboring nodes. The confirmed detection was then relayed back to the base station using an appropriately designed network routing algorithm. At the base station, a graphical user interface (GUI) would display the information regarding the nodes that detected and the route of the confirmed detection (See Fig. 2.33).

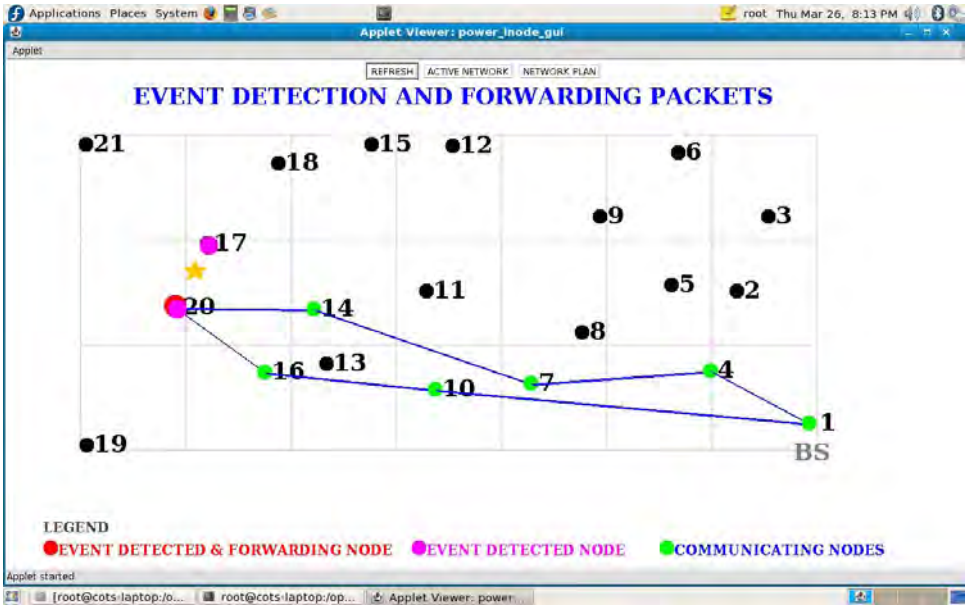


Figure 2.33: The graphical user interface (GUI) at the base station.

When tested over a period of several hours across several weeks, the network performed flawlessly by detecting every intrusion at speeds ranging from that of a crawl to a sprint at 5m/sec. There were also no false alarms in the period over which testing was conducted.

2.16 Wireless Trip-Wire

To cover an area of a few square kilometers, tens of thousands of sensing nodes would be required. This may not be practical in terms of cost. In that case a linear arrangement of the sensing nodes serving as a wireless trip-wire would be preferred (See Fig. 2.34) as it requires far less number of sensing nodes. A large area can be covered by interlacing many such wireless trip wires. We assume that the intruder walking radially towards the

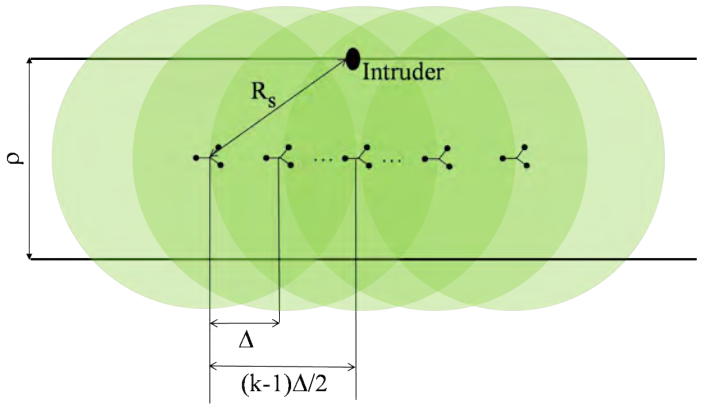


Figure 2.34: Line geometry of the wireless trip-wire.

sensor will go undetected. Let the spacing between the nodes be Δ. Let the trip wire provide us k-coverage for a width of $\frac{\rho}{2}$ on either sides. Then,

$$\left(\frac{(k-1)\Delta}{2}\right)^2 + \left(\frac{\rho}{2}\right)^2 = R_s^2 \tag{2.2}$$

For this arrangement, area covered per node, $(a-p-n)$,

$$\frac{(k-1)\Delta \times \rho}{k} \tag{2.3}$$

It can be proven that, $\Delta = \frac{\sqrt{2}R_s}{k-1}$ and $\rho = \sqrt{2}R_s$ maximize the $(a-p-n)$

$$(a-p-n)_{max} = \frac{2R_s^2}{k} \tag{2.4}$$

Note that the theoretical maximum value of $(a-p-n)$ is $\frac{\pi R_s^2}{k}$. The difference is because in the practical case, there are many regions that are less than k-covered and not taken into consideration (See Fig. 2.34).

Excluding the case when the intruder walking on/along the line of the sensing nodes, the intruder can be radial to at most one sensing node. The nodes within the sensing range use majority logic to reach a consensus in declaring an intrusion.

2.17 Limitations of the Algorithm

The performance results reported above was for the period October 2008 to March 2009 (Winter data). However, when we carried out testing around noontime in April 2009, at the height of the summer in Bangalore, we noticed a significantly higher false alarm rate. We observed that in summer, the energy in all the frequencies of the clutter signal had gone up resembling the spectral signature of that of the intruder. We suspect that this could at least in part, be due to the fact that the ambient temperature was in this period, close to the temperature of the human body, as can be seen from the fact that the recorded maximum temperature in shade was 34° Celsius. Thus, heat radiated by surrounding objects would not be filtered out by the optical filter placed in front of the quad sensor element array that is part of the Panasonic Analog Motion Sensor. Fig. 2.35 shows a sample waveform recorded in this period. When such summer noontime data was also included in the training set (Combined Winter-Summer data), linear SVM recorded a training performance of $60/275 = 21.8\%$ misses and $22/275 = 8\%$ false alarms. Testing performance recorded $30/300 = 10\%$ misses and $6100/100000 = 6.1\%$ false alarms. Quadratic SVM, replacing the linear SVM, was able to improve the record on training data to $47/275 = 17\%$ misses and $15/275 = 5.5\%$ false alarms. The improvement with regard to testing data was far more pronounced. This is because the SVM was intentionally fed with data from both the classes that was difficult to separate. Change detection on the stream of γ values supplied by the output of the SVM was attempted,

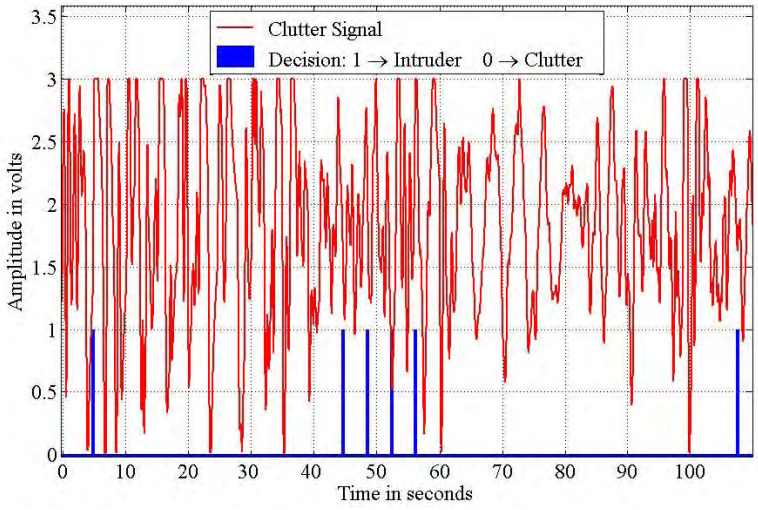


Figure 2.35: Quadratic SVM on summer clutter data.

but found to offer little improvement. The overall training performance on winter and combined winter-summer data is summarized in Fig. 2.36. We feel that the use of

Training Data Set	Linear SVM	Quadratic SVM
Winter Data	95%	-
Winter & Summer Data	85%	88%

Figure 2.36: Overall training performance.

multiple sensors that employ majority logic for consensus would permit reduction of the false alarm rate since the false alarms are often caused by the movement of vegetation in close proximity to a sensor. An alternative approach would be to detect the intruder based on a better understanding of the signature waveform of the intruder and clutter.

The next chapter provides some analysis in this direction. A closed-form analytical expression as a good approximation to the intruder waveform is provided and it is shown how this can be used to determine how to optimally place multiple sensors so as to estimate the path of a human being in motion.

Chapter 3

Analytical Model For Intruder Signature

3.1 Idealized Intruder Waveform Analysis

The intruder is assumed to move with a uniform velocity along a straight line path in the vicinity of the sensor. This is a reasonable assumption as the sensing range of the sensor is limited only to a few meters. The beams generated by the multilens are assumed to be uniform and equally spaced. We assume that we operate well within the inherent bandwidth of the sensor and there is no clipping of the waveform in the sensor's output. The amplitude is assumed to remain constant within region of interest.

When an intruder cuts across the beams, the infrared signal incident on the sensing element can be modeled as a triangular waveform whose period is inversely proportional to the angular velocity ω of the intruder. The sensing element acts upon this infrared signal as a band-pass filter, which we will assume filters out all but the fundamental frequency component, whose frequency is proportional to ω. Let κ denote the constant of proportionality.

In the case of this sensor, the human will cut 15 beams while executing a circular motion

with linear velocity 'v'm/s and radius 'r'm. As a result, the sinusoid will have 7.5 cycles that correspond to a duration of 'T's.

Frequency of the sinusoid,

$$f = \frac{7.5}{T}.$$

Angular velocity,

$$\omega = \frac{v}{r} = \frac{1.92}{T} \; [As\ 110^\circ = 1.92rads].$$

$$\Rightarrow f = \left(\frac{7.5}{1.92}\right)\omega \approx 4\omega.$$

The constant of proportionality, κ, for this sensor, was experimentally verified to be very close to 4.

Now, let v be the speed of the intruder walking along a straight line making an angle ϕ with the sensor's axis and at a distance of d from the sensor. Let 2θ (equal to 110° for the sensor under study) be the horizontal angular coverage of the sensor. The sensor responds to the intrusion for $t > 0$.

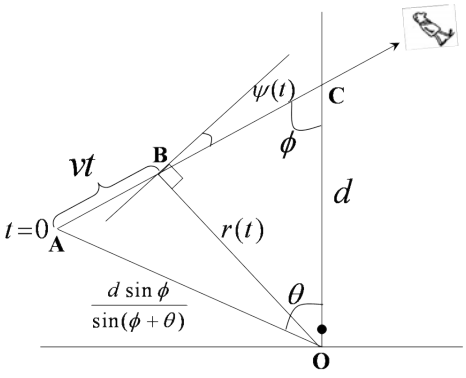


Figure 3.1: Geometry used for modeling intruder signature.

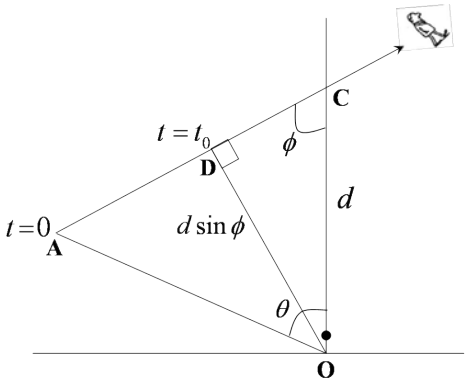


Figure 3.2: Illustrating t_0 and d_{min}.

From Fig. 3.1,

$$r^2(t) = v^2t^2 + d^2\frac{\sin^2\phi}{\sin^2(\phi+\theta)} + 2vtd\sin\phi\cot(\phi+\theta). \tag{3.1}$$

and

$$\frac{d}{\sin(90^\circ - \psi(t))} = \frac{r(t)}{\sin\phi} \Longleftrightarrow \cos\psi(t) = \frac{d\sin\phi}{r(t)}. \tag{3.2}$$

The instantaneous frequency $f(t)$ of the intruder signature is then from $\triangle OBC$ given by,

$$f(t) = \kappa\ \omega(t) = \kappa\frac{v\cos\psi(t)}{r(t)} \ = \ \kappa\frac{vd\sin\phi}{r^2(t)}. \tag{3.3}$$

Using $\triangle OAB$ we can substitute for $r(t)$ to get

$$f(t) = \frac{\kappa vd\sin\phi}{v^2t^2 + d^2\frac{\sin^2\phi}{\sin^2(\phi+\theta)} + 2vtd\sin\phi\cot(\phi+\theta)} \tag{3.4}$$

Let $\lambda = \frac{v}{d\sin\phi} = \frac{v}{d_{min}}$ and $\cot(\phi+\theta) = -\lambda t_0$. Thus $0 < \lambda < \infty$ when $v > 0$,$d > 0$ and $0 < \phi < \pi$. Also, d_{min} is the distance of the closest approach of the intruder to the sensor (see Fig. 3.2). We can now write

$$f(t) = \frac{\kappa\lambda}{1 + (\lambda(t - t_0))^2} \tag{3.5}$$

The maximum frequency, $f_{max} = \kappa\lambda$, occurs at $t_0 = -\frac{\cot(\phi+\theta)}{\lambda}$.

The intruder signature, $s(t) = \sin\left(2\pi\int_0^t f(t)dt\right)$ [Assuming the sine-like resemblance of the intruder signature.]

$$\begin{aligned} Let\ \chi(t) &= 2\pi\int_0^t f(t)dt \\ &= 2\pi\kappa\int_0^t \frac{\lambda}{1 + (\lambda(t - t_0))^2}dt \end{aligned}$$

Let $u = \lambda(t - t_0)$

$$\begin{aligned} &= 2\pi\kappa \int_{\lambda t_0}^{\lambda(t-t_0)} \frac{du}{1+u^2} \\ &= 2\pi\kappa \left[\arctan u\right]_{\lambda t_0}^{\lambda(t-t_0)} \\ &= 2\pi\kappa \left[\arctan(\lambda(t-t_0)) - \arctan(\lambda t_0)\right] \\ \therefore \chi(t) &= 2\pi\kappa \arctan\left[\frac{\lambda t}{1-\lambda^2 t_0(t-t_0)}\right] \end{aligned}$$

The intruder signature is thus given by,

$$s(t) = \sin\left(2\pi\kappa \arctan\left[\frac{\lambda t}{1-\lambda^2 t_0(t-t_0)}\right]\right), \tag{3.6}$$

$0 \leq t \leq \frac{1}{v}\left[min\left(d\sin\theta\left(\csc(\phi+\theta)+\csc(\phi-\theta)\right), d\sin\theta\csc(\phi+\theta) - d\cos\phi + \sqrt{R_s^2 - d^2\sin^2\phi}\right)\right]$.
The duration of the signature is limited either by the sensing range of the sensor or the angular coverage of the sensor. Note that $s(t)$ cannot be written as a rational function involving $\lambda,\ t_0,\ t$ as π is an irrational number.

3.2 Comparison of Actual and Analytical Intruder Signatures

Fig. 3.3 and Fig. 3.4 show a close match between analytical and actual signals for two distinct intruder signatures. There is not a perfect match between the two waveforms in certain regions since the intruder need not have moved with constant speed.

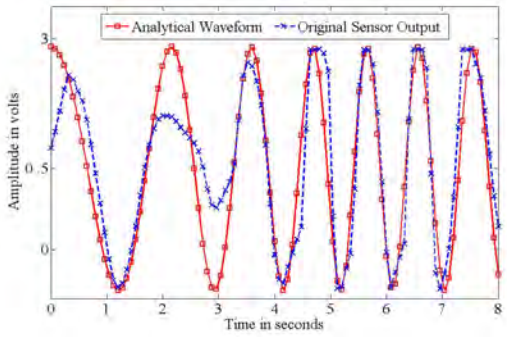


Figure 3.3: Intruder signature for $(v, d, \phi) = (0.7, 3, 90°)$.

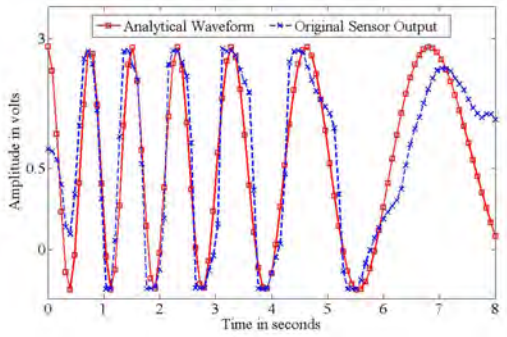


Figure 3.4: Intruder signature for $(v, d, \phi) = (0.3, 2, 50°)$.

3.3 Observations made from the Model

The following observations can be made from the model:

κ is the constant which corresponds to the density of the beams in the plane of the intruder motion. Hence the analytical expression naturally extends to other differential Passive Infra-Red (PIR) sensors in general as κ abstracts the lens.

λ and t_0 determine the intruder's analytical waveform. λ contains all the information required to track the intruder, but λ for different triplets of (v, d, ϕ) can be the same. This implies that the velocity and direction of motion information cannot be extracted from a single sensor. Hence, a single sensor cannot localize motion within its field of view.

The λ corresponding to colocated sensors will be identical as d_{min} is the same for these sensors. This implies that information about the intruder's velocity and direction of motion cannot be obtained from multiple sensors on the same node. So, to track the intruder, many sensing nodes spaced apart will be required.

Chapter 4

Optimal Positioning of the Sensing Nodes for Tracking the Intruder

4.1 Problem Description

The Passive Infra-Red (PIR) sensor under study gives a reliable sensing range of 6m. Hence, in order to cover large areas, multiple sensing nodes are required. In [16], PIR sensors are characterized for monitoring the occupancy pattern in an indoor environment. In [17] and [33], the angular characteristics of the PIR sensor and the lens are used to track an intruder in an indoor environment. The problem now at hand is to reliably track the intruder motion making use of the idealized intruder waveform model. We would like to know whether it is possible to track the intruder from the λ estimates. If so, the number of sensing nodes required and the optimal positioning of these nodes is to be determined to get a reliable estimate of the localization information (velocity and direction of motion).

4.2 Tracking

The goal in this section is to determine an optimal positioning of three sensing nodes, having coordinates (x_i, y_i) respectively, that will enable them to reliably estimate the

velocity of the intruder. (It can be shown that 2 sensors will not suffice). Set $\eta_i = 1/\lambda_i$. On the basis of (3.6), it can be assumed that the ith sensor node has available with it a reliable estimate of η_i. We make a reasonable assumption that the intruder moves in a more or less straight line path in the field of view of the sensor as the field of view of the sensor is limited only to a few meters. Let the intruder path equation be $ax+by+c=0$, which with $r = \frac{\sqrt{a^2+b^2}}{c}$ and $\alpha = \arctan\left(\frac{a}{b}\right)$, can be rewritten as: $xr\sin\alpha + yr\cos\alpha + 1 = 0$. Note that $\eta_i = \frac{d_{\min,i}}{v}$, where $d_{\min,i}$ is the distance from the ith node to the straight-line path of the intruder. For a node at (x_1, y_1),

$$d_{min,1} = \frac{ax_1 + by_1 + c}{\sqrt{a^2+b^2}} \tag{4.1}$$

$$\Rightarrow \eta_1 = \frac{d_{min,1}}{v} = \frac{\sin\alpha}{v}x_1 + \frac{\cos\alpha}{v}y_1 + \frac{1}{vr} \tag{4.2}$$

We have 3 unknowns, r, α, v but just one equation. Thus, we require two more equations to solve for r, α and v.

$$\eta_2 = \frac{\sin\alpha}{v}x_2 + \frac{\cos\alpha}{v}y_2 + \frac{1}{vr}$$

$$\eta_3 = \frac{\sin\alpha}{v}x_3 + \frac{\cos\alpha}{v}y_3 + \frac{1}{vr}$$

Now, we have 3 equations in 3 unknowns. After some work, it can be shown that

$$v = \frac{1}{\sqrt{s^2+c^2}} \tag{4.3}$$

$$\alpha = \arctan\left(\frac{s}{c}\right) \tag{4.4}$$

$$r = \frac{1}{v\left(\eta_3 - sx_3 - cy_3\right)} \tag{4.5}$$

where

$$\begin{aligned} s &= \frac{\sin\alpha}{v} = \frac{(\eta_1-\eta_2)(y_1-y_3)-(\eta_1-\eta_3)(y_1-y_2)}{(x_1-x_2)(y_1-y_3)-(x_1-x_3)(y_1-y_2)} \\ c &= \frac{\cos\alpha}{v} = \frac{-(\eta_1-\eta_2)(x_1-x_3)+(\eta_1-\eta_3)(x_1-x_2)}{(x_1-x_2)(y_1-y_3)-(x_1-x_3)(y_1-y_2)} \end{aligned}$$

Hence, 3 sensing nodes with η_i estimates and location coordinates (x_i, y_i) will suffice in reliably tracking the intruder.

4.3 Optimal Positioning of the 3 Sensing Nodes

Tracking involves the transformation: $(\eta_1, \eta_2, \eta_3) \rightarrow (r, \alpha, v)$. The impact of error in the estimates of η_i's on r, α and v should be kept minimum for reliable tracking. Equivalently, the Jacobian of the transformation carrying out the mapping: $(r, \alpha, v) \rightarrow (\eta_1, \eta_2, \eta_3)$ should be maximized.

$$J \doteq \begin{vmatrix} \frac{\partial\eta_1}{\partial r} & \frac{\partial\eta_2}{\partial r} & \frac{\partial\eta_3}{\partial r} \\ \frac{\partial\eta_1}{\partial\alpha} & \frac{\partial\eta_2}{\partial\alpha} & \frac{\partial\eta_3}{\partial\alpha} \\ \frac{\partial\eta_1}{\partial v} & \frac{\partial\eta_2}{\partial v} & \frac{\partial\eta_3}{\partial v} \end{vmatrix} \tag{4.6}$$

Without loss of generality, lets assume a coordinate system whose origin is equidistant from the three sensors. Each sensor then is at a constant distance R from the origin. With this, we obtain the below system of 3 equations in the 3 unknowns r, α and v:

$$\eta_i = \frac{\sin\alpha}{v}x_i + \frac{\cos\alpha}{v}y_i + \frac{1}{vr}, \quad 1 \leq i \leq 3. \tag{4.7}$$

We change variables by writing $x_i = R\cos(\beta_i)$, $y_i = R\sin(\beta_i)$, where $\beta_i = \arctan(\frac{y_i}{x_i})$. With this, we have

$$\eta_i = \frac{R}{v}\sin(\alpha+\beta_i) + \frac{1}{vr}, \quad 1 \leq i \leq 3. \tag{4.8}$$

After some work, it can be shown that this Jacobian is given by

$$J \doteq \begin{vmatrix} \frac{\partial \eta_1}{\partial r} & \frac{\partial \eta_2}{\partial r} & \frac{\partial \eta_3}{\partial r} \\ \frac{\partial \eta_1}{\partial \alpha} & \frac{\partial \eta_2}{\partial \alpha} & \frac{\partial \eta_3}{\partial \alpha} \\ \frac{\partial \eta_1}{\partial v} & \frac{\partial \eta_2}{\partial v} & \frac{\partial \eta_3}{\partial v} \end{vmatrix} \tag{4.9}$$

$$= \frac{R^2}{r^2 v^2} \left[\sin(\beta_3 - \beta_2) + \sin(\beta_1 - \beta_3) + \sin(\beta_2 - \beta_1) \right]. \tag{4.10}$$

The value of J is clearly maximized when $\beta_3 - \beta_2 = \beta_1 - \beta_3 = \beta_2 - \beta_1 = \frac{2\pi}{3}$ and when R is made as large as possible. This suggests that the nodes should be arranged in an equilateral triangle with R as large as possible, subject to the desired node density (See Fig. 4.1).

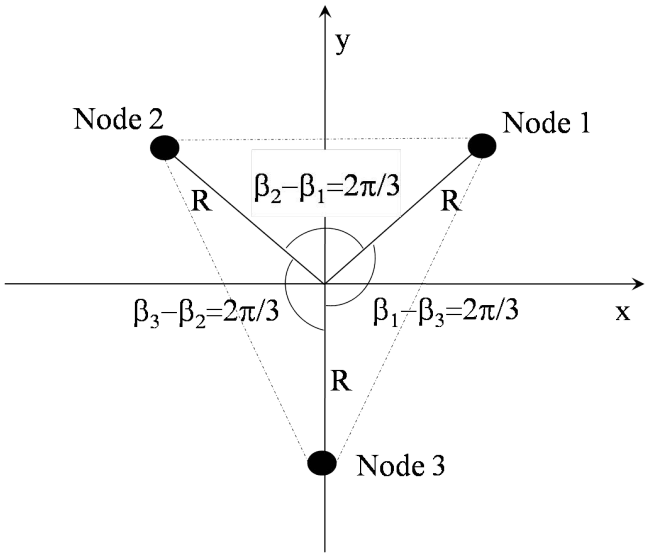


Figure 4.1: Equilateral triangle node arrangement.

Chapter 5

Conclusion and Future Work

In this thesis, application of Wireless Sensor Network (WSN) to the problem of intrusion detection was studied. It is our belief that the challenges have been reasonably met. The surveillance application will continue to be an actively researched area as more and more sophisticated sensors are required.

Infrared imagers in near future will replace the presently available multi-pixel differential-mode infrared sensors in this application. The cost of miniaturized infrared imagers is constantly reducing partly because of the mobile phone market. These imagers will have excellent performance in terms of power and clutter rejection.

It would be very useful to come up with a simple rule for estimating the parameter η from the signature. A reliable estimate of η would bring about an efficient method for tracking the intruder.

Though the algorithm presented in this thesis has good clutter rejection capability and it certainly improves upon the existing algorithms in the literature, finding what best trade-off between detection probability and false alarm probability an algorithm can achieve still remains an open problem. A good model for the clutter signature will help formulate the above problem into a proper detection problem. The solution to this detection

problem will certainly reveal the best tradeoff an algorithm can achieve. Also, it will help in better handling the summer clutter data. From our experimental observations, we believe that the amplitude of the clutter signal depends upon proximity, size of the vegetation and the ambient (or background) temperature; the frequency depends upon stems stiffness of the vegetation as well as the wind speed. A simple model consisting of small, random number of sinusoidal oscillators with random initial phase and frequency of oscillation could be a good starting point for modeling clutter.

The individual nodes (that are currently available) would last only for a few days on a pair of AA batteries. The network is desired to operate at least one year in surveillance application. One popular approach to extend the network's lifetime is to put the nodes on low duty cycle. Duty cycle is the fraction of the time a node is active. For this approach to be feasible, a number of issues need to be addressed [41]. The algorithm presented in the thesis needs to be tweaked to accommodate sleep wake cycling.

It would be interesting to employ compressive sensing techniques and see whether an intrusion in the presence of clutter can still be detected with just a few samples from the signature rather than the entire duration of the signature.

It would also be interesting to find out whether single or multiple intrusions occurred from the signature. A simple starting point would be to make use of the signature duration.

The characteristics of the Passive Infra-Red (PIR) sensor are largely dictated by the characteristics of the multilens. Angular velocity response and sensing range can greatly be improved by a good lens design (size, number and orientation of lenses) [37]. With the current multilens, many beams are not in play - this holds regardless of whether the sensor is placed on the ground (preferred placement) or a few meters above the ground. So, it would be beneficial to optimize the lens system for this application.

Appendix A

Source Code for the Intrusion Detection Algorithm

```
/************************************/

/* CONSTANTS USED IN THE ALGORITHM */

#define ANALOG_DET_WNDW 128
/* =N, size of window to compute Haar transform. Should be a power of 2 */
#define BINS 7
/* BINS = log2(ANALOG_DET_WNDW), number of frequency bins */
#define TEMP_WINDOW 16
/* =L, Number of samples to slide. Should be a power of 2 */
#define AVERAGE_2 306
/* 2*DC(=2450) / 2^SAMPLE_NORMALISE */
#define GAMMA_THRESHOLD 0
/* the projection value is compared against this */
#define SAMPLE_NORMALISE 4
/* the sample value from sensor is normalised by 2^SAMPLE_NORMALISE  */
#define GAMMA_NORMALISE 65536
/* 4096^2 / 2^( 2 * SAMPLE_NORMALISE  ) */
#define CHANNELS 3
/* Number of sensor channels */
#define SUPRESS_THRESHOLD 0.0
/* to declare a new intrusion (to supress multiple successive decisions), the
projected value should have gone below this threshold */
#define NORMALISE ( GAMMA_NORMALISE * ANALOG_DET_WNDW )
/* Besides normalisation of the sample by 4096, this constant also
normalises the squared (unnormalised) HT output by ANALOG_DET_WNDW */
/************************************/

/* HYPERPLANE PARAMETERS */

float W[]={-398.0744,-13.1683,-50.5246,-52.7896,-31.7128,-207.1104,790.8064,
21296.1216};
/* In this W, normalisation of the HT output is taken care*/
float B = -1.0538;
/************************************/
```

```
/* functions and variables used in INTRUSION DETECTION ALGORITHM */
bool firstTime[ CHANNELS ] = { true, true, true };
int sampleArray[ CHANNELS ][ ANALOG_DET_WNDW ];
int circ[ CHANNELS ] = { 0, 0, 0 };
int sampleIndex[ CHANNELS ] = { 0, 0, 0 };
bool supressFlag[ CHANNELS ] = { true, true, true };

int Classify( int );
int Analog_Classify( int, int );
/************************************/

int Analog_Classify( int data, int channel ) /* channel is 0,1,2,... */
{
    int cmp = TEMP_WINDOW, ledon = 2;
    /* ledon = 0 -> Clutter, 1 -> Intruder, 2 -> No decision  */

    sampleArray[ channel ][ circ[ channel ] ] = ( data >> SAMPLE_NORMALISE );
    circ[ channel ]++;
    /* circ[channel] indicates current position of the sample in the array */
    sampleIndex[ channel ]++;
    /* sampleIndex[channel] indicates the number of samples collected for the
    next computation */

    /* clearing up the pointers */
    if( circ[ channel ] == ANALOG_DET_WNDW )
    {
          circ[ channel ] = 0;
    /* wrapping around the array for filling up new samples from the beginning
    of the array*/
    }
    if( firstTime[ channel ] )
    /* buffering 'ANALOG_DET_WNDW' number of samples for the first time,
    thereafter 'TEMP_WINDOW' number of samples */
    {
        cmp = ANALOG_DET_WNDW;
    }
    if( sampleIndex[ channel ] == cmp )
    /* Indicates that buffer is filled with valid data */
    {
        sampleIndex[ channel ] = 0;
    }

    /* calling the detection algorithm */
    if( sampleIndex[ channel ] == 0 )
    {
        ledon = Classify( channel );
        /* Valid samples in tha array start from circ[channel] wrapping
        back at the end of the array and ends at circ[channel]-1 */
        firstTime[ channel ] = false;
    }

    return ledon;
}

int Classify( int channel )
{
    int i = 0, j = 0, k = 0, l = 0, m = 0, p = 0, n = 0, x = 0, y = 0, Lgj = 0,
decision = 0;
    int HT[ ANALOG_DET_WNDW ], sum = 0, diff = 0;
    float binsqr[ BINS+1 ];
    /* binsqr size should be log2(N)+1 */
    float bval = B;

    /* Fast Haar Transform cum Frequency Binning. Output (HT[]) is permuted.
    Binning is done simulatneously in binsqr[] */
    i = 1;
    /* 0.5 * the width of Haar wavelet in each bin */
    j = 2;
    /* the width of Haar wavelet in each bin */
```

```
m = ANALOG_DET_WNDW;
Lgj = 0;

for( l = BINS;  l > 0;  l-- )
/* Bins are computed from BINS to 1 */
{
    binsqr[l] = 0.0;
    m = m >> 1;
    /* Number of components in each bin */
    Lgj++;
    for( k = 0;  k < m;  k++ )
    {
        p = k << Lgj;
        /* time offset for each component in a bin */
        if( l == BINS )
        /* only for the first stage in FHT */
        {
            n = p + circ[ channel ];
            /* updating the starting position of valid samples in the
            array */
            x = n;
            y = n + i;
            if( n >= ANALOG_DET_WNDW )
            {
                x = n - ANALOG_DET_WNDW;
                /* wrapping around the array */
            }
            if( ( n + i ) >= ANALOG_DET_WNDW )
            {
                y = n + i - ANALOG_DET_WNDW;
                /* wrapping around the array */
            }
            sum=sampleArray[channel][x]+sampleArray[channel][y]-(AVERAGE_2);
            diff=sampleArray[channel][x]-sampleArray[channel][y];
        }
        else
        {
            sum = HT[ p ] + HT[ p + i ];
            diff = HT[ p ] - HT[ p + i ];
        }
        HT[ p ] = sum;
        HT[ p + i ] = diff;
        binsqr[ l ] = binsqr[ l ] + (float) HT[ p+i ] * (float) HT[ p+i ];
    }
    i = j;
    j = j << 1;
}
binsqr[ 0 ] = ( float ) HT[ 0 ] * ( float ) HT[ 0 ];

/* gamma=w'x+b with normalisation */
for( i = 0; i <= BINS; i++ )
{
    bval = bval + ( W[ i ] * binsqr[ i ] / NORMALISE );
}

/* to supress independently the multiple successive decisions of the 3
channels */
if( !supressFlag[ channel ] )
{
    if( bval >= GAMMA_THRESHOLD )
    {
        supressFlag[ channel ] = true;
        decision = 1;
    }
}
else
{
    if( bval < SUPRESS_THRESHOLD )
    {
```

```
            supressFlag[ channel ] = false;
        }
    }

    return decision;
}
```

Appendix B

Field Data



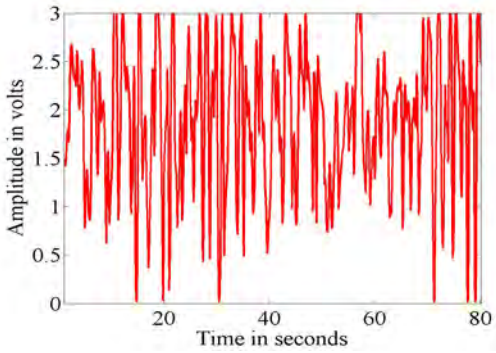


Figure B.1: Sensor facing vegetation and the corresponding data at an ambient temperature of $42°C$. This experiment was performed at IIT Madras.



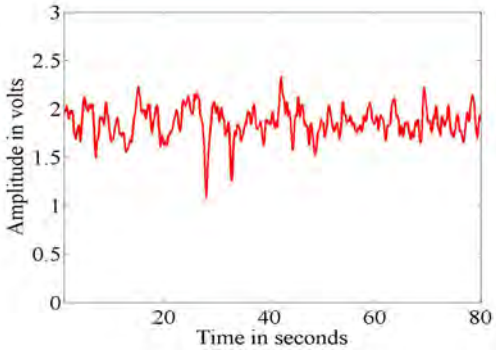


Figure B.2: Sensor facing an open ground and the corresponding data at an ambient temperature of $42°C$. This experiment was performed at IIT Madras.

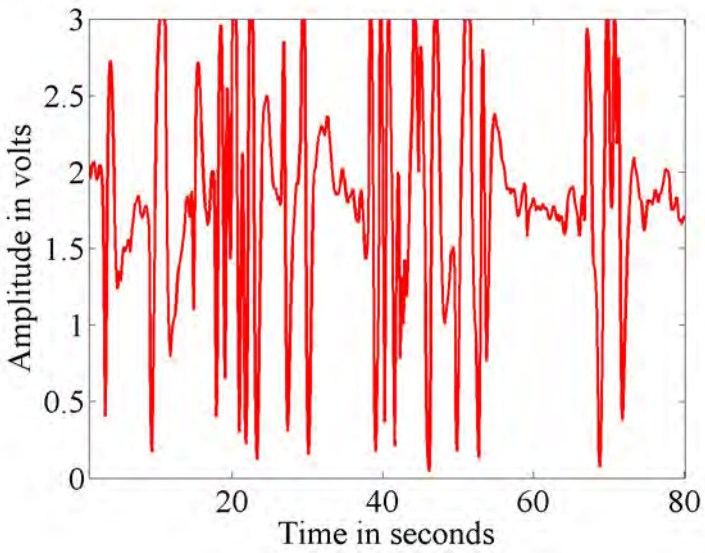


Figure B.3: Sensor data recorded with the sensor facing an open ground with moving clouds at an ambient temperature of $34°C$. This experiment was performed at the air-strip in IISc Bangalore.

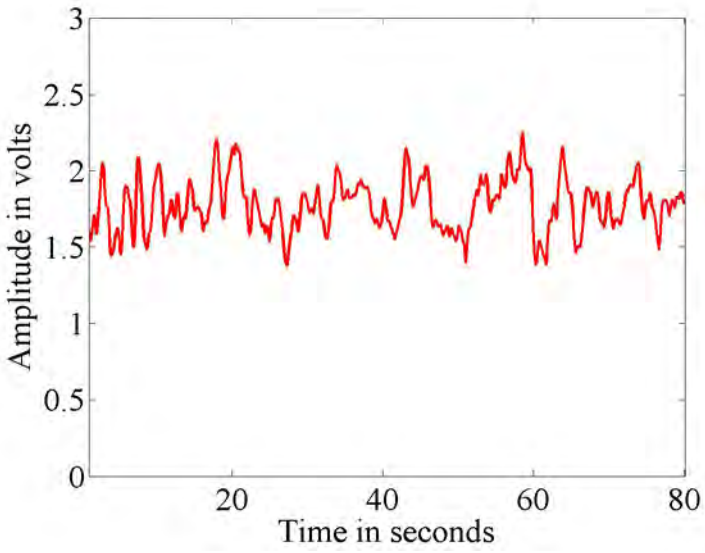


Figure B.4: Sensor data recorded with the sensor facing an open ground at an ambient temperature of $34°C$. This experiment was performed at the air-strip in IISc Bangalore.



Figure B.5: Sensor placed amidst dense grass.



Figure B.6: Sensor placed at a distance of 1 foot from the grass.

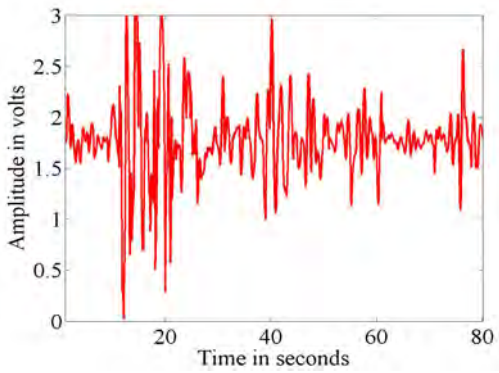


Figure B.7: Data corresponding to Fig. B.5.

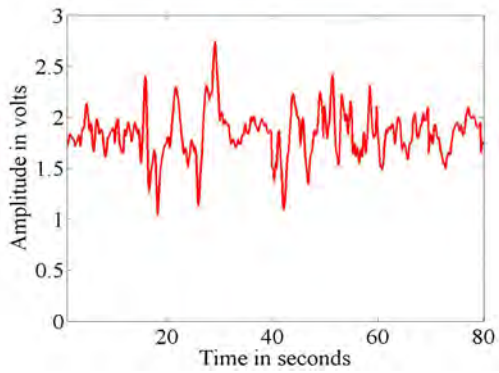


Figure B.8: Data corresponding to Fig. B.6.

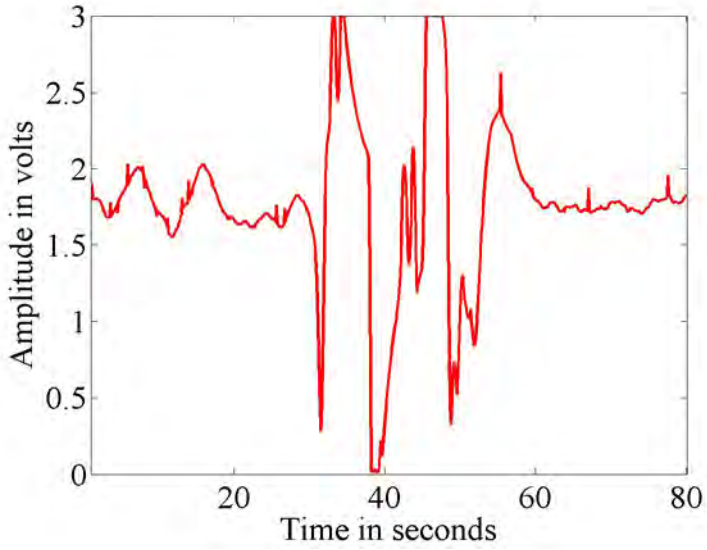


Figure B.9: Sensor data recorded when a dog moved in the vicinity of the sensor.

References

[1] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting Wireless Sensor Networks for Real-Time Surveillance," in *Proc. of the International Conference on Robotics and Automation*, May 2006.

[2] R. Powers, "Batteries for Low Power Electronics," in *Proc. of the IEEE*, pages 687-693, Apr. 1995.

[3] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, et al., "Exscal: Elements of an extreme scale wireless sensor network," in *IEEE RTCSA*, 2005.

[4] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking" , Ohio State University, 2003.

[5] Guillermo Barrenetxea, Franois Ingelrest, Gunnar Schaefer and Martin Vetterli, "Wireless Sensor Networks for Environmental Monitoring: The SensorScope Experience," in *The 20th IEEE International Zurich Seminar on Communications (IZS 2008)*, 2008.

[6] "MP Motion Sensor (AMN 1,2,4) Data Sheet," Panasonic Electric Works Corporation of America, New Jersey, USA.

[7] Sidney B. Lang, "Pyroelectricity: From Ancient Curiosity to Modern Imaging Tool," *Physics Today*, pages 31-36, Aug. 2005.

[8] Thomas M. Andrews, "Installation of External Intrusion Detectors in Harsh Environments," *International Carnahan Conference on Security Technology*, 1993.

[9] *http://en.wikipedia.org/wiki/Passive_infrared_sensor#PIR-based_motion_detector*

[10] Hans J. Keller, "30 Years of Passive Infrared Motion Detectors - A Technology Review," in *OPTO2000 Conference*, 2000.

[11] Shinji Kirihata, Katsuhiro Uchisawa and Masao Yamaguchi, "Low-Profile Dome-Shaped Multilens System," U.S. Patent 6,051,836, Apr. 18, 2000.

[12] Walter Kuster and Hans J. Keller, "Domed Segmented Lens System," U.S. Patent 4,930,864, Jun. 5, 1990.

[13] Herbert L. Berman, "Infrared Intrusion Detector System," U.S. Patent 3,703,718, Nov. 21, 1972.

[14] Scott R. Santoro, "Catoptric Optical System for Detecting Motion of a Radiant Source," U.S. Patent 6,051,836, Apr. 18, 2000.

[15] "Frequency Range for Pyroelectric Detectors," Application Note, PerkinElmer Optoelectronics, http://www.perkinelmer.com/opto.

[16] Alka R Kaushik, Branko G Celler, "Characterization of Passive Infrared Sensors for Monitoring Occupancy Pattern," in *Proc. of the 28th IEEE EMBS Annual International Conference*, Sept. 2006.

[17] M. Kastek, H. Madura, M. Morawski, T. Piatkowski, E. Powiada and H. Polakowski, "Test bed for measurement of angular parameters of passive infrared sensors," in *Elsevier: Infrared Physics & Technology*, vol. 49, pages 198201, July 2006.

[18] Akram Hossain and Muhammad H. Rashid, "Pyroelectric Detectors and Their Applications," in *IEEE Transactions on Industry Applications*, vol. 27, no. 5, pages 824-829, Sep. 1991.

[19] "Telos (Rev B) Datasheet," Moteiv Corporation, http://www.moteiv.com, Dec. 2004.

[20] Joseph Polastre, Robert Szewczyk and David Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.

[21] Prabal Dutta, Mike Grimmer, Anish Arora, Steven Bibyk and David Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random and Ephemeral Events," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.

[22] Xiaofan Jiang, Joseph Polastre and David Culler, "Perpetual Environmentally Powered Sensor Networks," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.

[23] Jonathan W. Hui and David Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004.

[24] Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, pages 144-152, ACM Press, 1992.

[25] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: a Library for Support Vector Machines," Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[26] Zhiqiang Zhang, Xuebin Gao, Jit Biswas and Jian Kang Wu "Moving Targets Detection and Localization in Passive Infrared Sensor Networks," in *The 10th International Conference on Information Fusion*, July 2007.

[27] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, T. He, A. Tirumala, Q. Cao, J. Stankovic, T. Abdelzaher and B. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *SenSys*, Nov. 2005.

[28] Bhaskaran Raman and Kameswari Chebrolu, "Censor Networks: A Critique of "Sensor Networks" from a Systems Perspective," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pages 75-78, July 2008.

[29] Peter Harrop, "Wireless Sensor Networks 2009-2019," IDTechEx, Nov. 2008.

[30] Robert R. Tenney and Nils R. Sandell, Jr., "Detection With Distributed Sensors," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-17, no. 4, pages 501-510, July 1981.

[31] R. Völkel, M. Eisner, K.J. Weible, "Miniaturized Imaging Systems," in *Microelectronic Engineering* 67-68, pages 461-472, 2003.

[32] D.K. Aravind and K.J. Wong, "Speckled Computing: Disruptive Technology for Networked Information Appliances," in *IEEE International Symposium on Consumer Electronics (2004)*, pages 219-223.

[33] Mohan Shankar, John B. Burchett, Qi Hao, Bob D. Guenther, David J. Brady, "Human-Tracking Systems Using Pyroelectric Infrared Detectors," in *Optical Engineering*, vol. 45, Oct. 2006.

[34] *MSP430 Ultra-Low-Power Microcontrollers*, Texas Instruments, 2008.

[35] *http://en.wikipedia.org/wiki/Haar_transform#Haar_transform*

[36] *http://www.glolab.com/pirparts/infrared.html*

[37] *http://www.fresneltech.com/*

[38] *http://www.wsnblog.com/*

[39] *http://www.libelium.com/*

[40] *http://www.xbow.com/*

[41] Santosh Kumar, Ten H. Lai, József Balogh, "On k–Coverage in a Mostly Sleeping Sensor Network," in *Mobicom '04*, Sep. 2004.