

Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN INGEGNERIA ELETTRONICA,
INFORMATICA E DELLE TELECOMUNICAZIONI

Ciclo XXI

**PROGETTO DI RETI DI SENSORI
WIRELESS E TECNICHE DI FUSIONE
SENSORIALE**

Tesi di Dottorato di:

PIERO ZAPPI

Relatori :

Chiar. mo Prof. Ing. **LUCA BENINI**

Coordinatore:

Chiar. ma Prof.ssa Ing. **PAOLA MELLO**

Settore Scientifico Disciplinare: ING/INF01 Elettronica

Esame finale anno 2009

**PROGETTO DI RETI DI SENSORI
WIRELESS E TECNICHE DI FUSIONE
SENSORIALE**

A dissertation submitted to the
DEPARTEMENT OF ELECTRONICS, COMPUTER SCIENCE AND SYSTEMS
OF UNIVERSITY OF BOLOGNA

for the degree of Doctor of Philosophy

presented by
PIERO ZAPPI
born October 18, 1980

March 2009

Keywords

Ambient intelligence (AmI)

Sensor Networks

Data Fusion

Classifiers

Multimodal Surveillance

Dynamic Networks

Gesture Recognition

Acknowledgements

The author would like to thank all the people that supported him during these years of Ph.D. course.

A special thanks goes to my parents that always trust me and my choices. Without your support and encouragement probably I would not be able to reach this wonderful goal. Thanks for all the time that you've been patient with me and accepted my lacking answer at your questions.

Thanks to prof. Luca Benini for giving me the opportunity to express myself with my research. Thanks for all the times that you believed in my ideas and for your help when I needed it. Thanks also to all the people at WSN group at MicrelLab. Elisabetta, Augusto, Davide, Luciano, Paolo, Sergio, Michele, Omar, Marco, Carlo and Alberto I am so happy I could share so much of my work and my life with you.

Thanks to all the people at Wearable lab at ETH (Zurich). Thanks to prof. Gerhard Tröster and Daniel Roggen for giving me the opportunity to spend my six month internship within your group. It enriched both my scientific career and my-self. A special thanks to: Pius, Maggy, Martin, Andrea, Clemens, Conny, Thomas and Nag that made my experience in Zurich unforgettable.

Last but not least, a huge thanks and hug goes to all my friend that made me enjoy the moments outside the lab. A special thanks to Benedetta, probably the only one who have seen how hard it has been for me sometimes.

To all of you: *Grazie!*

Contents

| | |
|--|------------|
| List of Figures | XI |
| List of Tables | XIV |
| Abstract | 1 |
| 1 Introduction | 3 |
| Introduction | 3 |
| 1.1 Ambient Intelligence: a vision on the future of electronic systems | 3 |
| 1.2 Thesis Organization Outline | 6 |
| 2 Ambient Intelligence | 9 |
| Ambient Intelligence | 9 |
| 2.1 Ambient Intelligence: general definitions | 9 |
| 2.2 Ambient Intelligence projects | 14 |
| 3 Wireless Sensor Networks | 17 |
| Wireless Sensor Networks | 17 |
| 3.1 Wireless Sensor Networks overview | 17 |
| 3.1.1 Wireless Sensor Network applications | 18 |
| 3.2 Wireless Sensor Nodes | 21 |
| 3.2.1 Computational Unit | 23 |
| 3.2.2 Sensor and Actuator Unit | 25 |
| 3.2.3 Communication Unit | 25 |
| 3.2.4 Power Unit | 26 |
| 3.3 State of the art | 27 |
| 3.3.1 Smart Dust | 27 |
| 3.3.2 Intel mote | 28 |
| 3.3.3 Mica Mote | 29 |
| 3.3.4 Tmote Sky | 29 |

| | | |
|----------|--|-----------|
| 3.3.5 | BT Node | 30 |
| 3.3.6 | System on chip | 30 |
| 3.4 | Protocols for sensor networks | 32 |
| 3.4.1 | Zigbee motivations | 32 |
| 3.4.2 | IEEE 802.15.4 Overview | 34 |
| 3.4.3 | Physical Layer | 35 |
| 3.4.4 | Media Access Controller (MAC) Layer | 37 |
| 3.4.5 | Zigbee | 39 |
| 3.4.6 | Security | 47 |
| 4 | Data Fusion and Pattern Recognition | 51 |
| | Data Fusion and Pattern Recognition | 51 |
| 4.1 | Data Fusion overview | 51 |
| 4.2 | Direct fusion of sensor data | 53 |
| 4.2.1 | Kalman filter | 55 |
| 4.3 | Fusion of features vectors | 57 |
| 4.3.1 | Naïve Bayes Classifier | 60 |
| 4.3.2 | Support Vector Machines | 61 |
| 4.3.3 | K-Nearest Neighbors | 64 |
| 4.4 | Fusion of high level inferences | 65 |
| 4.4.1 | Majority voting | 66 |
| 4.4.2 | Borda Count | 67 |
| 4.5 | Time variant classifiers | 67 |
| 4.5.1 | Hidden Markov Models | 67 |
| 5 | Multimodal surveillance | 73 |
| | Multimodal surveillance | 73 |
| 5.1 | Overview | 73 |
| 5.2 | Pyroelectric InfraRed (PIR) Sensors | 75 |
| 5.2.1 | Related work | 79 |
| 5.3 | Direction and number of people detection | 81 |
| 5.3.1 | System description | 81 |
| 5.3.2 | Model and system analysis | 84 |
| 5.3.3 | Data fusion | 89 |
| 5.3.4 | Evaluation | 90 |
| 5.3.5 | Conclusion | 92 |
| 5.4 | Distance estimation | 92 |
| 5.4.1 | System description | 92 |
| 5.4.2 | Model analysis | 93 |

| | | |
|----------|---|------------|
| 5.4.3 | Test and results | 96 |
| 5.4.4 | Conclusion | 97 |
| 5.5 | Multimodal sensor network for video surveillance | 98 |
| 5.5.1 | Integrated multimodal sensor networks | 98 |
| 5.5.2 | PIR sensor network | 100 |
| 5.5.3 | Sensing and data collection | 101 |
| 5.5.4 | Vision system | 104 |
| 5.5.5 | Multi modal integration | 107 |
| 5.5.6 | Conclusion | 110 |
| 5.6 | A solar-powered video sensor node for energy efficient multi-modal surveillance | 110 |
| 5.6.1 | Related work | 112 |
| 5.6.2 | System architecture | 112 |
| 5.6.3 | System analysis | 118 |
| 5.6.4 | Dynamic adjustment of the detection area | 120 |
| 5.6.5 | Conclusion | 124 |
| 6 | Activity Recognition in Redundant and Dynamic Sensor Networks | 125 |
| | Gesture Recognition In Redundant and Dynamic Sensor Networks | 125 |
| 6.1 | Overview | 125 |
| 6.2 | Related work | 128 |
| 6.2.1 | Gesture recognition | 128 |
| 6.2.2 | Tangible interfaces | 131 |
| 6.2.3 | Overview of available datasets | 132 |
| 6.3 | Activity recognition from body worn sensors: scalability and robustness | 136 |
| 6.3.1 | System architecture | 138 |
| 6.3.2 | Experimental setup | 141 |
| 6.3.3 | Test and results | 144 |
| 6.3.4 | Discussion | 152 |
| 6.3.5 | Conclusion | 154 |
| 6.4 | Activity recognition accuracy-power trade-off by dynamic sensor selection | 156 |
| 6.4.1 | Dynamic sensor selection | 157 |
| 6.4.2 | Accuracy of cluster performance estimation | 160 |
| 6.4.3 | Power-performance tradeoff characterization | 161 |
| 6.4.4 | Implementation using Tiny Task Networks (Titan) | 167 |
| 6.4.5 | Discussion | 169 |
| 6.4.6 | Conclusion | 171 |

| | | |
|----------|---|------------|
| 6.5 | Tangerine SMCube: a smart device for human computer interaction | 172 |
| 6.5.1 | The TANGerINE framework | 172 |
| 6.5.2 | SMCube overview | 174 |
| 6.5.3 | Gesture detection algorithm | 176 |
| 6.5.4 | HMM for SMCube, a feasibility study | 179 |
| 6.5.5 | Conclusion | 182 |
| 6.6 | Pervasive datasets | 184 |
| 6.6.1 | Experiment setup | 185 |
| 6.6.2 | Benchmark for context recognition | 188 |
| 6.6.3 | Conclusion | 189 |
| 7 | Conclusion | 191 |
| | Conclusion | 191 |
| | Bibliography | 192 |

List of Figures

| | | |
|------|---|----|
| 2.1 | Trends in computing | 11 |
| 2.2 | Distributed communication network | 12 |
| 2.3 | Intelligent Natural Interfaces (<i>Photo:Philips</i>) | 13 |
| 3.1 | WSN Application on battlefield | 18 |
| 3.2 | Structure of the WSN for habitat monitoring on Great Duck Island | 19 |
| 3.3 | Audio bio-feedback for impaired people support | 20 |
| 3.4 | WSN can be used for logistic support | 21 |
| 3.5 | Generic architecture of a sensor node | 24 |
| 3.6 | WiMoCA wireless sensor node | 24 |
| 3.7 | A diagram of the Smart Dust mote | 27 |
| 3.8 | The intel mote | 28 |
| 3.9 | The intel mote 2 | 29 |
| 3.10 | The Mica mote | 29 |
| 3.11 | The Tmote Sky mote | 30 |
| 3.12 | The BTnode mote | 31 |
| 3.13 | Comparision of Zigbee and other standards | 33 |
| 3.14 | Zigbee Stack development contribution | 33 |
| 3.15 | Zibee application range | 34 |
| 3.16 | IEEE 802.15.4 Network topologies | 35 |
| 3.17 | IEEE 802.15.4 superframe structure | 37 |
| 3.18 | Data transfer between a device and the coordinator in a beacon enabled network | 38 |
| 3.19 | Data transfer between the coordinator and a device in a beacon enabled network | 38 |
| 3.20 | Data transfer between a device and the coordinator in a non bea- con enabled network | 39 |
| 3.21 | Data transfer between the coordinator and a device in a non bea- con enabled network | 39 |
| 3.22 | Zigbee Stack overview | 39 |

| | |
|--|----|
| 3.23 Zigbee Network topologies | 40 |
| 3.24 Example of Zigbee address allocation | 42 |
| 3.25 Example of Zigbee tree routing | 43 |
| 3.26 Example of Zigbee neighbor routing | 43 |
| 3.27 Example of Zigbee mesh routing | 43 |
| 3.28 Example of Zigbee application, endpoint and clusters. | 46 |
| 3.29 Example of binding process. | 47 |
| 4.1 JDL Data Fusion model | 52 |
| 4.2 Fusion of sensor data | 53 |
| 4.3 Fusion of features vectors | 54 |
| 4.4 Fusion of high level inferences | 54 |
| 4.5 Pattern recognition steps | 58 |
| 4.6 Best separating hyperplane in the separable case (feature space = 2) | 62 |
| 4.7 Two options for building a set of separating hyperplanes in the multiple class example | 63 |
| 4.8 Separating hyperplanes in case of non separable data. | 63 |
| 4.9 K-NN mapping in the two class case, with $d=2$ and $k=1$ | 64 |
| 4.10 Diagram of the possible classifier fusion methods | 66 |
| 4.11 Example of HMM with $N = 3$ and $M = 3$ | 69 |
| 5.1 black body radiation curve at 37°C | 76 |
| 5.2 Schematic illustration of a PIR sensor | 77 |
| 5.3 Chain conversion for PIR sensors | 77 |
| 5.4 PIR preamplification modes. | 78 |
| 5.5 Schematic of a COTS PIR | 79 |
| 5.6 Output of a PIR sensor used in conjunction with array of Fresnel lenses | 79 |
| 5.7 Modified FOV of the PIR sensor | 83 |
| 5.8 Output of the PIR sensor when a single lens is used and a person moves back and forth in front of it. | 83 |
| 5.9 Setup of the PIR network used to detect number of person and direction of movement. | 84 |
| 5.10 Events of interests: passages of 1, 2 or 3 persons in sequence or in parallel. | 85 |
| 5.11 Reactiveness-noise immunity trade off in threshold selection. | 87 |
| 5.12 Output of the "correct" side PIRs when 2 people are passing. | 88 |
| 5.13 Output of the "wrong" side PIRs when 2 people are passing. Shadowing is highlighted. | 88 |

| | | |
|------|---|-----|
| 5.14 | Output of the middle pir when 1,2 and 3 people are passing in a row. | 91 |
| 5.15 | Output the three PIR when three people are passing side by side. PIR 1 is in the "best" position, the central PIR (PIR 2) shows an output similar to the one in the case of single person, PIR 3 output is affected by shielding effect. | 91 |
| 5.16 | The building block of the network is made up of two PIRs that autonomously monitor a slice of the AoI. The space between them is divided into three zones. | 93 |
| 5.17 | Output of a PIR sensor in case of passages at different distances. | 93 |
| 5.18 | Schematic of a typical C.O.T.S. PIR. Two sensing elements are used in series with opposite polarization, the output is pre-amplified through a built in MOS transistor. Highlighted with shading, the FoV of each sensing element. Notice how, in proximity of the device, the two FoVs are overlapped. | 94 |
| 5.19 | Task allocation for distance detection. | 95 |
| 5.20 | Mapping of input vector in the two dimensional feature space. The three classes are located into partially overlapped areas of the space. | 95 |
| 5.21 | Map of the test bed system. | 99 |
| 5.22 | Software architecture of the system. | 99 |
| 5.23 | General architecture of the PIR sensor node. | 100 |
| 5.24 | Schematic of the Fresnel lens used in this project. | 101 |
| 5.25 | Signals detected by sensor: when a person passes through the area under control from left to right the first peak is negative and from right to left the first peak is positive. | 101 |
| 5.26 | Event acquisition and Sensor Data Conditioning and Processing. | 102 |
| 5.27 | Sensor node composed by two PIRs | 103 |
| 5.28 | PIR activation sequence. | 103 |
| 5.29 | Communication protocol between nodes and sensor manager. . | 104 |
| 5.30 | Bird-eye-view description of our test bed environment. | 107 |
| 5.31 | Example of consistent labelling between three views. | 107 |
| 5.32 | Opening and closing doors make unreliable background suppression techniques. | 108 |
| 5.33 | Sensor-guided background update. | 109 |
| 5.34 | Consistent labelling after an occlusion exploiting a PIR node to detect direction changes | 110 |
| 5.35 | Hierarchical design of the video sensor node, with three different layers for the alert system. | 111 |
| 5.36 | Video sensor node architecture. | 113 |

| | | |
|------|--|-----|
| 5.37 | Developed prototype of the video sensor node. | 114 |
| 5.38 | Characteristic of the photovoltaic module. | 115 |
| 5.39 | Conceptual schematic of solar harvester: buck power converter and MPP tracker. | 116 |
| 5.40 | Output of a PIR sensor when a person moves at different distances | 117 |
| 5.41 | Schematics for trigger generation using PIR output signal. . . . | 118 |
| 5.42 | Flow chart of the human detection application. | 119 |
| 5.43 | Autonomy of the system varying the capacity of the reservoirs without environmental harvested energy | 121 |
| 5.44 | Amplitude of the PIR Output signal as function of the distance of the object. | 122 |
| 5.45 | Simulation results of the energy efficiency using a dynamic PIR sensitivity threshold. | 123 |
| 6.1 | Floor plan and pictures of the interior of the PlaceLab. | 134 |
| 6.2 | Example of image analysis and information extraction. | 137 |
| 6.3 | Activity recognition architecture. Features extracted from the sensor data are classified by competing hidden Markov models, each one trained to model one activity class. The most likely model yields the class label. These labels are fused to obtain an overall classification result. A naive Bayesian scheme, borda count scheme and a majority voting scheme are used. | 139 |
| 6.4 | Placement of the acceleration sensor nodes on the user. Ten nodes are uniformly distributed on each user arms (20 nodes in total). No specific assumption has been made on sensor position and orientation. | 144 |
| 6.5 | Average node level classifier (HMM) classification accuracy as a function of the threshold used to ternarize the input data from accelerometers when single axis acceleration features (left) or ac- celeration magnitude (right) features are used. | 145 |
| 6.6 | Sensor scalability. Average classification accuracy as a function of the number of nodes participating to the classification when different network level fusion methods are used. Vertical bars indicate classification variance. In each sub-plot single axis ac- celeration features and acceleration magnitude features are com- pared. Dashed lines indicate the best performance achieved us- ing a single node, the fusion of an increasing number of sensors quickly results in better performance. | 145 |

- 6.7 Sensor scalability. Average classification accuracy as a function of the number of nodes participating to the classification. Vertical bars indicate classification variance. On each plot different network level classifiers are compared. 146
- 6.8 Graphical representation of the confusion matrix when 1, 7, 13 or 20 sensor are participating to the classification. Naïve Bayes fusion and single axis acceleration features are used. Darker spots indicated higher number of classified instances. As can be seen from these plots, increasing the number of sensors results in a drift toward the main diagonal meaning higher classification accuracy. 148
- 6.9 Robustness to rotational noise. Average correct classification ration of a system with 1, 5, 10 or 20 active sensors as a function of the rotational noise added. Increasing the rotational noise results in decreasing classification accuracy. This degradation can be compensated by increasing the number of active nodes. . . . 149
- 6.10 Average classification accuracy of the system as a function of both level of rotational noise added to accelerometers output and number of sensors participating to classification. Naïve Bayes fusion method and single axis acceleration features used. 150
- 6.11 Average classification accuracy when 1, 5 or 20 nodes participate to the classification. The performance using different fusion methods are compared as the rotational noise added to the accelerometer output increases. 150
- 6.12 Robustness to random noise. Average classification accuracy of a system with 20 active nodes with an increasing number of nodes affected by random noise. Single axis acceleration features and acceleration magnitude features are compared. 151
- 6.13 Robustness to random noise. Average classification accuracy of a system with 20 active nodes with increasing number of nodes affected by random noise. The majority voting, Borda count and naïve Bayes fusion methods are compared using different features. 152
- 6.14 Comparison of the average classification accuracy of a cluster of 20 nodes when the number of active nodes is progressively decreased (variable size set), and when the number of nodes affected by random noise is progressively increased. The detection and exclusion of faulty nodes results in higher system performance. 153

- 6.15 The dynamic sensor selection algorithm starts by evaluating the existing cluster of nodes. If it does not fulfill the performance requirement, it looks for a new cluster of nodes by iteratively evaluating clusters of increasing size until the performance requirements are fulfilled. 159
- 6.16 Probability density function of the performance prediction error as a function of: the number of nodes in the cluster(a), the reference set size (b) 161
- 6.17 Average network lifetime in multiple of μ as a function of the target performance with Borda count as fusion method. The DSS sensor selection algorithm *best*, *first* and *random* are compared with the reference case where *all* the nodes are active together (no DSS algorithm). The DSS algorithm enables longer network lifetime thanks to the better management of available resources. 163
- 6.18 Average network lifetime in multiple of μ as a function of the target performance with single accelerometer axis features. The majority voting, Borda count and naïve Bayes fusion methods are compared. The use of the Borda count method results in longer network lifetime in comparison to the other fusion methods. 163
- 6.19 Average network lifetime in multiple of μ as a function of target performance for single acceleration axis and acceleration magnitude features. Borda count fusion method is used. The use of single accelerometer axis features results in longer lifetime. . . 164
- 6.20 Evolution of the classification accuracy over time, as a function of the target performance. The Borda count fusion method and single accelerometer axis features with *first* sensor selection criteria are used. The continuous line indicates the performance with the DSS algorithm selecting the nodes participating to activity recognition. The dashed line indicates the performance of the system where all the nodes are active simultaneously. Vertical segments indicate when a sensor is turned off and replaced. The dotted line indicates the target performance. When all the nodes are simultaneously used, the performance of the system quickly drops under the target threshold once time approaches the average node lifetime. With the DSS algorithm a better management of the active nodes allows to lengthen the system operation time. 165

| | | |
|------|--|-----|
| 6.21 | Node activation sequence corresponding to figure 6.20. The vertical axes represents the time (increasing toward the bottom). Each column corresponds to a node. Dark spots indicate when a node is active. Whenever a node is turned off it is replaced by one or more idle nodes until all nodes of the network have been used. | 166 |
| 6.22 | Titan configures an application task graph by assigning parts of the graph to participating sensor nodes depending on their processing capabilities | 168 |
| 6.23 | Setup for the Tangerine framework and identification of the three contexts. | 173 |
| 6.24 | The Tangerine SMCube and its use within the tabletop environment | 175 |
| 6.25 | Accelerometer output waveform when a <i>tap</i> event happen. . . . | 177 |
| 6.26 | Micrel SMCube application. | 177 |
| 6.27 | Micrel SMCube tasks. | 178 |
| 6.28 | Labeling software with a) atomic label sequence, b) composite label sequence c) start/stop button, and d) sensor health status. | 187 |

List of Tables

| | | |
|------|---|-----|
| 3.1 | Popular sensors and their output. | 26 |
| 3.2 | IEEE 802.15.4 frequency bands, modulation and spreading. . . . | 36 |
| 5.1 | Murata IRA E710 PIR sensor characteristics. | 82 |
| 5.2 | average, max and min duration of 1st and 2nd peak when one, two or three people are passing. | 87 |
| 5.3 | Experimental results. | 90 |
| 5.4 | Classifiers performance (correct classification ratio). | 96 |
| 5.5 | Classifiers computational effort to perform the classification of a single instance and memory cost (number of double) to imple- ment the classifier. $N_{sv1} = 257$, $N_{sv1} = 235$ and $T = 300$ | 97 |
| 5.6 | Naïve Bayes classifier's confusion matrix | 97 |
| 5.7 | Adopted codes | 104 |
| 5.8 | Power consumption of the video sensor node. | 119 |
| 5.9 | Energy requirement | 120 |
| 6.1 | List of activity classes to recognize from body-worn sensors. . . | 143 |
| 6.2 | Performance comparison between different fusion methods, fea- tures, and number of active nodes. | 147 |
| 6.3 | Computational effort | 154 |
| 6.4 | Average network lifetime (single accelerometer axis features). . | 164 |
| 6.5 | Average network lifetime (acceleration magnitude features). . . | 165 |
| 6.6 | Dynamic sensor selection algorithm computational effort when 85% minimum classification accuracy is needed. The values in the table represent the average number of equivalent clusters of size 1 that are evaluated during the network evolution. | 166 |
| 6.7 | Latency in gesture recognition. | 179 |
| 6.8 | Computational complexity | 181 |
| 6.9 | Algorithm complexity | 181 |
| 6.10 | Classification performances | 182 |
| 6.11 | Performance and cost comparison | 182 |

| | | |
|------|---|-----|
| 6.12 | Number of composite and atomic activities and total number of activity occurrences. | 186 |
| 6.13 | Overall experiment message loss. | 188 |

Abstract

Ambient Intelligence (AmI) envisions a world where smart, electronic environments are aware and responsive to their context. People moving into these settings engage many computational devices and systems simultaneously even if they are not aware of their presence. AmI stems from the convergence of three key technologies: *ubiquitous computing*, *ubiquitous communication* and *natural interfaces*.

The dependence on a large amount of fixed and mobile sensors embedded into the environment makes of *Wireless Sensor Networks* one of the most relevant enabling technologies for AmI. WSN are complex systems made up of a number of sensor nodes, simple devices that typically embed a low power computational unit (microcontrollers, FPGAs etc.), a wireless communication unit, one or more sensors and a some form of energy supply (either batteries or energy scavenger modules). Low-cost, low-computational power, low energy consumption and small size are characteristics that must be taken into consideration when designing and dealing with WSNs. In order to handle the large amount of data generated by a WSN several multi sensor data fusion techniques have been developed. The aim of multisensor data fusion is to combine data to achieve better accuracy and inferences than could be achieved by the use of a single sensor alone.

In this dissertation we present our results in building several AmI applications suitable for a WSN implementation. The work can be divided into two main areas: *Multimodal Surveillance* and *Activity Recognition*.

Novel techniques to handle data from a network of low-cost, low-power *Pyroelectric InfraRed* (PIR) sensors are presented. Such techniques allow the detection of the number of people moving in the environment, their direction of movement and their position. We discuss how a mesh of PIR sensors can be integrated with a video surveillance system to increase its performance in people tracking. Furthermore we embed a PIR sensor within the design of a *Wireless Video Sensor Node* (WVSN) to extend its lifetime.

Activity recognition is a fundamental block in natural interfaces. A challenging objective is to design an activity recognition system that is able to

exploit a redundant but unreliable WSN. We present our activity in building a novel activity recognition architecture for such a dynamic system. The architecture has a hierarchical structure where simple nodes performs gesture classification and a high level meta classifiers fuses a changing number of classifier outputs. We demonstrate the benefit of such architecture in terms of increased recognition performance, and fault and noise robustness. Furthermore we show how we can extend network lifetime by performing a performance-power trade-off.

Smart objects can enhance user experience within smart environments. We present our work in extending the capabilities of the *Smart Micrel Cube* (SM-Cube), a smart object used as tangible interface within a tangible computing framework, through the development of a gesture recognition algorithm suitable for this limited computational power device.

Finally the development of activity recognition techniques can greatly benefit from the availability of shared dataset. We report our experience in building a dataset for activity recognition. Such dataset is freely available to the scientific community for research purposes and can be used as a testbench for developing, testing and comparing different activity recognition techniques.

Chapter 1

Introduction

1.1 Ambient Intelligence: a vision on the future of electronic systems

The term *Ambient Intelligence (AmI)* refers to a vision on the future of the information society where smart, electronic environment are sensitive and responsive to the presence of people and their activities (*Context awareness*). In such smart environment, technology is invisible and embedded into the surrounding. People moving into this settings engage many computational devices and systems simultaneously even if they are not aware of their presence.

The concept of ambient intelligence has been developed in the late 90s. The basic idea derives from the concept that, while computers purported to serve people, being hard to use and not aware of our needs, they have actually forced humans to serve them. In the future, instead, the computation will be human-centered and will be brought to us whenever and wherever we need it through hand-held devices or embedded in the environment. Moreover, people will not have to learn how to use electronic devices, but they will interact with them in a more natural and intuitive way.

The ISTAG (Information Society Technology Advisory Group) has defined four scenarios to offer glimpse of futures that can be realized. Each scenario highlights a number of technological challenges that have to be tackled by researchers. Among them:

- Very unobtrusive hardware. This includes battery size reduction or self powered devices, sensor and actuator integrated into everyday object and new displays.
- A seamless mobile/fixed communications infrastructure. Complex heterogeneous networks need to function and to communicate in a seamless

and interoperable way.

- Dynamic and massively distributed device networks. An uncountable number of wireless, wired and mobile devices will coexist in the same space. Thus the network should be configurable on an ad hoc basis and dynamically adapt to changes.
- Natural interaction. System should be intuitive, like normal human behaviors.
- Dependability and security. AmI system will control many critical activities and handle a considerable amount of sensitive data. Thus this technologies should be tested to make sure they are safe for use.

The dependence on a large amount of fixed and mobile sensors embedded into the environment makes of *Wireless Sensor Networks* (WSN) one of the most relevant enabling technologies for AmI.

WSN are complex systems made up of a number of devices called sensor nodes. Each sensor node typically includes one or more sensors, a wireless radio, an energy source (either batteries or some energy harvesting unit) and a microcontroller with enough computational power to collect data from the sensors and perform some sort of computation with it. WSN design space is very wide and spans from small, fixed *Body Area Networks* (BAN) for rehabilitation composed of a handful of sensors nodes placed over the body, to large, dynamic networks for environment or animals monitoring consisting of thousand of nodes.

Despite the design of the sensor network and its sensor nodes is strictly application dependent, a number of constraints should almost always be considered. Among them:

- Small form factor to reduce nodes intrusiveness.
- Low power consumption to reduce battery size and to extend nodes lifetime.
- Low cost for a widespread diffusion.

These limitations typically result in the adoption of low power, low cost devices such as low power microcontrollers with few kilobytes of RAM and tenth of kilobytes of program memory with whom only simple data processing algorithms can be implemented. However the overall computational power of the WNS can be very large since the network presents a high degree of parallelism that can be exploited through the adoption of ad-hoc techniques. Furthermore through the fusion of information from the dense mesh of sensors even complex phenomena can be monitored.

Multi sensor data fusion is an emerging technology that has received significant attention for several human activities. When we talk about data fusion techniques, we refer to methods that aim at combining data from multiple sources (sensors, databases etc.) to achieve better accuracy and inferences than could be achieved by the use of a single sensor alone.

In general, using an efficient fusion schema several advantages can be expected.

- Improve the knowledge's confidence thanks to complementary informations.
- Improve the detection of phenomenon characteristics.
- Increase robustness to noise in adverse environmental conditions.

It is clear how AmI systems bring together several aspects of research. For this reason the Ph.D. research activity presented in this dissertation can be divided into two main topics.

Multimodal surveillance

Information from the environment can be collected through different sensor modalities. Video surveillance and other security-related applications have gained many credits during the last years. Several industrial and academic projects have recently started to increase the accuracy of (semi) automatic surveillance systems. In addition, the abatement of hardware costs allows the deployment of thousands of cameras for surveillance purposes at a reasonable cost. Despite the efforts made by the researchers in developing a robust multi-camera vision system, computer vision algorithms have proved their limits to work in complex and cluttered environments. One of the reason of such limits is that non-visible areas can not be processed by the system. In this context, the marriage between a widely distributed low-cost WSN and the coarsely distributed higher level of intelligence that can be exploited by computer vision systems may overcome many troubles in a complete tracking of large areas. For this reason several techniques to monitor the environment through Pyroelectric Infra-Red (PIR) sensors have been developed. Being low-cost, passive (thus low-power) and presenting a limited form factor, PIR sensors are well suited for WSN applications. In particular we developed novel modalities to detect direction of movement, to count the number and to track position of people moving within a smart environment. Such techniques have proved to increase the tracking accuracy of the multimodal system.

In several setup the use of wired video cameras may not be possible. For

this reason building an energy efficient wireless vision network for monitoring and surveillance is one of the major efforts in the sensor network community. The PIR sensors have been used to extend the lifetime of a solar-powered video sensor node by providing an energy level dependent trigger to the video camera and the wireless module. Such approach has shown to be able extend node lifetime and possibly result in continuous operation of the node.

Activity recognition in redundant and dynamic sensor networks

AmI envisions the large scale deployment of highly miniaturized, unobtrusive and interconnected (wireless) sensor nodes. This unobtrusive yet widespread sensing permits pervasive and wearable computing systems that provide transparent and natural human-computer interfaces (HCI) and smart assistance to users according to their context. Current approaches to activity recognition tend to assume static body-worn (wireless) sensor networks. Usually a fixed, often minimal, set of sensors nodes placed at well defined body locations is used. Nodes and their interconnections do not fail and sensor data is not affected by noise. However, under realistic conditions, on-body sensor networks tend to be dynamic. We developed an activity recognition signal processing chain suited for such dynamic sensor networks. It takes advantage of multiple sensors to cope with failures, noise and enable power-performance management. This approach is suited for other application domains where a large number of sensor nodes is used to monitor areas of interests. It is independent of specific sensors and classifiers used, and it is suitable for distributed execution.

Tangible interfaces play a fundamental role within AmI applications since they provide a natural way to interact with smart environment. Furthermore smart object used as tangible interface provide redundant information about user activity. Thus we developed the Smart Micrel Cube (SMCube) a smart object able to recognize the gestures of the user. The SMCube can complement other activity recognition techniques and increase the user experience.

1.2 Thesis Organization Outline

The remainder of the dissertation is organized as follows.

Chapter 2 introduces the basic concepts of Ambient Intelligence (AmI). It provides a general definition of the main building blocks and defines the critical factors common to AmI applications. Several example AmI projects are

presented to provide an insight to the current research in this field.

Chapter 3 describes WSNs. The chapter highlights the characteristics of WSN and the main application scenarios. A more detailed description of the building block of a WSN, the *Wireless Sensor Node*, is provided together with an overview of the state of the art of such devices. The chapter is concluded with the description of an emerging standard for WSN wireless protocol: *Zigbee*. This protocol has been developed for low-power, low-cost, low-throughput sensor networks even if recent studies point out the possibility to use it even for streaming applications.

In order to extract information within a WSN several techniques have been proposed in the literature. In 1988 the Joint Directors of Laboratories (JDL) Data Fusion Working Group, began an effort to codify the terminology related to data fusion. Chapter 4 presents the JDL data fusion models and provides the basic theory of several data fusion techniques that will be used in the works presented in this thesis.

Several techniques to track people through the use of a WSN based on simple, low-cost, low-power Pyroelectric InfraRed (PIR) sensors are presented in chapter 5. Moreover the chapter shows how such sensors can be integrated within a video surveillance network to augment its performance and to overcome some limitations of the video systems. Finally, the chapter describe how PIR sensors can be used in conjunction with Wireless Video Sensor Nodes (WVSN) and photovoltaic energy harvesting modules to extend node lifetime.

Thanks to technological advances soon it will be possible to embody a large number of sensor nodes within the environment, the objects that we use and our garments. Chapter 6 presents the work carried out to develop activity recognition techniques in such redundant and dynamic networks. Furthermore within this chapter we present our work in developing smart objects for natural interaction within smart environments.

Chapter 7 concludes the dissertation summarizing the results presented in this thesis.

Chapter 2

Ambient Intelligence

2.1 Ambient Intelligence: general definitions

In the AmI vision, humans will be surrounded by smart devices embedded in everyday objects such as furniture, clothes, vehicles, roads and smart materials. Devices are aware of human presence and activities, take care of his needs and are capable of responding intelligently to spoken or gestured indications of desire. Furthermore they are unobtrusive, often invisible: nowhere unless we need them. Interaction should be relaxing and enjoyable for the citizen, and not involve a steep learning curve [226].

The ISTAG (Information Society Technology Advisory Group) is a team that has been set up to advise the European Commission on the overall strategy to be followed in carrying out the IST thematic priority under the European framework programme for research. The ISTAG reflects and advises on the definition and implementation of a coherent policy for research in ICT in Europe. This policy should ensure the mastering of technology and its applications, and should help strengthen industrial competitiveness and address the main European societal challenges [92].

The first ISTAG meeting took place in 1999 and defined the objective of the group as

start creating an ambient intelligence landscape (for seamless delivery of services and applications) in Europe relying also upon testbeds and open source software, develop user-friendliness, and develop and converge the networking infrastructure in Europe to world-class

— ISTAG, “Orientations for Workprogramme 2000 and beyond”

The ISTAG promotes the creation of pervasive environment improving the

quality of life of the occupants and enhancing the human experience. Such smart, electronic environment are proactive to the presence of people and their activities. *Context awareness* is a key factor of this vision. Computer react based on their environment. Devices collect information about the circumstances under which they operate and react accordingly [178, 179].

Ambient Intelligence stems from the convergence of three key technologies:

Ubiquitous Computing

The vision of ubiquitous computing emerged in the late 80s at Xerox Palo Alto Research Center (PARC) when a heterogeneous group of researcher developed a novel paradigm of interaction between human and computers [224]. The term *ubiquitous computing* has been forged by Mark Weiser few years later [222] and refers to omnipresent computers that serve people in their everyday lives at home and at work, functioning invisibly and unobtrusively in the background and freeing people to a large extent from tedious routine tasks. Ubiquitous computing has as its goal the enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user [223]. The technology required for ubiquitous computing is three-fold: cheap, low-power electronic devices, a network that ties them all together, and software systems implementing ubiquitous applications. Human-smart environment interaction is possible through hand held devices that collect information from the environment or context aware services that are aware of people presence, understand their activities and react in a proactive manner. Some people say that ubiquitous computing is the Third Wave of Computing, where the First Wave was many people, one computer (mainframe), the Second Wave is the era of one person, many computers (Personal Computers). The Third Wave will be the era of many computers per person [2] (see figure 2.1).

Ubiquitous Communication

An important factor to fully exploit the power of ubiquitous system and to provide information everywhere it is needed is the presence of a rich wired and wireless communication infrastructure. Wireless communication is well suited for dynamic environment where the users moves within smart ambients. In order to realize demands for ubiquitous communication and pervasive computing, a change from the traditional approach of centralized, planned wireless communication networks such as GSM, toward an adaptive, self-organizing, multi-user, multi-system distributed wireless communications platform is essential [162] (see figure 2.2). To implement wireless technology on a wide level, however,

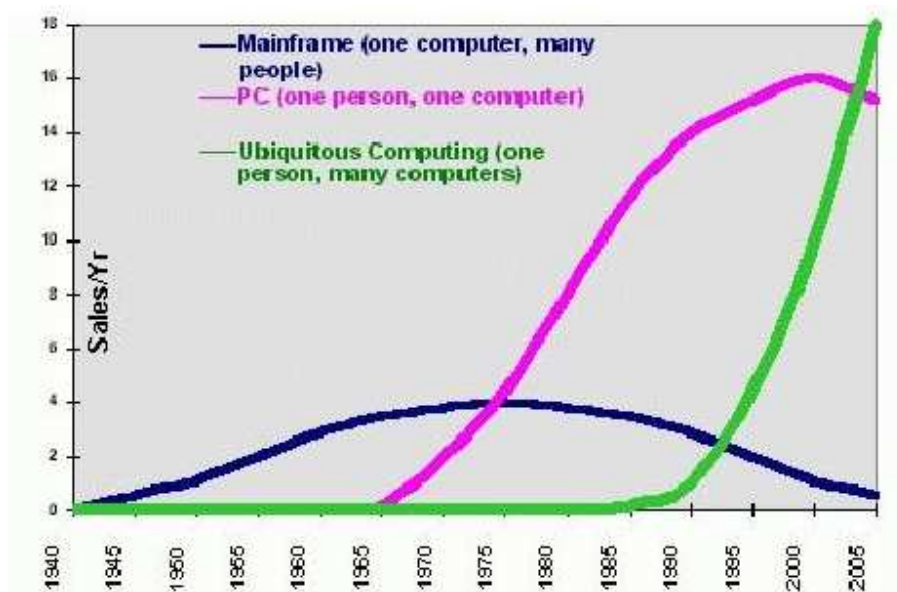


Figure 2.1: Trends in computing

the wireless hardware itself must meet several criteria on the one hand, while easy integration and administration as well as security of the network must be ensured on the other. Some of the unique features that the ambient intelligence scenario presents and that must be considered are: very large networks (hundred or thousands of nodes), both mobile and fixed nodes, node failure must be kept in mind, small battery size (for easier integration) and data centric communication (i.e. redundant data can be aggregated, compressed, dropped etc.). Incorporating these unique features into protocol design is important in order to efficiently utilize the resources of the environment [159].

Intelligent User Friendly Interfaces

Intelligent user interface have a fundamental role in ambient intelligence. These interfaces go beyond the traditional keyboard, mouse, and display paradigm to improve human computer interaction by making it more intuitive, efficient, and secure. Thus, Ubiquitous computing inspires application development that is off the desktop. In addition to suggesting a freedom from well-defined spaces, this vision assumes that physical interaction between humans and computation will be more like the way humans interact with the physical world. Input has moved beyond the explicit nature of textual input (keyboards) and selection (pointing devices) to a greater variety of data types. This has resulted in not only a greater variety of input technologies but also a shift from explicit means



Figure 2.2: Distributed communication network

of human input to more implicit forms of input. Computer interfaces that support more natural human forms of communication (such as handwriting, speech, and gestures) are beginning to supplement traditional interfaces. Intelligent human computer interaction promises to support more sophisticated and natural input and output, to enable users to perform potentially complex tasks more quickly, with greater accuracy, and to improve user satisfaction.

In 2001, two years later the first meeting, the ISTAG group has published a final report where four scenarios are described in order to offer provocative glimpses of futures that can be realized [53]. Each scenario contains positive and negative aspects that allow for a composite, even contrasted, picture of the future.

The analysis of these scenarios allow to identify the critical factors in building AmI systems. The factors are divided into 3 main topics.

Socio-political factors AmI should facilitate human contact and be oriented toward community and cultural enhancement. However to be acceptable AmI should inspire trust and confidence and thus needs to be driven by humanistic concerns, not technological ones since people do not accept everything that is technologically possible and available [154]. A major criticism came from the observation that being immersive, personalized,



Figure 2.3: Intelligent Natural Interfaces (*Photo:Philips*)

context-aware and anticipatory it brings up social, political and cultural concerns about the loss of privacy, the power concentration in large private companies and fear for an increasingly individualized, fragmented society [232]. This criticism should be kept in mind for a widespread acceptance of this new technology.

AmI also should exploit its great potential to enhance education and learning. Everyday life skills will grow because of rising opportunities and means of personal expression and interaction [52].

Business and industrial models Economic aspects of AmI are a fundamental factor for the diffusion of this technology. The most important questions are related to how translate technological and social changes into potential business models. However a number of elements emerged from the scenario that highlight several potentialities of AmI. Among them: enhancements in the productivity and the quality of products and services, comprehensive methods of monitoring and extracting information on real-world, reducing reaction times in unforeseen circumstances, new products and new services.

Technology requirements Five main technology requirements emerge from the analysis of the scenarios [53]:

1. Very unobtrusive hardware. Miniaturization is necessary to achieve dense dissemination of devices and to develop new sensors and

smart materials. In addition self-generating power and micro-power usage will be necessary due to poor scaling capability of batteries technology and new displays and smart surfaces should be developed to provide satisfactory interaction with the environment.

2. A seamless mobile/fixed communications infrastructure. Complex heterogeneous networks need to function and to communicate in a seamless and interoperable way. This implies a complete integration of mobile and fixed and radio and wired networks. Advanced techniques for dynamic network management will be necessary.
3. Dynamic and massively distributed device networks. A huge amount of sensors will be spread in the environment. This networks should be self configurable according to its specific, dynamic status and the current task with variable actors and components. Databases should be accessible on demand from anywhere in the system.
4. Natural feeling human interfaces. The design of novel multimodal, multi-user, and multi purpose interface for speech, gesture, and pattern recognition adaptive to user requirements is required.
5. Dependability and security. Technology should be safe for user both from the physical and psychological point of view. Thus technology should be tested and both hardware and software should be robust. For this reason there is likely to be an emerging emphasis on self-testing and self-organizing systems.

Ambient Intelligence will be brought to us with the promise of an enhanced and more satisfying lifestyle. However, its social benefits cannot be realized unless a number of requirements regarding socio political-issues, business model and technology development have been met. Several field of research will be involved in this change and furthermore novel interdisciplinary approaches will be necessary. Issues such as environmental and social sustainability, privacy, social robustness and fault tolerance will determine the take up of AmI.

2.2 Ambient Intelligence projects

A number of leading technological organizations are exploring pervasive computing apart from Xeroxs Palo Alto Research Center (PARC).

The Laboratory for Computer Science (LCS), the Artificial Intelligence Laboratory (AIL) at the Massachusetts Institute of Technology (MIT) together with several industrial partner have started the project *Oxygen* [142]. The mission of the project is to *bring an abundance of computation and communication within easy reach of humans through natural perceptual interfaces of speech and vision so*

computation blends into peoples lives enabling them to easily do tasks they want to do collaborate, access knowledge, automate routine tasks and their environment. The project focus on *network technologies* to connect dynamically changing configurations of self-identifying mobile and stationary devices to form collaborative regions, on *software technologies* to develop software systems able to adapt to users, to the environment, to change and to failure with minimal user intervention and without interruption to the services they provide, on *perceptual technologies* to build multimodal interaction with the electronic environment, and on *user technologies* for user support.

IBM created a living laboratory, called Planet Blue, to understand how people will interact with the emerging world of the wireless Internet [88]. The applications developed within this laboratory aim at highlight the requirements of the underlying infrastructure needed to support workers. The objective of Planet Blue is to define the future of post-PC personal computing and drive IBM's research in information access devices. The project focus on the development of dynamic personal portals, enhanced Personal Information Management (PIM) and smart meetings.

Carnegie Mellon University has started *Project Aura* that focuses on user attention [26]. The project motivation come from the observation that also user attention is a (limited) resource in a computer system. Aura's goal is to provide each user with an invisible halo of computing and information services that persists regardless of location and support it. Aura's related project includes: distributed real-time object system and interactive media, mobile file access, application-aware networking, wearable computers and cognitive assistance for everyday computing.

The regione Emilia Romagna (IT) founded the project *LAICA* [117](Laboratorio di Ambient Intelligence per una Città Amica - Ambient Intelligence Laboratory for a Friendly City). The objective of this project is to develop ambient intelligence solution at a city level through a set of demonstrators deployed in the city of Reggio Emilia (IT). The project bring together 3 university (University of Bologna, University of Modena and Reggio Emilia and University of Parma) and 6 industrial partners. Information from the demonstrator are collected to a middleware that present them both to public offices, police and citizens with different details.

Chapter 3

Wireless Sensor Networks

3.1 Wireless Sensor Networks overview

Advances in the fields of micro electronics, wireless communication, embedded microprocessors and micro-fabrication allowed the the birth of one of the most rapidly evolving research and development fields: *Wireless Sensor Networks* (WSN) [41, 241]. WSN are complex system consisting of spatially distributed autonomous devices, called *Sensor Nodes*, that collaborate to monitor physical or environmental conditions at different locations. Design, implementation, and deployment of a WSN involves a wide range of disciplines and considerations for numerous application-specific constraints [13]. In the last five years, significant progress has been made in the development of WSNs, and some WSN-based commercial products have already appeared on the market.

Even if WSN are strictly application dependent, it is possible to define a list of basic features [90].

- Self-organizing capabilities.
- Short-range broadcast communication and multihop routing.
- Dense deployment and cooperative effort of sensor nodes.
- Frequently changing topology due to fading and node failures.
- Limitations in energy, transmit power, memory, and computing power.

These characteristics make WSN different from other wireless systems and make them one of the most important enabling technologies for several applications.

3.1.1 Wireless Sensor Network applications

Historically WSNs were developed for military applications [30], however there has been a significant interest also in several other fields of human activities [166]. Following a list of application is discussed.

Military

Being capable of self organization a large number of sensor nodes could be rapidly deployed along defensive perimeter or into battlefields (for example by dropping them from a helicopter as shown in figure 3.1). Once on the field they would establish an ad hoc network and monitor for hostile military units. For example in [131] a wireless network of many low-cost acoustic sensors is used to determine both a snipers's location and the bullet's trajectory. Furthermore even if the loss of some sensors is likely to happen the ability to adapt to a changing topology will not prevent a redundant network to work properly. Clearly, fusing the information from a heterogeneous set of sensors can improve the precision and the number of inferences about the activity going on [81].

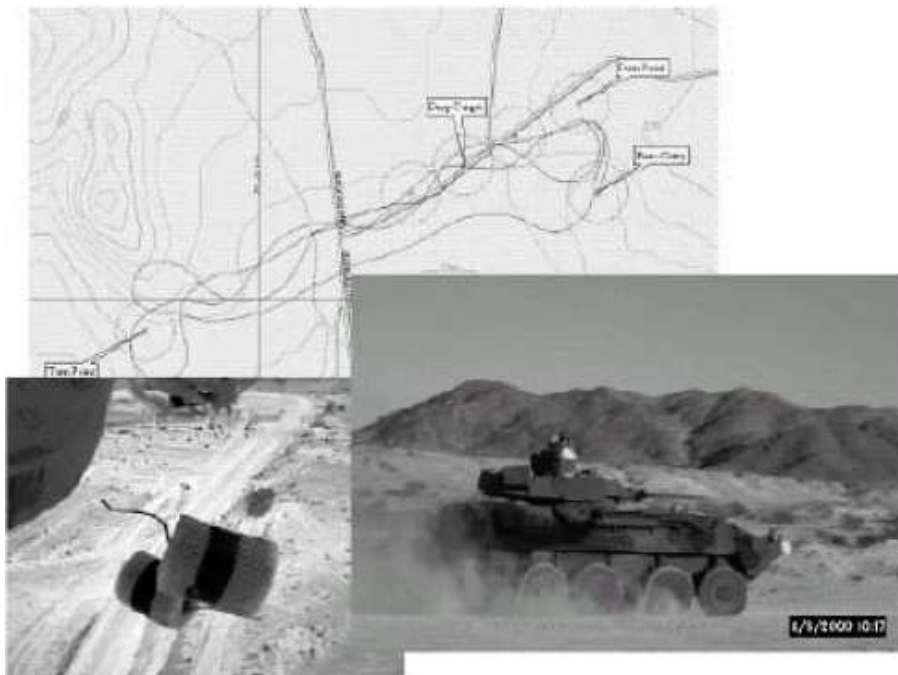


Figure 3.1: WSN Application on battlefield

Environmental and habitat monitoring

WSN have shown to provide an effective means to monitor geographically remote areas. Thanks to the ability of transmit collected data to a

data repository on a server, WSNs have been a great improvement in traditional monitoring systems where data required manual downloading by a maintenance team [132]. Some applications of environment monitoring through WSN include the the Environmental Observation and Forecasting Systems (EOFS) project which is large-scale distributed system designed to monitor, model, and forecast wide-area physical processes such as river systems like the Columbia river estuary [192] and the Sensor Web Project [152] which is a systems used to implement a global surveillance program to study volcanoes. The system uses a network of sensors linked by software and the internet to a satellite and has been designed with a flexible, modular, architecture to facilitate expansion in sensors, customization of trigger conditions, and customization of responses. Examples of WSNs applications for habitat monitoring include the Berkeleys habitat modeling at Great Duck Island [200] (see figure 3.2).

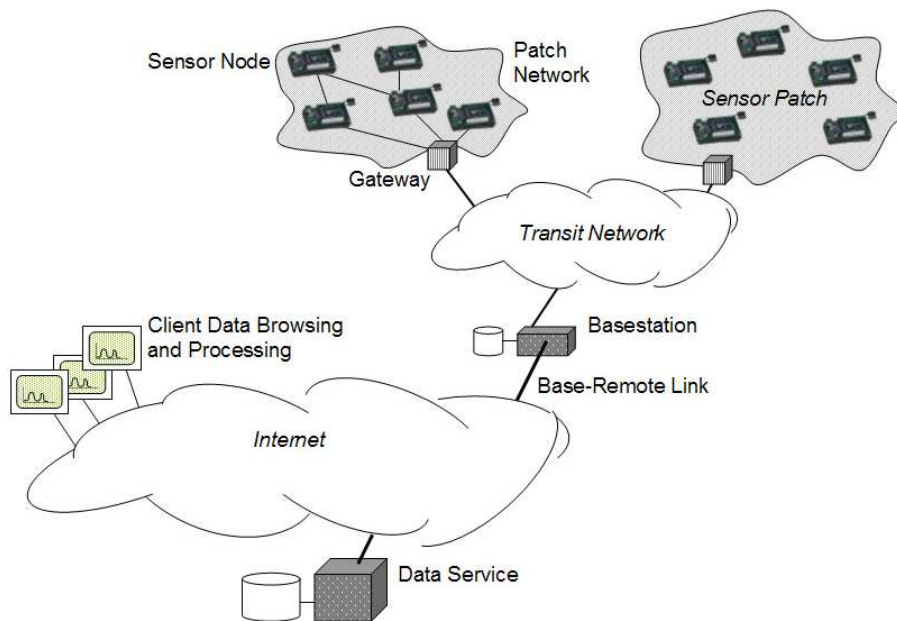


Figure 3.2: Structure of the WSN for habitat monitoring on Great Duck Island

Health care

Patient monitoring systems can be used to collect patient physical status related data at home and, in some cases, in outdoor scenarios, facilitate disease management, diagnosis, prediction and follow-up. Use of WSN can bring great benefit to this activity since the monitoring of people in their natural environments is not practical when it is necessary to use cables to connect the sensors with the processing and communication units

[146]. Some example application includes elderly care [17], post stroke rehabilitation [155] and support of people who suffer of physical disability in order to provide imminent feedbacks when occurs [21] (see 3.3).

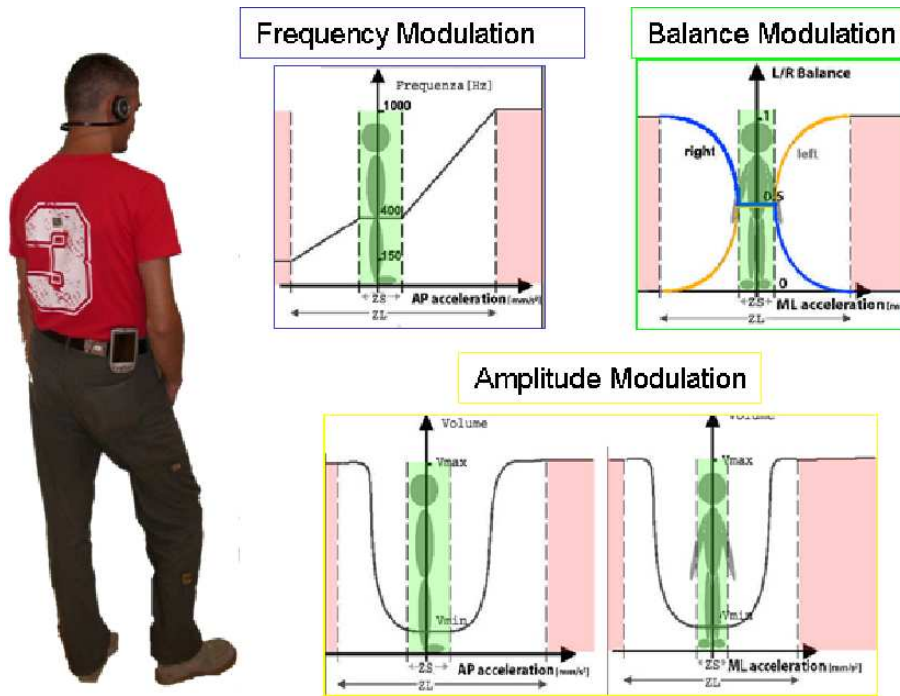


Figure 3.3: Audio bio-feedback for impaired people support

Domotic

Home automation is a field within building automation that focus on the application of automatic techniques for the comfort and security of home residents. The possibility to embed a large number of sensors into everyday objects allow the continuous monitoring of the home status. This results in a more efficient tuning of systems such as the heating, ventilating, and air conditioning (HVAC) and the easy and natural interface with electronic devices [163].

Logistic

Tracking of goods is one of the most important aspect for modern companies. In a globalized world, production process is distributed among several country and many actors take part of it. WSN provide opportunities for the control and management of transport and logistics processes, since sensor nodes can be associates with goods and track their path, who used them and eventually report misuse. An overview of issues and pos-

sible approaches can be found in [61].

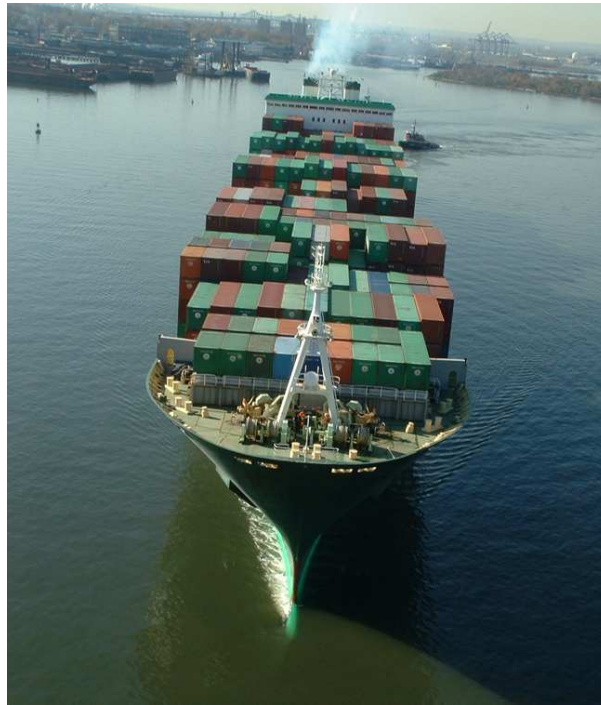


Figure 3.4: WSN can be used for logistic support

Surveillance

As for military application WSN can be used to monitor the access to building, restricted areas and other critical infrastructure such as power and telecoms grids or roads and motorway. Heterogeneous systems that comprise lower-cost sensors, such as presence or acoustic sensors, can support more bulky and expensive sensors such as imagers, in order to provide cost effective and efficient systems. The use of this setup is even more effective if we consider that it is rather difficult for security guards to continuously watch a set of video monitors when most of the time nothing occurs is considered. Thus low-cost sensor can help to focus their attention only where it is necessary [242].

3.2 Wireless Sensor Nodes

WSN basic building blocks are called *Wireless sensor nodes* or *sensor nodes*. A sensor node is a device capable to collect data from one or more sensors, perform some sort of computation with it, than (wirelessly) send this data to other nodes or system for further analysis.

The major characteristics and requirements of a sensor node can be listed in the following [174]:

Low cost

WSN may consist of hundred or thousand of sensor nodes, thus single sensor node cost should be kept low. Also, it is likely that sensor node will be embedded into everyday object, therefore, for a widespread diffusion of sensor network, their cost should not be excessive.

Low cost requirement results in the adoption of low level components such as low power microcontrollers with limited amount of data and program memory available. As a consequence, even if, due to the high number of nodes working in parallel within the network, the overall computational power and memory available to the network can be quite high, single node capabilities are strictly limited. Thus, application for WSN should be made up of many simple tasks done in parallel by the nodes of the network.

Limited size

Sensor nodes will be embedded into the surroundings, into object and even into user garments. For this reason, unobtrusiveness is a critical point in order not to impair normal activities. A consequence of miniaturization is the evolution of sensor nodes from dedicated embedded devices where commercial off the shelf components with emphasis on small form factor, low-power processing and communication, share a common board to system on chip sensor nodes where on a common die coexist an MCU, a wireless transceiver and sensors.

Low power

Power consumption is one of the biggest issues in the design of WSNs. Nodes, typically, are equipped with batteries, thus they have a limited amount of available energy. Often a frequent change of batteries can be unfeasible, specially in large WSN, or can not be possible when, for example, nodes are placed in harsh environment. In many application scenarios, the target node lifetime should be several years long. This imposes drastic constraints on power consumption that can drop down to an average of few tenth of microwatts.

Limited power consumption usually is achieved using low power hardware or performing several trade off between the energy consumption and other network characteristics such as: quality of service, latency, sensing accuracy, reactivity to changes in topology, node size (since batteries do not scale as quickly as integrated circuits).

Another approach is to rely on energy scavenging systems to extend node

lifetime. However energy harvesting, typically, provide a non constant amount of energy that must be carefully managed to assure the desired service.

Wireless

Wireless is a key factor for many applications that rely on mobile nodes, and in order to reduce WSN cost. In fact, sensor nodes, even if fixed, may be placed in environment where communication infrastructure are not present. In this situation the cost of wiring sensor nodes can be too high and result in sensor network rejection.

Scalability and self organization

Wireless sensor nodes should be able to autonomously organize themselves and to adapt to changes in their setup and number. This characteristic is fundamental since often WSN are deployed without a precise control of nodes position (for example, when dropped on battle field) and also because, due to the low cost hardware used, nodes failure can be rather common. For this reason sensor network should be able to provide a graceful degradation as the number of nodes decrease. Furthermore, self organization is necessary where mobile nodes move within different regions and interact with a multitude of different other nodes.

Figure 3.5 presents the system architecture of a generic sensor node which, typically, is made up of four basic building blocks.

- Sensing Unit.
- Computational Unit.
- Communication Unit.
- Power Unit.

An example of wireless sensor node is presented in figure 3.6 [65].

3.2.1 Computational Unit

Sensor nodes should collect data from the environment, process it and communicate. For this reason a central processing unit is needed. The CPU should be able to manage the sensor node activity while meeting the energy consumption, size and cost constraints. There are a large number of available microcontroller, microprocessors and FPGA that can be integrated within sensor nodes, which allow a high degree of flexibility [213, 1].

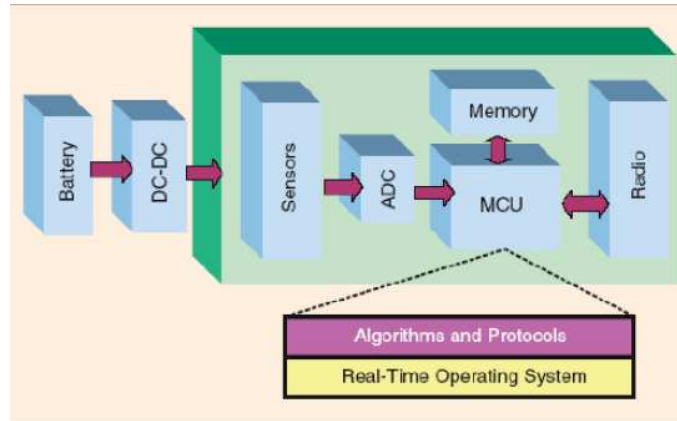


Figure 3.5: Generic architecture of a sensor node

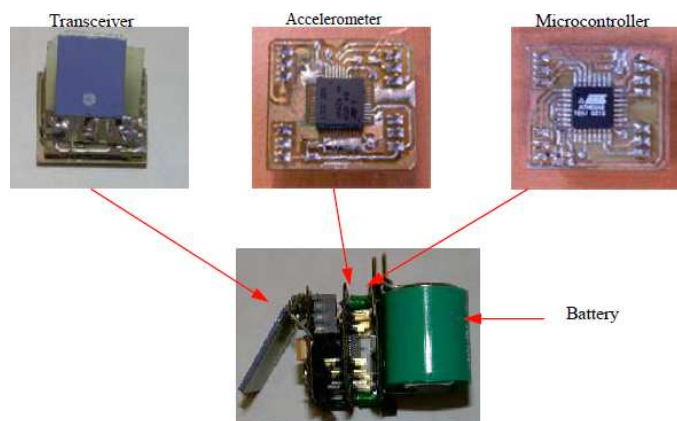


Figure 3.6: WiMoCA wireless sensor node

Microcontrollers

Nowadays, microcontroller includes a sufficient amount of memory and enough computational power to iterate with sensors and communication devices such as short-range radio to compose a sensor node. Furthermore they provide non-volatile memory for data storage and several other devices such as: ADC, UART, SPI, counters and timers.

There are many types of microcontrollers, ranging from 4 to 32 bits, varying the number of timers, bits of ADC and power consumption. In particular they provide several different operating modes that allow to save energy when the sensor node is idle.

FPGA

Field Programmable Gate Array (FPGA) presents some disadvantages with respect to microcontrollers. The most important is related to power consumption, which is not as low as microcontrollers one. However the development of ultra low power FPGA can make these devices a suitable solution for sensor node.

3.2.2 Sensor and Actuator Unit

A *sensor* is a device that converts a physical phenomenon into an electrical signal. On the other hand, an *actuator* convert an electrical signal into physical phenomena. The first decade of the 21st century has been called as the "Sensor Decade" for the dramatic increase in sensor R&D over the past years [229]. Sensors are used to measure various physical properties sch as temperature, force, pressure, flow, position, light intensity, acceleration, incident infrared radiation, etc. [182].

Sensors may be classified in a number of ways. One useful way is to classify sensors either as active or passive. The former require an external source of power, thus they consume power even when nothing is detected. The latter generate their electrical output signal without requiring external voltage or current. A list of popular sensors is presented in table 3.1.

Most sensors require an output conditioning circuit to amplify and filter their output in order to be processed by a microcontroller. Typical sensor conditioning circuits include amplifier, filtering, level translation, impedance transformation.

3.2.3 Communication Unit

The wireless communication channel enables to transfer signals from sensors to exterior world, and also an internal mechanism of communication to establish and maintain of WSN. This medium needs to be bidirectional, to be

| Property | Sensor | Active/Passive | Output |
|--------------------|-----------------------|----------------|-----------------|
| Temperature | Thermocouple | Passive | Voltage |
| | Silicon | Active | Voltage/Current |
| | Thermistor | Active | Resistance |
| Force/Pressure | Strain Gage | Active | Resistance |
| | Piezoelectric | Passive | Voltage |
| Accelerometer | Accelerometer | Active | Capacitance |
| Infrared radiation | Pyroelectric InfraRed | Passive | Voltage/Current |
| light intensity | Photodiode | Passive | Current |

Table 3.1: Popular sensors and their output.

energy-efficient, and have relatively slow data rate. Two basic techniques are used: optical communication and radio frequency communication [214].

Optical communication

Two main technologies are available for optical communication: laser and infrared.

Laser communication consumes less energy than RF over larger range, is secure, since upon interception the signal is interrupted, and do not need antennas. However it requires line of sight and alignment between transmitter and receiver and this is a major drawback since several applications presents randomly deployed nodes.

Also infrared is directional and requires line of sight between 2 communicating nodes. It allows only short range (less than 10 meters), but do not require antennas. An interesting solution is presented with the PushPin project [122] in order to achieve omni-directional infrared communication on a single plane.

Radio frequency communication

Based on electromagnetic waves, one of the most important challenges for this typology of communication is antenna design and size. However RF communication present several advantages. It is easy to use, to integrate and it is a well established technology. Power consumption of RF communication is affected by type of modulation, data rate and transmission power. An important aspect to consider when working with RF transceiver is that idle state (radio active but not transmitting, nether receiving) drawn as much current as receive mode. Thus wireless protocols must reduce as much as possible this waste of energy.

3.2.4 Power Unit

Power supply unit usually consists of a battery and a dc-dc converter. Thus, the power needs of large wireless sensors network (maybe deployed in harsh

environment) is the current biggest impediment that keeps them from becoming completely autonomous, forcing them to be either connected to an external power source or have lifecycles that are curtailed by batteries. Furthermore, in some application like gesture recognition, where sensor are embedded into user garments, battery size is the most relevant factor when seeking unobtrusiveness since battery technology tends to be a limiting factor in miniaturization [157].

For this reason in the last years, energy harvesting has emerged as one alternative to provide perpetual power solution to sensor network.

3.3 State of the art

In this section a we present a series of commercial and academic solutions of wireless sensor nodes and their main features.

3.3.1 Smart Dust

The goal of the Smart Dust project, founded by DARPA (Defense Advanced Research Projects Agency), is to demonstrate that a complete sensor communication system can be integrated into a cubic millimeter package. This involves both evolutionary and revolutionary advances in miniaturization, integration, and energy management [15, 220]. A conceptual diagram of a Smart Dust mote is presented in figure 3.7.

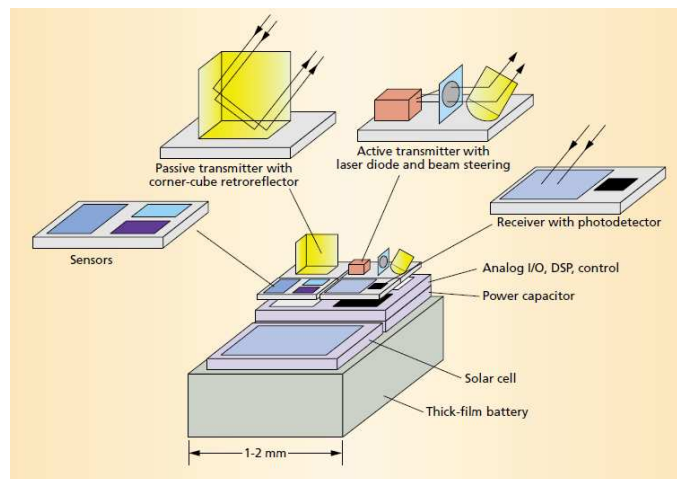


Figure 3.7: A diagram of the Smart Dust mote

Many sensors, including temperature, pressure, and acceleration sensors, from MEMS and CMOS processes can be attached to a mote. In contrast to

typical computing systems, in an autonomous cubic-millimeter package computation must focus on minimizing a given tasks energy consumption. This is achieved through frequency and voltage scaling, since the computation requirement for this motes are limited. Communication is possible by means of two approaches: passive reflective systems between nodes and the base stations and active steered laser systems between motes. The power system consists either of a thick-film battery, or a solar cell with a charge-integrating capacitor for periods of darkness, or both.

3.3.2 Intel mote

The Intel Mote is a new sensor node platform motivated by several design goals: increased CPU performance for data compression as well as initial classification and analysis, improved radio bandwidth and reliability, and the usage of commercial off-the-shelf components in order to maintain cost-effectiveness. An important aspect of the platform design was to increase performance while preserve battery life. To satisfy these requirements, Intel chose a system on chip from Zeevo Inc. including a CMOS Bluetooth radio and an ARM7TDMI core operating at 12MHz and with 64KB SRAM and 512KB FLASH [151].

The Intel Mote is built on a 3×3 cm circuit board that integrates the Zeevo module, a surface-mount 2.4GHz antenna, various digital I/O options using stackable connectors and a multi-color status LED (see figure 3.8).

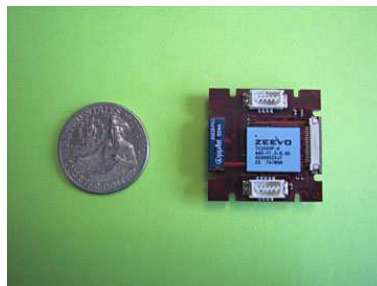


Figure 3.8: The intel mote

Intel second generation of sensor nodes are the Intel Mote 2. This motes are based on an Intel PXA270 XScale CPU with 32 MB of flash and 32 MB of SDRAM resulting in high performance processing capabilities. The processor integrates a DSP co processor, a security co processor and an expanded set of I/O interfaces. The platform also provides an on-board 802.15.4 radio and the option to add other wireless standards such as Bluetooth and 802.11b via an SDIO interface. The complete platform is hosted on a single 36×48 mm printed circuit board [109, 184](see figure 3.9).



Figure 3.9: The intel mote 2

3.3.3 Mica Mote

MICA Motes (see figure 3.10), developed by UC Berkeley research group on wireless sensors, is a mote module used for research and development of low power, wireless, sensor networks. The motes measures 3.16×6.35 cm and is created using off-the-shelf hardware, but the architecture and its capabilities could be implemented in just a few square millimeters of custom silicon. The main microcontroller is an Atmel ATMEGA128 running at 4MHz with 128kB of FLASH and 4kB of RAM. The radio module is based on an RF TR1000 transceiver operating at 916.5 MHz. Several sensor extension board can be connected to the base board, such as: thermal temperature, barometric pressure, magnetic fields, light, passive infrared, acceleration, vibration, and acoustics [84].



Figure 3.10: The Mica mote

An evolution of the Mica motes are the Mica2 mote [34] and the the MICAz [35] mote from Crossbow [36]. The latter, in particular, is a 2.4 GHz, IEEE 802.15.4/ZigBee, board used for low-power, wireless, sensor networks.

3.3.4 Tmote Sky

Tmote Sky [183] is an ultra low power wireless module for use in sensor networks, monitoring applications, and rapid application prototyping. On a sin-

gle $3,22 \times 6.55$ cm board it integrates an ultra low power microcontroller (MSP430 from TI), sensors (Humidity, temperature and light sensors), a Zig-
bee compliant radio (CC2420 from Chipcon), antenna and programming capabilities (see figure 3.11). Tmotesky offers a robust solution with hardware protected external 1MB flash, in the event of a malfunctioning program, the module loads a protected image from flash to restore proper operation.



Figure 3.11: The Tmote Sky mote

3.3.5 BT Node

The BTnode (see figure 3.12) is an autonomous wireless sensor platform developed at ETH Zurich by the Computer Engineering and Networks Laboratory (TIK) and the Research Group for Distributed Systems [60]. The mote is based on a Bluetooth radio and a microcontroller. It serves as a demonstration platform for research in mobile and ad-hoc connected networks (MANETs) and distributed sensor networks. Currently the latest version is revision 3 which includes a core CPU Atmel ATmega128L with 4kByte EEPROM, 64kByte SRAM, 128kByte Flash and a dual radio device composed of a Zeevo ZV4002 Bluetooth radio and a low power Chipcon CC1000 radio operating at 868 MHz. The BTnode rev3 is compatible to the old BTnode rev2 and the Berkeley Motes. This twin device can operate both radios simultaneously or shut them down independently when not in use.

3.3.6 System on chip

One of the main limitations of the platforms presented in the previous sections is that they are built using commodity chips, which themselves are not specifically designed for wireless sensor network applications. As a result, they suffer several inefficiencies that lead to limited functional capabilities, high power consumption, and limited operational lifetimes [64]. A breakthrough innovation happened when the whole sensor node has been integrated on a single chip. In the following sections we present the solutions proposed by 2 Original Equipment Manufacturers (OEM).



Figure 3.12: The BTnode mote

Freescal solutions

With the mission of making the world a smarter place with leading embedded semiconductor solutions for cars, mobile phones, networks and many more, Freescale is a leading company that develops and produces electronic devices for many applications: automotive, computer networks, communications infrastructure, office buildings, factories, industrial equipment, tools, mobile phones, home appliances and everyday consumer products. Freescale has joined the Zigbee alliance in 2004 as a promoter and, since then it has developed several solutions for Zigbee.

In particular 2 system on chips have been developed for WSN.

MC1322x Platform in a Package (PiP)

The MC1322xV [68] is Freescale's third-generation ZigBee platform which incorporates a complete, low power, 2.4 GHz radio frequency transceiver, 32-bit ARM7 core based MCU, hardware acceleration for both the IEEE 802.15.4 MAC and AES security, and a full set of MCU peripherals into a 9.5×9.5 mm Platform-in-Package (PiP). The MC13224V solution includes a fully functional 32-bit TDMI ARM7 processor, 128KB FLASH, 96 KB RAM and, 80K ROM containing boot code, all device drivers and fully compliant IEEE 802.15.4 MAC. Typical power consumption is 21mA in Rx mode and 29mA in Tx mode and drops to less than $1\mu\text{A}$ in stop mode. This device can be used for wireless applications ranging from simple proprietary point-to-point connectivity to complete ZigBee mesh networking in order to provide a highly integrated, total solution, with premier processing capabilities and very low power consumption.

MC1321x System in Package (SiP)

The MC1321x family is Freescale's second-generation ZigBee platform which incorporates an 8 bit MCU (MC9S08GT) with a Zigbee compliant transceiver (MC1320x) into a single 9×9 mm package [68]. The MC13213

provides 60 K Flash memory and 4 K of RAM and can operate at up to 40MHz. It consumes 35mA in Tx mode and 42mA in Rx mode when the MCU operates at 16MHz. By using the IEEE 802.15.4 Compliant MAC, or BeeStack ZigBee Protocol Stack, the MC1321x solution can be used for wireless applications from simple proprietary point-to-point connectivity to a complete ZigBee mesh network.

Ember solutions

Ember's mission is to be the leading provider of wireless sensor and control network technologies that enable dramatic energy efficiency improvements for businesses, homes, and the utilities that serve them. For this reason Ember joined the Zigbee Alliance in 2003 as a promoter and developed several devices and tools to develop Zigbee based applications [57].

Since 2005 ember produces the SN250, system on chip for Zigbee based WSN. The EM250 combines a 2.4GHz IEEE 802.15.4 compliant radio transceiver with a 16-bit microprocessor with 128kB Flash and 5kB RAM in a 7×7 mm package. Requiring 28mA in RX mode and 24 in TX mode and being able to drop power consumption down to $1\mu\text{A}$, it is optimized for designs requiring long battery life and low external component count.

3.4 Protocols for sensor networks

Typically a sensor node implements radio frequency communication. In fact, the development of novel protocols and standard for wireless sensor network is a very active field of research [46, 10].

Within the work of this thesis we focused on the development of Zigbee based WSN. In this section we present this protocol.

3.4.1 Zigbee motivations

The Zigbee alliance [243] has been created in 2002 to meet markets which require longer battery life, lower data rates and less complexity than available from existing wireless standards (see figure 3.13). At that time, for such wireless application a standard has been developed by the IEEE. The IEEE 802.15 TG4 was chartered to investigate a low data rate solution with multi-month to multi-year battery life, very low complexity and operating in an unlicensed, international frequency band [89].

As shown in figure 3.14, the scope of the task group 4 is to define the physical layer (PHY) and the media access controller (MAC) of the protocol, upon which lay the upper layers defined by the Zigbee Alliance.

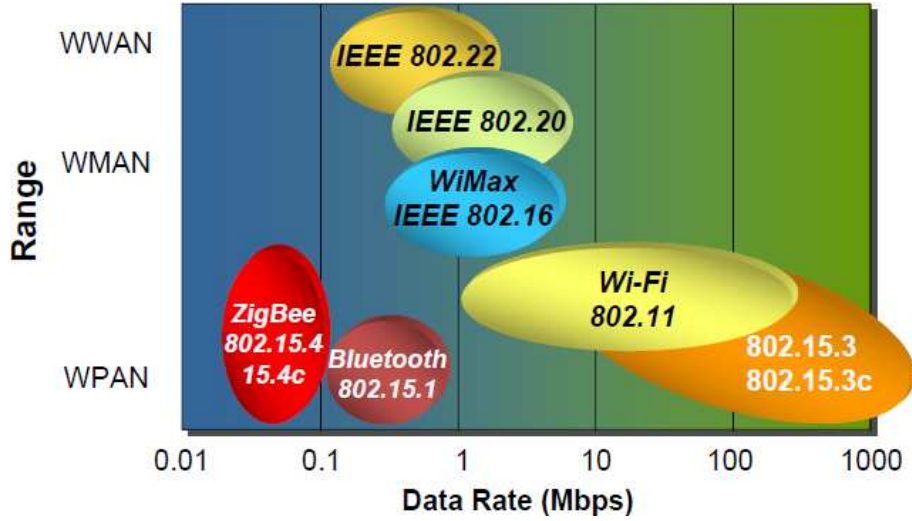


Figure 3.13: Comparison of Zigbee and other standards

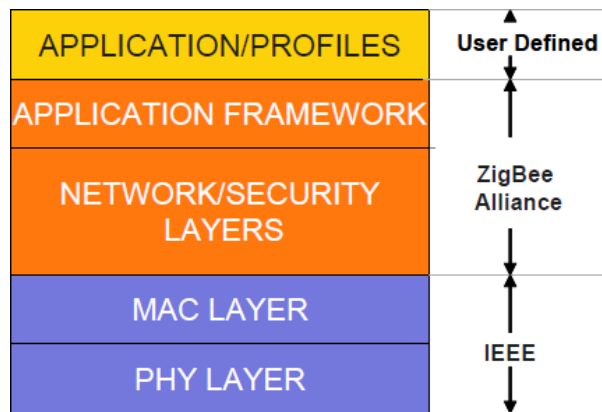


Figure 3.14: Zigbee Stack development contribution

The Zigbee protocol addresses those application that rely on large autonomous networks that needs to operate reliably for years without any operator intervention. For such application power consumption is one of the most important constraints together with cost reduction (both low cost devices and low cost setup and maintenance), while data rate and QoS has are less relevant. Furthermore being a non proprietary solutions it emphasize multivendor interoperability.

A list of possible applications is presented in figure 3.15.

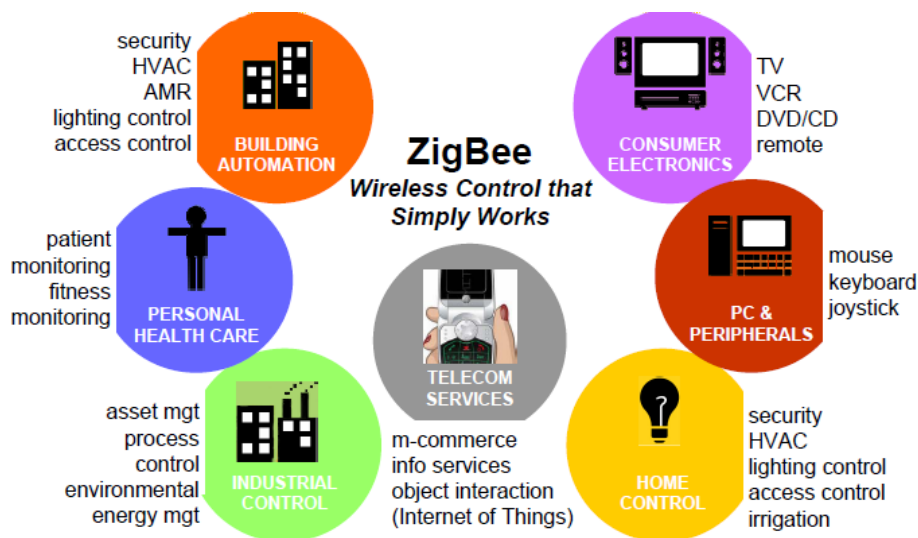


Figure 3.15: Zibee application range

3.4.2 IEEE 802.15.4 Overview

A low rate wireless personal area network (LR-WPAN) is a simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. The main objectives of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol.

IEEE Standard 802.15.4 defines the PHY layer and MAC sublayer specifications for LR-WPAN with fixed, portable, and moving devices with no battery or very limited battery consumption requirements typically operating in the personal operating space (POS).

The first version of the protocol has been ratified in May 2003. A revision process was then initiated to incorporate additional features and enhancements as well as some simplifications to the 2003 edition of this standard. Since

May 2004 the TG4 put itself into hibernation, thus the revision was completed by the new task group 4b that ratified the 2006 version of the protocol in June 2006. Following a successive amendment published in June 2007 has added the specification of 2 new PHY layers: the Ultra-wide band (UWB) PHY at frequencies of 3 GHz to 5 GHz, 6 GHz to 10 GHz with data rates ranging from 110 kb/s to 27.24 Mb/s, and Chirp spread spectrum (CSS) PHY at 2450 MHz with data rate of 250 kb/s and 1000 kb/s. In the following sections we will focus on the 2006 specifications.

Two different device types can participate in an IEEE 802.15.4 network: a full-function device (FFD) and a reduced-function device (RFD). The former has more capabilities than the latter being able to communicate to all other devices in its range. RFD, instead, can communicate and may be associated only with a single FFD. Consequently, the RFD can be implemented using minimal resources and memory capacity.

Depending on the application requirements, a network may operate in either of two topologies: the star topology or the peer-to-peer (mesh) topology (see figure 3.16). The first defines a structure where the communication is established between devices and a single central controller, called the *PAN coordinator*. A PAN coordinator is used to initiate, terminate, or route communication around the network and is the primary controller of the PAN. Mesh networks also have a PAN coordinator; however, in these any device may communicate with any other device as long as they are in range. Up to 2^{32} devices can coexist in the same network

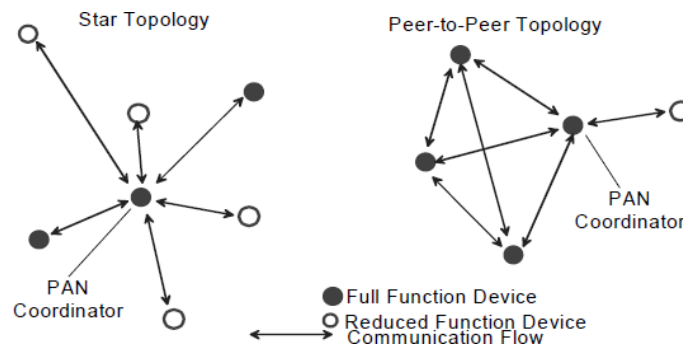


Figure 3.16: IEEE 802.15.4 Network topologies

3.4.3 Physical Layer

The PHY layer is responsible for the following tasks:

- Activation and deactivation of the radio transceiver.

- Energy detection (ED) within the current channel.
- Link quality indicator (LQI) for received packets, which represent the strength of the received signal.
- Clear channel assessment (CCA) for carrier sense multiple access with collision avoidance (CSMA-CA).
- Channel frequency selection.
- Data transmission and reception

A compliant device shall operate in one or several frequency bands using the modulation and spreading formats listed in table 3.2. A total of 32 channels are available to set up a network.

| PHY (MHz) | Frequency band (MHz) | Spreading parameters | | Data parameters | | |
|--------------------|----------------------|----------------------|------------|-----------------|-------------------------|-------------------|
| | | Chip rate (kchip/s) | Modulation | Bit rate (kb/s) | Symbol rate (ksymbol/s) | Symbols |
| 868/915 | 868–868.6 | 300 | BPSK | 20 | 20 | Binary |
| | 902–928 | 600 | BPSK | 40 | 40 | Binary |
| 868/915 (optional) | 868–868.6 | 400 | ASK | 250 | 12.5 | 20-bit PSSS |
| | 902–928 | 1600 | ASK | 250 | 50 | 5-bit PSSS |
| 868/915 (optional) | 868–868.6 | 400 | O-QPSK | 100 | 25 | 16-ary Orthogonal |
| | 902–928 | 1000 | O-QPSK | 250 | 62.5 | 16-ary Orthogonal |
| 2450 | 2400–2483.5 | 2000 | O-QPSK | 250 | 62.5 | 16-ary Orthogonal |

Table 3.2: IEEE 802.15.4 frequency bands, modulation and spreading.

The receiver ED measurement is intended for use by a network layer as part of a channel selection algorithm. It is an estimate of the received signal power within the bandwidth of the channel. No attempt is made to identify or decode signals on the channel. Once the coordinator has detected the energy on every channels, it start the network on the channel with less energy detected to limit interferences.

Prior to any transmission a node should perform a CCA to avoid collision with other communications. CCA algorithm should return an indication of channel busy according to the selected CCA modality.

- *Mode 1* Energy above threshold.
- *Mode 2* Carrier sense only. CCA shall report a busy medium only upon the detection of any signal compliant with this standard with the same modulation and spreading characteristics of the PHY that is currently in use by the device.

- *Mode 3* Carrier sense with energy above threshold. CCA shall report a busy medium if a signal with the modulation and spreading characteristics of this standard is detected and it has an energy above a threshold.

3.4.4 Media Access Controller (MAC) Layer

The MAC layer handles all access to the physical radio channel. This standard allows the optional use of a superframe structure defined at MAC level. The format of the superframe is defined by the coordinator, an example is presented in figure 3.17. From figure 3.17, we can define the structure of the superframe.

- *Beacon*. The beacon is a special packet issued by the coordinator of the PAN used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes.
- *Contention Access Period*. The Contention Access Period (CAP) starts immediately following the beacon. Any device wishing to communicate during the contention access period (CAP) competes with other devices.
- *Contention Free Period*. The optional Contention Free Period (CFP) is used to guarantee specific data bandwidth in low latency applications. This period is divided into several Guarantee Time Slots (GTS) that, upon request, can be reserved by specific devices.
- *Inactive*. The last part of the superframe is communication free. During this time, devices can turn off their radio to save energy.

Networks that do not use a superframe structure, require either to implement ad hoc synchronization techniques or to keep some node always in RX mode to avoid message loss.

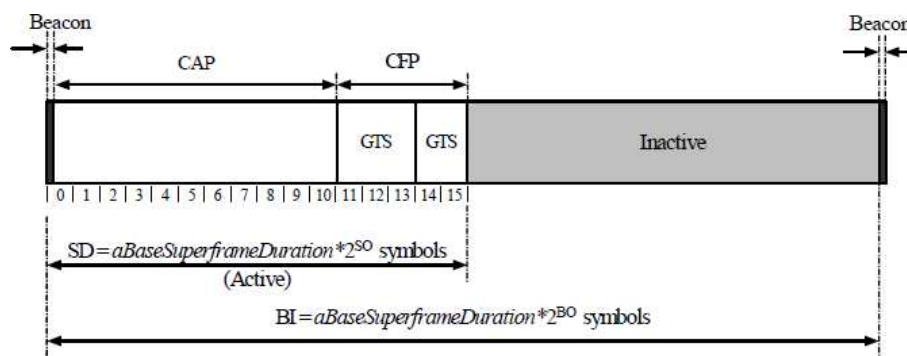


Figure 3.17: IEEE 802.15.4 superframe structure

MAC layer is responsible for the following tasks:

- Generating network beacons if the device is a coordinator
- Synchronizing to network beacons
- Supporting PAN association and disassociation
- Supporting device security
- Employing the CSMA-CA mechanism for channel access
- Handling and maintaining the GTS mechanism
- Providing a reliable link between two peer MAC entities

At MAC level three types of data transfer transactions exist, according to the structure of the network (with or without beacon).

In a beacon-enabled PAN, data transfer are synchronized through the beacons. A device wishing to transfer data to a coordinator first listens for the network beacon in order to synchronize to the superframe structure, then transmits its data frame during the appropriate period. An optional acknowledgment packet can be sent to acknowledge successful data reception (see figure 3.18). Communication between coordinator and associated device is indicated in the beacon message. When the device receives the beacon it knows if there are pending packets that have to be received, then it transmits a MAC command requesting the data, and wait for the packet from the coordinator (see figure 3.19).

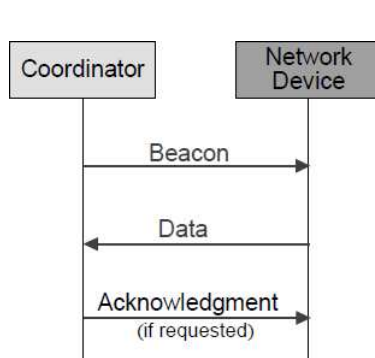


Figure 3.18: Data transfer between a device and the coordinator in a beacon enabled network

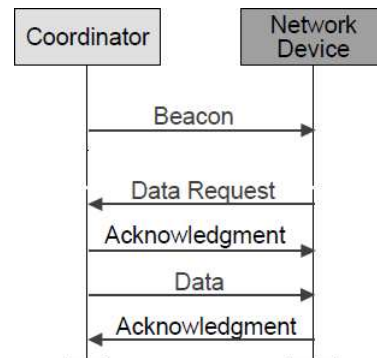


Figure 3.19: Data transfer between the coordinator and a device in a beacon enabled network

In not beacon enabled PAN, it is assumed that either an ad hoc synchronization techniques is implemented or the coordinator is always listening. In the latter case, a device wishing to transmit a message, once performed a CCA, simply send it, the coordinator eventually acknowledge the correct reception

of data (see figure 3.20). In the opposite direction the communication is device driven, thus is the device that periodically poll the coordinator for pending messages. If any packet is pending the coordinator either do not acknowledge the request or send a packet with payload equal to zero, otherwise, once acknowledged the request, it send the data and eventually wait for an acknowledgment frame (see figure 3.21).

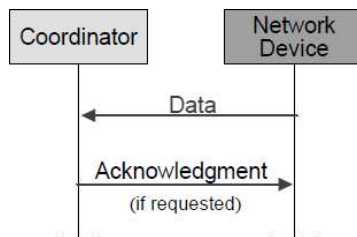


Figure 3.20: Data transfer between a device and the coordinator in a non beacon enabled network

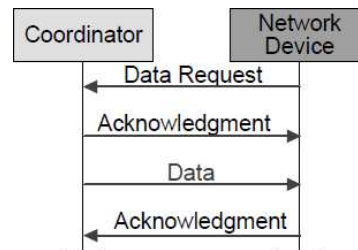


Figure 3.21: Data transfer between the coordinator and a device in a non beacon enabled network

3.4.5 Zigbee

The Zigbee Alliance specified the upper layers of the Zigbee standard. A complete overview of the structure of the protocol is presented in figure 3.22.

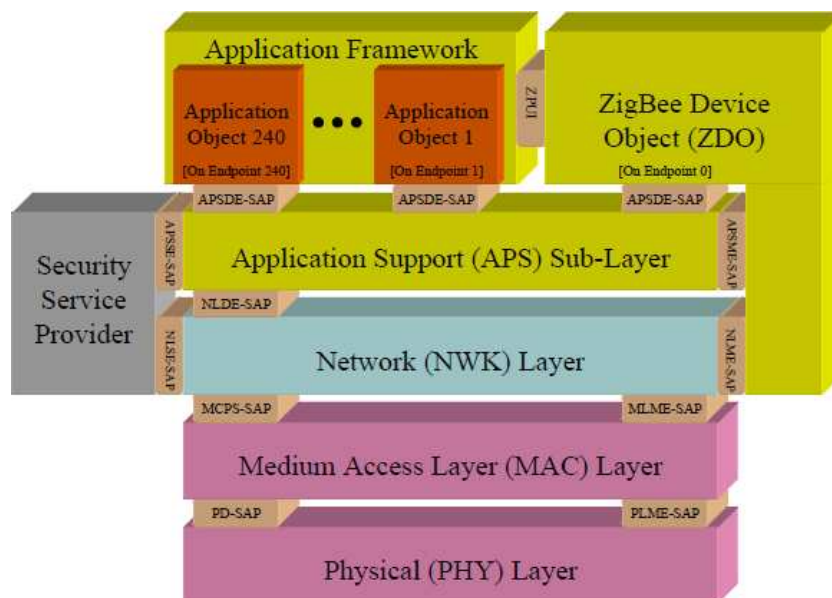


Figure 3.22: Zigbee Stack overview

The first version of the Zigbee stack was released in 2004 and supported

only one profile: home control lighting. This stack was never extensively deployed with customers and is no longer supported. Two years later the second version of the stack (Zigbee 2006, the one discussed below) has been ratified. This version supports a single stack known as the Zigbee stack. The latest version of the stack, Zigbee 2007, has been released in 2007 and includes 2 stacks: the Zigbee stack and the Zigbee Pro stack [58].

The Zigbee and Zigbee Pro stacks are complete implementations of the networking layer, security services and the application framework. Devices implementing Zigbee and Zigbee Pro can interoperate by acting as end devices in the other type of network, that is: in a Zigbee Pro network, Zigbee devices can only operate as end devices and vice versa. The reason is that the 2 stacks use different addressing techniques: the Zigbee uses a tree addressing while the Zigbee Pro uses a stochastic addressing.

Since the Zigbee protocol is based on the IEEE 802.15.4 MAC and PHY layers it rely on two device types too: FFD and RFD (see session 3.4.2). Furthermore it defines three possible roles: network coordinator, router, end device. Up to 2^{16} devices can coexist in the same network.

As for IEEE 802.15.4 the coordinator of the network is unique and is responsible for initiating and maintaining the devices on the network. Routers are FFD that have the ability to communicate with all other routers within their communication range and to move data and control messages through the network. End devices are low power low cost nodes that can communicate only with their parent router/coordinator. While, usually the coordinator and the routers are main powered, end devices are battery powered.

The specification allows three topologies of network: Star (with a single coordinator as the central node), cluster tree (where end device are the leaf of the tree) and mesh (typically composed only of routers and the coordinator) (see figure 3.23).

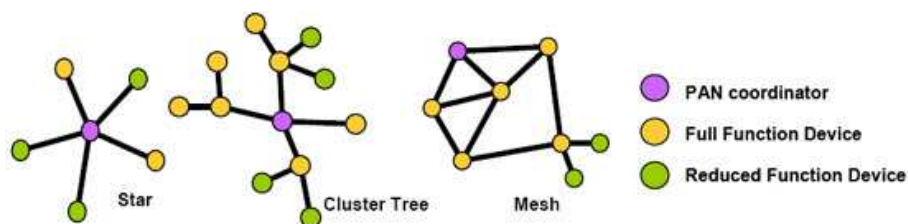


Figure 3.23: Zigbee Network topologies

The Zigbee specification defines the following layers.

- *Network Layer.* The network layer (NWL) is responsible of: join and leave a network, route frames to their destinations, discover and main-

tain routes between devices, discover neighbors, store neighbor information, apply security to packets.

- *Application Layer*. The application layer (APL) is made up of 3 parts: *Application Support Sublayer (APS)*, *Zigbee Device Object (ZDO)* and the user specific applications (that are hosted within the *Application Framework, AF*). The former provide a common interface to user applications in order to access all zigbee services. It is responsible of: maintaining binding tables (which are tables that contains the id of devices that match based on their services and their needs), forwarding messages between bound devices and fragmentation, reassembly and reliable data transport. The ZDO is a standard application built over the APS that is in charge to setup and manage the device role within the network. Its duties are: define the role of the device within the network (coordinator, router, end device), handle binding requests, discovering devices on the network, establish secure connections between devices.

Following a more detailed description of the two layers.

Network Layer

The network layer is required to provide functionality to ensure correct operation of the IEEE 802.15.4-2003 MAC sub-layer and to provide a suitable service interface to the application layer.

To all devices it should allow to join and leave the network, while only router and the coordinator can permit devices to join and leave the network, participate in assignment of logical network addresses and maintain a list of neighboring devices.

A node wishing to associate to the network should perform a join request to a router or the coordinator of the network. Upon acceptance it will be assigned a unique network address. The ZigBee stack uses tree addressing. This address is calculated according to three constants shared by all nodes of the network that have routing capabilities.

1. *Max Depth (MD)*. This parameter specify the maximum allowed distance (in hop) between a device and the coordinator of the network. Each device will keep track of its depth. Coordinator have depth 0 and the maximum depth is MD-1.
2. *Max Children (MC)*. This parameter specify how many node can be associate to a router or to the coordinator.
3. *Max Router (MR)*. This parameter specify how many associated devices can have routing capabilities.

These constants are used to build a vector called *CSkip table* whose elements are calculated as follow [118].

$$CSkip(d) = \begin{cases} 1 + MC \cdot (MD - d - 1), & MR = 1 \\ \frac{1 + MC - MR - MC \cdot MR^{MD-d-1}}{1 - MR}, & MR \neq 1 \end{cases}$$

Each device with routing capabilities will have its CSkip value according to its depth. With this value, when it receives an association request that can be accepted (i.e. there is enough room for this router/end device) it assigns the new address in a sequential manner according to the following

$$\begin{cases} A = A_{parent} + n \cdot CSkip(d), & \text{If device is a Router} \\ A = A_{parent} + n, & \text{if device is not a Router} \end{cases}$$

Where A is the address of the new device, A_{parent} is the address of the device that receives the association request, and n is the sequential order of the request (i.e. the n_{th} router or end device that ask for association). A schematic example with $Maxdepth = 3$, $MaxChildren = 2$, and $MaxRouter = 2$ is presented in figure 3.24.

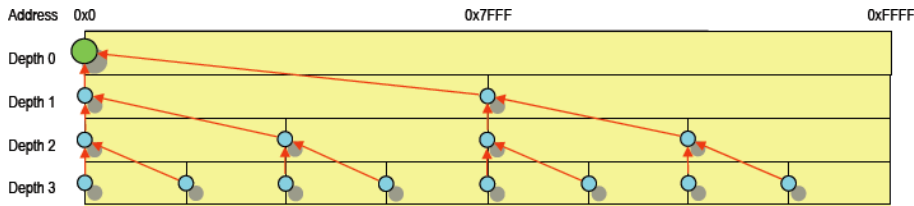


Figure 3.24: Example of Zigbee address allocation

Packet routing can be done in one of the three following ways.

Tree Routing

The destination address tells you where the destination is located within the association tree. In fact given the local address and the destination address it is possible to define a simple mechanism that route the packet up or down the tree branches. If the destination address is larger than the local address and smaller than the local address plus the value of the CSkip value of the local node level than the packet should be routed down. Otherwise it should be routed up. Figure 3.25 present this routing technique.

Neighbour Routing

The coordinator and Routers maintain a table with the addresses of the devices within its range of communication. These tables are used to send

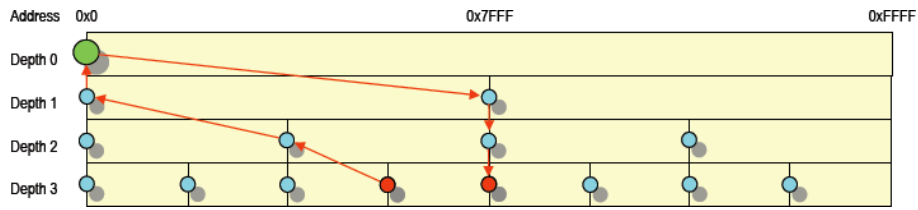


Figure 3.25: Example of Zigbee tree routing

messages directly to the destination (see figure 3.26)

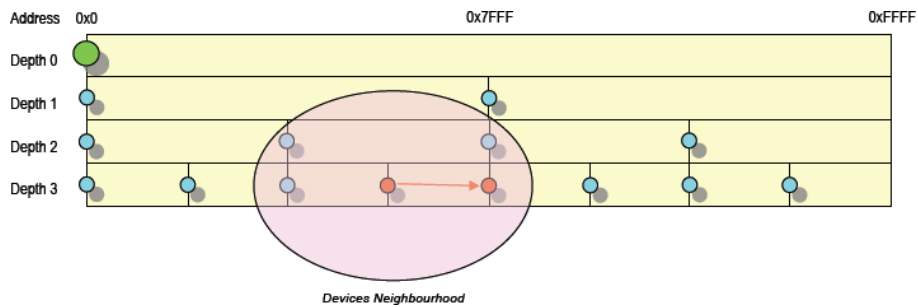


Figure 3.26: Example of Zigbee neighbor routing

Mesh Routing

The coordinator and Routers maintain the routing table of next hop neighbor. If neighbor devices have an entry for the destination the message can be forwarded through it (see figure 3.27)

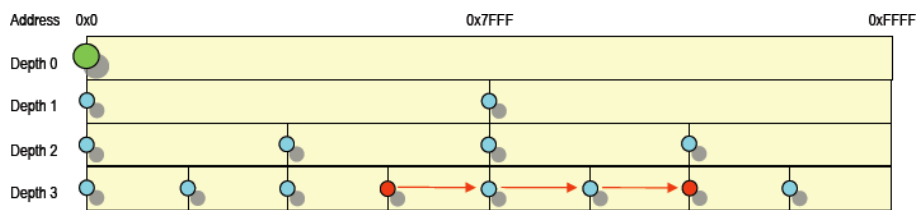


Figure 3.27: Example of Zigbee mesh routing

Usually the tree routing is used when both the neighbor and the mesh routing do not find the destination. Eventually, a route discovery procedure can be initiated to find a suitable path to the destination. The route discovery is performed through a sequence of broadcast messages. If a better route is found the routing tables of the nodes involved in the discovery are updated.

Application Layer

In order to explain the details of the APL some definitions are needed (see also figure 3.28).

Endpoint

The endpoint is a logical extension of the device address. Up to 240 user application can be hosted on a Zigbee device, each one is identified by a unique endpoint id. For example, a single zigbee device can control 4 lamps, each lamp will be associated to a different endpoint. Two endpoints are used for special usage: endpoint 0 for the ZDO and endpoint 255 to broadcast messages to all applications. Endpoints 241-254 are reserved for future use.

Attributes

Physical devices associated to a Zigbee device are identified through the use of attributes. For example, a the lamp might contain an output attribute turned on which represents the current status of the lamp.

Profile

Application profiles are agreements for messages, message formats and processing actions that enable applications to create an interoperable, distributed application between applications that reside on separate devices. Through the use of certified profiles developers from different companies can develop new products that can interoperate with other vendors devices. In the example of the lamp, a home lightning profile can be developed define the behavior of devices that are part of the lightning system such as switches and lamps.

Following a list of profiles that have been ratified:

- Home Automation (HA) defining devices for typical residential and small commercial installations.
- Commercial Building Automation (CBA) defining devices for large commercial buildings and networks.
- Advanced Metering Initiative (AMI) For utility meter reading and interaction with household devices
- Telecom Application (TA) Wireless applications within the telecoms area.
- Wireless Sensor Network Applications (WSN) Wireless sensor networks.
- Personal Home Health Care (PHHC) Monitoring of personal health in the home environment.

Cluster

A Cluster is identified by a cluster identifier, which is associated with data flowing out of, or into, the device. Binding decisions are taken by matching an output cluster identifier to an input cluster identifier, assuming both devices are within the same profile. For example the lamp can contain an input cluster id 21 to change its status from off to on while the switch will have an output cluster 21 to change the lamp status.

Descriptor

ZigBee devices describe themselves using descriptor data structures. Each device have a set of descriptors that are used to match devices and their services. The *node* descriptor contains information about the capabilities of the ZigBee node like device logical type, MAC capabilities and frequency band. It is mandatory for each node. The *node power* descriptor gives an indication of the power status of the node and is mandatory for each node. Each endpoint should have a *simple* descriptor that contains information specific to each endpoint on this node such as profile id, input clusters and output clusters. It is mandatory for each endpoint present in the node. Finally the *complex* descriptor contains extended information about the devices connected to a zigbee node such as manufacturer and model name and serial number. The use of the complex descriptor is optional.

Application Support Sublayer

The APS provides three basics services : *device discovery*, *service discovery*, *binding*.

Device discovery

Device discovery is the process whereby a ZigBee device can discover other ZigBee devices by initiating queries that are broadcast (of any broadcast address type) or unicast addressed. This service is used to find either the 16 bit Network address or the 32 bit IEEE 802.15.4 address.

Service discovery

Service discovery is the process whereby services available on endpoints at the receiving device are discovered by external devices. There are 2 modalities to issue a service discovery. The first is accomplished issuing a query to a specific endpoint on a given device, the second is called a match request that compares the descriptors of the 2 devices to see if they can match. Service discovery is a key process to interface and connect devices within the network. Through both direct and broadcast re-

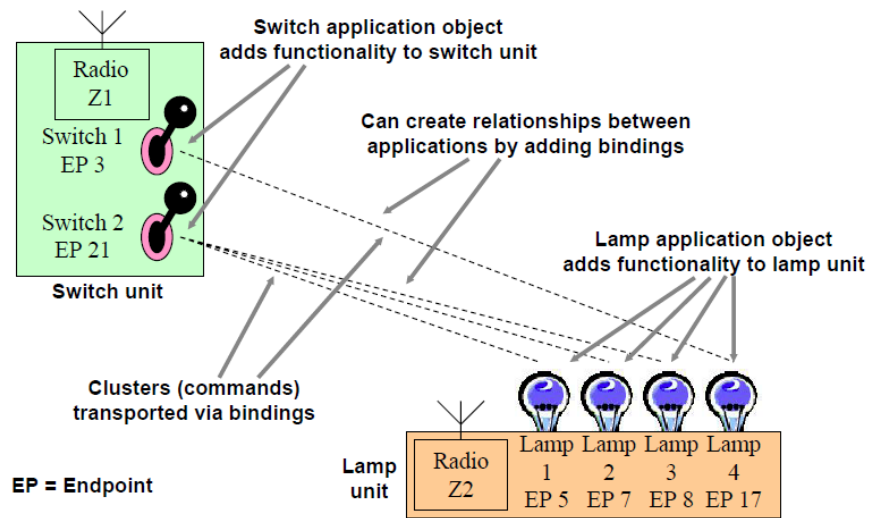


Figure 3.28: Example of Zigbee application, endpoint and clusters.

quests for service matching a range of options for commissioning tools and applications is available.

Binding

Binding is the creation of logical links between complementary application devices and endpoints. It is always performed after a communications link has been established. The binding is performed once a service match between two endpoints on two devices has been found. The nodes can decide to communicate directly through their network address (*direct* addressing) or to rely on a third node. The information about which cluster is bound between the two nodes is stored in a *binding table* stored within a device designated as the binding table cache.

Binding tables are used to perform *indirect* addressing. The use of direct addressing requires the controlling device to have knowledge of destination address, endpoint and cluster id. Such information may not be stored on the sending device, however after binding they are stored in the binding table. When a source device wishes to send a command to a destination using indirect addressing it simply send their messages to the device that contains the binding table that will use source address, endpoint and cluster id to find those of the destination device and rely the message to the indicated destination. Figure 3.29 clarify this process.

Note that one source address, endpoint and cluster can be bound to

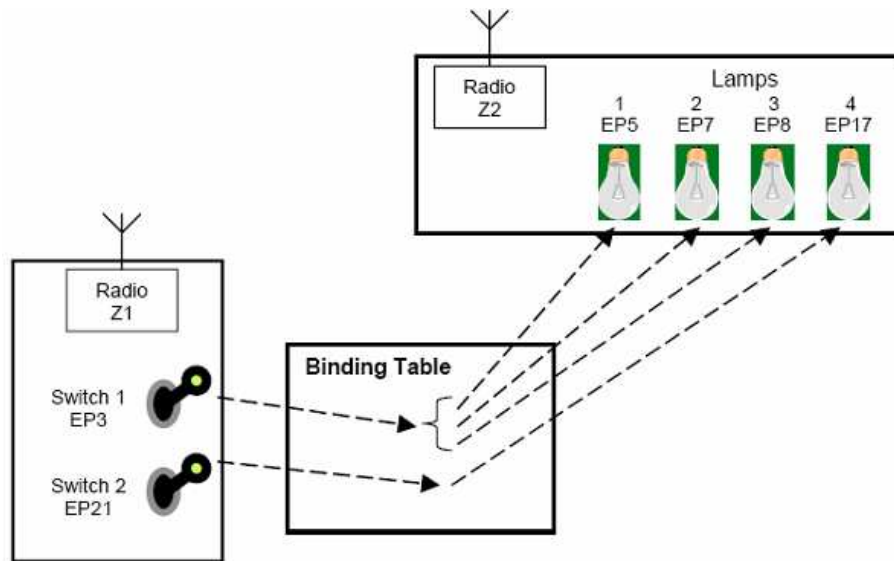


Figure 3.29: Example of binding process.

more than one destination ones.

Zigbee Device Object

ZigBee Device Object is an application which employs network and application support layer primitives to implement ZigBee End Devices, ZigBee Routers and ZigBee Coordinators. It has its own profile, ZigBee Device Profile (ZDP), and occupy reserved endpoint 0. Its public interface provides address management of the device, discovery, binding, and security functions within the application framework layer of the ZigBee protocol stack.

The ZDO is responsible for the following functions:

- Initializing the Application Support Sublayer (APS), Network Layer (NWK), Security Service Provider (SSP) and any other ZigBee device layer other than the end applications residing over Endpoints 1-240.
- Assembling configuration information from the end applications to determine and implement the following functions: device initialization, service and device discovery, security management, network management, binding management and node management.

3.4.6 Security

Security within the Zigbee stack is applied through the use of four techniques:

- Freshness check prevents replay attacks that can cause undesired network behavior. Each Zigbee device maintain a 32 bit freshness counter for both incoming and outgoing messages.
- A 0, 32, 64 or 128 bit message integrity code (default 62) is added at the end of the message to prevent an attacker from modifying the message in transit. The length of the integrity code is a trade-off between the message payload length and the protection level (the probability that a random guess of the code would be correct).
- Message Encryption using a standard 128 bit AES algorithm prevents an eavesdropper from listening to messages. Two level of encryption are possible through the use of two different keys: network level and device level. This protection is independent from the others, thus low power devices can decide not to implement it.
- Device authentication provides assurance about the originator of the message. Also authentication can be performed at network and device level. The latter, however, requires to store a pair of keys for each link, increasing the memory requirement for a node.

Such techniques are implemented both at MAC, NWK, and APL layer.

As highlighted in the previous list, security makes use of three symmetric keys:

- *Master key*. It is the basis for long-term security between the two devices, used during the execution of a symmetric-key key establishment protocol and to generate link keys. It is the basic key for secure communication and it can be either pre-installed or sent over the air if eavesdropping can be prevented.
- *Network key*. Broadcast communication is secured through this key shared by all devices of the network. As for the master key it can be sent over the air or pre-installed.
- *Link key*. Link keys are shared between couples of devices and are used for direct messages. Link keys are generated from the master key using the APL key establishment services based on the Symmetric-Key Key Establishment (SKKE) protocol.

To handle keys and to manage secure device authentication the Zigbee protocol defines the role of the *Trust center*. The trust center is a device trusted by all the other devices within a network. All members of the network shall recognize exactly one trust center, and there shall be exactly one trust center in each secure network. Typically the network coordinator is also the trust center.

The functions performed by the trust center can be subdivided into three subroles: trust manager, network manager, and configuration manager. The trust manager are used by the devices of the network to identify other devices of the network. The network manager is responsible for the network and distributes and maintains the Network key. Finally the configuration manager is responsible for binding two applications and enabling end-to-end security between devices of the network (for example, by distributing link keys).

Several design principles are involved in Zigbee security.

1. The layer that originates a frame is responsible for initially securing it. For example, MAC messages to handle a device disassociation must be secured at MAC level.
2. If protection is required all messages should be secured except the ones between a router and a newly joined device, until the new device receive the keys.
3. Keys can be reused between layers.
4. If two devices have a link key, it is always used instead of the network key.
5. All devices in the same network should use the same security level.

However some policy decisions are left to the real implementation.

1. Handle error conditions.
2. Out of band methods for key setup.
3. Handle loss of counter or key conditions.
4. Policy for expiration and update of keys.
5. Policy for new devices acceptance.

Chapter 4

Data Fusion and Pattern Recognition

4.1 Data Fusion overview

A concise definition of *Data fusion* have been proposed to highlight the fact that similar problems of data association and combination occur in a wide range of engineering, analysis, and cognitive situations. According to this definition data fusion is the process of combining data or information to estimate or predict entity states [193]. Often we refer to data fusion also as *Sensor Fusion*. In this case we refer to the use of techniques that combine data from multiple sources (sensors or high level inferences), and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone [76].

The concept of multi-sensor data fusion is not a novel idea. Humans and animals use multiple senses to improve their ability to survive. Nowadays the development of new sensors, hardware and processing techniques make real-time fusion of data possible.

Fusing data from multiple sensors offers some advantages over standard algorithms [129]:

1. Improved confidence due to complementary and redundant information;
2. Robustness and reliability in adverse conditions;
3. Increased coverage in space and time,
4. Better discrimination between hypotheses due to more complete information;

5. System being operational even if one or several sensors are malfunctioning;
6. Possible solution to the vast amount available information.

In 1988 the Joint Directors of Laboratories (JDL) Data Fusion Working Group, began an effort to codify the terminology related to data fusion. The result of that effort was the creation of a process model for data fusion and a data fusion lexicon [228]. Since then several revision have been proposed to improve that model [193, 18, 124]. The JDL process model identifies the processes, functions, categories of techniques, and specific techniques applicable to data fusion. The results is presented in figure 4.1.

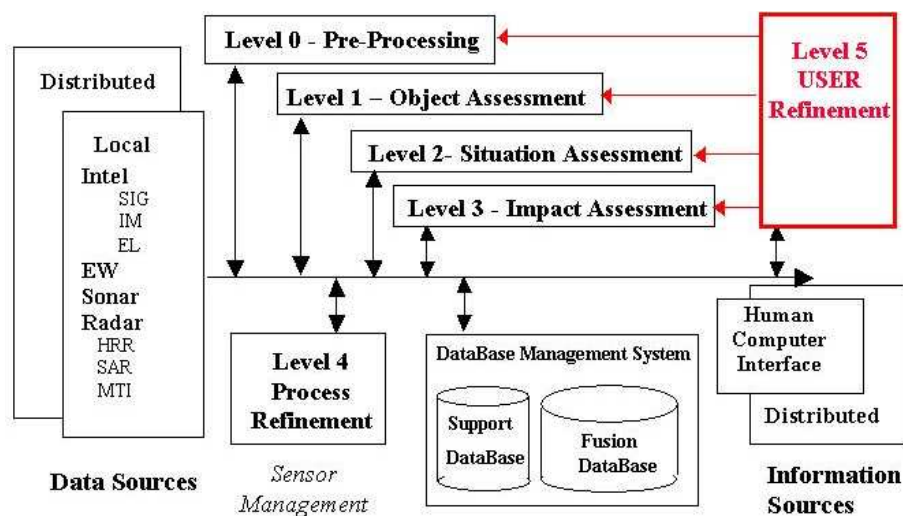


Figure 4.1: JDL Data Fusion model

The model shows a two-layer hierarchy. The top level is made up of four blocks: sensor inputs, human-computer interaction, database management and source preprocessing. At lower level 6 subprocesses are defined [18]:

Level 0 - Sub-Object Data Assessment . Estimation and prediction of signal/object observable states on the basis of signal level data association (i.e. data pre-processing).

Level 1 - Object Assessment . Estimation and prediction of entity states on the basis of observation-to-track association, continuous state estimation and discrete state estimation (i.e. combining data to estimate entity attributes and identity).

Level 2 - Situation Assessment . Estimation and prediction of relations among entities in the context of their environment.

Level 3 - Impact Assessment . Estimation and prediction of effects on situations of planned or estimated actions by the participants.

Level 4 - Process Refinement . Adaptive data acquisition and processing to support mission objectives. This is a meta-process that monitor the overall data fusion process to improve its performances

Level 5 - User Refinement . Adaptive determination of who handle information and adaptive data displaying to support cognitive decision making and actions (i.e. which data display to support user decision).

For each of these subprocesses the hierarchical JDL model identifies specific functions and categories of techniques. Three basic alternatives can be used for multisensor data: direct fusion of sensor data 4.2, representation of sensor data via feature vectors , and fusion of the feature vectors 4.3, or independent processing of each sensor to achieve high-level inferences, which are subsequently combined 4.4.

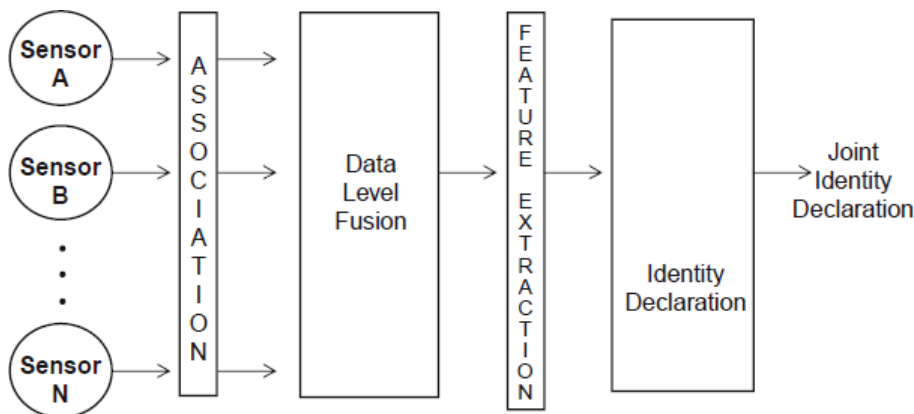


Figure 4.2: Fusion of sensor data

Several techniques have been proposed for data fusion [177, 66, 153, 123]. Several of them rely on pattern recognition techniques or on filtering. In the following sections we present an overview of algorithms.

4.2 Direct fusion of sensor data

Direct fusion of sensor data refers to the combination of input signals from a (heterogeneous) group of sensors in order to provide an output signal that is usually of the same form as the original signals, but of greater quality. The signals from sensors can be modeled as random variables corrupted by un-

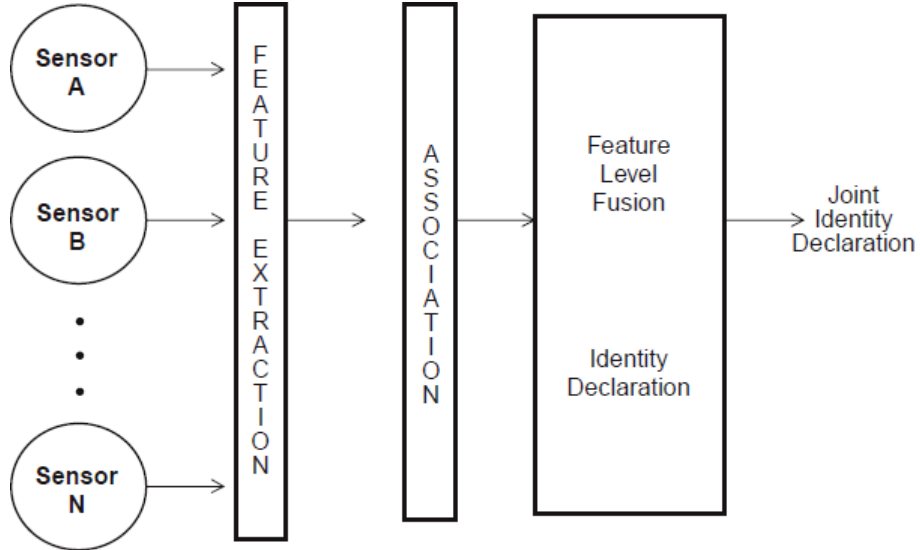


Figure 4.3: Fusion of features vectors

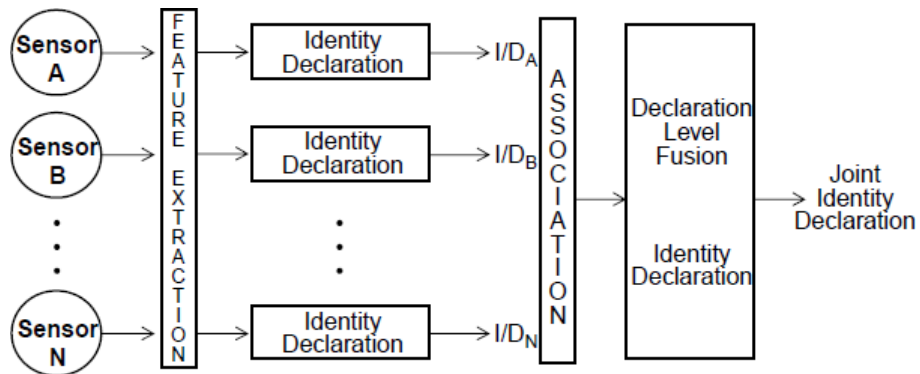


Figure 4.4: Fusion of high level inferences

correlated noise, and the fusion process can be considered as an estimation procedure.

Predictive filters are widespread tools in modern science. They perform state prediction and parameter estimation in fields such as robotics, computer vision, and computer graphics. They belong to the class of Bayesian filters, since they apply the Bayesian rule of conditional probability to combine a predicted behavior with some corrupted indirect observation [71].

As compared to the other types of fusion, fusion of sensor data requires a higher degree of synchronization between data streams from the sensors. The most common techniques for this kind of fusion consist of weighted averaging and Kalman filtering.

4.2.1 Kalman filter

In 1960, R.E. Kalman published a paper describing a recursive solution to the discrete data linear filtering problem [103]. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application.

The *Kalman filter* is the simplest example of a predictive filter. It represents uncertainties as Gaussian random variables, fully described by a mean and a covariance matrix, and models the system with linear dynamics and observations. Since Gaussians are preserved under linear transformation, the Kalman filter's implementation uses only linear algebra operations.

It can be shown that the Kalman filter is an optimal recursive data processing algorithm. One aspect of this optimality is that the Kalman filter uses all information that can be provided to it. It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest. Furthermore it use the knowledge of the system and measurement device dynamics, the statistical description of the system noises, measurement errors, and uncertainty in the dynamics models, and any available information about initial conditions of the variables of interest [135].

[225] The Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by the following set of linear equations 4.1

$$\begin{cases} x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ z_k &= Hx_k + v_k \end{cases} \quad (4.1)$$

Where x_k is the state of the process and z_k is the, noisy, measurement. w_k and v_k represent, respectively, the process noise and the measurement noise and are assumed to have a normal probability distribution with the parameters

presented in equation 4.2.

$$\begin{aligned} P(w) &\sim N(0, Q) \\ P(v) &\sim N(0, R) \end{aligned} \quad (4.2)$$

Where Q is the process noise covariance matrix and R is the measurement noise covariance matrix.

A , B and H are the equations that specifies how the state of the process evolves and is related to the measurement.

The filter estimates a process by using a form of feedback control. The process evaluate the state at some time and then obtains feedback in the form of (noisy) measurements. A set of *time update* equations (or *predictor* equations) are responsible for projecting forward the current state and error covariance estimates to obtain the *a priori estimates* for the next time step. The *measurement update* equations (or *corrector* equations) generate a feedback used to incorporate a new measurement into the *a priori* estimate and obtain an improved *a posteriori estimate*.

Equations 4.3 and 4.4 present respectively the prediction and the correction equations

$$\bar{x}^- = A\bar{x}_{k-1} + Bu_{k-1} \quad (4.3)$$

$$P_k^- = AP_{k-1}A^T + Q$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\bar{x}_k = \bar{x}_k^- + K_k(z_k - H\bar{x}_k^-) \quad (4.4)$$

$$P_k = (I - K_k H)P_k^-$$

Where with the over line we indicate the estimated value of the status of the process, $P = E[e_k e_k^T]$ is the *error covariance* matrix where the error is calculated as $e_k = x_k - \bar{x}_k$, and K_k is the *Kalman gain* that decides how much the *a priori* estimates should be corrected by the k -th observation (note how the large the measurement noise, R , the smaller the correction).

In the actual implementation of the filter, the measure of the noise covariance matrix, R , is generally possible prior to operation of the filter, since we should be able to measure the process to estimate its state. The determination of the process noise covariance, Q , is generally more difficult as often we do

not have the ability to directly observe the process we are estimating. For this reason we rely on an off line tuning of the parameter of the filter. This tuning often take the name of *system identification* or *training*.

If the hypothesis on the linearity of the process and the pdf of the noise are respected, the Kalman filter can provide an exact solution for the estimation of the process. However, one of the main criticism to the Kalman filter is that the hypothesis on the linearity of process and on the noises models are too restrictive. For this reason many other models have been proposed.

- Extended Kalman filters.
- Unscented Kalman filters.
- Particle filters.

The objective of these models is to provide an approximate solution for an exact model rather than an exact solution for an approximate model.

4.3 Fusion of features vectors

Fusion of features vectors combines distinctive relevant characteristics (*features*) from sensor readings to extract useful information or to classify a particular phenomenon. Those features may come from several raw data sources (several sensors, different moments, etc.) or from the same raw data.

Several techniques to fuse features vectors take the name of *pattern recognition* techniques. Pattern recognition can be defined as the act of taking in raw data and taking an action based on the “category” of the pattern [54]. In general a pattern recognition system establishes a mapping between the measurement space and the space of potential meanings (*classes*). This mapping is performed in six steps (see figure 4.5).

Sensing Data is collected from the one or more sensors.

Pre processing Pre processing includes all the steps necessary to condition the signal for further processing. Typically this step includes a filter to reduce signal noise.

Segmentation Segmentation is a critical step in the pattern recognition chain. Sensors provide a continuous stream of data, segmentation aims at extracting only the data related to a single entity to classify.

Feature extraction This step aims at reducing data dimension. The objective here is to extract quantities that are distinctive of a certain class.

Classification Classification uses the information provided by the features to assign the object to a category.

Post processing Exploit further context information other than from the target pattern itself to improve performance.

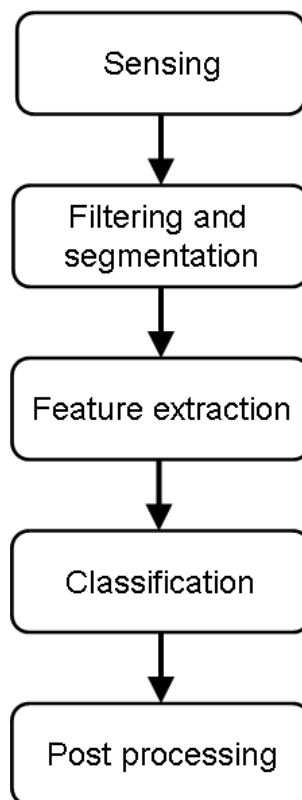


Figure 4.5: Pattern recognition steps

To perform the classification step we rely on a set of tools called *classifiers*. A large number of classifiers have been developed to address several problems in pattern recognition. In general we can sort them into two categories.

Supervised classifiers In supervised learning, a teacher provides a category label or cost for each pattern in a *training set*, and we seek to reduce the sum of the costs for these patterns. The classifier is trained off line using this set of samples. Typically training is a computationally expensive operation while normal classification is much more lightweight and suited for real time operation.

Unsupervised classifiers In unsupervised learning or clustering there is no explicit teacher, and the system forms clusters out of the input patterns.

Unsupervised classifiers are used when a training set is not available or too expensive to be created. Typically they rely on a set of assumption on the underlying probability densities and assume that the only thing that must be learned is the value of an unknown parameter vector.

The development of pattern classification systems rises a number of issues. Many are domain or problem specific, and their solution will depend upon the knowledge and insights of the designer.

Feature Extraction

The task of feature extraction is strictly problem and domain dependent. In general we can place a conceptual boundary between feature extraction and classification since an ideal feature extractor would yield a representation that makes the job of the classifier trivial and vice versa, a perfect classifier would not need the help of a feature extractor. Typically it is not possible to define features that are good for all problems and developer experience play an important role.

Overfitting

For supervised classifiers may sound obvious the idea that a larger training set will result in a more complex, but more performing classifier. Experience showed that increasing the complexity of the classifiers may result in poorer performance during normal operation. In fact while an overly complex model may allow perfect classification of the training samples, it is unlikely to give good classification of novel patterns. This situation is known as *overfitting*. One of the most important areas of research in statistical pattern classification is determining how to adjust the complexity of the model.

Model Selection

Tenth of model have been developed for classification. In general is hard to know when a hypothesized model differs significantly from the true model underlying our patterns.

Prior Knowledge, Context awareness

Incorporating prior knowledge and context awareness to improve the classification accuracy. However context can be highly complex and abstract and often came from different spaces than our features vectors.

Segmentation

Segmentation is one of the deepest problems in those pattern recognition application where continuous stream of data are handled like speech recognition, hand written recognition etc. In such application is essential

to extract part of the signal related to a single word or letter in order to perform the classification. Such procedure is not trivial but fundamental to improve classifier accuracy.

Computational Complexity

The computational complexity of different algorithms is of great importance, especially for practical applications where limited resources are available. Algorithm scaling with the number of features or performance-complexity trade off are very important for pattern recognition techniques evaluation.

In the following section a set of pattern recognition techniques are presented as examples. The list focus on the algorithm used during the work presented in this thesis.

4.3.1 Naïve Bayes Classifier

The naïve Bayes classifier is a simple probabilistic classifier based on Bayes' theorem. The classifier combines the Bayes probabilistic model with a decision rule. A common rule is to classify an input instance as belonging to the class that maximize the *a posteriori* probability [173]. Previous work [172] showed that naïve Bayesian classifiers perform well even if the independence assumption is not met.

Formally, given the conditional model $P(C|A_1, A_2, \dots, A_N)$, where C denotes the final classification output class and A_k are N input variables (the features vector) and using Bayes theorem we can define:

$$P(C|A_1, A_2, \dots, A_n) = \frac{P(A_1, A_2, \dots, A_n|C) P(C)}{P(A_1, A_2, \dots, A_n)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Marginal}}$$

Where the *Posterior* is the probability of a class given the input sequence, *Likelihood* is the conditional probability of a sequence given a certain class, *Prior* is the prior probability of the selected class, and *Marginal* is the probability of the input sequence.

Applying the assumption that the input attributes are independent we can write:

$$P(C|A_1, A_2, \dots, A_n) = \frac{P(C) \prod_{i=1}^n P(a_i|C)}{P(A_1, A_2, \dots, A_n)} \quad (4.5)$$

According to the proposed decision rule we can finally state:

$$C_{out}(a_1, a_2, \dots, a_n) = \underset{c}{\operatorname{argmax}} \frac{P(C = c) \prod_{i=1}^n P(A_i = a_i | C = c)}{P(A_1 = a_1, A_2 = a_2, \dots, A_n = a_n)} \quad (4.6)$$

Since the denominator in equation 4.6 is constant for every class we only need to compute the numerator. The *prior* can be easily derived knowing the percentage of each class in the training set. The *likelihood* is obtained during the training phase, by defining $P(A_i = a_i | C = c) = \frac{t_c}{t}$, where t_c is the number of training instances for which the class $C = c$ and the attribute $A_i = a_i$ and t is the number of training instances for class c . We must note that when for a class c we do not have a sample for which $A_i = a_i$, $\prod_{i=1}^n P(A_i = a_i | C = c)$ for that class is always zero, despite the value of the other input attribute. For this reason often the M-estimate of the likelihood is used. Its formula is presented in the following equation:

$$P(A_i = a_i | C = c) = \frac{t_c + mp}{t + m} \quad (4.7)$$

where p is an *a priori* estimation of $P(A_i = a_i | C = c)$ and m is a user specific value. Typical choice for p is $\frac{1}{\#A \text{ values}}$, while a typical value for m is $m = 1$.

4.3.2 Support Vector Machines

Support Vector Machines (SVM) is a supervised classifiers belonging to the class of linear discriminant classifiers. Such classifiers build discriminant functions that are a combination (either linear or not linear) of the input vectors' components. Geometrically, a discriminant function defines an hyperplane that separates two classes [54]. Several solution have been proposed to deal also with non-separable data.

The original idea about SVM has been developed since 1979 by Vladimir Vapnik [210, 211, 212]. Recently there has been an explosion in the number of research papers on the topic of SVM. SVMs have been successfully applied to a number of applications ranging from particle identification, face identification, and text categorization to engine knock detection, bioinformatics, and database marketing [14].

The simplest case deal with 2 classes linearly separable data. If we call x_i the vector with the features, and $y_i = \pm 1$ the label of each input vector. A discriminant function that is a linear combination of the components of x can

be written as:

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{w} + b &> +1 & y_i &= 1 \\ \mathbf{x}_i \cdot \mathbf{w} + b &< -1 & y_i &= -1 \end{aligned} \quad (4.8)$$

Where \mathbf{w} is a weight vector that determines the orientation of the separating hyperplane and b is a bias that indicate the distance from the origin of the separating hyperplane see figure 4.6.

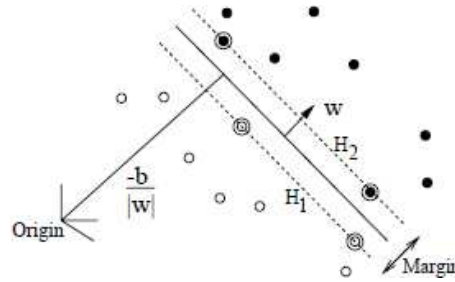


Figure 4.6: Best separating hyperplane in the separable case (feature space = 2)

It is clear that infinite planes can be defined to separate the two sets of samples. A smart choice is to select the one that presents higher margin. The hyperplane with higher margin can be found if we consider the points where the equality in equation 4.8 holds. Such points lay on 2 hyperplanes (H_1 , H_2) that share the same normal vector \mathbf{w} and relative distance (margin) equal to $\frac{1}{\|\mathbf{w}\|}$. Thus we can find the optimal hyperplane (the one with maximum margin) by minimizing $\|\mathbf{w}\|^2$ subject to constraints 4.8.

Note how the only points needed to build the separating hyperplane are the one that lay on H_1 and H_2 . Such points are called *support vectors*.

In a more complex case, where we have to distinguish between more than 2 classes, 2 solutions are possible: build an hyperplane that separate each class from all the other, build an hyperplane for each couple of classes (see figure 4.7).

This approach can be extended to handle non separable data. The idea is to relax the constraints in equation 4.8, but only when necessary. In order to do it we introduce a further cost called *slack variables*, ξ_i .

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{w} + b &> +1 - \xi_i & y_i &= 1 \\ \mathbf{x}_i \cdot \mathbf{w} + b &< -1 + \xi_i & y_i &= -1 \\ \xi_i &\leq 0 & & \forall i \end{aligned} \quad (4.9)$$

In equation 4.9 for an error to occur ξ_i must be greater than 1, hence $\sum \xi_i$ is an upper bound of the training error. We can take this contribution into account by changing the objective function to be minimized to $\frac{\|\mathbf{w}\|^2}{2} + C(\sum \xi_i)^k$ [23], where C is a user defined constant. The higher is C the higher is the penalty

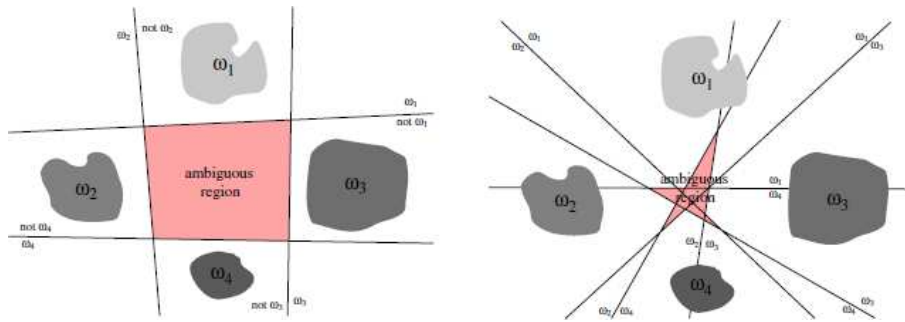


Figure 4.7: Two options for building a set of separating hyperplanes in the multiple class example

assigned to errors. A graphical representation of the use of slack variables is presented in figure 4.8.

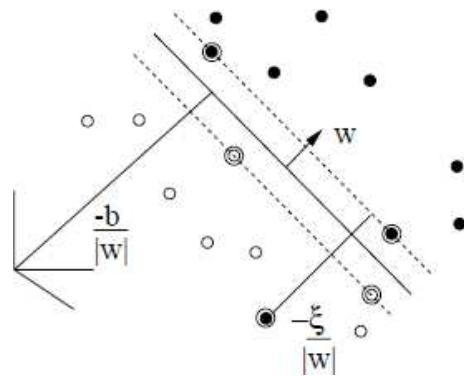


Figure 4.8: Separating hyperplanes in case of non separable data.

The concept above can be further extended to non linear hyperplanes. The basic idea is to map the input feature vector into a space with much higher dimensionality ($n \gg m$) where they can be easily separated.

$$\Phi : R^m \rightarrow R^n \quad (4.10)$$

It can be shown that in the training steps the vector of features appears always as a product of vectors $(\mathbf{x}_i \cdot \mathbf{x}_j)$, thus if we are able to find a *Kernel function* $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ we will use only such functions and we do not even need to know Φ .

Some example of kernel are presented in 4.11.

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \quad (4.11)$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$

4.3.3 K-Nearest Neighbors

The *k-Nearest Neighbors* (k-NN) algorithm is amongst the simplest of all machine learning algorithms. It belongs to the class of non parametric classification techniques.

In its classical form, given a reference dataset of n elements $D_n = \{(\mathbf{x}_i, y_i), 1 \leq i \leq n\}$ where \mathbf{x}_i are the samples and y_i the respective class, the *Nearest Neighbor* assigns any new input feature vector to the class of the nearest vector. More in general the k-NN algorithm maps any new feature vector to the most represented class within the labels of the k nearest reference vectors [115].

Formally, for a two class problem, given a metrics $d(x, x')$ on \mathbb{R}^d and an integer k , the k-nearest neighbor classifier generates a map from \mathbb{R}^d to $\{0, 1\}$ as a function of the reference samples D_n wherein each point $\mathbf{x} \in \mathbb{R}^d$ is mapped into one of the two classes according to the majority of the labels of its k -nearest neighbors in the reference sample see figure 4.9.

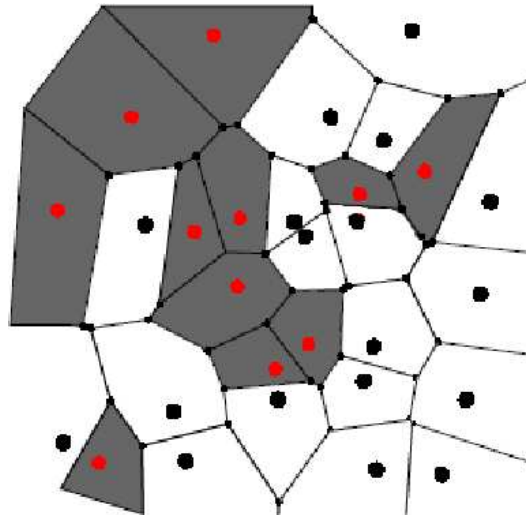


Figure 4.9: K-NN mapping in the two class case, with $d=2$ and $k=1$

Two main issues in developing a k-NN classifier are: the choice of the metric, since features can present dynamics that differ of several degree of magni-

tude, and tie handling. A way to handle ties is to assign the new feature vector to the most represented class.

Despite its simplicity, versions of this non parametric algorithm are asymptotically consistent with a Bayes classifier, and are competitive with other popular classifiers in practical settings. However special attention must be placed when implementing this technique on an embedded, low-power, low-cost device, since in order to classify a new instance the whole training set D_n must be stored on the device.

4.4 Fusion of high level inferences

The data fusion systems presented in the previous sections use a single feature descriptor and a particular classification procedure to determine the true class of a given pattern. However, for problems involving a large number of classes and noisy inputs, perfect solutions are often difficult to achieve. For this reason a number of methods have been developed for classifier fusion, we often refer to this method as *mixture of experts*.

There are two general groups of classifier fusion techniques [175]:

1. *Method operating on classifiers*. They aim at finding the single best classifier among a group of classifiers and take its output as the final decision for further processing [231] (*Dinamc Classifier Selection*).
2. *Method fusing classifiers output*. Produce an output which is a combination of the output of the input classifiers outputs [233].

The method operating on classifiers output can be further divided according to the type of the output produced by individual classifiers [175].

- *Single labels*. Classifiers producing crisp, single class labels (SCL) provide the least amount of useful information for the combination process.
- *Ranked classes*. Classifiers can provide a ranking of the possible classes. Two main methods can be used to fuse such ranking: class set reduction, that aim a reducing the number of possible classes while ensuring that the correct class is still present in the reduced set; class set reordering, that aim at ranking the correct class at the top of the results.
- *Soft output*. Soft outputs are numbers in the range $[0, 1]$ that cover all known measures of evidence: probability, possibility, necessity, belief and plausibility [110].

This taxonomy is presented in figure 4.10.

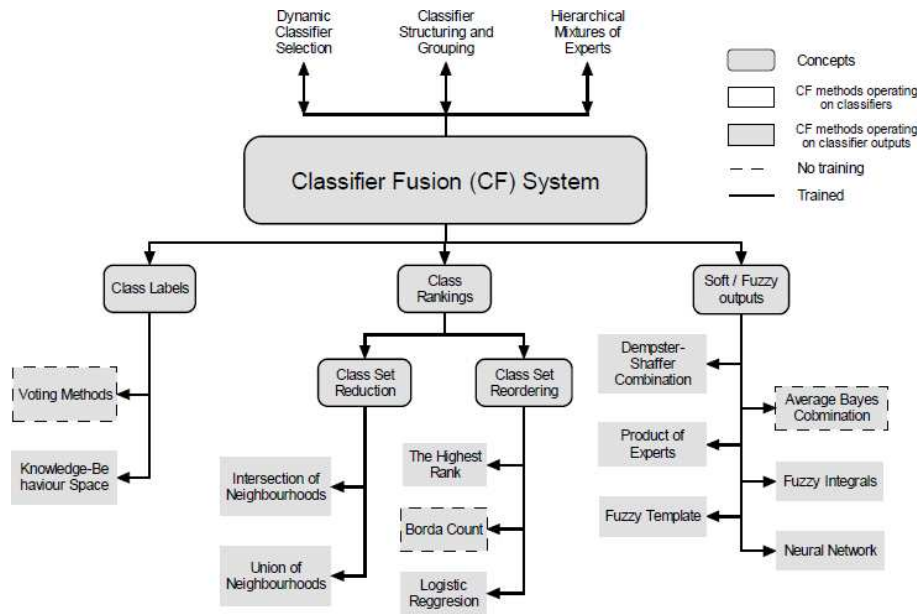


Figure 4.10: Diagram of the possible classifier fusion methods

4.4.1 Majority voting

Majority voting [119] is the simplest single labels classifier fusion methods. It does not assume prior knowledge of the individual classifiers, and it does not require training.

The majority voting technique can be used when we are in presence of n experts that produce a unique decision regarding the identity of the sample. In combining the decisions of the n experts, the sample is assigned the class for which there is a consensus, or when at least k of the experts are agreed on the identity, where

$$k = \begin{cases} \frac{n}{2} + 1 & \text{if } n \text{ is even} \\ \frac{n+1}{2} & \text{if } n \text{ is odd} \end{cases} \quad (4.12)$$

According to this rule the combined decision is correct when a majority of experts is correct, while it is wrong when the majority of experts is wrong *and agrees*. The strength of this method stems from the fact that in order to produce an error the majority of expert must be *wrong and make the same mistake*, which is unlikely.

Despite its simplicity, this approach showed as much effectiveness as other more complicated approaches such as: Bayesian classifier, logistic regression, fuzzy integral, and neural network [120].

4.4.2 Borda Count

The Borda count is an election method based on the ranking of the candidates. It belongs to the class reduction methods of the rank classes classifiers and can be considered a generalization of the majority vote method.

Each expert generates a ranked list of potential classes where the most likely class is placed at the top of the list. The Borda count for a class is the sum of the number of classes ranked below it (i.e. less likely) by each classifier. The output class is the one with the highest Borda count [86].

The Borda count method is simple to implement and requires no training. However, it does not take into account the differences in the individual classifier capabilities. All classifiers are treated equally, which may not be preferable when we know that certain classifiers are more likely to be correct than others.

4.5 Time variant classifiers

In the previous sections, data from each class were assumed drawn from a single generating distribution, and independently of each other (independent, identical distributed samples, *i.i.d.*). However, in many field of research we deal with sequential data. In contrast to *i.i.d.*, sequential data present a high degree of correlation between successive samples, and the information is contained in the sequence of samples. For this reason, a sequence of samples are the actual input of our classifiers, and each sample may be generated out of a different distribution.

Several techniques have been developed to classify sequential data [51]. Common approaches include hidden Markov models (HMMs) [55], dynamic time warping [141] and neural networks [221, 143].

4.5.1 Hidden Markov Models

The Hidden Markov Model(HMM) is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing.

The HMM belong to the class of the Markov processes, which are models used to describe the evolution of a system. A Markov process describes a system which at any given time t can be in one of N states S_1, S_2, \dots, S_N . At each time step, the system changes its state according to a set of probabilities associated with the actual state. The output of the process is the set of states at each instant of time, where each state corresponds to a physical event, thus we refer to this model also as observable Markov model [168].

In many cases of interest, the state of the system cannot be directly observed, but inferred through measurements of other variables called *observation*. This implies that the state of a system at any given time can be treated as a *hidden* random variable that generates the observables that we measure. A HMM is a probabilistic model used to describe sequences of observations $O = \{o_1, o_2, \dots, o_T\}$ and their corresponding hidden state $Q = \{q_1, q_2, \dots, q_T\}$.

Two fundamental hypotheses are given:

1. The state of the system at any given time t depend only on the state at time $t - 1$.

$$p(q_t | q_{t-1}, o_{t-1}, q_{t-2}, o_{t-2}, \dots, q_1, o_1) = p(q_t | q_{t-1}) \quad (4.13)$$

2. The observable at any given time t depend only on the state at time t .

$$p(o_t | q_t, q_{t-1}, o_{t-1}, q_{t-2}, o_{t-2}, \dots, q_1, o_1) = p(o_t | q_t) \quad (4.14)$$

A Discrete HMM is characterized by the following parameters:

- A set of N states $S = \{s_1, s_2, \dots, s_N\}$. Although they are hidden, often they are related to some physical significance.
- A set of M discrete observables $V = \{v_1, v_2, \dots, v_M\}$ which represent the physical values out of the system.
- The state transition probability matrix $A = \{a_{ij}\} = P(q_{t+1} = s_j | q_t = s_i)$. Each element a_{ij} of the matrix defines the probability of being in state s_i at time t and in state s_j at time $t + 1$.
- The observation probability matrix $B = \{b_i(k)\} = P(o_t = v_k | q_t = s_i)$. Each element $b_i(k)$ of the matrix defines the probability of seeing symbol k in state i .
- The initial state distribution vector $\Pi = \{\pi_i\} = P(q_1 = s_i)$. Each element π_i of the vector defines the probability of being in state i at the beginning of the sequence.

The compact notation of a HMM is $\lambda = (A, B, \Pi)$. In figure 4.11 an example with $N = 3$ and $M = 3$ is presented.

Continuous HMM differ from Discrete HMM only because the observables can assume continuous value. In this case B typically is represented through a mixture of gaussian, thus this matrix is replaced by one vector of mean and one covariance matrix for each state.

There are three main problems associated with HMMs.

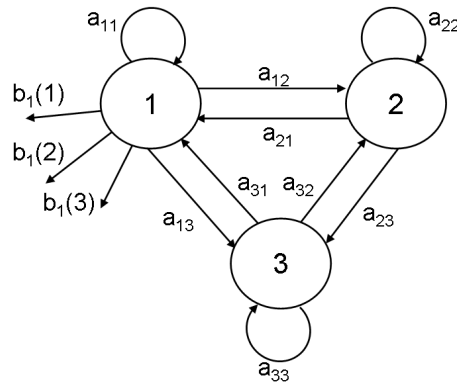


Figure 4.11: Example of HMM with $N = 3$ and $M = 3$

1. Given a sequence of observation $O = \{o_1, o_2, \dots, o_T\}$ and a model λ find the probability that the model generated that sequence $P(O|\lambda)$. This is also called the *evaluation* problem. The solution of this problem is equivalent to perform the classification of data.
2. Given a sequence of observation $O = \{o_1, o_2, \dots, o_T\}$ and a model λ find the probability the most probable sequence of states that generated that sequence. This is also called the *decoding* problem. Since a physical status can be associated to each state of the model, the solution of this problem is equivalent to filter out the noise on the observations.
3. Given a set of observations O_1, O_2, \dots, O_l find the model λ that best describes that observations. This is the *estimation*. The solution of this problem optimize, by training, a model for solving problems 1 and 2.

In the following section we present the solutions to the three problems for discrete HMM.

Evaluation problem

The most straightforward way of calculating the probability of a sequence $O = \{o_1, o_2, \dots, o_T\}$ given a model λ is through enumerating every possible state sequence of length T , Q . Assuming the statistical independence of observations, the probability of each sequence Q is:

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) \quad (4.15)$$

The final probability of the sequence can be obtained by summing the products between probability in 4.15 times the probability of the sequence Q over

all possible sequences.

$$\begin{aligned}
 P(O|\lambda) &= \sum_{\text{all } Q} P(O_t|Q, \lambda)P(Q|\lambda) \\
 &= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1, q_2} b_{q_2}(O_2) \dots a_{q_{T-1}, q_T} b_{q_T}(O_T)
 \end{aligned} \tag{4.16}$$

However the calculation presented in 4.16 involves the order of $2 \cdot T \cdot N^T$ calculations. This is computational unfeasible, since T in most cases is in the order of hundreds or thousands of samples.

For this reason we rely on a more efficient, recursive, procedure called the *forward* algorithm.

This procedure uses the forward variables α_t defined as the probability of the partial observation sequence, $O = \{o_1, o_2, \dots, o_t\}$ and state s_i , at time t , given the model λ , $\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i|\lambda)$. This procedure is made up of three steps.

1. Initialization: $\alpha_1(i) = \pi_i(O_1)b_i(O_1), 1 \leq i \leq N$
2. Induction: $\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i)a_{ij}]b_j(O_{t+1}), 1 \leq j \leq N$ and $1 \leq t \leq T - 1$
3. Termination: $P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$

The computational cost is $N^2 \cdot T$.

Decoding problem

The decoding problem is solved through the *Viterbi* algorithm which is a dynamic programming method.

The viterbi algorithm defines the following quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t|\lambda]P(O|Q, \lambda) = \prod_{t=1}^T P(O_t|q_t, \lambda) \tag{4.17}$$

$\delta_t(i)$ is the highest probability along a single path, at time t , which accounts for the first t observation and ends in state S_i . By induction $\delta_{t+1}(j) = [\max_i \delta_t(i)a_{ij}] \cdot b_j(O_{t+1})$. To retrieve the state sequence, we need to keep track of the argument which maximized equation 4.17.

The complete procedure is made up of the following steps:

1. Initialization: $\delta_1(i) = \pi_i b_i(O_1)$
2. Induction: $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i)a_{ij}]b_j(O_t)$
state at time t : $q_t = \arg \max_{1 \leq i \leq N} [\delta_t]$

3. Termination: probability of the path $P = \max_{1 \leq i \leq N} [\delta_t]$

Note that this procedure at each instant t finds the most probable state q_t given the model and the sequence. However the resulting final sequence may not be possible. Infact, suppose that $a_{ij} = 0$ for a given couple of state $\{i, j\}$. It may happen that $\delta_t = s_i$ and $\delta_{t+1} = s_j$ because 2 different path present the highest probability at 2 successive steps. However, the resulting state sequence is the one that maximizes the probability at each time step t of seeing the previous sequence of symbols.

Estimation problem

The third problem of HMMs is to adjust the model parameters (A, B, λ) to maximize the probability of a set of observation sequences (training set). There is no known way to analytically solve this problem, thus we rely on a iterative procedure called the *Baum-Welch* algorithm. The Baum-Welch algorithm belongs to the class of the Expectation-Maximization (EM) algorithms.

In order to describe this algorithm we must define three other variables, the backward variable $\beta_t(i)$, $\xi_t(i, j)$ and $\gamma_t(i)$.

The backward variable $\beta_t(i)$ is similar to the forward variable $\alpha_t(i)$ except that it is calculated from the last sample. It represents the probability of the partial observation sequence from $t + 1$ to the end T , given the actual state S_i at time t and the model λ .

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i, \lambda) \quad (4.18)$$

As for the α variable we can compute it in a recursive manner.

1. Initialization: $\beta_T(i) = 1, 1 \leq i \leq N$
2. Induction: $\beta_t(j) = \sum_{i=1}^N \beta_{t+1}(i) a_{ij}, 1 \leq j \leq N$ and $t = T - 1, T - 2, \dots, 1$

The $\xi_t(i, j)$ variable represent the probability of being in state S_i at time t , and state S_j at time $t + 1$, given the model and the observation. In equation 4.19 is described how to calculate $\xi_t(i, j)$ as a function of α, β, a_{ij} and $b_i(k)$.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1} \beta_{t+1}(j))}{\sum_{i=1}^N \sum_{j=i}^N \alpha_i a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (4.19)$$

Finally the $\gamma_t(i)$ is the probability of being in state S_i at time t given the observation sequence O and the model λ . $\gamma_t(i)$ can be computed using both the α and the β or the $\xi_t(i, j)$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} = \sum_{j=1}^N \xi_t(i, j) \quad (4.20)$$

Notice that $\sum_{t=1}^{T-1} \gamma_t(i)$ represent the expected number of transition from S_i and $\sum_{t=1}^{T-1} \xi_t(i, j)$ represent the expected number of transition from S_i to S_j . Thus a method for reestimation of the parameter of an HMM ($\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$) is the following.

$$\bar{\pi}_i = \text{expected number of state } S_i \text{ at time } t=1 = \gamma_1(i) \quad (4.21)$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transition from state } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (4.22)$$

$$\begin{aligned} \bar{b}_i(k) &= \frac{\text{expected number of times in state } i \text{ and observing symbol } v_k}{\text{expected number of times in state } i} \\ &= \frac{(\sum_{t=1}^{T-1} \gamma_t(i))_{s.t.O_t=v_k}}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (4.23)$$

It has been proved that $P(O|\bar{\lambda}) \geq P(O|\lambda_0)$, where λ_0 is a starting, random model, and that iterating the estimation of the model parameter, this iteration converge to a local maxima.

Since only local maxima can be obtained, this procedure should be repeated several times, each time starting from a different random guess of the parameters. The best solution is kept.

Chapter 5

Multimodal surveillance

5.1 Overview

Human detection and motion tracking have always gathered much attention in field as surveillance, industrial applications and, in general, smart environments. Conventional tracking techniques use cameras and process large amounts of data to extract features such as number of people, position and direction [19].

Video surveillance and other security-related applications have gained many credits due to the terroristic threats of the last years. Several industrial and academic projects have recently started to increase the accuracy of (semi) automatic surveillance systems. In addition, the abatement of hardware costs allows the deployment of thousands of cameras for surveillance purposes at a reasonable cost.

The ever-increasing demand of security and the low cost of cameras contributed to the diffusion of the research in distributed multi-camera surveillance systems. Multiple cameras enable the surveillance of wider areas and the exploitation of redundant information (provided by the different viewpoints) might solve classical limitations of single-camera systems, such as occlusions.

Despite the efforts made by the researchers in developing a robust multi-camera vision system, computer vision algorithms have proved their limits to work in complex and cluttered environments [207].

These limits are mainly due to two classes of problems. The first is that non-visible areas can not be processed by the system. This trivial statement is of particular importance in cluttered scenes and can be partially lessened by using multiple sensors (not only cameras). The second class of problems, instead, is due to the limited resolution of cameras. Having infinite resolution and zooming capabilities would make the job easier, but, in addition to be

unfeasible, it would exponentially increase the computational load and it is typically too expensive.

An interesting solution is that of using simple but effective specialized sensors to solve the specific problems of the vision systems. In this way, vision would still provide high-level information, and low-level sensors would assure higher accuracy. In this context, the marriage between a widely distributed low-cost wireless sensor network and the coarsely distributed higher level of intelligence that can be exploited by computer vision systems may overcome many troubles in a complete tracking of large areas.

Pyroelectric InfraRed (PIR) detectors take advantage of pyroelectricity, which is the electrical response of a polar, dielectric material to a change in its temperature, to detect a body at thermal disequilibrium with the surrounding environment. These sensors are typically used in commercial applications to detect presence of individuals to trigger alarms. However, human tracking with cameras can greatly benefit from the integration of PIR detectors within the video system. Being low-cost, low-power and presenting a small form factor, PIR sensors are well suited for WSN based application.

In this chapter we present the design and development of a low cost wireless sensor network which, by means of PIR detectors, is able to extract the number, direction of movement and the position of individuals moving through a gate or a section of a hallway [236, 238, 237]. In this way, we explored a novel use of PIR sensors for advanced tracking.

The proposed algorithms have low computational requirements, they are therefore well suited for systems with limited computational resources, such those available in sensor nodes (usually equipped with 8-bit microcontrollers). Moreover, they enable fast recognition of occurring events, to obtain a highly reactive system. We propose a multilevel data analysis, where processing is distributed among nodes: the end-nodes extract the features, while a coordinator infers the event happened.

In this chapter we will show how these techniques have been integrated with a video surveillance system to increase its effectiveness [165, 40].

PIR sensors can be integrated within a video surveillance network also to increase the lifetime of Wireless Video Sensor Nodes (WVSN). Low-cost video surveillance systems based on wireless sensor networks will hit the market with the promise of flexibility, quickly deployment and providing accurate real-time visual data. However, many technical problems have to be still overcome for a widespread diffusion of such a technology. For instance, even if research continues to develop higher energy-density batteries, capacity constraints limit the lifespan of common wireless sensor nodes. For this reason, energy-aware design and maximization of the sensor network lifetime become

the major key research challenges for WWSN and their applications.

To enhance vision sensor networks, two successful strategies can be adopted:

1. exploiting alternative power sources which increase the autonomy of the nodes considerably;
2. exploring multi-modal sensor integration which can save on-board power consumption

Recently, several researchers have proposed alternative power sources and Energy Scavenging techniques to extract and convert power from the surrounding environment and to replenish energy buffers like batteries or supercapacitors. In particular, photovoltaic (PV) harvesters are the most promising to enable perpetual operation of WSNs [20, 187]. Unfortunately if the power consumption of a device can be estimated, the power generated by a PV module changes non-linearly under varying temperature or solar irradiance and techniques which automatically tune the operating point of the solar cell should be considered to provide the maximum output power.

From the sensor capability point of view, CMOS imagers are generally high-power consuming devices and accuracy of the information increases the required power. Therefore they should be activated very carefully in order to save energy and their functions could be replaced by low-power low-level vision devices during the idle intervals, when the density of the events or the energy stored is low. Being able to detect variations of incident infrared radiation, due to movement of bodies not at thermal equilibrium compared the environment, the use of a network of PIR may lead to the extraction of more complex data such as object direction of movements, speed, distance from sensor and other characteristics [185]. The combination of several vision devices with heterogeneous features allows the development of multimodal surveillance applications with efficient energy policies. In fact, video would still provide high-level information when required, and PIR sensors would assure a continuous monitoring service triggering the CMOS camera when an event is detected.

In this chapter we present the design, implementation and characterization of a self powered video sensor node, able to detect people and supported by PIR sensors to enhance energy efficiency [128, 127].

5.2 Pyroelectric InfraRed (PIR) Sensors

Detectors that measure radiation by means of the change of temperature of an absorbing material are classified as thermal detectors. Thermal detectors respond to any wavelength radiation that is absorbed, and when an appropriate

absorbing material is applied to the detector element surface, they can be made to respond over a selected range of wavelengths [93].

Pyroelectricity is the electrical response of a polar, dielectric material to a change in its temperature. For pyroelectric sensor applications two classes of material are used: ionic crystals (like $LiTaO_3$ or $LiNb_3$) and molecular crystals or polymers (for example Polyvinylidene Fluoride, PVDF) [227, 11]. In crystalline matter, pyroelectricity occurs in all materials with symmetries that allow the existence of a polar direction, in polymers, the polar features are polar molecules or groups. The wavelengths of interest are mainly in the range of the infrared window at $8 \div 14\mu m$, in which the IR emission of room temperature bodies also peaks (see figure 5.1).

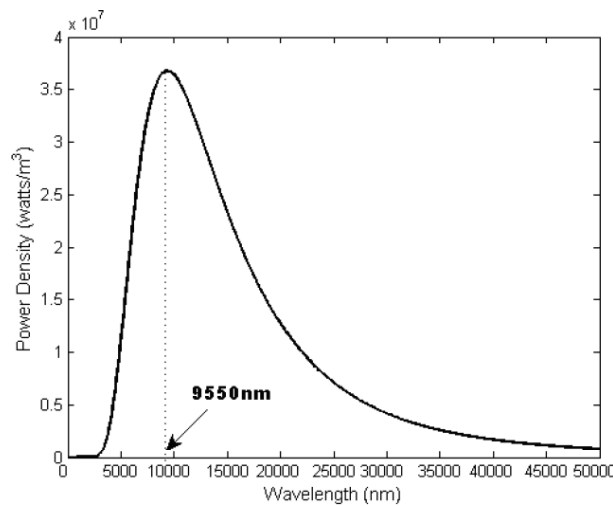


Figure 5.1: black body radiation curve at 37°C

The basic structure of a pyroelectric sensing element is a planar capacitor whose charge Q varies according to:

$$\Delta Q = A \cdot p \cdot \Delta T \quad (5.1)$$

where A is the area of the sensing element and p the pyroelectric coefficient specific for that material. The origin of this effect lies in polar features that are lined up with the same orientation along at least one direction in the material [156].

This charge usually is measured using electrodes as a current through a capacitor surface (see figure 5.2). In fact, calculating the time derivative of the pyroelectric charge, we obtain the following:

$$I = A \cdot p \cdot \frac{dT}{dt} \quad (5.2)$$

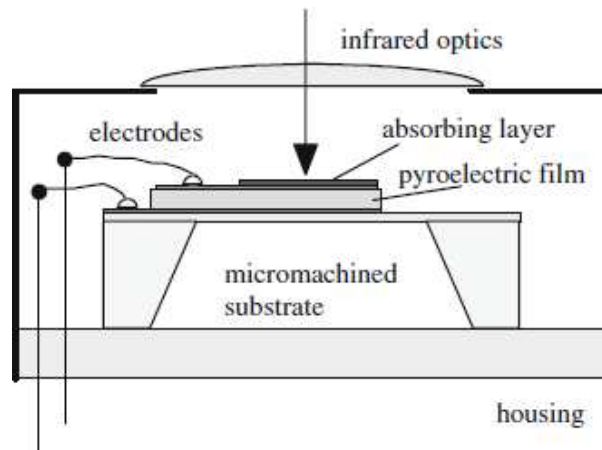


Figure 5.2: Schematic illustration of a PIR sensor

This extremely low current must be amplified with a high impedance preamplifier, two alternatives are possible: voltage and current mode. The complete chain conversion is presented in figure 5.3 [93].

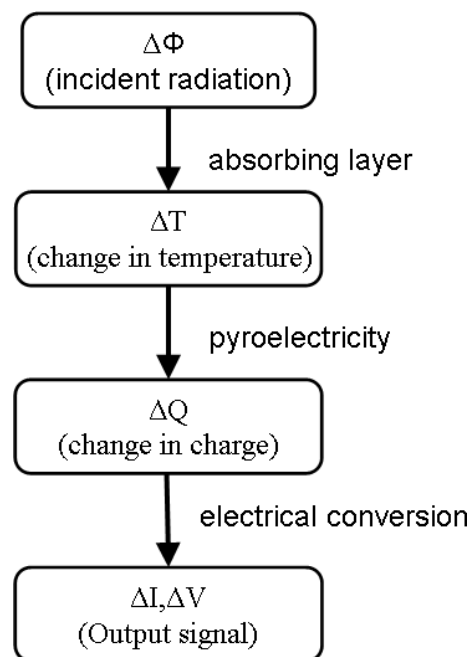


Figure 5.3: Chain conversion for PIR sensors

Due to its simplicity, voltage mode is the most commonly used operating mode for pyroelectric detectors. In the simplest case the preamplifier is made up of a JFET transistor configured as source follower. The gate resistor and the JFET are integrated into the detector housing. The resistor in the source line is

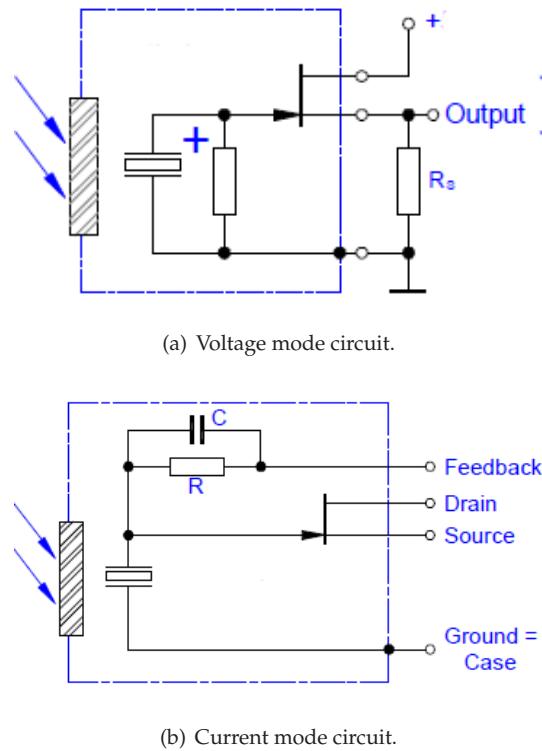


Figure 5.4: PIR preamplification modes.

placed outside the detector housing 5.4(a).

Current mode pyroelectric detectors are not as widely available as voltage mode ones. The reason for this is that elementary pyroelectric detectors are mass produced for light switches and motion detectors. Due to the complexity of the preamplifier the use of this kind of preamplification is limited to few applications. In the simplest case the housing containing only the pyroelectric element, more complex solutions include the pyroelectric element, JFET and feedback resistor 5.4(b).

Nowadays commercially available pyroelectric materials are stable, uniform and durable. This development has made practical the large scale production and application of cost effective, high performance pyroelectric infrared detectors into a wide variety of commercial, industrial and military applications.

Commercial Off The Shelf (COTS) PIR sensor usually include two or four sensitive elements in order to improve the immunity to changes in the background temperature and achieve a shorter settling time. The resulting schematic is presented in figure 5.5.

Furthermore, In order to shape the Field of View (FOV) of the sensor, the

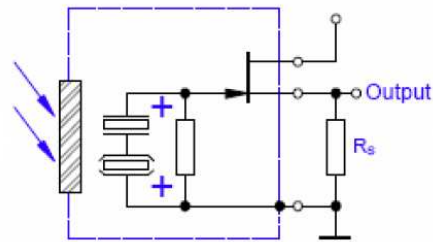


Figure 5.5: Schematic of a COTS PIR

detector is used together with a package equipped with Fresnel lens. Fresnel lenses are good energy collectors and are used in several applications. They can be obtained molding inexpensive and lightweight plastic materials with transmission characteristics suited for the desired wavelength range. Array of such lenses are designed to divide the detection area in distinct zones. As a body moves through such cones of view, incident radiation changes and the resulting output clearly indicates the presence of a person (see figure 5.6).

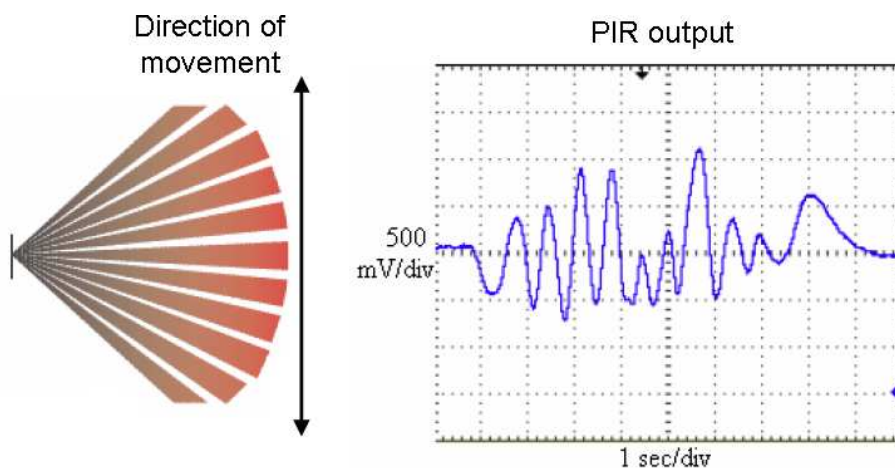


Figure 5.6: Output of a PIR sensor used in conjunction with array of Fresnel lenses

5.2.1 Related work

Thanks to their ability to detect body not at thermal equilibrium with the environment PIR sensor are widely used to in surveillance systems [145] and automatic light switching systems.

PIR sensors are also used in much more complex applications such as thermal imaging [4], radiometry [164], thermometers [203] and biometry [62, 63].

Several other works explore the usage of PIR sensor arrays to track people movement. In Gopinathan et al. [73] a pyroelectric motion tracking system able to detect the path of a single person moving in an area and based on coded apertures has been developed. The apertures are designed to modulate the visibility of four PIR detectors over a 1.6×1.6 m area such that the position of a source among 15 resolution cells may be discriminated using 4 measurements.

in Shankar et al. [185] a low cost sensor cluster is used to extract velocity as well as the path of a single person. While a modified PIR sensor is used to classify 5 different human motion events in [206]. In Song et al. [188] analyzes the performance and the applicability of the PIR sensors for security systems and propose a region-based human tracking algorithm. This technique has been implemented and tested in real environment. Results shows that the human tracking algorithm based on the PIR sensors performs very well with the proposed sensor deployment. De Vlaam in his master thesis [45] uses a wireless PIR sensor network to detect objects and humans for security applications and provide an estimation of the direction of movements. The network is implemented using Mica2 nodes and data gathered by a base station. Tracking algorithms are implemented on the nodes and speed calculation provided accurately, even if influenced by the orientation of the sensors. Slightly different is the work of Hashimoto et al. [79] where an array of PIR is used to count the number of people moving through a gate. Since the sensor can only detect temperature changes, the incident radiation flow is modulated by a chopper wheel that temporarily obstruct the PIR Field Of View (FOV). The data from the sensors is processed by a PC.

Sensor networks implemented with PIR are useful where privacy must be preserved together with security. In [170] cameras and PIR sensors are deployed respectively in public and private areas, and their information combined to correlate events such as tracking human motion and undesired access or presence in private areas, such as theft. This work demonstrate benefits of reducing camera deployment in favor of PIR sensors and reports results from a survey on 60 people, stating that people consider motion sensors less invasive for their privacy than cameras.

PIR sensors are often combined with vision systems and other kind of sensors in research focused on robot navigation and localization. In Sekmen et al. [181] a sound source localizer and a motion detector system are implemented on a human service robot called ISAC, with the purpose of redirect the attention of ISAC. The motion detector system use an infrared sensor array of five PIR sensors and it is integrated with the vision system of ISAC to perform real-time human tracking, in an inexpensive way.

Combining PIR sensor with video systems is a common approach to im-

prove video analysis. In [22] PIR sensors are used to provide a trigger event in a motion-detection application based on cameras for tracking events at night. The appearance of an infrared radiating body set off the PIR sensor, which turns on a floodlight enabling the cameras to capture clearly an event such as animals passing by an outdoor detected area. In Araujo et al. [3] PIR sensors are used to distinguish a still person from the background and then perform a correct tracking of people. Bai et al. present a system based on an ARM board with a Camera module being triggered by a Pyroelectric Infrared Sensor (PIR) which senses changes in the external temperature from an intruder. The system captures the relevant images and send them to a remote server. Finally, a number of publications exploit the integration of PIR sensor to improve camera based localization [6, 32, 33].

5.3 Direction and number of people detection

In this section we present the design and development of a low power and low cost wireless sensor network which, by means of PIR detectors, is able to extract the number and direction of movement of individuals moving through a gate or a section of a hallway. The network used as test-bench consists of four wireless nodes: three sensor nodes, equipped with a PIR detector, and a coordinator node, which gathers, analyzes and sends via RS232 to a PC the number and direction of people passing through. The wireless infrastructure is based on Zigbee protocol [243].

The proposed approach has very low computational requirements, it is therefore well suited for systems with limited computational resources, such as 8-bit microcontrollers, typically used on sensor nodes. Moreover, it enables fast recognition of occurring events, to obtain a highly reactive system. To achieve this we propose an efficient multilevel data analysis, where processing is distributed among nodes: the end-nodes extract the features, while the coordinator infers the event happened. This allow the exploitation of the intrinsic parallelism within the sensor network.

5.3.1 System description

The wireless sensor node we implemented is built on top of a Zigbee developer board (SARD) [69] which already includes all the necessary components to implement a Zigbee node. The board uses a GT60 microcontroller of the 8-bit family HCS08 by Freescale together with the MC13192 transceiver. The detector used for our sensor nodes is Murata IRA E710 [150], which present the characteristic shown in table 5.1. The output of this sensor must be ampli-

fied several hundred of times in order to achieve signal amplitude that can be handled by the microcontroller with a supply voltage of 3V and a 8-bit ADC. In particular, our signal conditioning circuit is a double stage amplifier which achieve a total amplification of about 1400 and operates as a band-pass filter between 0.57Hz and 11Hz which is suitable for detecting moving people [160]. Furthermore it bias the output voltage at $\frac{V_{dd}}{2}$ when no movement is detected. The conditioning circuit board includes also a low power voltage regulator used to decouple power supply lines from the transceiver ones.

| | |
|---|-----------|
| Responsivity (500K, 1Hz, 1Hz) (mV p.p.) | 4.3 |
| Field of View ($^{\circ}$) | 90 |
| Spectral Response (μm) | 5-14 |
| Supply Voltage (V) | 2-5 |
| Operating Temperature ($^{\circ}C$) | -40 to 70 |

Table 5.1: Murata IRA E710 PIR sensor characteristics.

Commercial presence detection systems currently in use aim to cover wide area with a single PIR. The only thing that is required is to obtain an indication whether somebody is moving in the covered area or not. Thus, Fresnel lens arrays are made of a number of elements and span over several tens of degree width. In this situation the output signal depends on all the components of incident radiation through any lens.

The novel idea behind our system is to reduce the number of Fresnel lenses used for the array and to augment the number of PIR sensors placed in the area. In particular we choose to reduce at minimum the horizontal span while keeping a wide vertical span. To achieve this we used the package of a COTS PIR presence detector, IS-215T [87], and shielded the unwanted elements of the provided lens with metallic tape. The lenses left uncovered were chosen as the three central ones, on the top, in the center and on the bottom (see figure 5.7).

In case of dual element sensor, as Murata IRA E710, each cone of the FOV associated to a lens must be divided into two adjacent sub cones. As a body moves, the elements see the change in radiation flux in sequence causing two opposite peaks (see figure 5.6). Here, since only one lens is not shielded, as the body moves in front of the PIR only a couple of peaks will be produced. An example of PIR output as a person moves back and forth is presented in figure 5.8.

The communication within the nodes of the network is based on Zigbee protocol [243]. The whole network is made up of four wireless nodes: a coordinator node and three sensor nodes placed in a row with different orientations (see figure 5.9).

The choice of a wireless solution is motivated by three reasons:

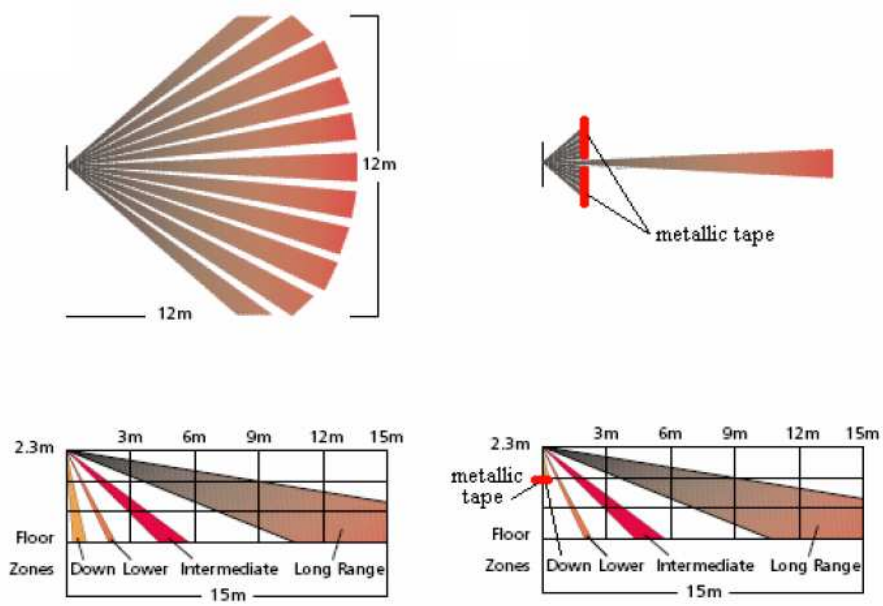


Figure 5.7: Modified FOV of the PIR sensor

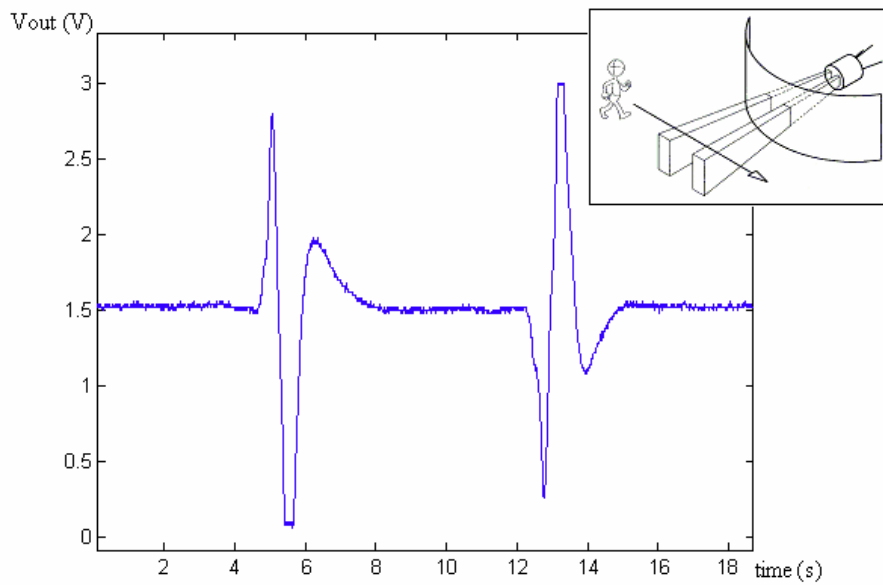


Figure 5.8: Output of the PIR sensor when a single lens is used and a person moves back and forth in front of it.

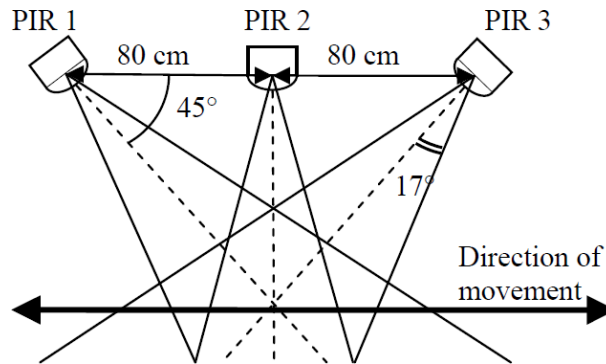


Figure 5.9: Setup of the PIR network used to detect number of person and direction of movement.

1. we plan to include the cluster of nodes presented here within a complex and heterogeneous system made of different kind of sensors;
2. we need flexibility in the number and position of nodes;
3. to increase scalability we want to keep some computation within the node and send only higher level information, therefore we dont need high bandwidth;

Among wireless protocols the one that fits better our needs is Zigbee. This protocol has low power, low data rate characteristics, it allows high flexibility and it enables coverage of a relatively wide area, by means of multi-hopping. According to Zigbee specification the coordinator node, after initialization, starts the network. On the other end the sensor nodes associate with the coordinator and send the data from PIR sensor. The coordinator acts also as a sink node toward a PC via RS232. On the PC a simple application collects the packets from the network.

5.3.2 Model and system analysis

As mentioned above, the objectives of our project are to recognize the number of people and direction of movement through a gate. At present, we tested the system to detect direction and number of people in five situations: one, two or three people passing in line or side by side (see figure 5.10).

This system will be included in a wireless sensor network. Thus the approach that we propose is suited for devices with limited computational power. To accomplish its objective the 8-bit microcontroller placed on the SARD detects the number of peaks at the output of the sensor conditioning circuit and the duration of one fixed peak, as will be clarified in next paragraphs. This task can be achieved quickly while the person walks through; therefore as soon as

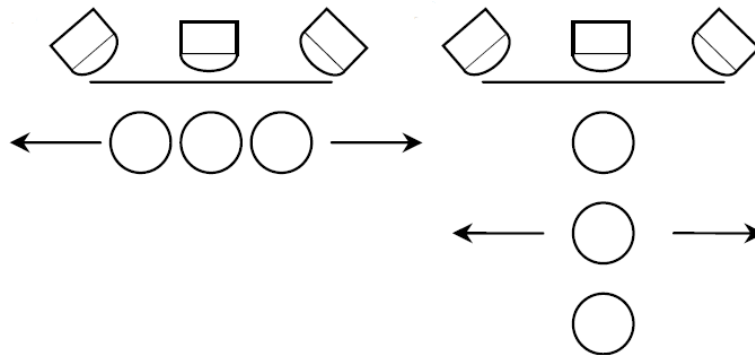


Figure 5.10: Events of interests: passages of 1, 2 or 3 persons in sequence or in parallel.

the end of an event is detected the sensor node can send its features at the coordinator which, by means of few additional controls on data, can detect the number of individuals and direction of their movement. The resulting system is then highly reactive and can produce an indication within few hundreds of millisecond after the passage. This time delay depends mainly on the time required to the sensor nodes to access to the wireless channel and send 2 or 4 bytes of data (depending on their position) at 250kb/s, but can be considered negligible due to the fact that each passage duration is in the order of seconds.

Single person moving

The PIR output greatly depends on how the people move within the field of view of the detector. Speed and distance from sensor and size of the person can heavily influence the output of the sensor. Also, when two people walk in line, the resulting signal depends on several variables such as distance from sensor, reciprocal distance, speed, etc.

Narrowing the FOV of the detector to only one column of lenses has two positive effects:

1. reduces the area where a person affects the output signal helping in distinguishing different people moving in series and movement back and forth;
2. each passage produces only a couple of peaks where the first can be seen as an indication of direction of movement (see figure 5.8).

With this simple consideration we are able to detect direction of movement of a single person by means of a single detector by looking at first peak direction. However, some issues arise: noise immunity, segmentation of events (detection of the beginning and end of an event).

There are three main sources of noises

1. *period of adjustment.* When a person exits from the FOV of a sensor a negative change in radiation flux occurs causing oscillation around bias output voltage;
2. *background temperature change.* Due to normal daily fluctuation of temperatures;
3. *power supply.* Due to the rapid change from active to off state of the wireless unit which induces spikes on Vdd lines.

Use of a dual element PIR detector helps to reduce the period of adjustment and also the influence of changes in background temperature.

The spikes on Vdd lines due to the wireless module are amplified by the sensor conditioning circuits and produces heavy changes in the output signal. To overcome this problem alimantation decoupling is then needed.

Segmentation of events is obtained simply using two thresholds 0.3V above and below the bias output voltage, respectively at 1.2V and 1.8V. An event starts when one of these two thresholds is broken while it stops when, for a 1 second period, the output signal stays between the two thresholds. The choice of 0.3V is a compromise between the sensitivity of the detector and the ability to distinguish successive events. In fact, as it can be seen from figure 5.11, a higher threshold may cause the impossibility to recognize events generated by an individual moving at high speed or far from the sensors, while the event duration can be much shorter. On the other hand a lower threshold will allow far or faster people to be detected but will result in a longer duration of the event. Not to be forgotten, when thresholds are too low, noise may cause unwanted events.

Group of people moving

People moving in line. When people walk in front of the PIR detectors in line we expect to see a number of peaks proportional to their number. In this situation the choice of a narrow FOV results in PIR 2 output (see figure 5.9) depending only on one person at a time. Consequently, for each one we see a couple of peaks. Obviously, the direction of the first peak indicates the queue direction of movement.

People moving side by side. In this case we expect that the output of the sensor is highly influenced by how the group moves and by the individuals body size. The main effects are two:

1. shielding of the closest person on the other ones;
2. oscillations due to more than one person in the FOV.

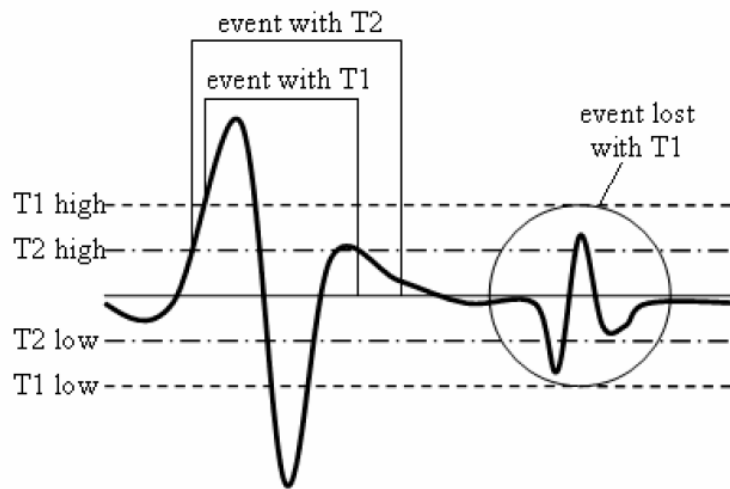


Figure 5.11: Reactiveness-noise immunity trade off in threshold selection.

When the group is walking exactly one beside the other, we expect that the PIR 2 output presents the same shape than in one person case. The only observed difference is that the dynamic of the signal has a longer duration. Preliminary tests showed that the duration of the peaks can be used to distinguish between one person and many people traversing, but cannot give a clear indication on the exact number of individuals passed. In table 5.2 we present the maximum, minimum and average duration of the two peaks of the waveform collected from PIR 2 during our tests.

| | 1 st peak (ms) | | | 2 nd peak (ms) | | |
|---|---------------------------|------|------|---------------------------|------|------|
| | min. | avr. | max. | min. | avr. | max. |
| 1 | 300 | 450 | 630 | 470 | 590 | 680 |
| 2 | 310 | 480 | 640 | 640 | 710 | 870 |
| 3 | 480 | 560 | 650 | 680 | 800 | 880 |

Table 5.2: average, max and min duration of 1st and 2nd peak when one, two or three people are passing.

As can be seen from the table, the mean value of the second peak distinguishes clearly between the presence of one individual and many individuals, but range is high. However, this feature can be useful to differentiate situations where people are crossing the FOV in group from individuals traversing in sequence.

Using the setup presented in figure 5.9 helps to overcome the problem of closer people shadowing the farther ones. In fact, with such configuration at least one PIR sees the people moving side by side almost like they are moving

in series. The output of the PIR whose orientation is orthogonal to direction of movement is similar to the case of a queue of people passing, especially if the 2 users are walking far one from each other. This is a consequence of the fact that they enter in the field of view of the PIR in successive instants, thus they produce two partially overlapped outputs as the case of one person only (see figure 5.12).

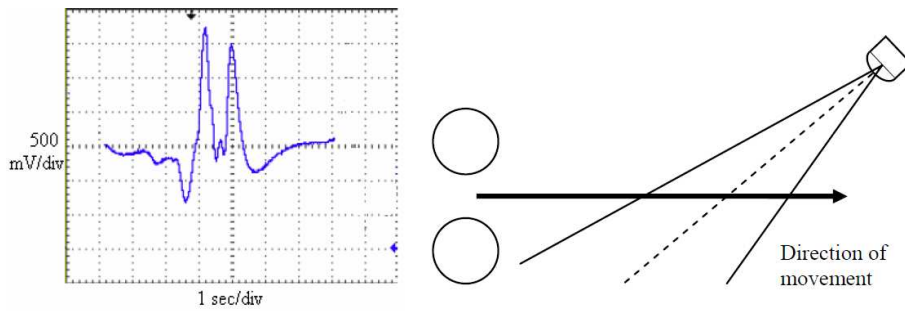


Figure 5.12: Output of the "correct" side PIRs when 2 people are passing.

We used three sensors because, depending on the direction of movement and the relative position, we expect to lose information due to the shadowing of the tester closer to the sensors on the others, as shown in figure 5.13.

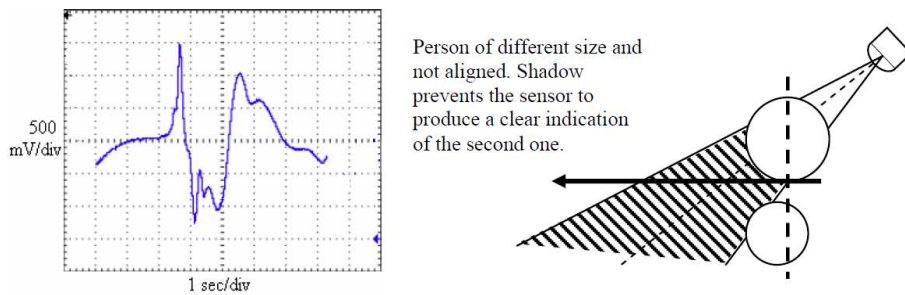


Figure 5.13: Output of the "wrong" side PIRs when 2 people are passing. Shadowing is highlighted.

Use of an array of three PIR detectors placed as shown in figure 5.9 allows detecting direction of movement and number of users moving in both directions by looking at the number of peaks and the direction of the first peak. However some issues arise in the implementing phase:

1. detecting which sensor gives the right information on the number of people;
2. detecting each peak.

Depending on direction of movement and relative position we assume that only one out of three sensors is in the best position to recognize the number of people passing. According to this only one RF module needs to send its information. However, each sensor itself is not able to know if it is the one or not. In such situation we decided to use a multilevel data analysis: the single sensor detects direction (e.g. upper or lower peak) and number of peaks then sends such information to the coordinator, which works as a cluster head. Having information from all the nodes, the coordinator understands how the testers are passing.

Peak detection is made using threshold. In particular, a positive peak is detected when the signal is lower than a certain threshold under a previous detected local maximum. Vice versa a negative peak is detected when the signal breaks a certain threshold above a previous local minimum. Tests have shown that an optimal threshold is 0.7V. After we detected a maximum we look for a minimum and vice versa. The first peak indicates the direction of movement of the individuals in the FOV.

5.3.3 Data fusion

We propose a simple method to distinguish between the five proposed situations.

According to the previous considerations, each wireless sensor node samples the output of the PIR detector and then identifies the number of peak pairs (one positive and one negative) and the direction of the first peak. Moreover the sensor placed in the middle (PIR 2 in the configuration of figure 5.9) detects the duration of the second peak. These eight features are sent to the coordinator which infers the number of people and direction of movement. We assume that the coordinator knows the relative position of the three nodes.

Direction of movement is detected looking at the indication of the three sensors. In the event of contradictory inputs, the direction is considered the one suggested by at least two out of three nodes.

Number of people is extracted by means of four features: number of peaks detected by each node and duration of the second peak measured by the central node. Firstly, the coordinator looks at the length of the second peak measured by PIR 2. According to Table 2 we specified a threshold of 0.64 sec: if the period is shorter than 0.64 sec it means that people are moving in line. As a consequence, the number of couples of peaks detected by the central node indicates the number of people passing. Otherwise more than one person is passing side by side. Depending on direction, the number of people is now indicated by the number of pair recognized by one of the two sensors on the side.

The coordinator, knowing the relative position, can then choose the correct one.

5.3.4 Evaluation

To confirm our hypothesis we performed five sets of testing set. During each test, one, two or three people moved forward and back ten times along the longer direction of the room. In multiple people cases, testers walked both side by side and in sequence. For every situation we performed 20 tests, 10 in one direction and 10 in the other, for a total of 100 experiments.

Table 5.3 summarizes the results.

| Number of people | Correct direction | Correct number |
|------------------|-------------------|----------------|
| 1 | 20/20 (100%) | 19/20 (95%) |
| 2 sequence | 20/20 (100%) | 20/20(100%) |
| 2 side by side | 20/20 (100%) | 17/20 (85%) |
| 3 sequence | 20/20 (100%) | 20/20(100%) |
| 3 side by side | 20/20 (100%) | 13/20(65%) |

Table 5.3: Experimental results.

As can be seen from table 5.3, we achieved 100% correct recognition of direction of movement applying the rules proposed and using the threshold indicated to identify the start of an event. It worth notice that the middle sensor node always infer the correct direction, therefore it is sufficient for the extraction of this parameter.

Following the rules proposed we achieved 89% accuracy on detecting the number of people passing. However, it is necessary to distinguish between the tests where people are walking in line and when they move side by side. We had 59 experiments (98.3%), within the first subset, where the correct number of people is extracted. In figure 5.14 the output of the middle sensor when one, two or three people are passing is illustrated.

The accuracy drastically reduces to 75% in the subset of experiments where people walk side by side (30 correct identification over 40 tests). In figure 5.15 the output of the three detectors is shown when three people are walking through. In this situation the accuracy results to be much lower than in the other. This is mainly due to two reasons: shielding effect and handmade sensors. In fact, the closest tester walks only few centimetres away from the sensor, thus slight differences in reciprocal position caused a shielding effect that undermined the gathered data. Moreover, the board with the sensor conditioning circuit was built in our lab, while the package and the lenses are the one from COTS presence detectors: slight misalignment and a non accurate distance between the sensor and the lenses compromised the correct detection of people.

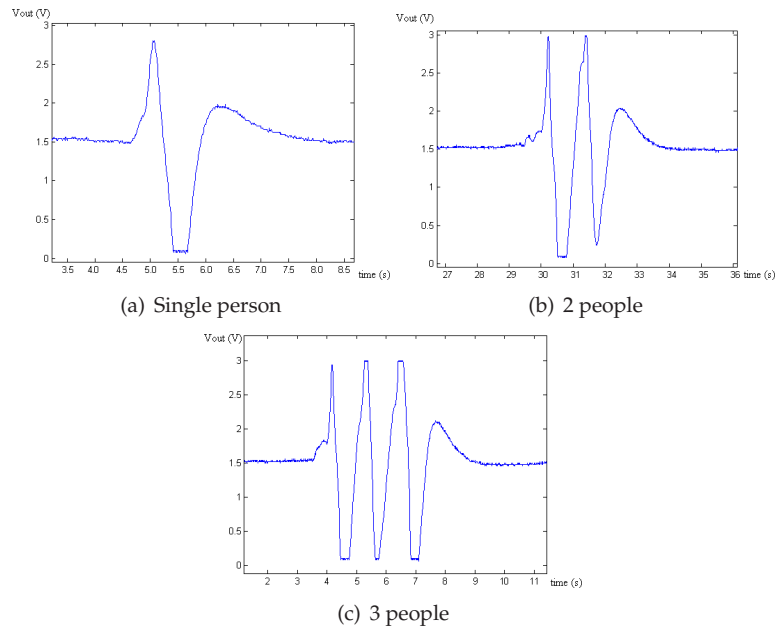


Figure 5.14: Output of the middle pir when 1,2 and 3 people are passing in a row.

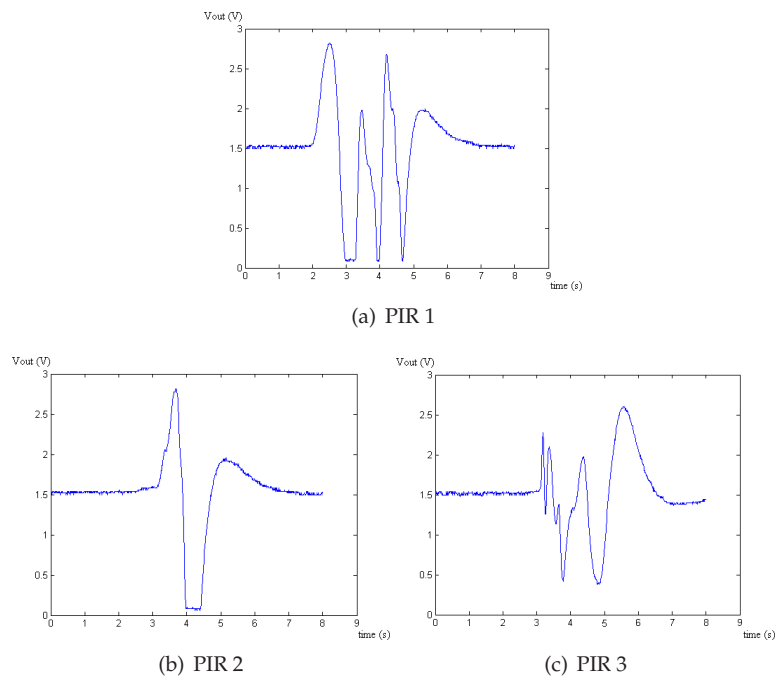


Figure 5.15: Output of the three PIRs when three people are passing side by side. PIR 1 is in the "best" position, the central PIR (PIR 2) shows an output similar to the one in the case of single person, PIR 3 output is affected by shielding effect.

5.3.5 Conclusion

In this chapter we introduced a novel approach to detect direction of movement and number of people passing in a hallway or through a gate by means of a set of three PIR detectors. We proposed a novel configuration for the sensors and a set of features that can be easily calculated by typical wireless sensor nodes based on low power microcontrollers (e.g. 8 bit ones). The detectors are located along a wall of a hallway. A coordinator node, which works also as a sink node, collects the features sent from the three wireless sensor nodes and it can infer direction of movement and number of people.

The technique proposed allows high reactivity. Results of the detection are available few hundreds of milliseconds after the event is ended. Tests show that in 100% of the cases our system has been able to detect correctly direction of movement, while number of users has been identified correctly in 89% of our tests.

5.4 Distance estimation

In this section a novel technique to detect person position through the use of an array of PIR detectors is proposed. The basic block of the network includes two PIR detectors placed on opposite walls of a hallway or a gate and facing each other. Each block is able to autonomously classify in real-time passages between the PIRs into three classes according to the distance of the person from the sensors, thus resulting in high system scalability and flexibility. The use of three different classifiers, namely Naïve Bayes, Support Vector Machines (SVM) and k-Nearest Neighbor (k-NN), is evaluated and performance vs. cost trade offs are explored.

5.4.1 System description

Our objective is to detect the distance of a moving person from the walls of an hallway. Since we do not need a precise estimation of the position we roughly divide the hallway into three separate zones: close to sensor 1, middle, close to sensor 2 (see figure 5.16). Each couple of PIR detectors monitors the passage through a thin section of the hallway.

We used the same hardware described in section 5.3.1. Also here, we shielded all the lenses of the IS-215T package except the central ones in order to narrow the FoV of the sensors.

The whole AoI is covered with several couples of PIR sensors, according to the application need.

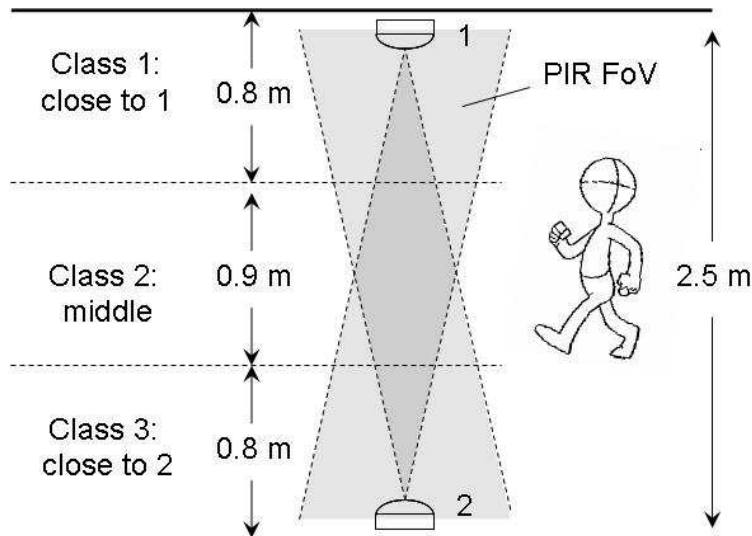


Figure 5.16: The building block of the network is made up of two PIRs that autonomously monitor a slice of the AoI. The space between them is divided into three zones.

5.4.2 Model analysis

Figure 5.17 shows the PIR output as a function of distance.

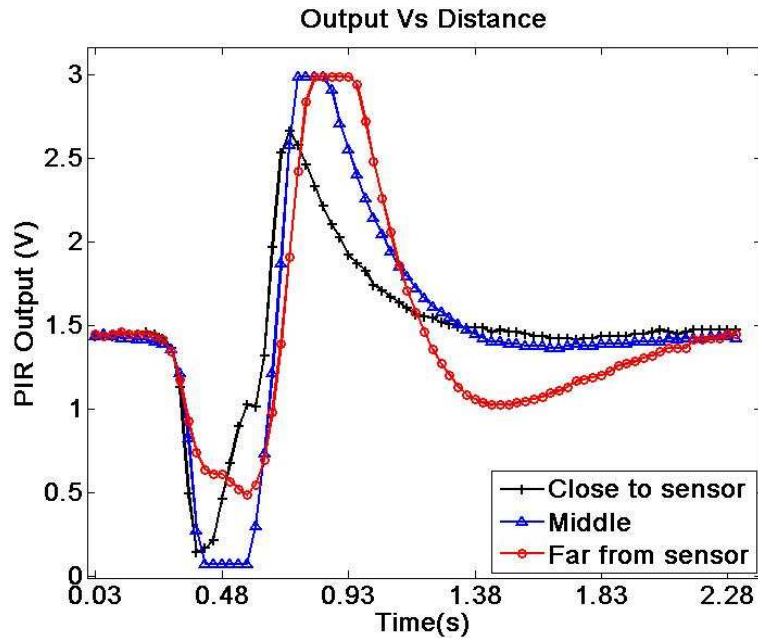


Figure 5.17: Output of a PIR sensor in case of passages at different distances.

From this plot, we can see how signal duration increases with distance while signal amplitude is at a maximum for passages in the middle position.

Signal duration increase is due to the FoV conic shape. In fact, a PIR is mostly sensitive to entrances and exits from its FoV and these two instants are more distant when a person walks far from the sensors.

Output peak-to-peak amplitude decreases with distance because far bodies result in a smaller change in the incident radiation. Amplitude reduction for closer passages is due to the interaction of the two sensitive elements. In figure 5.18 we highlighted each elements' FoV. In proximity of the sensor the two FoVs are overlapped, thus compensating each other.

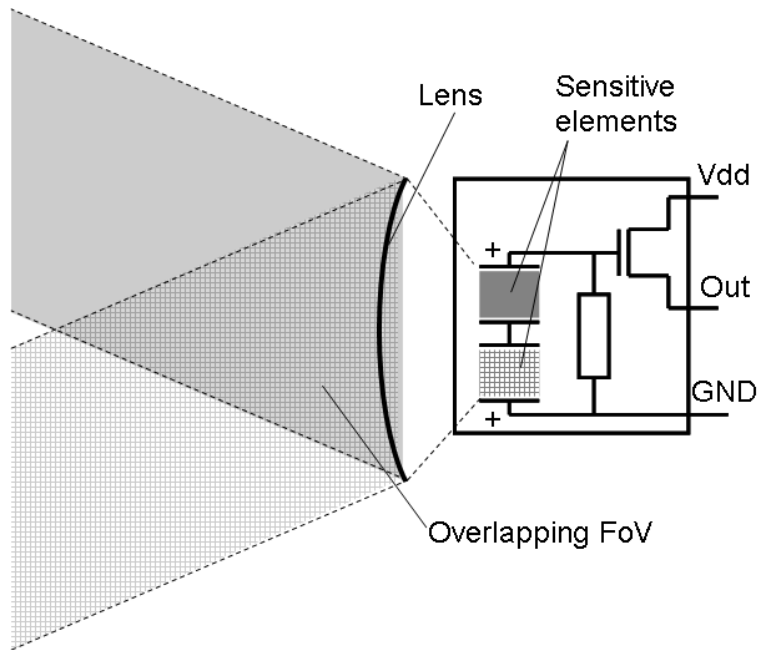


Figure 5.18: Schematic of a typical C.O.T.S. PIR. Two sensing elements are used in series with opposite polarization, the output is pre-amplified through a built in MOS transistor. Highlighted with shading, the FoV of each sensing element. Notice how, in proximity of the device, the two FoVs are overlapped.

In case of isolated people, each passage can be easily segmented using two thresholds above and below $\frac{V_{dd}}{2}$. The starting of the passage is detected when one of the threshold is broken, the end when the PIR output remains between the threshold for a certain time T . According to results from previous work [238], we placed the thresholds at $\frac{V_{dd}}{2} \pm 300mV$ and $T = 1sec$.

When a passage is detected, each sensor extracts its duration and the PIR output amplitude. These two features are wirelessly sent to a central unit in order to evaluate the distance of passage, thus reducing the power consumption related to wireless communication and the bandwidth required. The central unit calculates the ratio between homogeneous features (duration and amplitude). Therefore each passage results in a two-elements vector of features (relative duration and relative amplitude) with whom we estimate the position of

the person ((see figure 5.19)).

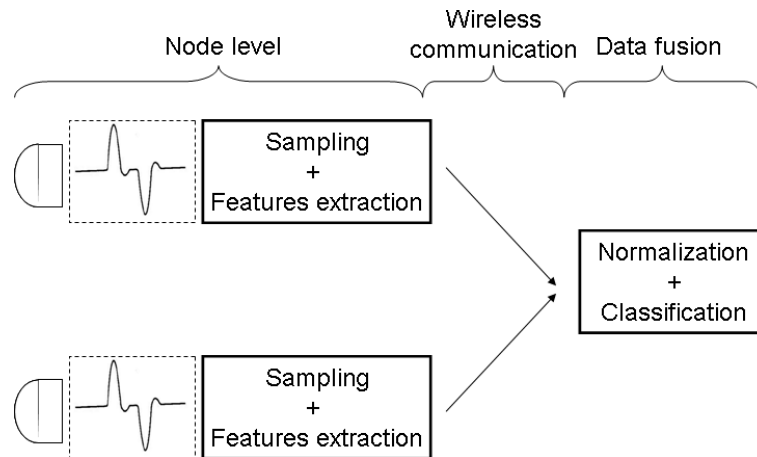


Figure 5.19: Task allocation for distance detection.

In figure 5.20, we plotted such vectors for a subset of samples from passages at different distances. As can be seen from this figure, it is not possible to define well separated region of the space for each distance of passage, so we decided to rely on a classifier in order to estimate it.

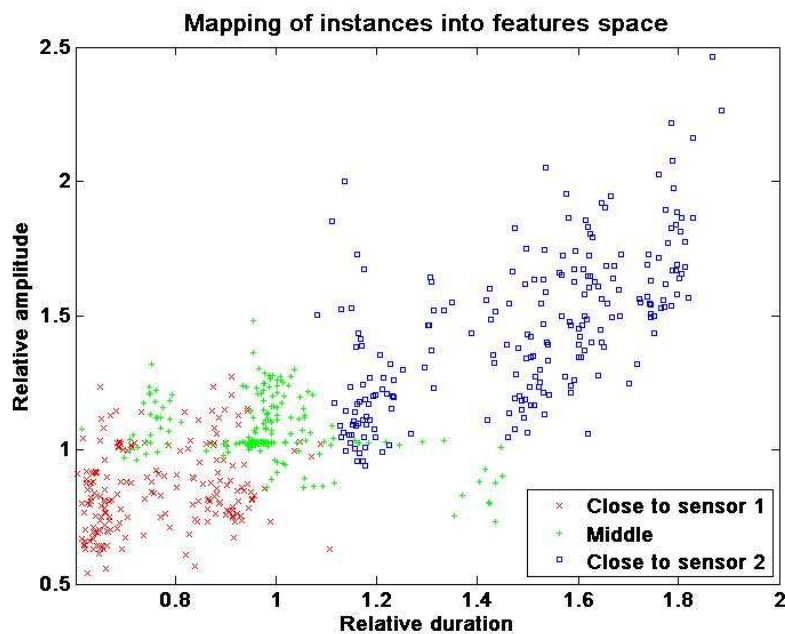


Figure 5.20: Mapping of input vector in the two dimensional feature space. The three classes are located into partially overlapped areas of the space.

We tested and compared the use of three classifiers: Naïve Bayes, Support Vector Machines (SVM) and k-Nearest Neighbor (k-NN) (see sections 4.3.1,

4.3.2 and 4.3.3).

These classifiers fall into the category of supervised classifiers. Such classifiers require a starting, off-line, usually computational heavy training phase during which their parameters are learned from a set of training instances (*training set*). On the other hand, classification of instances during normal operative phase is a lightweight task that can be implemented in real-time on low cost, low power devices, thus allowing distributed implementation through the sensor network.

5.4.3 Test and results

To validate our approach we collected about 200 instances for each class. In order to test the selected classification algorithm we used the Waikato Environment for Knowledge Analysis (WEKA) software developed at the University of Waikato [230]. The algorithm used are: NaiveBayesSimple for Naïve Bayes, SMO with polynomial kernel for SVM, and IBk for k-NN.

To evaluate the results we used 4-folds cross validation, consequently the available instances from each class have been divided into four groups (folds), three of them have been used to train the classifier and one to validate it. The training and validation steps are repeated four times, each one using a different fold for validation.

Classifiers performance, calculated as correct classification ratio, are presented in table 5.4.

| Classifier | Correct classification ratio (%) |
|---------------|----------------------------------|
| Naïve Bayes | 83.49 |
| 1-NN | 92.47 |
| 3-NN | 93.75 |
| 5-NN | 92.95 |
| Linear SVM | 86.06 |
| Quadratic SVM | 86.06 |
| Cubic SVM | 87.50 |

Table 5.4: Classifiers performance (correct classification ratio).

In table 5.5 the computational costs to perform the classification of one instance is reported. Here N_{sv1} and N_{sv2} are, respectively, the number of support vectors for the quadratic and cubic SVM classifier and T is the number of reference instances for the k-NN classifier.

From these tables we can see how k-NN achieves higher accuracy with respect to the other classifiers. However, this classifier is the one with highest memory requirements to store the reference instances. Moreover, whenever a passage is detected, it is necessary to compute the distance between the new

| Classifier | Computational cost | Memory cost |
|---------------|--|-------------------|
| Naïve Bayes | 3 sum, 3sqr., 3 exp., 6 mul., 1 max | 15 |
| 1-NN | $T \cdot (3 \text{ sum}, 2 \text{ sqr.}, 1 \text{ sqrt.})$, 1 max | $2 \cdot T$ |
| 3-NN | $T \cdot (3 \text{ sum}, 2 \text{ sqr.}, 1 \text{ sqrt.})$, 3 max | $2 \cdot T$ |
| 5-NN | $T \cdot (3 \text{ sum}, 2 \text{ sqr.}, 1 \text{ sqrt.})$, 5 max | $2 \cdot T$ |
| Linear SVM | 6 mul., 6 sum, 2 max | 6 |
| Quadratic SVM | $2 \cdot N_{sv1} \cdot (1 \text{ mul.}, 1 \text{ sum}), 2 \text{ max}$ | $2 \cdot N_{sv1}$ |
| Cubic SVM | $2 \cdot N_{sv2} \cdot (1 \text{ mul.}, 1 \text{ sum}), 2 \text{ max}$ | $2 \cdot N_{sv2}$ |

Table 5.5: Classifiers computational effort to perform the classification of a single instance and memory cost (number of double) to implement the classifier. $N_{sv1} = 257$, $N_{sv2} = 235$ and $T = 300$

input vector and all references, thus resulting in higher latency in reporting the distance estimation than the other classifiers.

SVMs performs well, but from tables 5.4 and 5.5 we can see that an increase in classifier complexity results in little or null increase in classification performance. Thus the use of high complexity classifiers is not justified for this application. Finally Naïve Bayes method shows the worst performance, however it has little memory requirements and computational effort.

A deeper understanding of classification performance can be gathered looking at its confusion matrix. Table 5.6 presents, as an example, the confusion matrix when using Naïve Bayes classifier. By looking at the matrix we can see how instances from classes *close to 1* and *close to 2* are never confused, indicating limited uncertainty in position estimation. The other classifiers present this characteristic, too.

| | classified as | | |
|------------|---------------|--------|------------|
| | close to 1 | middle | close to 2 |
| close to 1 | 165 | 33 | 0 |
| middle | 15 | 180 | 12 |
| close to 2 | 0 | 43 | 176 |

Table 5.6: Naïve Bayes classifier's confusion matrix

5.4.4 Conclusion

Wireless sensor networks will provide great opportunities for researchers and developers. However several technical issues must be addressed when dealing with such systems, in particular size, cost and power consumption should be reduced and a large amount of data must be efficiently handled.

In this section we proposed a novel approach to estimate, using PIR detector, people position within a section of an hallway or a gate. Being passive, small and low cost PIR detectors are well suited within wireless sensor net-

works.

As a person moves within PIR's FoV, the sensor node locally calculate passage duration and PIR output maximum amplitude. With such information coming from a couple of detectors facing each other we estimated the position of a person moving between them. Estimation is performed with a classifier that classifies the passages into a set of three classes: close to one PIR, middle, close to the second PIR. Since our approach relies on a set of identical, autonomous building blocks it shows high flexibility and scalability.

We tested the use of three classifiers. We compared the classifiers in terms of correct classification ratio, computational cost and memory cost. We found that k-NN has better performance (up to 93.75% correct classification ratio) but requires more memory and computational effort than the other classifiers. On the other hand, Naïve Bayes and linear SVM result in lower performance (respectively 83.49% and 86.06% correct classification ratio) but have much more relaxed requirements on memory and computational cost.

5.5 Multimodal sensor network for video surveillance

This work reports the joint research of MicrelLab [139], which is part of DEIS, University of Bologna ([49, 204]), and the Image Lab [91], which is part of University of Modena and Reggio Emilia [205], developing a multimodal sensor network that integrates a wireless network of PIR-based sensors with a traditional vision system to provide drastically improved (in accuracy and robustness) tracking of people. It is worth noting that people surveillance is more interesting from the researchs point of view w.r.t. to vehicle tracking, because of the intrinsic complexity in detecting, tracking, and understanding human behavior: changes in posture and gestures, human interaction, presence of multiple people, and so on, make the problem challenging and interesting for the computer vision community.

5.5.1 Integrated multimodal sensor networks

Vision systems achieve good accuracy when working alone, but they definitely could benefit from the multi-modal integration with PIR sensors. For testing the integration, a test bed has been created at our campus. Figure 5.21 shows the location of cameras and PIR sensors. The system we implemented is composed by several modules, working in parallel on different threads (see figure 5.22). In particular, a thread is generated for each camera, devoted to compute the list of people present in the scene exploiting a two stage processing

(segmentation and tracking). All the camera threads are tightly connected to a coordinator thread, that detects if the same person is visible in more than one camera, and, in such a situation, it labels the corresponding tracks with the same identifier.

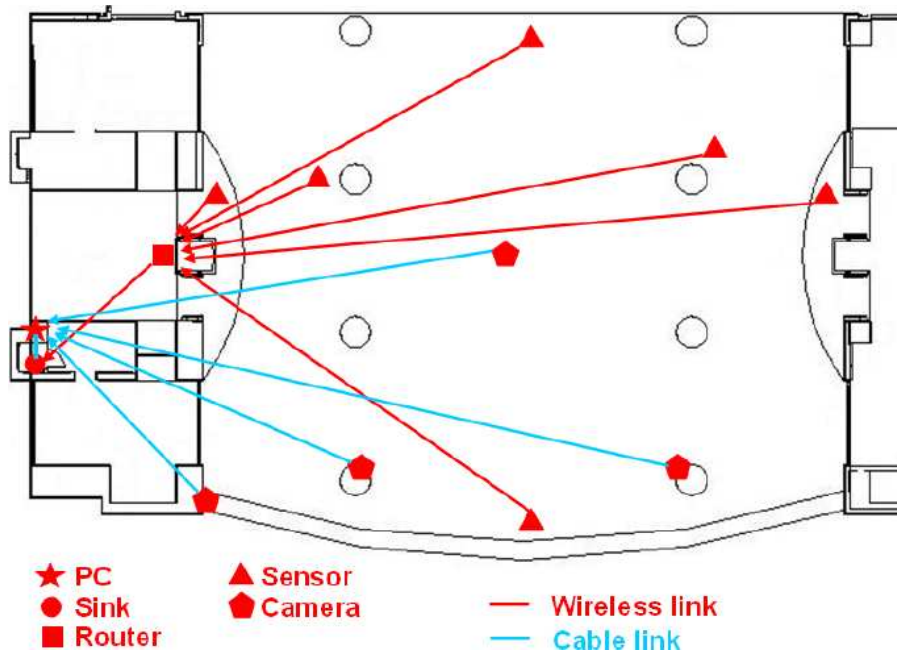


Figure 5.21: Map of the test bed system.

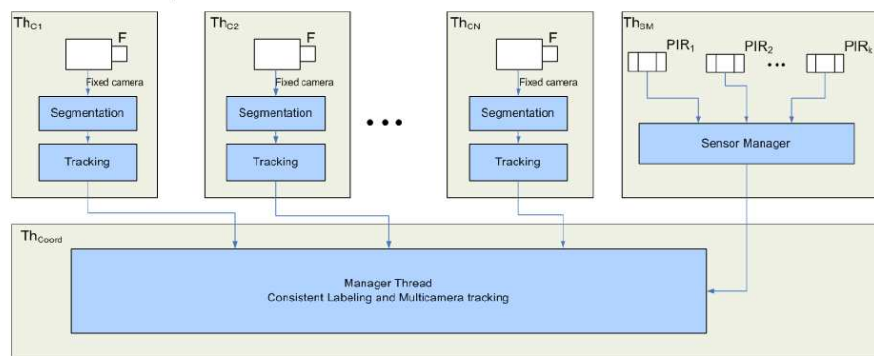


Figure 5.22: Software architecture of the system.

At the same time, a sensor manager coordinates the network of sensors distributed over the monitored area. Observing the output of a couple of PIR, the microcontroller integrated on the sensor node is able to detect both presence and direction of movement of the person walking by it. When such situation is detected the microcontroller creates and wirelessly sends a message to a spe-

cial node which acts as a sink. The sink then forwards the message to the sensor manager via RS232 cable in order to make the information available to the tracking and labeling algorithms.

Eventually, data coming from cameras and sensors are collected and managed by a supervisor thread. The coordination between cameras and sensors is twofold. Each time the vision system requires more detailed or reliable information about the presence of people in the zones monitored by the sensors, it sends requests to the supervisor thread. Contemporaneously, when the sensor network detects a particular event, the manager takes care to inform the involved cameras.

5.5.2 PIR sensor network

The wireless sensor node that we used in this project is slightly different than the one presented in 5.3.1.

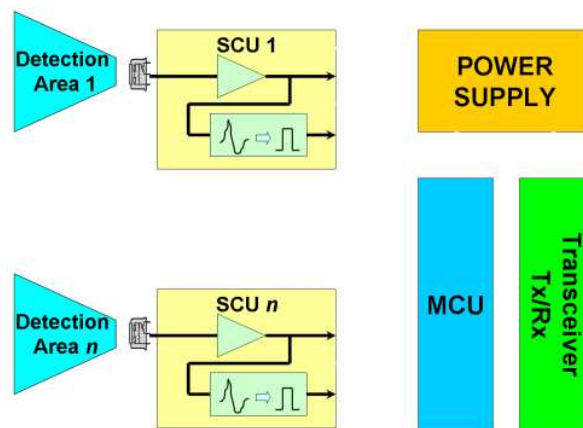


Figure 5.23: General architecture of the PIR sensor node.

The general architecture of a single node is shown in figure 5.23. More than one PIR can be connected to the microcontroller unit (MCU) through the signal conditioning unit (SCU). Each PIR sensor is equipped with a Fresnel lens from Murata [150] (see figure 5.24), which is used to shape the detection area while IR filter is used to limit incoming radiation between 8 and $14\mu\text{m}$, typical of human body radiation range. By suitably shading its Fresnel lenses, we were able to obtain a cone of coverage with a vertical angle of 60 degree and an horizontal angle of 38 degree.

Furthermore, a single PIR sensor can detect the direction of movement. Figure 5.25 shows the signal detected by sensor when a person passes through the area under control from left to right (the first peak is negative) and from right to left (first peak is positive). The PIR output sensor conditioning circuit as well

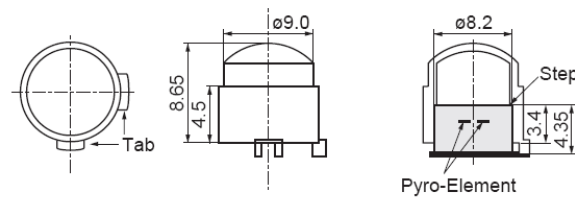


Figure 5.24: Schematic of the Fresnel lens used in this project.

as the microprocessor and transceiver (TX/RX) units are the same as in 5.3.1.

Digital output is obtained through use of a programmable comparator, with I2C serial interface, where a threshold can be set. However, in our work analog output is also considered to extract more complex information than the simple presence.

In our setup, each node includes two PIR sensors.

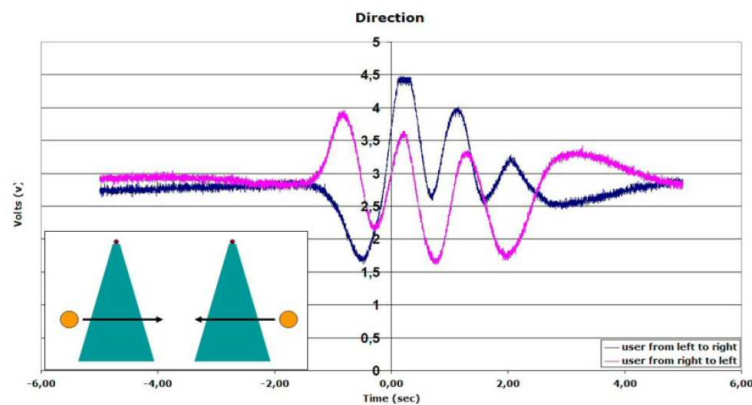


Figure 5.25: Signals detected by sensor: when a person passes through the area under control from left to right the first peak is negative and from right to left the first peak is positive.

5.5.3 Sensing and data collection

Figure 5.26 shows the processing data flow from event acquisition to generation of the packet which will be sent by the wireless nodes.

We are interested in detecting precisely presence and direction of movement, but also more complex movements such as changes in direction within the covered area. In fact, these are information that can be exploited by the vision system for enhancing the accuracy of the video surveillance application, in which presence and direction of movement (of people) are key information.

As outlined above, we augment the information produced by a single node by using 2 PIR sensors (figure 5.27(a)) per node. The typical sensors output

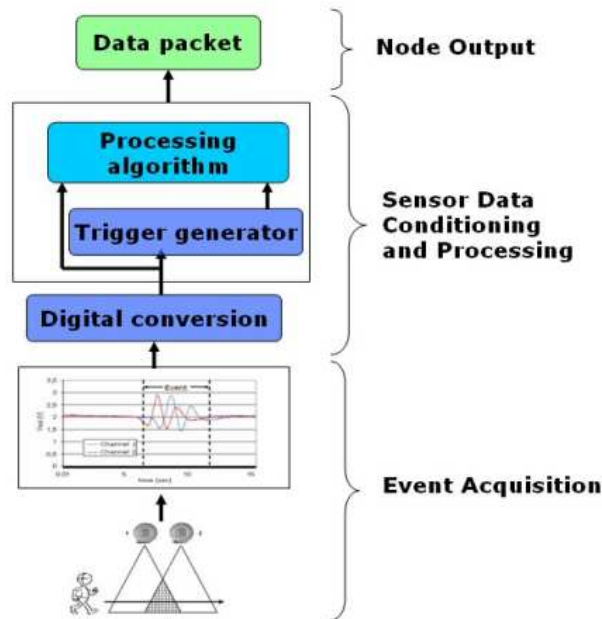


Figure 5.26: Event acquisition and Sensor Data Conditioning and Processing.

when a person is walking through the sensor area is the one presented in figure 5.27(b). The signal collected by the sensors is digitally converted to be processed by the microcontroller. When a person crosses the monitored area each of the two sensors generates a waveform similar to the one in figure 5.25 depending on the direction of movement. We consider interesting events those stimulating a significant variation of the signal (figure 5.27(b)): when the input coming from the digital converter exceeds a lower or an upper threshold, a trigger is generated to start the processing algorithm in charge of extracting information from the signal.

The analysis, as mentioned above, is aimed at understanding the direction of a person walking in the covered area. Assuming that one person is moving from left to right as in figure 5.27(a), he will be detected first by PIR_i then by both PIR_i and PIR_j and at last only by PIR_j as it is lightened in figure 5.28. In general, a different activation sequence can help identifying changes in direction of movement within the area covered by the array of sensors. Results from the processing is a message containing information about the presence and/or direction of movements in the selected area.

Note that the trigger generator is disabled for a period to be set depending on the application after the detection of an event, avoiding redundant information to be sent. In our case, the period is set at 2 seconds. This choice has been verified as not influencing correct analysis, because it does not cause loss of events.

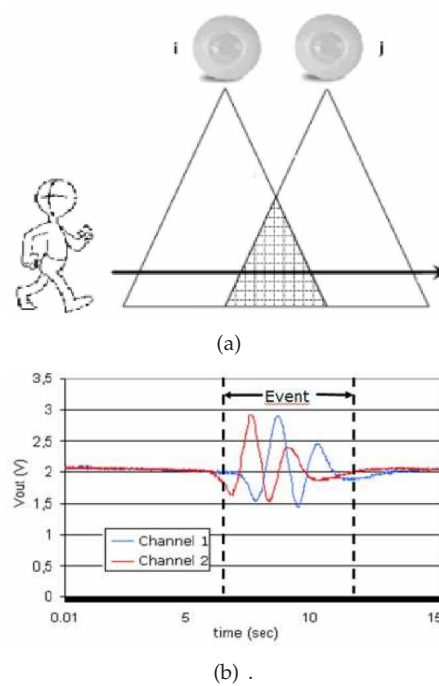


Figure 5.27: Sensor node composed by two PIRs

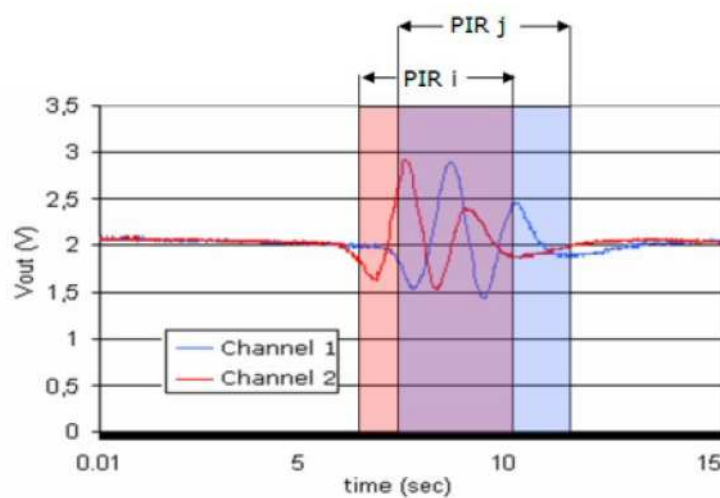


Figure 5.28: PIR activation sequence.

The network has a star topology, i.e., all the nodes are end devices and communicate only with a central one, the coordinator. The central node (bridge) collects data from the sensor nodes and sends them to another node (sink) which communicate through its RS232 interface to a PC (see Fig. 1). Hence, in our application the bridge is the network coordinator while the other nodes are end devices. The sensor nodes, which are located in the courtyard, are

battery-powered while the bridge and the sink are main powered.

This topology is suitable to the characteristics of the monitored area. In fact, the sensor are located in a courtyard outside the building while the PC, due to privacy issues, is locked inside a small room, which must be kept closed within the building. Some tests shown that only the sensors close to the door of the building are able to communicate with a device inside the room, while all the courtyard can be covered by a receiver located close to the door.

As already mentioned, the information collected by the sensors are sent to the video processing server via RS232 cable. We decided to use an asynchronous communication, that is, the sensor network send data to the server as soon as it collects them. Each message is made up of a start byte (the ASCII code I), a sensor ID, an area ID, an indication of length (the number of following couples name-value, see table 5.7), several couples name-value and a stop bit (the ASCII code F) (see figure 5.29). Start and stop bit are used for synchronization. Area and sensor node ID are used to uniquely identify the node.



Figure 5.29: Communication protocol between nodes and sensor manager.

| Information | Code | Value | Code |
|-------------|------|---|------|
| Presence | 1 | Present | 1 |
| | | Area free | 16 |
| Direction | 2 | From PIR _i to PIR _j | 48 |
| | | From PIR _j to PIR _i | 192 |

Table 5.7: Adopted codes

5.5.4 Vision system

The vision system has been developed by the ImageLab at University of Modena and Reggio Emilia.

Single camera processing

Many approaches to people detection and tracking by single cameras have been proposed in the literature. Their schemes are often similar: first, to perform motion detection by separating points belonging to still parts from points belonging to moving parts (by means of background suppression, frame difference, or statistical analysis); then, blob analysis aims at grouping spatially correlated points into objects and characterizing them by visual features and

motion components; eventually, moving objects are tracked with the aim of keeping track of their identity to further analyze the behavior.

Our approach from single camera follows this scheme, and it is composed by two main modules: segmentation and tracking. The first module aims at extracting visual objects, that are entities that we are interesting in and that we want to analyze separately with respect to the background. Normally, the visual objects are objects detected in motion for almost one frame. To this aim, background suppression techniques are often adopted and operate by subtracting the current background model B^t from the current frame I^t . The points are extracted and grouped with a labeling process into a set FO^t of foreground objects at instant time t . This set contains both relevant objects and other outliers, such as shadows and noise. To identify shadow points, we used a deterministic approach, proposed in [37], based on the assumption that shadows have similar chromaticity but lower brightness than the background on which they are cast.

Objects in the set FO^t considered too small are discarded as noise. The set VO^t of visual objects obtained after the size-based validation is processed by the tracking module that computes for each frame t a set of tracks $Tt = \{T_1^t, \dots, T_m^t\}$.

In the case of people tracking, the basic tracking approaches (based on directional rules, or Kalman filters) are not suitable, since humans undergo to deformation in the shape, move with unpredictability and sudden changes in the main direction, and are likely to be occluded by objects or other people. For these reasons, we proposed a probabilistic and appearance-based tracking algorithm able to manage also large and long-lasting occlusions [38]. Despite its accuracy, our tracking fails in the case the person changes his direction when occluded, since the algorithm relies on the hypothesis of constancy of motion during occlusions (being any other hypotheses not reasonable). Since in absence of visibility cameras are useless, this is a concrete and interesting example in which PIR sensors can be useful.

The knowledge about VO_s and their status is exploited by a selective background model [37] in order to be both reactive to background changes and robust to noise. Selective update is obtained by, on the one hand, not considering moving pixels in the updating process, and, on the other hand, forcing inclusion of stopped objects (previously moving) into the background model. Unfortunately, the system sometimes misclassified moving objects (such as a person) with stopped objects (such as a door that has been opened). In these cases, the lack of enough resolution prevents the vision system to work properly and PIR sensors might help.

Eventually, scene understanding is a high-level module and heavily de-

depends on the specific application. In the case of video surveillance of people, it includes a posture classification module [38], capable to discriminate between four postures (standing, sitting, crouching, and laying) and, consequently, to detect interesting events, such as a persons fall.

Consistent labelling

Real video surveillance setups often require multiple cameras, both to cover wider areas and to solve occlusions by exploiting multiple viewpoints. The goals of the consistent labelling and multicamera tracking module are the detection of correspondences between people extracted from each single camera tracking module, and then the computation of a list composed by the best views (selected from the different cameras) of people present in the scene. This list is the input of higher level tasks, as posture classification, face detection, and recognition. We propose an approach of consistent labelling based on geometrical features and homographic transformations. For two overlapped cameras C_i and C_j , through a learning procedure in which a single track moves from one view to another, an automatic procedure computes the End of Field of Views (EoFoVs) that are exploited to keep consistent labels on the objects when they pass from one camera to the adjacent. By this, the homography that binds the ground planes on the two views can be easily computed. Full details can be found in [25].

Differently from other methods that check consistency only when objects pass through the edges of the field of views (camera handoff), we compute the assignment each time a new object is detected in the camera C_i in the overlapping. In this case its support point is projected in C_j by means of the homographic transformation. The coordinates of the projected point could not correspond to the support point of an actual object. Thus, we select for the match the object in C_j whose support point is at the minimum Euclidean distance in the 2D plane from these coordinates. This approach is an efficient tradeoff between classical techniques that verify correspondences at the camera handoff instant only (as in [107]), and complex methods of 3D reconstruction that find correspondences at each frame preventing any real time implementation (as in [144]). Figure 5.30 gives a bird-eye-view description of the area acquired by three different cameras; this representation is possible due to the homographic transformations between different views. The edges of field of view have been superimposed. The people can be detected by one, two, or even three cameras depending on their position. When a person is in the internal part (where three cameras are overlapped), three different views of the same person are available. In figure 5.31 an example of consistent labelling between three cameras is reported.

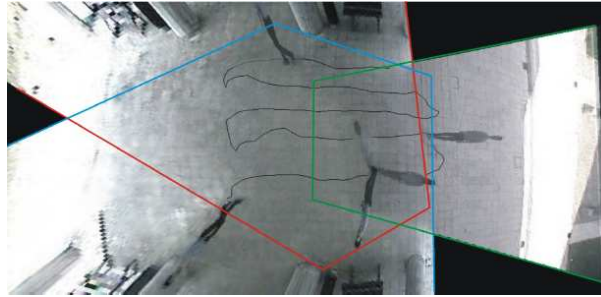


Figure 5.30: Bird-eye-view description of our test bed environment.

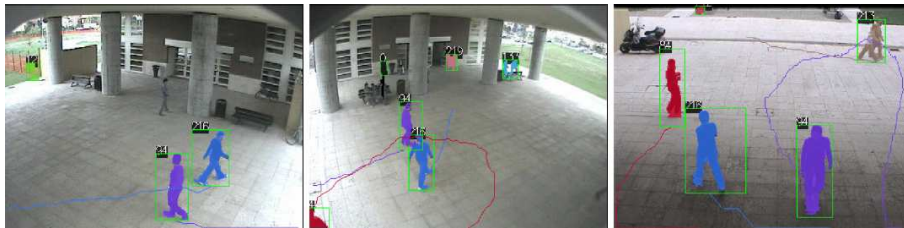


Figure 5.31: Example of consistent labelling between three views.

5.5.5 Multi modal integration

To test the system we have equipped the atrium of our faculty with four cameras and several PIRs, as depicted in figure 5.21. Detailed descriptions with particular experimental results of the sensor nodes and of the multicamera system are reported in [39].

Sensor-guided background update

Algorithms of motion capture based on background subtraction rely on a very crucial task: the update of the background, especially in presence of illumination changes and moved objects inside the scene. For example, when the doors in figure 5.32 are opened, the background scene changes and the detection of people in that area becomes unreliable. To this aim, we use sensors to monitor the area near the doors. If the single camera processing detects a visual object in the door area but the sensors do not capture events, then we assume that the motion is due to an incorrect background. In such a situation, the background is updated by forcing the area covered by the sensor directly with the input image.

More generally, each tracking system analyzes its list of detected objects. If an object is still for a long time, then the correspondent camera thread makes a request to the manager specifying the object location. The manager searches if the concerned zone is covered by a sensor and, in such a situation, it responds



Figure 5.32: Opening and closing doors make unreliable background suppression techniques.

with the relative state. If the computer vision and the sensor network are discordant, then the sensor is considered more reliable, and the vision system reacts consequently, for example updating the background.

In figure 5.33 some frames taken from a single camera that is capturing the entrance of our faculty are reported. The rows report, from top to bottom, the input frames, the output of the tracking system, and the background model. Initially (first column) the door was open. Some frames later a person closes the door and from this instant the background becomes inconsistent. In fact, the system erroneously detects the presence of a person in the area of the door (see figure 5.33(e)). When the PIR sensor placed near the door does not capture any events, the background is correctly updated (last column).

Detection of direction changes during occlusion

Occlusions are another problem that characterize video surveillance systems based on computer vision; for example, in the environments of figure 5.32, people can walk behind the columns, and, in such a situations, the system is likely to lose them. To face this problem, we have introduced some rules inside the tracking system. When a track disappears, it is not deleted immediately, but its appearance is kept unchanged and an estimation of the track position is computed exploiting a constant velocity assumption. If the person returns visible again with a similar appearance and a position near to the predicted one, then the system assigns the same label of the disappeared track. However, if the person changes direction during the occlusion, the system is not able to correctly assign the label anymore.

For this reason, we exploit a PIR sensor node placed behind the column.



Figure 5.33: Sensor-guided background update.

As above mentioned, these sensors detect not only the presence of a person, but also his direction. Then, we can detect a change of direction capturing couples of opposite direction events sent in a short temporal window. In such a situation, the direction of the motion applied to the track is inverted in order to estimate the position frame by frame.

In figure 5.34 an example of consistent labelling after an occlusion is reported. The person walks behind a column and, during the occlusion, inverts his direction. The computer vision tracking algorithm is not able to solve the consistent labelling because the person reappears too far with respect to the predicted position (computed with a constant velocity assumption). Using PIR sensors, instead, the change of direction is detected and the estimated track position can be properly updated. Then, when the person reappears, the tracker assigns the same label (24) assigned before the occlusion (see figure 5.34(b)).

Differently from the previous example, in this case the sensor network detects an event and the manager thread informs the involved cameras of it. Then, if a tracking system has detected an object in the corresponding position, the motion direction is changed accordingly.

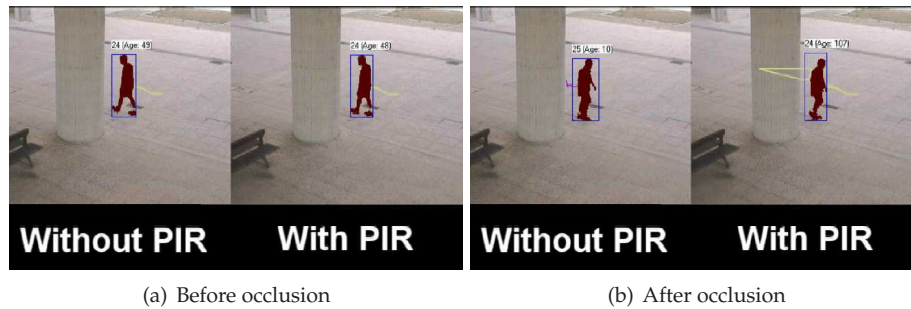


Figure 5.34: Consistent labelling after an occlusion exploiting a PIR node to detect direction changes

5.5.6 Conclusion

Distributed surveillance is a challenging task for which a robust solution, working on a 7/24 basis, is still missing. This section is meant to propose an innovative solution that integrates cameras and PIR (Passive InfraRed) sensors. The proposed multi-modal sensor network exploits simple outputs from the PIR sensor nodes (detecting the presence and the direction of movements of people in the scene) to improve the accuracy of the vision subsystem.

Two case studies are reported. In the first, the vision system, based on background suppression, fails due to a door that is opened. Since background is not immediately updated, the door is detected as a moving object (resolution is not sufficient to enable a correct motion detection). In this case, a PIR sensor is used to discriminate between the opened door and a real moving person. In the second case study, a person changes its direction when it is occluded by a column. The vision tracking algorithm relies on the constancy of the speed during occlusions and thus fails. A pair of PIR sensors are, instead, used to detect the change in direction and alerting the vision system.

The reported results demonstrate that using the integration between PIR sensors and cameras the accuracy can significantly be increased.

5.6 A solar-powered video sensor node for energy efficient multimodal surveillance

Building an energy efficient wireless vision network for monitoring and surveillance is one of the major efforts in the sensor network community. In this section we describe an application for people detection, which exploits both network architecture flexibility and on-board processing capabilities. The application, based on support vector machine engine (SVM), is able to detect events (e.g. when the environment is changed due to the movement of subject in the

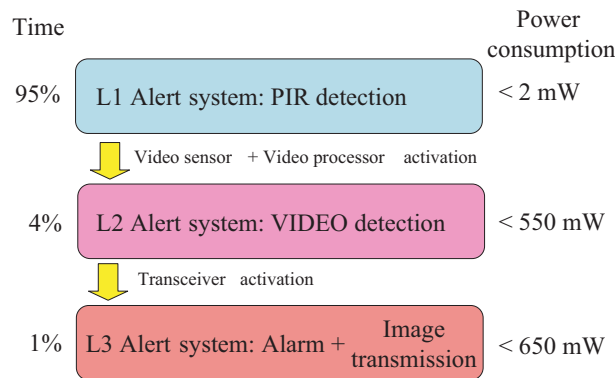


Figure 5.35: Hierarchical design of the video sensor node, with three different layers for the alert system.

scene), and distinguishes the presence of people or human bodies rather than objects or animals in the field of view before generating alarms or sending information through the wireless link. We focus on the design, implementation and characterization of a self-powered video sensor node, able to detect people and supported by PIR sensors to enhance energy efficiency.

The video sensor node is designed to support flexibility in terms of distribution of the processing tasks across the network and is powered by a solar scavenger using a 70 cm^2 photovoltaic panel. Keeping the nodes constantly active is clearly impracticable, because of the power consumption of components such as imager, transceiver and microprocessor. Therefore the proposed architecture follows a hardware/software hierarchical design with three layers which can be separately activated, as showed in figure 5.35.

The figure considers a hypothetical surveillance scenario where events occupy the 4% of the time and only 20% of them results in an alarm to report. The objective is to wake up the video acquisition only in presence of people and to reduce the number of not-interesting events in order to guarantee longer lifetime while the system is recharged by a fluctuating and unpredictable energy source. Once the video is waken-up, the node locally classifies input images and wirelessly sends to a base station only relevant ones, thus saving energy by reducing the amount of transmitted data.

We developed a novel method to modulate the status of each layer by exploiting a PIR based wake-up circuit and local image processing. The sensitivity of the trigger signal from the PIR detector is adjusted dynamically according to the available energy in the reservoirs, the average contrast of the images taken from the scene and the probability of seeing a person in the camera FOV.

5.6.1 Related work

Recent years witness a rapid growing of research and development of surveillance and multimodal applications using multiple sensors, including video and other kind of sensors. The aim of such systems is both to overcome some points of failure of a particular kind of sensor and to balance different parameters fixed by the application among which power consumption plays a central role.

Power management is a critical issue when dealing with wireless sensor networks and it is well known that batteries does not scale as much as electronic device [157] thus posing a severe limitation in the achievable unobtrusiveness. Also the cost of batteries often exceeds the one of nodes. At last, in some application, it may be not possible to reach the sensors (i.e. due to dangerous environment, like battlefields) in order to replace batteries.

In [74] the authors attempt to formalize and analyze the trade-off between power conservation and quality of surveillance in target tracking sensor networks. In [239] a dynamic sensor selection is applied to efficiently use available sensor energy and extend overall network life. Another attempt to extend network life by capitalizing on low power states of its node can be found in [12]. In this work the amount of data collected by the system is tuned in order to minimize power consumption while achieving high accuracy. Finally in [81], a distribute network of motes equipped with acoustic and magnetic sensors have been deployed in order to achieve longevity, adjustable sensitivity, stealthiness and effectiveness in a military surveillance application. Since in this paper the authors aim at achieving longevity through sensor selection techniques, they use a high number of low power nodes with low resolution (magnetic field detector) and network life extension is obtained by reducing number of active sensors when any activity is detected and successively wake them up. In contrast we have a unique sensor, which provides much more information and we modulate its activity through the use of another low power sensor.

In contrast to the work presented in this session none of the cited works attempted to reduce the node power consumption except using low power hardware, and they either do not consider a stochastic source of energy as the one provided by an energy scavenging system.

5.6.2 System architecture

The hardware architecture of the solar-powered video sensor is displayed in 5.36 and consists of several modules: the solar harvesting unit, the vision board which hosts both the CMOS imager and the PIR sensor with a common area

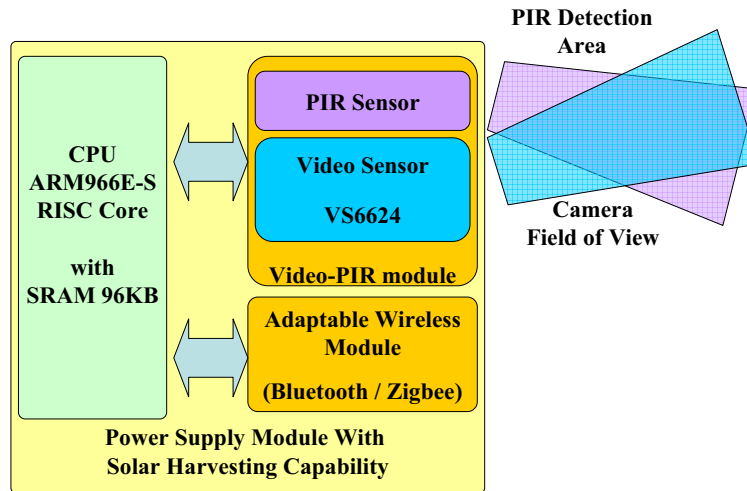


Figure 5.36: Video sensor node architecture.

under monitoring, the wireless module, the microprocessor and other peripherals.

Computational unit and CMOS imager

The core of the video node consists of an STR91xF microprocessor from STMicroelectronics with an ARM966E 16/32-bit RISC architecture, 96 MHz operating frequency, 96 KB SRAM and several peripheral interfaces that can be disabled if not used. The microprocessor provides the high-speed logic interface necessary to capture images from the camera and processing data for people detection or object classification, it also offers configurable and flexible power management control through operative frequency scaling.

The vision module includes a SXGA CMOS color digital camera targeted for mobile applications featuring low-size and low-power consumption and a Pyroelectric Infrared Detectors, which detection area is overlapped with the field of view of the video sensor.

The video sensing device is a VS6624 CMOS imager from STMicroelectronics. It supports up to 15 fps SXGA with progressive scan and up to 30 fps with VGA format. It operates at 2.8 V and 12 MHz frequency and the power consumption is 120 mW when active, while it decreases down to 23 mW when switched to standby. Although it supports SXGA resolution, only 160×120 pixels are enough to perform the human detection algorithm, and it allows to save time and energy for storing and processing data. CMOS camera can be programmed and controlled via internal registers using I^2C serial interface. It supports several output formats, in particular we adopt 8-bit grayscale images with YCbCr 4:0:0 format.

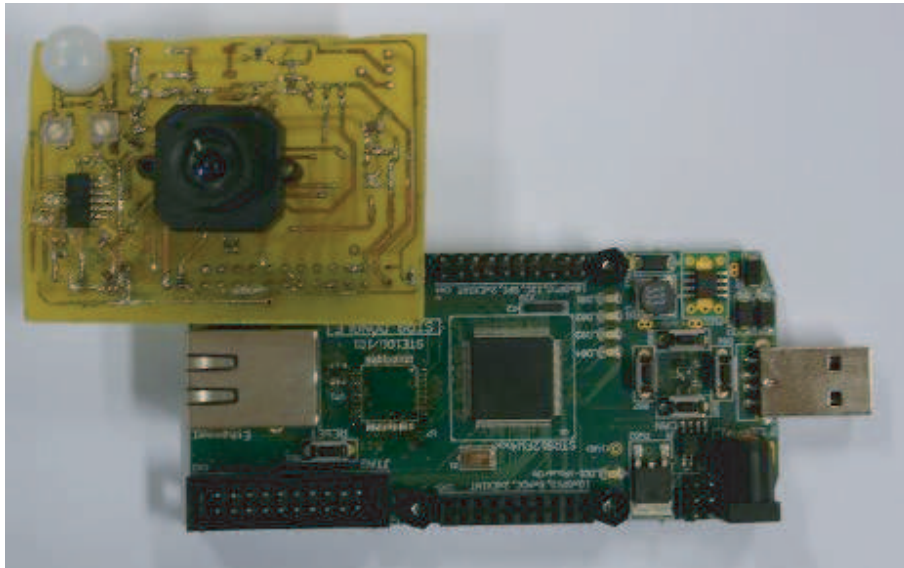


Figure 5.37: Developed prototype of the video sensor node.

Wireless communication capabilities have been supported through a suitable interface for both Zigbee and Bluetooth compliant transceiver. The module has a stackable design as the sensor node, hence the wireless layer is easy to replace. We implement hardware and software interfaces in order to host different wireless standard used in wireless sensor network community such as Zigbee and Bluetooth or proprietary protocols. All the performance and measurements discussed in this section are referred to the version with Bluetooth capability.

Figure 5.37 shows the first version of the developed prototype, the whole system is designed with low power consumption as the primary goal. The system is powered by an energy management module which hosts solar harvesting capability. The solar cell used to replenish the energy reservoirs has a nominal output power of 500 mW under full outdoor irradiance and a harvesting circuit extracts the maximum power available from the solar cell following the optimal operating point at the minimum energy cost.

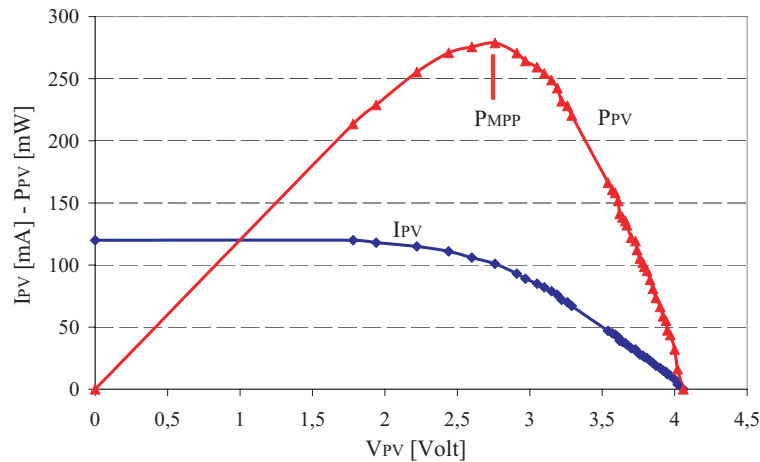
Energy harvesting unit

Energy harvesting is a low cost-effective operation, in term of energy harnessed, device size and efficiency. One of the primary issues to address is minimizing the power consumed by the harvester itself. Less power will require the circuit, faster will be the growth of the harvested energy in the accumulator.

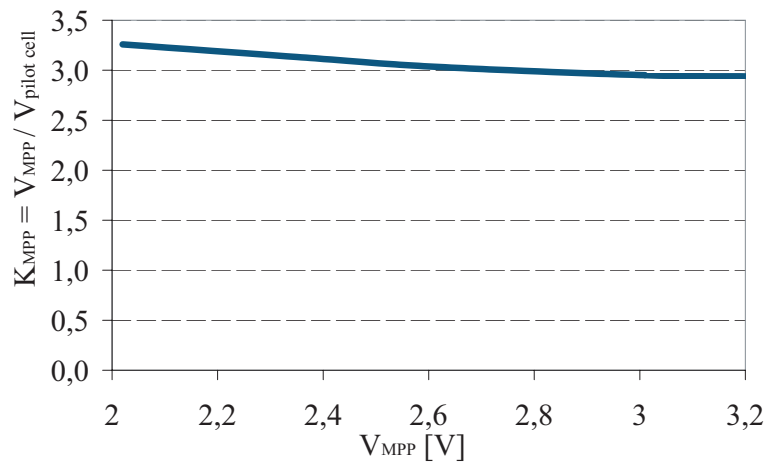
The I - V characteristic of a PV module is given by the following equation:

$$I_o = I_g - I_{sat} \left\{ e^{\frac{q}{AKT}(V_o + I_o R_s)} - 1 \right\} \quad (5.3)$$

where I_g is the generated current, I_{sat} is the reverse saturation current, q is the electronic charge, A is a dimensional factor, K is the Boltzmann constant, T the temperature in degree Kelvin, R_s the series resistance of the cell. The internal shunt resistance is neglected in this model. The plot of the PV module adopted in our solar harvester is shown in figure 5.38(a).



(a) I-V and P-V plots of the used photovoltaic module.



(b) Near linear relation between V_{MPP} and the pilot cell.

Figure 5.38: Characteristic of the photovoltaic module.

One key design challenge is how to optimize the efficiency of solar energy collection under non stationary light conditions and therefore maximum power point tracking techniques (MPPT) aim to automatically find the operating point (V_{PV} , I_{PV}) at which a PV module should operate to provide the maximum output power following it when light intensity changes. There are

several methods and algorithms to track the MPP [59], we adopt one based on Fractional Open-Circuit Voltage (FOCV) which is the most used and cost-effective in medium and small-scale solar harvester. This method exploits the nearly linear proportional relationship between the operating voltage at MPP (V_{MPP}) of the main photovoltaic module and the open circuit voltage of a small additional PV array used as pilot-cell ($V_{pilot\ cell}$) under the same light L and temperature T conditions (5.4).

$$V_{MPP}(T, L) \approx K_{MPP} \cdot V_{pilot\ cell}(T, L) \quad (5.4)$$

We adopt the CPC1824 from Clare, Inc. [137] for the pilot-cell. It is a monolithic photovoltaic module of only 9 mm^2 , and it works as irradiance sensor providing feedback information to the harvester. The pilot cell follows almost linearly the behavior of the main PV module during light variations. As shown in figure 5.38(b), the ratio between the operating voltage at the MPP of the main module and $V_{pilot\ cell}$ is almost constant under several solar intensities.

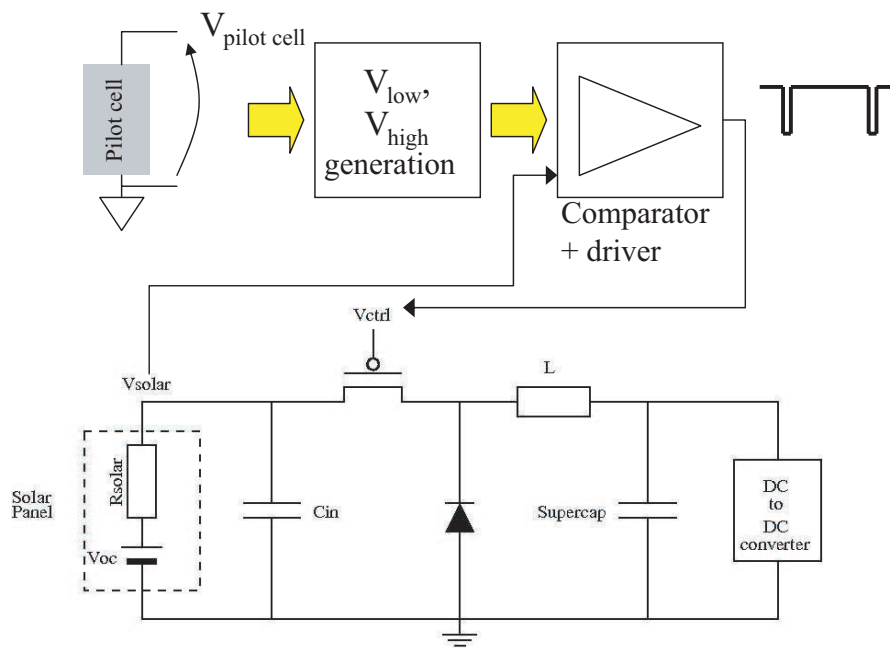


Figure 5.39: Conceptual schematic of solar harvester: buck power converter and MPP tracker.

Figure 5.39 depicts the schematic of the solar scavenging circuit for the video sensor node. By measuring the pilot-cell voltage the circuit estimates the MPP of the main module generating a lower and an upper threshold around its value. Then an ultra-low power comparator continuously checks the oper-

ating point of the main cell to the thresholds adjusting dynamically the duty cycle and the frequency of the control signal which drives the power converter circuit. Solar energy harvesters usually exploit buck configuration because the voltage level of the energy reservoirs is lower than the nominal operating voltage of the solar cell. In our implementations we exploit supercapacitors as energy storage devices, since they overcome many drawbacks of batteries that are critical in WSN applications and for long-live maintenance-free embedded systems. The harvester achieves an efficiency of the 80% and depending on solar irradiation can provide a maximum output power of about 500 *mW* while the power consumed by energy harvesting process is less than 1 *mW*.

PIR sensors wake-up unit

As in the other works presented in this section (see sections 5.3, 5.4 and 5.5) we used a commercial PIR detector that includes 2 sensitive elements placed in series with opposite polarization. The details of this device have been presented earlier in section 5.4.1, a schematic of this device is presented in figure 5.18. The Fresnel lenses adopted in this project are the ones described in section 5.5.2 (see figure 5.24).

In particular in this work we are interested in the amplitude of the output signal which, outside the area where the FoV of the 2 elements is overlapped (see figure 5.18), is inversely proportional to the distance from the detector as can be seen in figure 5.40.

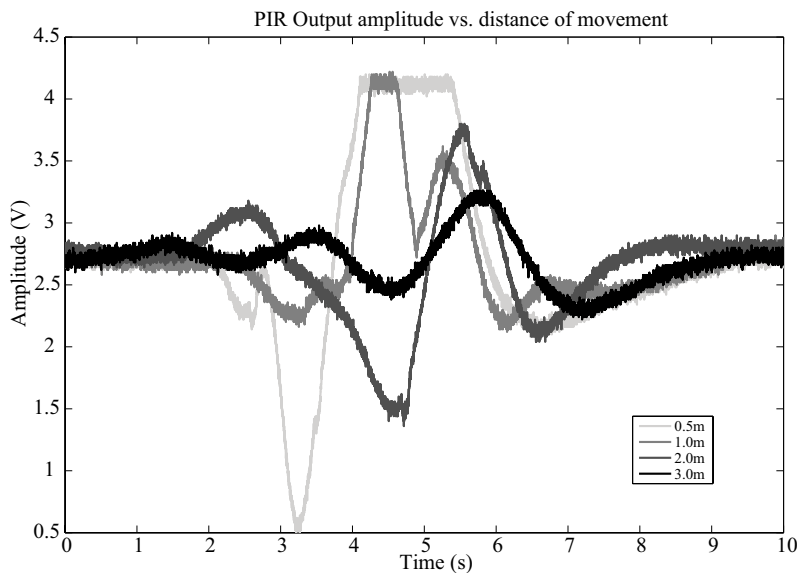


Figure 5.40: Output of a PIR sensor when a person moves at different distances

The sensor output signal is conditioned as in 5.3.1

In addition to the amplifier we designed a trigger with adjustable threshold. The schematic of the circuit is presented in figure 5.41. Here the series of R1, R2 (where R1=R2) and the digital potentiometer produces the 2 thresholds which are symmetrical to $\frac{V_{dd}}{2}$ and their reciprocal distance increases with the resistance of the digital potentiometer. When the amplified output breaks one threshold it generate an interrupt for the Video node core. Thus, by on-line programming the potentiometer we can adjust the sensitivity of the wake-up signal.

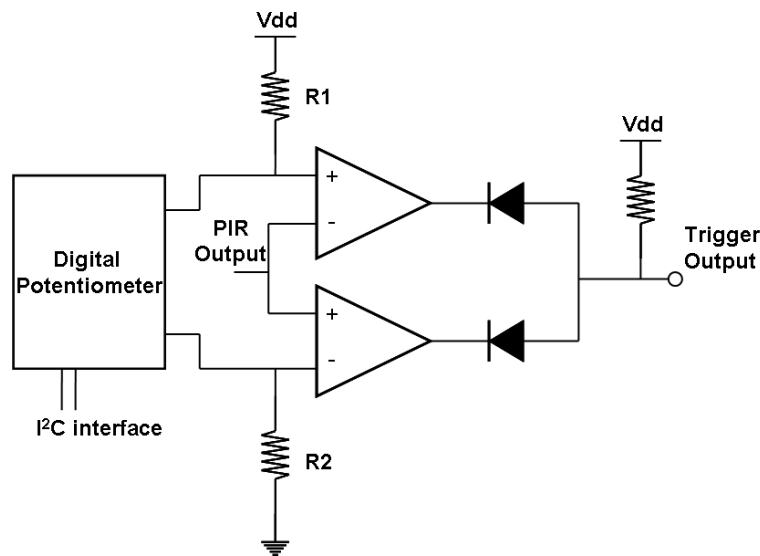


Figure 5.41: Schematics for trigger generation using PIR output signal.

5.6.3 System analysis

Sensor node characterization

The ARM microprocessor STR91x offers configurable and flexible power management control which allows dynamic power consumption reduction. It supports three global power control modes: RUN, IDLE and SLEEP. SLEEP mode is used by the video sensor node when no events are registered in the field of view. When triggered by an event from the PIR sensor, the system switches into RUN mode starting the detection application until the PIR trigger events or regions of interest are discovered in the current image, then the system switches back into SLEEP mode where the power consumption decreases up to 90% since only the PIR module operates. Power consumptions are reported in table 5.8.

| Component | Power [<i>mW</i>] |
|----------------------------|---------------------|
| ARM9 (RUN mode) | 450 |
| ARM9 (IDLE mode) | 49,5 |
| ARM9 (SLEEP mode) | 15 |
| Video sensor (ON mode) | 165 |
| Video sensor (IDLE mode) | 23 |
| TX/RX module (ACTIVE mode) | 98 |
| TX/RX module (IDLE mode) | 10 |
| PIR sensor | 1,5 |
| Solar Harvester | 0,98 |
| Video Node (Active) | 650 |
| Video Node (Sleep) | 50 |

Table 5.8: Power consumption of the video sensor node.

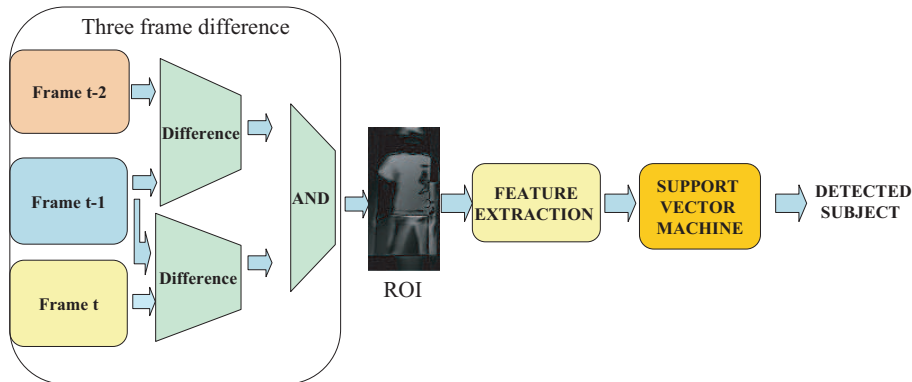


Figure 5.42: Flow chart of the human detection application.

Human detection application

Figure 5.42 presents the main steps of the implemented algorithm for human body detection. After triggered by the PIR sensor, all the system wakes up and the CMOS imager acquires and sends a frame to the microprocessor with YCbCr 4:0:0, grayscale, 8-bit format. In order to isolate a 128×64 region-of-interest (ROI) of the event we initially perform a background subtraction using the three-frame algorithm sub-image [101]. A pixel-by-pixel subtraction is performed using the first and second frame stored in the memory, then another pixel-by-pixel subtraction uses the second and third frame. Finally the two results pass in a logical AND to have a difference-image that allows to detect and track moving objects across different frames.

This new image is stored in SRAM and we use it to search and isolate region of interests (ROI) in a 128×64 sub-image. To obtain the vector of feature for the following classification step, we calculate the average values of gray for each column and row in ROI (which is equivalent to project the ROI image onto horizontal and vertical axes). Thus the size of the input vector for the

| Task | Energy [mJ] | time [ms] |
|------------------------|-------------|-----------|
| Three Frame Difference | 440 | 720 |
| ROI Extraction | 12,2 | 20 |
| Feature Extraction | 9,6 | 16 |
| SVM | 21,21 | 35 |

Table 5.9: Energy requirement

classifier is reduced from 8192 to 192 elements. Undoubtedly both smart ROI size and efficient feature extraction algorithm contribute significantly to save energy and time processing.

Regarding the classification function, a highly tuned SVM-like hardware oriented algorithm has been implemented for the STR91xF [106]. A detailed description of this algorithm and its performance in people recognition can be found in [105]. Being a "learning from examples" technique, SVM [211, 180] it is firstly trained on a set of available data known as *training set*. Such a computationally expensive training phase is performed off-line by a powerful base station, then the classification function are loaded to the nodes to classify the patterns under observation.

Thanks to background subtraction the training set is independent from the node position and orientation, thus all SVM can be trained at once using the same training set.

The output of the classification can be simply binary report of the presence of the human body in the field of view, or an image of the region of interest with the detected subject. This result can be sent via wireless to a controller unit.

Autonomy of the system

We considered a typical application scenario of an outdoor surveillance. Assuming a rate of events as presented at the beginning of this section we estimated the capacity necessary to perform a complete and effective service during the night using the energy harvested and saved during the day. Experimental results using different size supercapacitors without solar harvesting capabilities, show that the system can achieve autonomy of several hours (figure 5.43). Increasing the capacity up to 500 F it is possible to operate for about 8 hours, till the next morning.

5.6.4 Dynamic adjustment of the detection area

In a distributed vision network several nodes cooperate for an efficient surveillance service and the area under monitoring is covered by multiple nodes de-

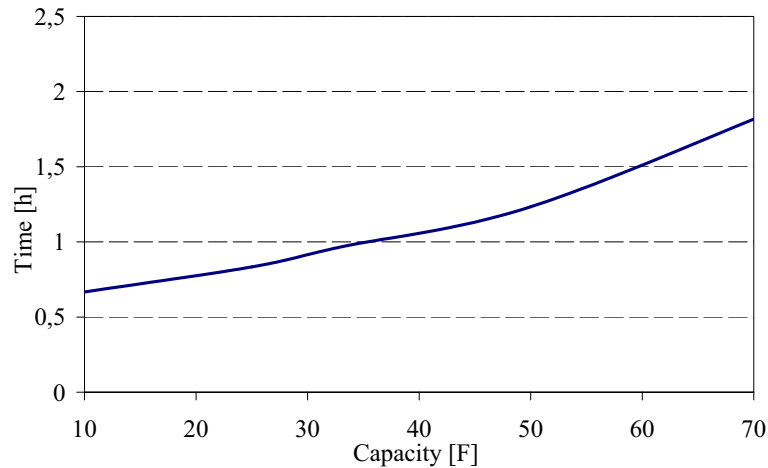


Figure 5.43: Autonomy of the system varying the capacity of the reservoirs without environmental harvested energy

ployed in the environment and the whose projections of camera field of views are usually overlapped. For this reason it is possible to develop distributed policies for smart dynamic coverage of the region under surveillance. For instance when a node is lacking of energy it could reduce its detection area and consequently its activity while other cooperative nodes compensate augmenting PIR sensitivity for longer distance events. In such a cooperative vision, a dynamic adjustment of the detection area on each single video is necessary.

Figure 5.44 shows the amplitude of the PIR signal as a function of the distance of the detected object. This result highlights how is possible to modulate the detection area by adjusting the thresholds used to generate a wake-up signal for the video node.

If we assume a uniform probability that a person moves in a certain point of the area of interest, by increasing the threshold we reduce the sensitivity of the trigger and the area covered by the PIR and consequently the probability to activate the camera.

For this reason the threshold (5.5) is regulated as a function of the following parameters:

- contrast of the image, C ;
- the energy available in the supercapacitor, E_{CAP} ;
- the probability of seeing a person moving in a certain point at a certain time, p .

$$V_{threshold} = \alpha \frac{p}{E_{CAP}} + \beta C \quad (5.5)$$

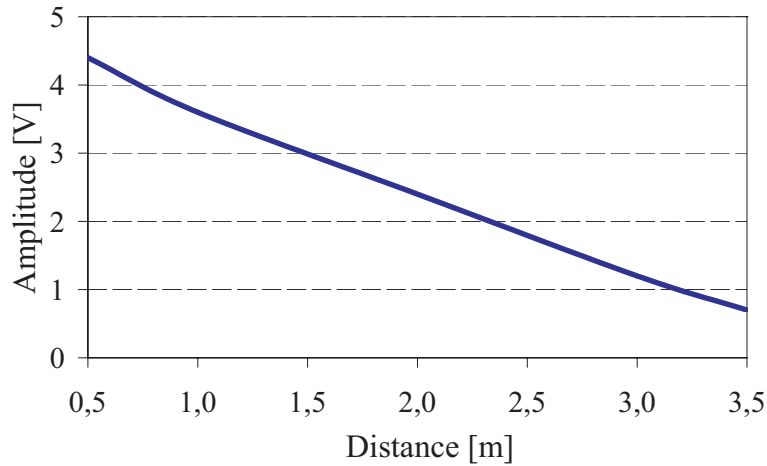
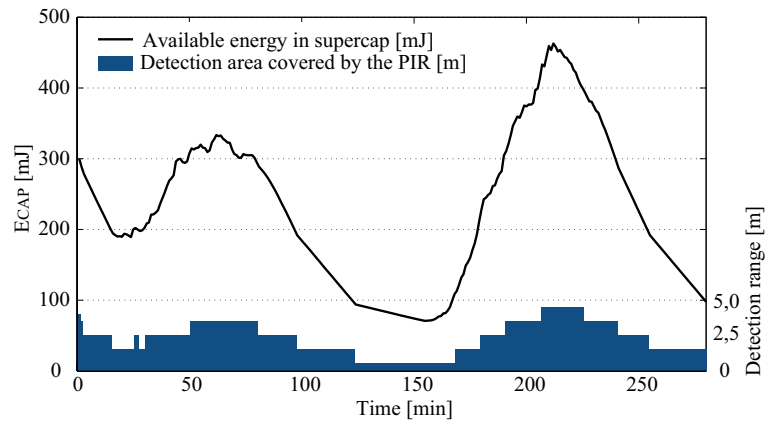


Figure 5.44: Amplitude of the PIR Output signal as function of the distance of the object.

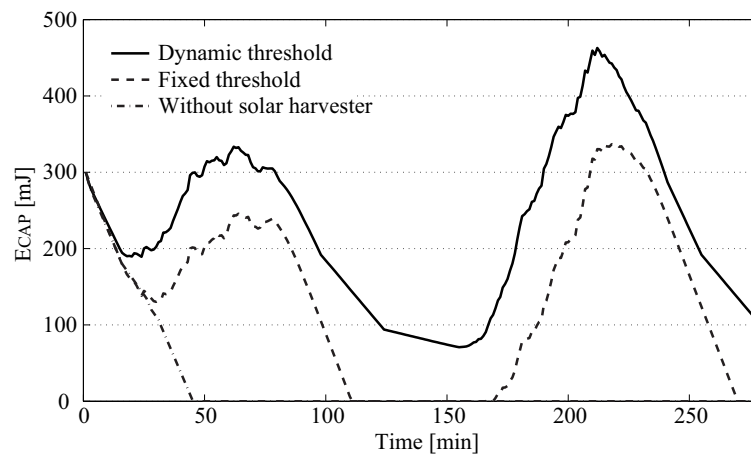
Images with low contrast C may result in a loss of accuracy of the SVM algorithm. Thus, it is better to suspend the vision algorithm saving energy when the contrast of the image is lower than a defined value $C < C_{th}$. Concurrently, when the contrast of the images is low, the threshold of the PIR could be reduced in order to extend the area under monitoring and sending alarms relying only on PIR detection. The value of the threshold should be inversely proportional to the energy available in the supercapacitor and directly proportional to the probability density of a people moving in the field of view. In fact when more energy is available a higher number of detection can be tolerated. On the other hand, if the probability of detecting a person is higher, lack of energy in the accumulator forces a higher reduction of the field of view of the PIR if we want to extend the lifetime.

A simulation to verify the performance of the proposed dynamic threshold is depicted in figure 5.45(a). The energy harnessed from the solar cell is powering the sensor node and replenishing the energy storage E_{CAP} with the exceeding energy. When the energy in the storage is enough to sustain the desired quality of service, the detection area covered by PIR sensor increases (up to $4m$ in our scenario). Similarly, as soon as the available energy decreases due to a reduction of the harvesting supplying, the threshold switches diminishing the area covered by PIR and consequently the rate of activation of the camera. The simulation covers about five hours of operation of the sensor node, and the threshold function is approximated using discrete values. It worth to notice that simulations are performed using energy storage devices with limited capacitance of $33F$ and a constant contrast C of the images higher than the threshold C_{th} . To prove the effectiveness of the dynamic adjustment of the monitored area, figure 5.45(b) illustrates the behavior of the node with differ-

ent configurations. The plot compares the energy stored in the supercapacitor in the same operating condition of figure 5.45(a) with the situation when the threshold of PIR sensor is fixed with a constant size of the area under monitoring of 3 m (dashed plot). Using a fixed threshold the trade-off between energy and sensitivity is off-line design parameter and wide detection areas increase the probability to be out of service because of the empty energy accumulator, as happens in the figure during the interval I_{OFF} [111, 168]. The plot shows also the performance of the video node without solar harvester and when no environmental energy is stored in the accumulator. Obviously in this case the video node has a limited lifetime as for all battery-operated systems.



(a) Variation of the area under monitoring as function of the stored energy.



(b) Comparison of the energy efficiency in different solution: with dynamic variation of the PIR sensitivity threshold, with fixed threshold and without solar harvester.

Figure 5.45: Simulation results of the energy efficiency using a dynamic PIR sensitivity threshold.

5.6.5 Conclusion

An integrated self-powered video sensor node for energy efficient surveillance has been proposed. The adoption of a solar harvester for supplying the node leads to several benefits such as the possibility to extend the lifetime of the vision sensor network. However since the amount of energy provided by the photovoltaic module cannot be predicted the status of the system must be dynamically adjusted. A multimodal platform equipped with different family of vision sensor with heterogeneous features of power consumption and resolution permits to adopt very effective energy management techniques reducing considerably the activation of the camera, the microprocessor and other power consuming devices. In the proposed system the sensitivity of a low power PIR based wake-up circuit is adjusted dynamically according to the available energy on-board, to the contrast and the probability of moving subjects enter the video node field of view. With such a technique, under a hypothetical surveillance scenario, we estimated that using a 500F super capacitor the wireless video node is able to operate for about 8 hours during nighttime.

Chapter 6

Activity Recognition in Redundant and Dynamic Sensor Networks

6.1 Overview

Technological advances enable the large scale deployment of highly miniaturized, unobtrusive and interconnected (wireless) sensor nodes (WSN) in our living environments, in our outfits, and in devices we carry with us. This unobtrusive yet widespread sensing permits pervasive and wearable computing systems that provide transparent and natural human-computer interfaces (HCI) and smart assistance to users according to their context and activities.

Human activities and manipulative gestures are an important aspect of context that supports *activity-based computing* [43] with application including gestural-based HCI [102], support of impaired people [83], or industrial worker's assistance [196].

The prevailing assumptions underlying traditional approaches to activity recognition are

- Sensors placed at an "optimal" body locations for the activities to detect. Variation in sensor placement over time are proscribed as they affect classification.
- Sensors are rarely available unless specifically provided for a desired application scenario and they are bulky, thus their number is minimized in order to reduce obtrusiveness (e.g. manipulative gestures may be detected from few IMUs placed on limbs and back).

- The sensors characteristics remain constant (i.e. no sensor degradation) and they do not fail.
- Sensor interconnections are reliable.

This leads to activity recognition signal processing chains optimized for statically defined sensor setups that do not allow a more flexible network structure. Moreover, due to their minimized number, every sensor node is a point of failure of the entire network. Thus, once one sensor runs out of energy or fails the system is not able to achieve its objective and maintenance is needed.

Yet, sensor nodes get smaller and cheaper. They become available in larger numbers, integrated in our outfits, in devices we carry around, and in our environments. Although this allows a dense on-body sensor placement, it may be at the expense of sensor accuracy or robustness, or interconnection reliability. Under realistic conditions, on-body sensor networks tend to be dynamic. This leads to real-world deployment issues. Textile sensing elements are subject to high mechanical stress (e.g. during washing or when worn) which may lead to sensor degradation and faults. Networks of miniature body-worn wireless sensors may suffer from radio interferences, as well as occlusions caused by body-parts, thereby causing data rate reduction or data loss. In order to avoid relative motion, sensors attached on the body require the use of tight-fitting clothes or relatively high attachment pressure, which limits comfort. Users take with them and leave instrumented devices (e.g. cellphone, PDA), and change sensor-augmented clothing. In a general setting the sensor network characteristics may thus change in unpredictable ways.

Furthermore, high classification accuracy is usually desired. This implies the use of several sensors distributed over the body, depending on the activities to detect. At the same time, a wearable system must be unobtrusive and operate during long periods of time. This requires power optimization to improve user acceptance, since batteries are a limiting factor in miniaturization [157], and to enable long term operation of pervasive computing environments. As a result, application-defined power-performance tradeoffs are beneficial.

Power minimization is mainly addressed by reducing the power consumption of single nodes. Energy use may be reduced by improved wireless protocols [10], careful hardware selection [84], or duty cycling to keep the hardware in a low-power state most of the time [42]. Energy harvesting techniques may also complement battery power [148], although the unpredictability of energy supply typical of harvesting makes it difficult to manage duty cycling schedules [215].

However, activity recognition requires fixed sensor sampling rate and continuous sensor node operation, since user gestures can occur at any time and

maximum classification accuracy is desired. Therefore adaptive sampling rate and unpredictable duty cycling can not be used to minimize energy use.

The objective of this chapter is to investigate activity recognition in this challenging context of large and dynamically changing sensor networks. Through the chapter we demonstrate the benefits for activity recognition brought about by the availability of a large number of sensor nodes, in terms of scalable system performance, fault and noise tolerance, and power-performance management opportunities.

To this end, we introduce an activity recognition signal processing chain suited for dynamic sensor networks. It takes advantage of multiple sensors to cope with failures, noise and enable power-performance management. It is based on a *dynamic classifier fusion core* that combines the information of simple activity classifiers operating on individual sensor nodes into a joint activity classification. The algorithm can be easily parallelized to best use the computational power of a sensor network. We validate this method using a set of activities from a quality assurance checkpoint of a car assembly line. We show that this approach allows scalable system performance, and intrinsic robustness to faults and noise.

Furthermore we investigate how to extend network life in an activity recognition system, while maintain a desired accuracy, by capitalizing on the network redundancy. A set of sensor nodes perform gesture recognition continuously and autonomously, while the others are kept in low power state. They are activated when their contribution is needed to keep the desired classification accuracy, such as when active nodes fail or turn off due to lack of energy. The number of sensors that contribute to activity recognition is modulated through *dynamic sensor selection* on the basis of a *system performance model* defined during system training.

We show that this approach enables runtime application-defined power-performance management at the network level. This approach is independent of specific sensors and classifiers used, and it is suitable for other application domains where a large number of sensor nodes is used to monitor areas of interests.

With the objective of enhancing the interaction with smart environment, smart objects can be used as tangible interfaces and play a fundamental role in improving human experience within interactive spaces for entertainment and education [95].

The Tangerine Smart Micrel Cube (SMCube) is a tangible smart object for Human Computer Interaction (HCI) equipped with sensors (digital tri-axes accelerometer as default) and actuators (infrared LEDs, vibro-motors) embedded in a wooden cube. Data from accelerometer is used to locally detect the ac-

tive face (the one directed upward) and a set of gesture performed by the user. These information are wirelessly sent to a base station for processing. Furthermore, through the LEDs the node can interact with a vision based system in a multi modal activity detection scenario.

The SMCube has been designed as a building block of the Tangerine application framework to provide a tangible smart interface within the *digital media table* scenario. Since it can be carried around and interact with different tables it can be used as a representative of the user and it can support data exchange and natural interaction.

In this chapter we present our attempt to augment the intelligence of the SMCube implementing an on board gesture recognition algorithm based on a decision tree. The algorithm is able to distinguish between four gestures performed by the user: *cube placed on the table*, *cube held*, *cube shake* and *tap* (the user tap on the upper face of the cube). The gestures recognized provoke reactions defined by the system or application the cube is interacting with.

One of the main challenges in developing activity recognition techniques is related to the large amount of data required for:

- build a new model;
- validate a novel approach;
- compare different techniques.

The last part of the chapter describes our experience in building a dataset for context recognition. The dataset is available for research purposes and is intended to be a common benchmark for design, evaluation and comparison of different activity recognition approaches. It includes several repetition of complex activities made up of atomic gestures, thus it presents an hierarchical structure suitable for multilevel data analysis. All the data streams are labeled and videos are available for a deeper understanding of the activities [240].

6.2 Related work

6.2.1 Gesture recognition

Human activities can be recognized from various kind of sensors in objects, the environment or on-body [221, 143]. The approaches can be divided into those relying on ambient infrastructure such as video camera and tracking systems [190, 191] and those relying on body-worn sensors [8]. In this chapter we rely on body-worn sensors (e.g. motion sensors). A standard approach to the technical integration of on-body sensors relies on a Body Area Network (BAN)

[100], with wired or wireless interconnections depending on the application needs. Objects instrumented with sensors can also provide insight into human activities.

Activity recognition is a sense and classify problem. Sensors are used to acquire signals related to the activities of interest. A wide range of body worn and object-integrated acceleration sensors have been used to recognize object use [102], hand gestures [161] or whole body activities [27]. Gestures can be sensed using miniature and low-cost MEMS acceleration sensors, simple ball switches [209] or Inertial Measurement Units (IMUs) that combine accelerometers, gyroscopes and magnetometers to provide higher accuracy [7, 82, 98]. Textile sensors (typically conductive elastomer used as stretch sensors) can be integrated into textiles without affecting usability. This allows unobtrusive motion-sensing garments [133] to monitor the movement of joints (elbows, shoulders, fingers etc.). Muscle activity sensed by electromyography provides information about muscle motion [29]. Microphones can provide an indication of user activities generating characteristic sounds [219]. Objects instrumented with motion sensors can complement body-worn motion sensors [130].

Above all, the recognition of complex real-world activities benefits from the combination of multiple sensor types. In [195] data from body worn inertial sensors is correlated with the hand position measured from ultrasonic sensors in order to detect gestures during an assembly or maintenance tasks. Additional sensing modalities such as force sensitive resistors informing about muscle activity, and instrumented objects and environments provide further insights into the activities taking place [196]. Furthermore the fusion of contribution from a several body worn sensor can improve noise and fault tolerance [218]. In this work the output of a fixed number of sensor is combined resulting in a set of strings, one for each class. Through error correcting codes the authors show how faults on nodes can be compensated.

Once acquired, sensor data is pre-processed to reduce noise, and segmented into sections likely to contain activities. A vector of features is extracted from each segment. A classifier operating on the features yields the activity class. Several design choices are available at each step, depending on the application scenario, the activities that have to be recognized, and the available computational power (see section 4.1).

When features are time invariant (e.g. zero crossing rate or frequency spectrum), simple time-independent classifiers can be used (see section 4.3). Some examples includes linear classifiers, such as Support Vector Machines (see section 4.3.2), or decision trees, such as C4.5. In a more general case features are time dependent, and classifiers suited for temporal pattern recognition are used (see section 4.5). HMMs are often used in activity recognition since they

tend to perform well with a wide range of sensor modalities [191, 195] (they are also used successfully in other problem domains, such as speech recognition, for which they were initially developed [147, 70, 168]).

Information from multiple sensors (possibly of various kinds) can be fused to improve classification accuracy [216, 171]. This can occur at various levels (see figures 4.2, 4.3 and 4.4).

Fusion at the classifier level (decision fusion) combines the result of individual classifiers operating on independent sensors and is commonly used in HCI and activity recognition as it allows the fusion of heterogeneous sensors. It is implemented by a meta-classifier that combines the decisions provided the sensor specific classifiers [108]. Various meta-classifiers are available (see section 4.4).

Most state of the art activity recognition systems reviewed above tend to aim at the best activity recognition performance. Power reduction is often considered as a by-product of careful hardware selection [84] and improved wireless protocols [208, 243].

Power-performance trade-offs is usually performed at node level by adapting parameters such as clock rate [114], or sample rate [99], as well as signal processing window sizes and overlap [16, 189]. These approaches are mostly applied to single sensor nodes and do not consider the sensor network as a whole.

Duty cycling [42] and energy harvesting [148] are typical approaches to extend node lifetime in WSN. However, the unpredictability of energy supply typical of harvesting makes it difficult to manage duty cycling schedules [215]. Furthermore, activity recognition systems require constant data sampling and processing, which makes such approaches inappropriate, unless additional information is used to wake up the system the moment an activity or gesture starts.

Power can be managed at the network level. This has been extensively studied in environmental and ambient monitoring with WSN. Examples include optimized routing algorithms [198, 169], clustering algorithms for redundant data reduction [31, 217] and data compression techniques [176]. In a dense mesh of nodes, a number of sensors can be turned off to extend the WSN lifetime while other nodes still cover the area of interest [28] or by turning on parts of the system only when an event of interest occurs [80]. However such optimizations typically focus on rare events [56, 138], or periodic data collection, rather than continuous data acquisition and classification required for activity recognition.

An attempt to balance power consumption and performance in a gesture recognition scenario is presented in [149]. Here gestures are grouped according

to pre-defined *Situation*. Each *Situation* can be easily recognized with only a subset of nodes, thus the others can be kept into low power state.

6.2.2 Tangible interfaces

The development of smart objects is an active field of research [202, 44, 48, 199, 72]. With the objective of enhancing the interaction with smart environment, smart objects can be used as tangible interfaces and play a fundamental role in improving human experience within interactive spaces for entertainment and education [95].

Interactive surfaces are a natural choice when developing applications that deal with browsing and exploration of multimedia contents. On these surfaces users can manipulate elements through direct and spontaneous actions. This research led to systems based on gesture recognition and analysis of users bare hands [158, 77, 9].

In the case of complex applications, featuring multiple options and actions, simple and spontaneous hand gestures turn out to be not enough. Solutions could be:

1. use an extended set of complex gestures to include an wide vocabulary of actions.
2. use specific interface elements such as menus and icons.

However, the former risk to distort the naturalness of interaction while the latter reduce the directness of interaction causing conflict between digital contents and interface elements, both sharing the same visualization area. The result is that such solutions could increase the user cognitive load without significantly improve the interaction level.

Tangible user interfaces (TUI) can be an alternative solution to the mentioned techniques. TUI are smart objects that the system interprets as part of the interaction language [67]. Users, manipulating those objects, inspired by their physical affordance, can have a more direct access to functions mapped to different objects [186].

TUIs have a broad literature; several systems approached the use of passive physical objects with recognizable shapes or encodings [97, 75] , as well as smart objects embedding sensors [113, 201].

Several examples include digital desks or tables as in the work of Mazalek et al. [136] or the recently presented Microsoft Surface Computing platform [140] where the focus of the interaction design is on the relationship between the physical and the digital object.

Other works exploit TUI for Mixed Reality (MR) applications. Kotranza et Lok presents the results of a pilot study in which eight ($n = 8$) physician-assistant students performed a clinical breast exam on the MRH patient [112]. In the work of Dias et al. novel uses for TUIs are proposed since such objects can trigger the functionalities of the virtual system [50]. Finally the work of Lee et al. shows that tangible MR provides more cost-effective and reliable visualization and simulation for the existing pervasive environment. They embed virtual objects into the real smart environment in a way that support the service provided by the smart home [121].

6.2.3 Overview of available datasets

UCI Machine Learning Repository

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The archive was created as an ftp archive in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets. It has been cited over 1000 times, making it one of the top 100 most cited "papers" in all of computer science. The current version of the web site was designed in 2007 by Arthur Asuncion and David Newman [5], and this project is in collaboration with Rexa.info at the University of Massachusetts Amherst.

Currently 177 data sets are maintained as a service to the machine learning community. Among the many we can find:

- **Iris Data Set.** This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day [54]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. The attributes are the measure width and length of sepal and petal of different class of iris flowers.
- **UJI Pen Characters Data Set.** One of the newest added datasets it contains samples sentences from 60 writers at two different sites in two phases. Each writer contributed with letters, digits, and other characters and two samples were collected for each pair writer/character. Writers were instructed to clear the content of the corresponding box by using an on-screen button and try again whenever they made a mistake or were unhappy with the writing of any character. Subjects were monitored only

when writing their first exemplars and every sample considered OK by its writer was accepted, even if some of its points lay out of the corresponding acquisition box. Only X and Y coordinate information was recorded along the strokes by the acquisition program, without, for instance, pressure level values or timing information. Thus, in multi-stroke samples, no information at all was recorded between strokes. Both coordinates were expressed as integer ink units, with the origin lying at the top left corner of the corresponding acquisition box. X values grow left-to-right and Y values grow downwards.

- **Census Income (KDD) Data Set.** This data set contains weighted census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau. The data contains 41 demographic and employment related variables.
- **Parkinsons Data Set.** The dataset was created by Max Little of the University of Oxford, in collaboration with the National Center for Voice and Speech, Denver, Colorado, who recorded the speech signals. This dataset is composed of 195 voice recordings of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). The main aim of the data is to discriminate healthy people from those with PD.

The PlaceLab at MIT

The PlaceLab [94] is a real home where the routine activities and interactions of everyday home life can be observed, recorded for later analysis, and experimentally manipulated. Volunteer research participants individually live in the PlaceLab for days or weeks, treating it as a temporary home. Meanwhile, a detailed description of their activities is recorded by sensing devices integrated into the fabric of the architecture.

The PlaceLab has been developed as a complement to existing tools and methodologies for gathering data on behavior and use of technology in home settings (e.g., laboratory user studies, surveys, interviews, ethnographic observation) since studying behavior in naturalistic living environments allows researchers to better understand how to create technologies that respond to and respect the complexity of life.

Figure 6.1 presents some image from the interior of the 1000 square foot lab and its floor plan. The lab consists of a living room, dining area, kitchen, small office, bedroom, full bath and half bath.

The PlaceLab is equipped with: 18 microphones, 9 color cameras, 9 infrared cameras, 8 switches to detect open-close events (such as opening of the refrigerator or the lighting of a stove top burner ecc.), 34 temperature sensors, 10



Figure 6.1: Floor plan and pictures of the interior of the PlaceLab.

humidity sensors, 5 light sensors, 1 barometric pressure sensor, 37 electrical current sensors, 11 water flow sensors and 2 gas flow sensors.

Several datasets have been collected in the PlaceLab, three of them are public:

- **PLIA1.** This sample PlaceLab dataset was recorded on Friday March 4, 2005 from 9 AM to 12 noon with a volunteer who was familiar with the PlaceLab, but not a creator of the core technical infrastructure. The researcher was asked to perform a set of common household activities during the four-hour period that included the following: preparing two recipes, doing a load of dishes, cleaning the kitchen, doing at least two loads of laundry, making the bed, and light cleaning around the apartment. The volunteer determined the sequence, pace, and concurrency of these activities and also integrated additional household tasks. The intent was to have a short test dataset of a manageable size that could be easily placed on the web without concerns about anonymity. The dataset shows a variety of activity types and activate as many sensors as possible, but in a natural way. In addition to the activities above, the researcher searches for items, struggles to use an appliance, talks on the phone, answers email, and performs other everyday tasks. The researcher wore two mobile accelerometers (one on the left thigh and the other on the right wrist) and a wireless heart rate monitor.
- **PLIA2.** It is the same as PLIA1 recorded one year later on Friday March 24, 2006 from 10AM to 2PM. During that year several improvement were

made to the PlaceLab infrastructure: a robust visualization and annotation tool has been created, high-frame rate still images with time synchronized audio snippets are saved instead than using video codec, the latency of all 1-wire sensors have been reduced, the sampling rates for temperature, humidity, light, etc. sensors has been increased, the activity ontology has been revised and small mobile phones replaced PDAs for self-report applications.

- **PLCouple1.** This dataset consists of all easily anonymized sensor data for a 2.5 month period when a couple stayed in the PlaceLab. The participants were encouraged to maintain as normal a routine as possible. They went to work, had visitors over, cooked meals, and worked on projects and leisure activities according to their own preferences. They brought objects such as small appliances, clothing, bedding, boxes of books and audio tapes, and food from their own home when they moved in. Although they were living away from home, the relatively long duration of the experiment allowed the residents to acclimate to the apartment [125].

ICDM Data Mining Contest on localization

The first IEEE ICDM Data Mining Contest (IEEE ICDM DMC07) [234] was held in conjunction with the 2007 IEEE International Conference on Data Mining (IEEE ICDM 2007). This contest is about indoor location estimation from radio signal strengths received by a client device from various WiFi Access Points (APs). This is a problem of practical significance and technical challenge. Indoor location estimation in wireless networks using Received Signal Strength (RSS) values has attracted great interests in data mining and machine learning communities. Many applications rely on this task, ranging from robotics to context-aware computing can now be realized with the help of distributed wireless networks, to security related applications, and to mobile commerce and health care for the sick and elderly. The problem can be visualized by considering the following scenario:

A person holding a wireless client device walks around a building floor. The client device (which can be a PDA) is equipped with a wireless card that can receive signals from many surrounding wireless access points (APs). Each of these APs is identifiable with a unique ID. Based on the collection of signal strength values (RSS values), a data mining algorithm running on the client device tries to figure out the current location of the user.

A typical way to do this task is through triangulation. However, triangulation methods cannot cope with the uncertainty associated with the RSS values.

Thus, for this contest some training data have been collected and data mining and machine learning methods have been used to locate the user. The ICDM contest data set will serve as a benchmark data set in comparing solutions for such a challenging and practical problem as localization.

All WiFi data are collected in approximately 200 locations, where each location is a grid. A grid has a size of about $1.5m \times 1.5m$. The RSS values include a set of IDs for the access points (AP) and their corresponding RSS values (received signal strength). The larger the RSS value is received from an access point AP₁, the closer to the AP₁ is the client device.

ImageParsing

Imageparsing.com (IP) is a web site affiliated with the Lotus Hill Institute (LHI)-an independent, non-profit, international research organization, established at EZhou, China, in 2005 by Dr. Song-Chun Zhu (Professor, Statistics and Computer Science, UCLA)

In order to advance the field of computer vision, the objective of the IP website is to provide large scale annotated ground truth data to the general vision community [235].

The ground truth data set intends to cover almost all aspects of the computer vision and pattern recognition research: edges, contours, contour attributes, segmentation, grouping, occluded contour completion, text, object category recognition, scene, 3D world frames, UAV images, Google Earth images, video, and cartoon. The ground truth data are stored in a unified data structure the And-Or graph representation and organized in a Database (MySQL) for retrieval and search. It has now over 3 million annotated object nodes by June, 2007.

Natural images consist of an overwhelming number of visual patterns generated by very diverse stochastic processes in nature. The objective of image understanding is to parse an input image into its constituent patterns. Figure 6.2 is an example of parsing a stadium scene hierarchically: human (face and clothes), sports field (a point process, a curve process, homogeneous color regions, text) and spectators (textures, persons).

6.3 Activity recognition from body worn sensors: scalability and robustness

In this section we wish to investigate activity recognition in the challenging context of large and dynamically changing sensor networks.

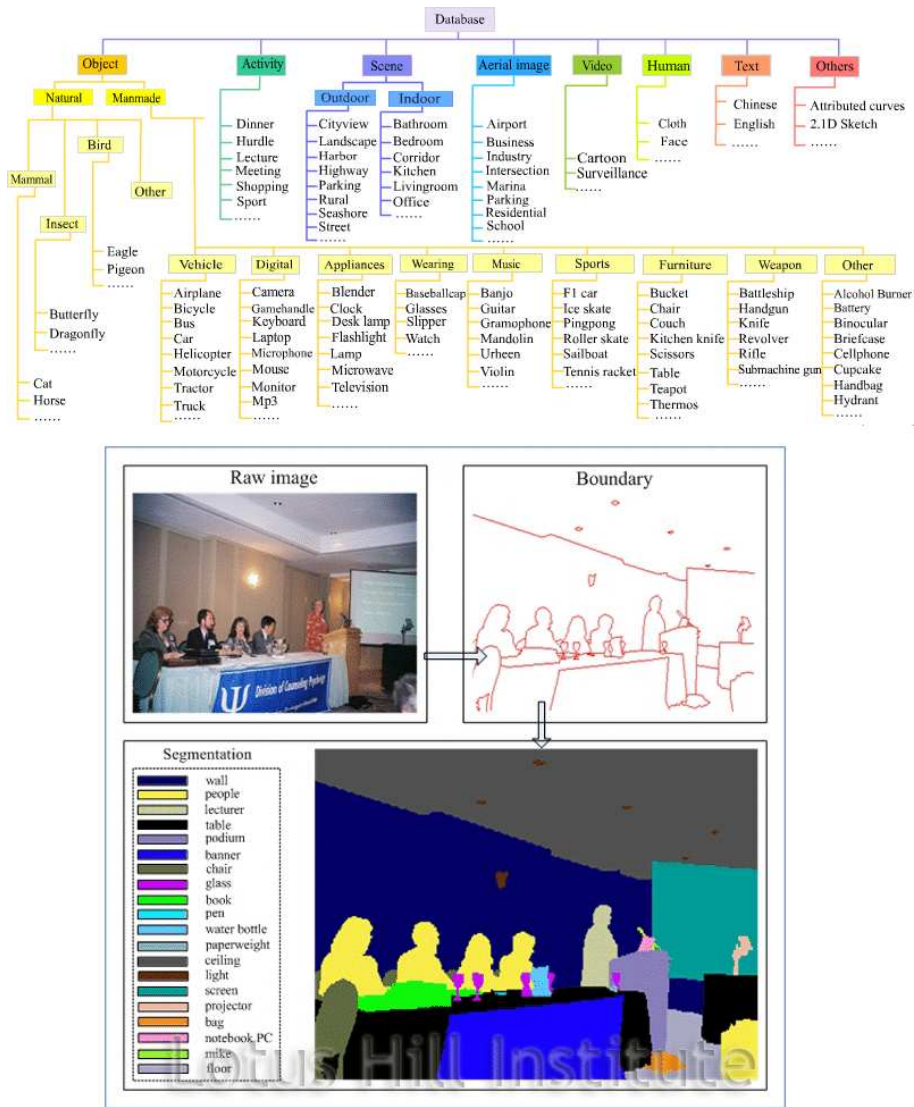


Figure 6.2: Example of image analysis and information extraction.

This work is part of an ongoing effort to explore activity recognition in large networks of simple sensors distributed on the body. Energy constraints favors low-power and miniature sensors, such as MEMS accelerometers or textile stretch sensors in contrast to inertial measurement units (which combine accelerometers with gyroscopes and magnetometers) which are more power hungry. These sensors may be unreliable or subject to faults and their placement may be subject to jitter during operation due to repeated body motion or the use of loose fitting clothes. We envision activity recognition in such *dynamic* sensor network where the number and type of nodes may even change over time, with sensors added or removed at run-time.

We investigate the use of sensor fusion techniques for gesture recognition. A meta-classifier fuses the information of simple classifiers operating on individual sensors. We investigate the outcomes of classifier fusion in function of the number of sensors on the recognition performance (*sensor scalability*), and on the robustness to faults (*robustness*).

We validate this method using a set of 10 activities from a quality assurance checkpoint of a car assembly line. We show that this approach allows scalable system performance, and intrinsic robustness to faults and noise.

6.3.1 System architecture

The outline of our architecture is illustrated in figure 6.3. Sensor data is first acquired and preprocessed. Preprocessing consists of feature extraction. Features are classified individually for each sensor, leading to class labels. Finally these class labels are fused, which yields the likely activity class corresponding to sensor data.

Classification of activities from accelerometers

We use three-axis accelerometers for activity recognition because they are small and inexpensive.

We consider two possible preprocessing variants.

In this work we consider isolated activity recognition, thus here preprocessing do not include segmentation. Rather we assume that sensor nodes are able to detect the beginning and the end of activity occurrences. Various methods are available to identify segments in data streams likely to contain an activity [104, 47, 111].

In the first variant (*individual acceleration axis*), we consider each acceleration sensor axis as a standalone one-dimensional "sensor" that provides a discrete signal s_t^{ij} (i indicates the sensor node, j one of the three axis x , y or z ; t denotes the signal sample). Thus, each sensor node provides three data streams (one for

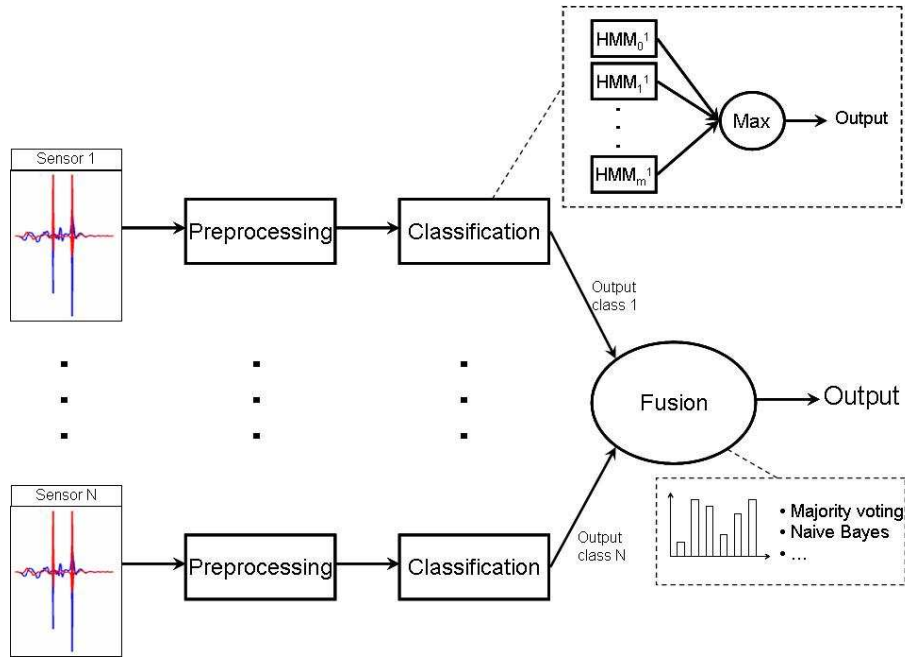


Figure 6.3: Activity recognition architecture. Features extracted from the sensor data are classified by competing hidden Markov models, each one trained to model one activity class. The most likely model yields the class label. These labels are fused to obtain an overall classification result. A naive Bayesian scheme, borda count scheme and a majority voting scheme are used.

each axis). Each data stream is processed by independent activity recognition signal processing chains up to node-level class output. In other words, each sensor node provides three class decision outputs.

In the second variant (*acceleration magnitude*), the three axis of the acceleration sensor ($s_t^{i,x}$, $s_t^{i,y}$, $s_t^{i,z}$) are combined into a single acceleration magnitude signal: $s_t^i = \sqrt{s_t^{i,x^2} + s_t^{i,y^2} + s_t^{i,z^2}}$. Thus, each sensor node provides one data stream (the acceleration magnitude), that is processed by a single activity recognition processing chain up to node-level class output. This reduces computational load, provides better robustness against sensor rotation, but may lead to information loss. In other words, when using acceleration magnitude, each sensor node provides a single class decision output.

Features are extracted from the sensor signal to reduce the input dimensionality to the classifier and highlight important signal properties. Discrete feature symbols that indicate the acceleration magnitude (negative acceleration, positive acceleration, and no acceleration) are obtained by *ternarizing* the acceleration amplitude with two thresholds. This allows to use HMMs with discrete observations that are significantly less computationally demanding than HMMs operating on continuous observations.

The conversion of the sensor signal s_t^i into a feature symbol f_t^i is done by

means of two thresholds $R - \Delta R$ and $R + \Delta R$ as follows:

$$f_t^i = \begin{cases} - & \text{for } s_t'^i < R - \Delta R \\ 0 & \text{for } R - \Delta R \leq s_t'^i \leq R + \Delta R \\ + & \text{for } s_t'^i > R + \Delta R \end{cases} \quad (6.1)$$

R is equal to $0g$ for individual acceleration axis and to $1g$ for the variant acceleration magnitude.

The sequence of features are then classified at node level using a set of HMM (see section 4.5.1). As many HMMs as activity classes are defined and trained to model the activity classes. These HMMs compete and the one modeling best the features indicates the class label (see figure 6.3).

We use ergodic (fully connected) discrete HMMs with 4 hidden states. The possible observations are the 3 acceleration features. Training is performed by optimizing the HMM parameters with the Baum-Welch algorithm starting from HMMs with randomly initialized parameters. For each model, optimization was repeated 15 times with a different random initialization and the HMM modeling best the target gesture was selected. The model likelihood is estimated using the Forward algorithm. We used the Kevin Murphy HMM Toolbox for this purpose.

The classification result c_{out}^i of node i is sent through the network to fuse it with the decisions of other nodes to obtain the network-level activity recognition C_{out} .

Classifier fusion

Among classifier fusion methods (see section 4.4), we consider a majority voting scheme (section 4.4.1), borda count ranking scheme (section 4.4.2) and a naive Bayesian fusion method (section 4.3.1).

These methods are tractable for wearable systems and cope with a change in the number of sensors, such as when a sensor fails, without needing any retraining.

When using a majority voting fusing scheme the final output class C_{out} is selected as the most represented among the node level classifier outputs c_{out}^i .

When using a borda count fusing scheme each node level classifiers need to produce a class ranking. This is done by sorting the classes c in ascending order, according to the probabilities $P(O|\lambda_c)$. Each node i thus provides a vector \vec{c}_{out}^i with the sorted list of classes.

When using the naive Bayes classifier for fusion of high level inferences we require a starting offline training phase to extract the *likelihood* values for each node level classifier output and each class ($P(c_{out}^k = c_j | C_{real} = c_i)$) for

$1 < k < M$ and $1 < i, j < C$, where k is the index of the nodes of the network, M in total, and i and j are the indexes of the class, C in total). In particular we build an $N \times N$ *Node Statistics* matrix NS_k which has the structure presented in equation 6.2, where N is the number of classes that have to be detected.

$$NS_k = \begin{pmatrix} t_{11}^k & t_{12}^k & \dots & t_{1N}^k \\ t_{21}^k & t_{22}^k & \dots & t_{2N}^k \\ \dots & \dots & \dots & \dots \\ t_{N1}^k & t_{N2}^k & \dots & t_{NN}^k \end{pmatrix} \quad (6.2)$$

Each element of the matrix is the M-estimate of the likelihood (see equation 4.7 section 4.3.1) with $m = 1$ and $p = \frac{1}{N=10} = 0.1$.

The naïve Bayes classifier classifies the gesture according to the following:

$$C_{out}(c_{out}^1, c_{out}^2, \dots, c_{out}^N) = \underset{c}{\operatorname{argmax}} \left[\frac{P(C = c) \prod_{k=1}^n t_{c, c_{out}^k}^k}{P(c_{out}^1, c_{out}^2, \dots, c_{out}^N)} \right] = \underset{c}{\operatorname{argmax}} \left[\prod_{k=1}^n t_{c, c_{out}^k}^k \right] \quad (6.3)$$

Where the last step is possible because the probability at the denominator in equation 6.3 is identical for all classes and we assume that the *Prior* probability $P(C)$ is constant for all classes.

6.3.2 Experimental setup

To assess our approach we apply it to an activity recognition scenario within the quality checkpoint of a car manufacturing plant [196]. The knowledge of workers' activities enables context-aware support [134, 194].

Within the quality checkpoint, workers must verify the functionality and quality of the car. This is done by visual and tactile inspection. For instance, all car parts must be operated and the presence of scratches and the smoothness of surfaces must be sensed. These activities translate into characteristic limb gestures. Out of the 46 activities performed in this checkpoint [197] we selected the subset of ten gestures listed in table 6.1.



The user holds a notepad with his left hand and writes down a short note with his right hand.



The user opens the hood with his left hand and blocks it with a stick kept with his right hand.



The user removes the stick with his right hand while keeping the hood with his left hand then closes the hood with his left hand.



The user checks the gaps on the front door by sliding his left and right hand over the gaps. The two hands move simultaneously.



The user grabs the car left front door with his left hand while it is closed and opens it completely.



The user grabs the car left front door with his left hand while it is open and closes it completely.





| | |
|--|---|
|  |  |
| <p>The user grabs the car left front and back doors with his left and right hands than open and close completely and at the same time the two doors.</p> | <p>The user checks the gaps on the trunk by sliding his left and right hand over the gaps. The two hands move simultaneously.</p> |
|  |  |
| <p>The user opens the trunk using both hands and then moves it up and down on the top of his head three times before closing it.</p> | <p>The user grabs the steering wheel with both hands and turns it clockwise and counter-clockwise three times.</p> |

Table 6.1: List of activity classes to recognize from body-worn sensors.

We equipped a subject with 20 sensor nodes containing a 3-axes accelerometer (Analog Device ADXL330) placed on the two arms (10 on each arm) as illustrated in figure 6.4. To ensure generality of the results, the sensors were placed to cover the two arms without any particular constraints (no specific position or orientation). We recorded a data set composed of 70 repetitions of each gesture listed in table 6.1, for a total of 700 gesture instances. Acceleration

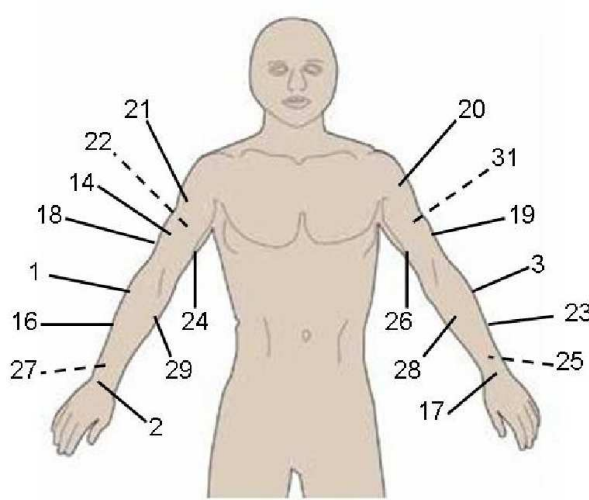


Figure 6.4: Placement of the acceleration sensor nodes on the user. Ten nodes are uniformly distributed on each user arms (20 nodes in total). No specific assumption has been made on sensor position and orientation.

data was sampled at 100Hz.

6.3.3 Test and results

In the remaining of the section we refer to the symbol sequences obtained from individual acceleration sensor axis after ternarization as *Single axis sequences*, and to the symbol sequences obtained from combining the three acceleration sensor axes of a node into a magnitude vector, after ternarization, as *Magnitude sequences*.

Node-level activity recognition performance

The pre-processing thresholds ΔR of equation 6.1 is optimized to maximize the classification accuracy of the node classifiers.

System performance is computed by classifying gesture instances using a single acceleration axis of a given node (in the single axis sequence case) or a triplet of acceleration axis from a given node (in the acceleration magnitude case). This is repeated for all axis and nodes, and averaged. Figure 6.5 illustrates the effect of ΔR on the system performance.

In the rest of this paper the values of ΔR providing the best performance is used: $R = 0g$ and $\Delta R = 400mg$ for the single axis pre-processing method; and $R = 1g$ and $\Delta R = 40mg$ for the magnitude pre-processing method.

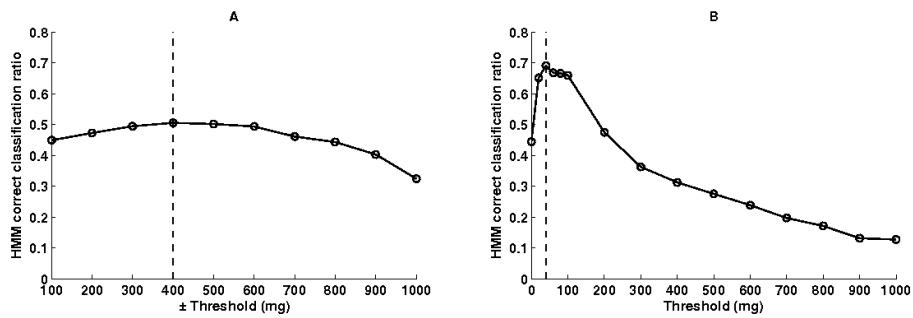


Figure 6.5: Average node level classifier (HMM) classification accuracy as a function of the threshold used to ternarize the input data from accelerometers when single axis acceleration features (left) or acceleration magnitude (right) features are used.

Network-level activity recognition performance and sensor scalability

To show how performance scales with the number of nodes that participate to the gesture recognition we evaluated the classification accuracy over the whole test set of a cluster of increasing size, from 1 to 20 sensors. For each cluster size we performed 50 trials, using randomly picked nodes, and we report the average accuracy. When single axis sequences are used, we select the sensor axes in a node-wise manner: when one node is selected, all of its three axes are used, but each of them is considered as an independent sensor.

In figure 6.6 we illustrate the average performance of the system as a function of the number of nodes within the network, for the various sensor fusion methods and pre-processing method. In figure 6.7 we compare the different fusion methods for a given pre-processing technique.

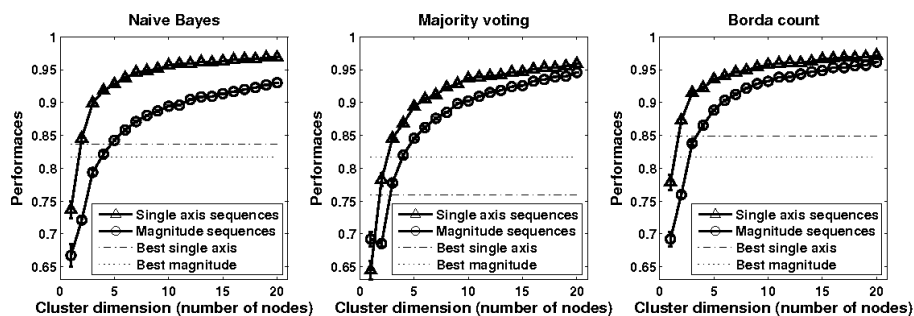


Figure 6.6: Sensor scalability. Average classification accuracy as a function of the number of nodes participating to the classification when different network level fusion methods are used. Vertical bars indicate classification variance. In each sub-plot single axis acceleration features and acceleration magnitude features are compared. Dashed lines indicate the best performance achieved using a single node, the fusion of an increasing number of sensors quickly results in better performance.

System performance increases with the number of nodes participating to

the classification. For the larger networks, the addition of one more node results in lower increase of performance than for smaller networks since the new sensor is more likely to provide redundant information and thus only slightly improves the overall system knowledge.

The use of single axis sequences results in higher performance. This can be understood because computing the magnitude of a vector is not an injective function. Thus, different vectors may produce the same magnitude, resulting in a loss of information.

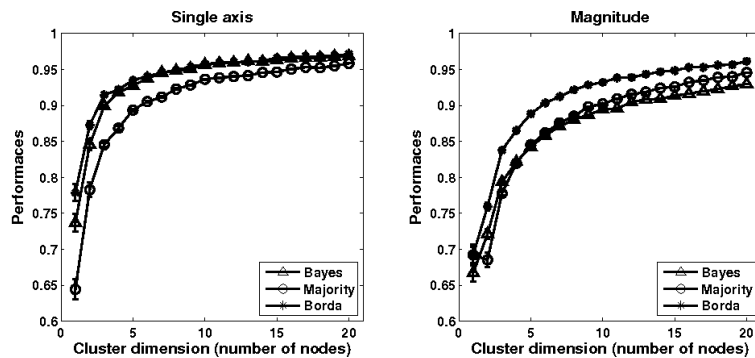


Figure 6.7: Sensor scalability. Average classification accuracy as a function of the number of nodes participating to the classification. Vertical bars indicate classification variance. On each plot different network level classifiers are compared.

The Borda count fusion method shows the best performance in comparison to the other two fusion methods, both with single axis acceleration and acceleration magnitude sequences. Moreover, the performance with naïve Bayes fusion decreases in comparison to the other fusion methods when using acceleration magnitude sequences, and it is outperformed by the other two fusion methods for cluster larger than five nodes.

Majority voting and Borda count present the same results for magnitude pre-processing with cluster of size 1 since the methods are identical when only one decision is available.

Table 6.1(a) and 6.1(b) summarize the results for all the methods and pre-processing techniques for cluster sizes 1, 5, 10 and 20. In Figure 6.8 we represent the confusion matrix obtained when using single axis sequences and naïve Bayes fusion method. As the number of sensor increases, the diagonal of the matrix is emphasized, reflecting the performance increase. By using 20 nodes we achieved more than 95% classification accuracy with all of the three fusion algorithms and single axis sequences. The use of magnitude sequences resulted in slightly lower classification performance, although with lower computational cost (a single instead of 3 classifiers per node).

(a) Single accelerometer axis features

| # | Fusion | Performance (%) |
|----|----------|-----------------|
| 1 | Majority | 64.4 |
| 1 | Borda | 77.9 |
| 1 | Bayes | 73.7 |
| 5 | Majority | 89.3 |
| 5 | Borda | 93.5 |
| 5 | Bayes | 92.8 |
| 10 | Majority | 93.6 |
| 10 | Borda | 95.7 |
| 10 | Bayes | 95.7 |
| 20 | Majority | 95.9 |
| 20 | Borda | 97.1 |
| 20 | Bayes | 96.9 |

(b) Acceleration magnitude features

| # | Fusion | Performance (%) |
|----|----------|-----------------|
| 1 | Majority | 69.2 |
| 1 | Borda | 69.2 |
| 1 | Bayes | 66.7 |
| 5 | Majority | 84.5 |
| 5 | Borda | 88.8 |
| 5 | Bayes | 84.2 |
| 10 | Majority | 90.2 |
| 10 | Borda | 93.2 |
| 10 | Bayes | 89.4 |
| 20 | Majority | 94.6 |
| 20 | Borda | 96.1 |
| 20 | Bayes | 93.0 |

Table 6.2: Performance comparison between different fusion methods, features, and number of active nodes.

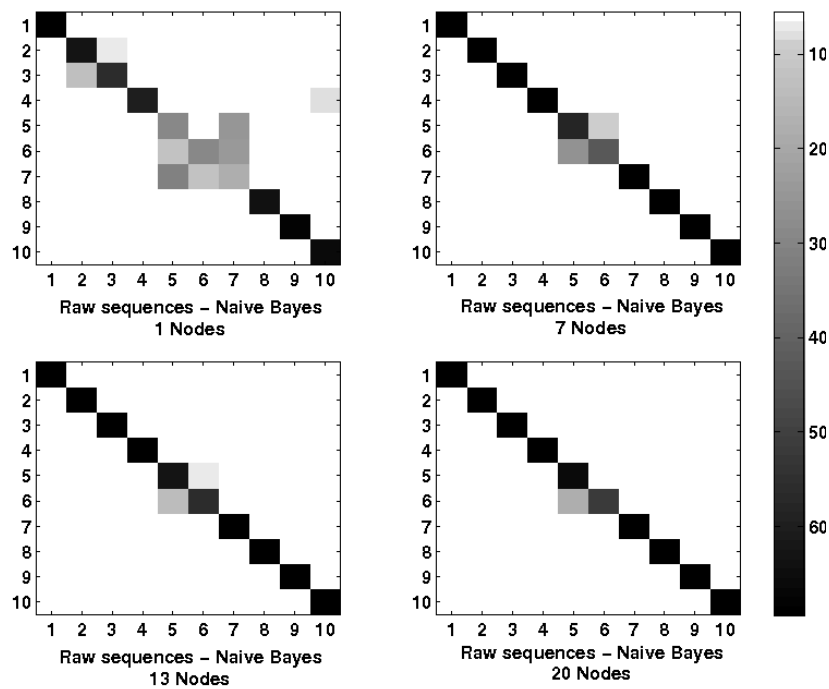


Figure 6.8: Graphical representation of the confusion matrix when 1, 7, 13 or 20 sensor are participating to the classification. Naïve Bayes fusion and single axis acceleration features are used. Darker spots indicated higher number of classified instances. As can be seen from these plots, increasing the number of sensors results in a drift toward the main diagonal meaning higher classification accuracy.

Robustness to noise and faults

During regular operation sensors may be affected by noise or faults that alter the input sequences. We evaluate the robustness of the gesture recognition system to different noise sources. To the best of our knowledge no previous effort tried to model the noise that may affect body worn accelerometers, thus we define 2 sources of noise likely to occur: rotational noise and random noise.

Rotation noise: the nodes, due to loose fitting garments that bend during normal activity, may change their position and orientation around their attachment point. At this stage we ignore changes in position and we focus on changes of orientation. We model this by a rotation of the coordinate system of the accelerometer. We define the new coordinate system by successive application of a rotation along the X, Y, and Z axes. The acceleration vector from the accelerometer is then projected onto this new coordinate system and subsequently processed as if it were the real recorded acceleration signal. The rotation along each axis is a random value within the range $[0^\circ; \alpha_{max}]$. We test the classification accuracy with α_{max} varying from 5° to 60° in 5° step. Note

that acceleration magnitude features are insensitive to the rotational noise.

As previously for the sensor scalability analysis, we repeat the test 50 times for each cluster size and noise level and present averages, with a randomly selected set of nodes.

Figure 6.9 illustrates the performance of the system with clusters of nodes of different size as a function of α_{max} . Dashed lines show the performance of the same system when using magnitude sequences. Figure 6.10 details as an example the performance of the system as a function of number of nodes and the noise level when using naïve Bayes. In figure 6.11 we compare the robustness of three fusion methods for different cluster sizes. Naïve Bayes fusion is more robust than the other two methods when the noise increases.

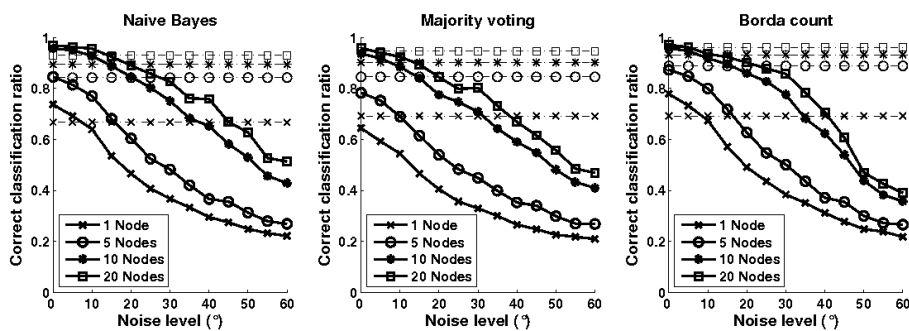


Figure 6.9: Robustness to rotational noise. Average correct classification ratio of a system with 1, 5, 10 or 20 active sensors as a function of the rotational noise added. Increasing the rotational noise results in decreasing classification accuracy. This degradation can be compensated by increasing the number of active nodes.

Noise decrease classification accuracy. This effect can be compensated by augmenting the number of nodes within the network. For example, using Borda count with a single node we can achieve almost 80% classification accuracy without noise. The higher the noise level, the larger the clusters should be to provide immunity to noise and maintain the initial performance. With rotational noise up to 10° , the initial performance can be maintained by increasing the network to 5 nodes. With 10 nodes we can tolerate up to 20° of rotational noise and almost 35° with 20 nodes while keeping the same performance.

Note that in these tests all the nodes of the network are affected by the same level of noise. In a real scenario, it is more likely that only a part of the active nodes is affected, thus resulting in better performance.

These results highlight that the use of magnitude sequences may be overall a better choice if accelerometer orientation in a system is likely to be variable since it outperform single axis accelerations performance even for low noise levels. This may prove important in the design of loose fitting garments with integrated sensors, where a small rotational noise and position variability is

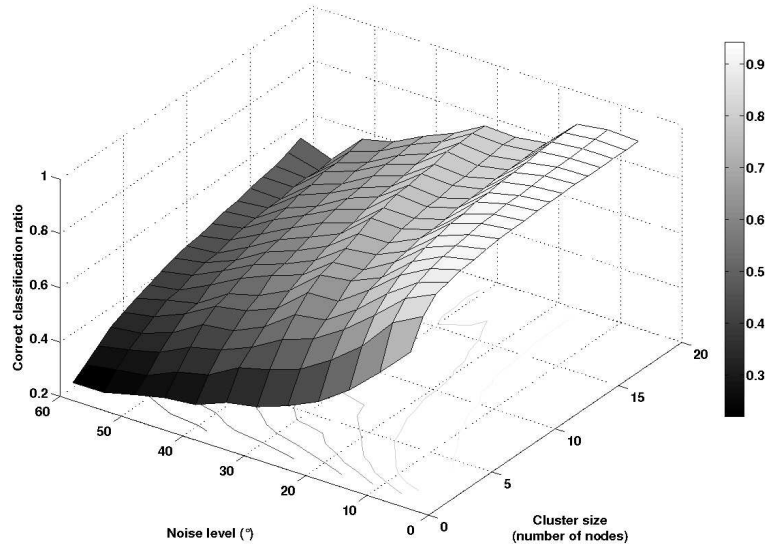


Figure 6.10: Average classification accuracy of the system as a function of both level of rotational noise added to accelerometers output and number of sensors participating to classification. Naïve Bayes fusion method and single axis acceleration features used.

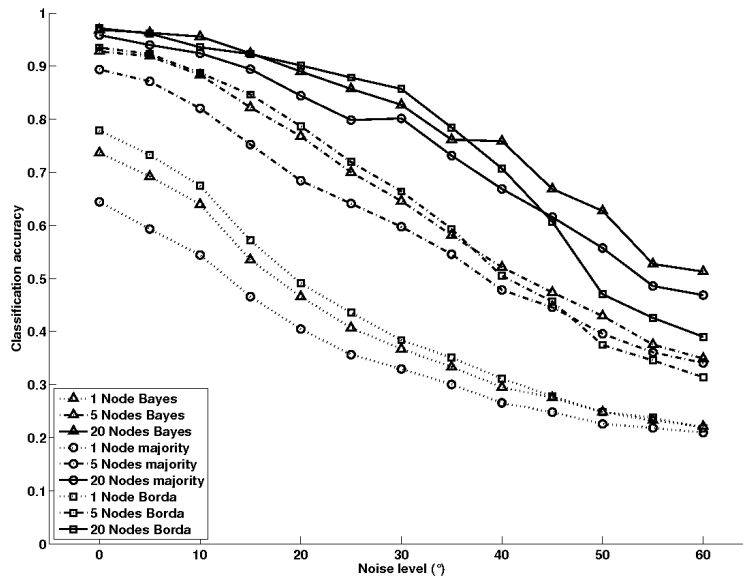


Figure 6.11: Average classification accuracy when 1, 5 or 20 nodes participate to the classification. The performance using different fusion methods are compared as the rotational noise added to the accelerometer output increases.

likely to be always present [78].

Faults: We assess robustness against failures seriously affecting the system, such as communication errors, or misclassification. We model these errors in a generic way by randomly changing the decision output of a classifier. This corrupted output is then processed normally (i.e. it is fused with the decisions of the other nodes). With the naïve Bayes and majority voting fusion, the decision output of a node is randomly changed. With the Borda count fusion the decision vector output (ranked list) is randomly mixed.

To determine fault tolerance to this kind of noise we calculate the classification accuracy of a cluster of 20 nodes, with an increasing number of nodes affected by random noise. We repeat the evaluation 50 times, each time picking a random set of nodes as faulty ones, and report the average performance.

Figure 6.12 shows the average performance of the system as a function of the number of nodes affected by noise for the various fusion and pre-processing methods.

The three methods have similar performance trends as the number of faulty nodes increases, but majority voting and naïve Bayes fusion show better robustness to this type of faults than Borda count (see figure Figure 6.13).

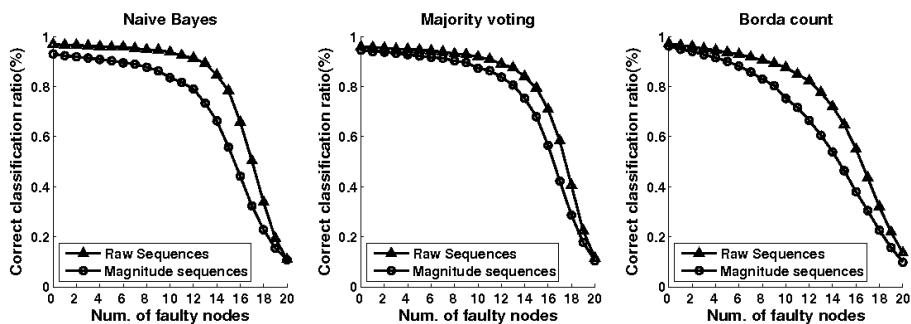


Figure 6.12: Robustness to random noise. Average classification accuracy of a system with 20 active nodes with an increasing number of nodes affected by random noise. Single axis acceleration features and acceleration magnitude features are compared.

Although larger noise levels decrease classification accuracy, sensor fusion allows to reduce the impact of faults in larger sensor sets and performance remains relatively constant up to high fault rates. In some cases, even when more than half the nodes are faulty the performance of the system shows little degradation. Interestingly, this robustness to faults is provided without an explicit fault detection mechanism, and is an inherent advantage conferred by having multiple nodes contributing to the overall classification.

All three methods show similar performance, around 10%, when all nodes of the network are faulty. This comes from the fact that all the classifier outputs are random, and the class decision is randomly distributed among the ten

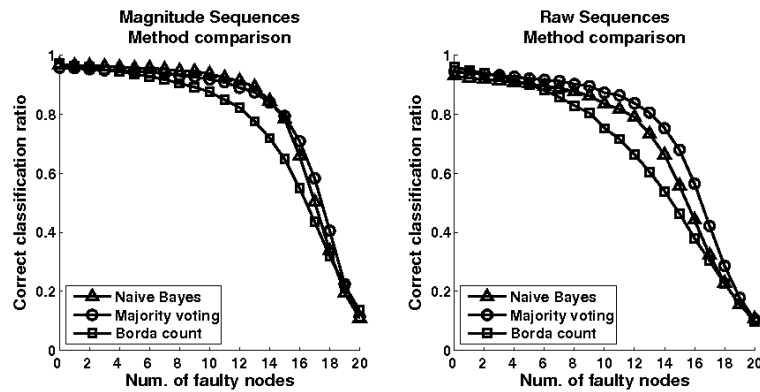


Figure 6.13: Robustness to random noise. Average classification accuracy of a system with 20 active nodes with increasing number of nodes affected by random noise. The majority voting, Borda count and naïve Bayes fusion methods are compared using different features.

possible classes.

6.3.4 Discussion

In previous sections we showed how it is possible to combine data from a large number of small and cheap accelerometers, easily integrable into garments, in order to achieve high gesture recognition accuracy even in presence of noisy data or unreliable sensors.

Such information can be exploited to vary the number of working sensor according to dynamic application constraint. For this objective we can augment the number of sensor used, and thus the correct classification ratio, only in critical situations and keep unused sensor in a low-power, idle state in order to increase network life.

A comparison of the results obtained for sensor scalability and tolerance to random noise indicate that a node providing a wrong decision has a stronger impact on system performance after decision fusion than not providing a decision at all.

In figure 6.14 we highlight this in case of single accelerometer axis sequences and naïve Bayes fusion method, with a cluster of 20 active nodes. The average performance of the system as the number of active nodes is reduced is compared to the average performance as the number of faulty nodes is increased. At high fault rates, the results show that it is beneficial to exclude sensors providing corrupted decisions (e.g. if data is corrupted during a transmission) from participating to the decision fusion, rather than to rely on decision fusion to compensate for the error.

Once the HMM models are trained, the classification of a sequence (using

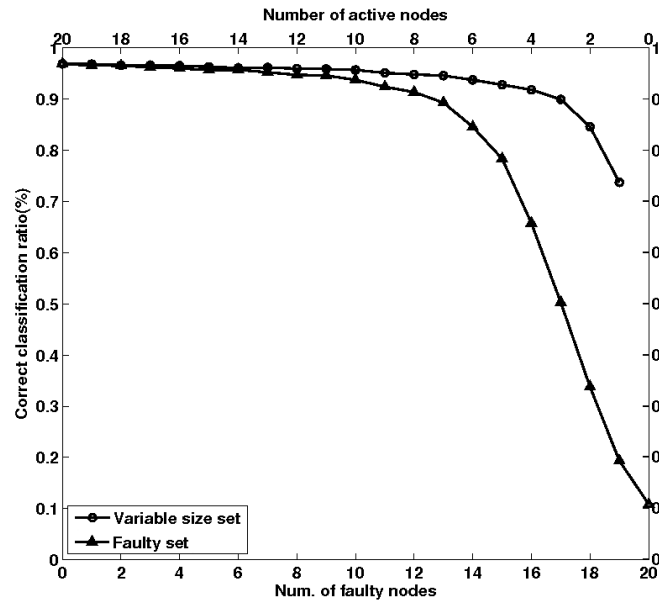


Figure 6.14: Comparison of the average classification accuracy of a cluster of 20 nodes when the number of active nodes is progressively decreased (variable size set), and when the number of nodes affected by random noise is progressively increased. The detection and exclusion of faulty nodes results in higher system performance.

the *forward algorithm*) is a lightweight task that can be computed online while the samples are collected. Each time a sample arrives the node must perform the following steps:

1. pre-process the sample (ternarize or extract magnitude from a triad of samples).
2. calculate one step of forward algorithm for all HMM models: $S^2 + S$ multiplications and S sums, where S is the number of hidden states. In our case we have 10 HMM and $S = 4$, thus we must compute 200 multiplications and 40 sums.
3. normalize through a shift partial results in order to avoid underflow.

As the node recognizes the end of a gesture it must perform S sums for each HMM and find the class with higher probability or sort the classes to build the rank for Borda count. The computational effort of the fusion node is presented in table 6.3 where C is the number of classes to recognize and N the number of nodes to fuse. These steps can be easily done by today's low power sensor nodes, as microcontrollers nowadays usually embed a multiplier.

| Fusion method | Multiplication | Sums |
|---------------|-------------------|-------------|
| Majority | 0 | $C \cdot N$ |
| Borda | 0 | N |
| Bayes | $C \cdot (N + 1)$ | 0 |

Table 6.3: Computational effort

6.3.5 Conclusion

Traditional human activity recognition systems rely on a static signal processing chain that processes the data of a fixed and often minimal set of sensors placed at well-defined body (or environment) locations. Yet, in real-world deployment failures and signal degradation occur. This leads to dynamically changing availability and characteristics of sensing resources. On the other hand, technological advances allows for sensors to become smaller and cheaper. They nowadays become pervasive: from sensor-augmented clothing and accessories, to smart wearable devices and widespread ambient sensing. Thus, given these advances, the problem of activity recognition has to be understood in the context of *large* and *dynamic* sensor networks. This calls for activity recognition methods suited for such systems, and that capitalize on their characteristics.

In this section we demonstrated the benefits for activity recognition brought about by the availability of a large number of sensor nodes, and how real-world challenges can be balanced by an adaptive use of resources. We introduced a *generic* hierarchical architecture to recognize activities in dynamic sensor networks composed of: *individual activity recognition sensor nodes* that operate independently of each other and (ii) a *dynamic classifier fusion core* that combines the decisions of a variable number of nodes into a joint activity classification.

The activity recognition system introduced here addresses a number of the pitfalls of traditional activity recognition systems. Thanks to our approach: (i) sensors are not single points of failure since they can be replaced dynamically; (ii) sensor fusion provides robustness to signal degradation; (iii) performance can be scaled dynamically, and even outperform that of systems using a minimal application-specific sensor set.

We assessed this system in a real-world case study. We used it to recognize a set of 10 activities performed by industrial workers at the quality assurance checkpoint of a car assembly factory. Activities were sensed with up to 20 3D acceleration sensors distributed over the arms of the worker. Within the generic architecture outlined above we used and compared: hidden Markov models for individual activity recognition and majority voting, Borda count and naïve Bayes as decision fusion algorithms.

We demonstrated that system performance can be scaled dynamically by

selecting the number of sensors participating to activity recognition. The three dynamic decision fusion algorithms show similar performance scaling trends, although with slightly different performance values and computational costs.

Borda count showed the best performance results. The classification accuracy is 77.9%, 95.7%, and 97.1% with respectively 1, 10 and 20 nodes when processing acceleration axis independently.

Majority voting is less computationally expensive than Borda count and results in slightly lower performance. The classification accuracy is respectively 64.4%, 93.6%, and 95.9% with 1, 10 and 20 nodes.

Naïve Bayes is the most computationally expensive method among the three. During training it requires a further step to build classification statistics and during recognition it needs to compute the posterior probability of activity classes. The performance of this approach is slightly higher than majority voting but slightly worse than Borda count. The classification accuracy is respectively 73.7%, 95.7%, and 96.9% with 1, 10 and 20 nodes.

Overall, fusion using Borda count offers a good trade off between performance and computational effort.

The fusion of multiple sensors allows to outperform the accuracy obtained from the single best node/classifier of the system (this aspect is analytically discussed in [171]). For instance, by using the single best node/classifier (fusion of the 3 axes of that node) of the system, the resulting performance is 84.9%, 75.9% and 83.7% with Borda count, majority voting, and Naïve Bayes decision fusion. By fusing the result of 10 nodes, the performance can be increased on average by more than 10% over the single best node.

We demonstrated that activity recognition in a sensor network provides intrinsic robustness to noise and faults. We modeled two sources of errors. The first one models the imperfect attachment of nodes. The second models a generic type of faults leading to misclassification (e.g. typically sensor failure, but also communication errors). We showed that there is implicit tolerance to errors provided by the fusion of the decision of multiple sensors in both cases. This may prove important in the design of loose fitting garments with integrated sensors, where the likely sensor placement and orientation variability may be compensated through data fusion.

In comparison to the intrinsic robustness provided by the system, we showed that when nodes misclassify gestures due to faults there is an additional performance benefit in explicitly discarding them from decision fusion. This benefit is larger with higher fault rates, however it becomes negligible with moderate to low fault rates. Consequently, if the system is expected to suffer from likely sensor degradation or failures there are advantages in building a fault detection mechanism in the sensor nodes. However, if sensor degradation or failures

are less likely, the system is able to compensate these faults at the algorithmic level thanks to decision fusion.

The results presented here are generalizable along three lines. First, the activities that are considered to assess the activity recognition system are representative of a wide range of human activities since they involve left and right upper and lower arm motion overall different scales (i.e. the hand trajectory is contained in a volume spanning tens of centimeters up to several meters). This type of activities are common not only in industrial manufacturing, but also e.g. in sports, entertainment, and health care/rehabilitation applications. As a consequence, the range of the results obtained here regarding classification accuracy and number of sensors may be extrapolated to other problem domains sharing similar characteristics.

Second, the activity recognition architecture is generic. While in this work we used hidden Markov models for activity recognition, other classifiers may be used without affecting the generality of the results. The characteristics of the system are brought about by the dynamic classifier fusion core and dynamic sensor selection algorithm. We compared the three families of decision fusion algorithms (single class label, class set reduction, and soft output [175]) with three sensor selection heuristics. In every case we observed similar results (e.g. identical trends when parameters are changed, or similar performance values). This leads us to believe that the results presented here can be extrapolated to other algorithms (classification, decision fusion, and dynamic sensor selection), within the architectural framework introduced here.

Third, the architecture introduced here can be seamlessly applied to activity and context recognition in ambient intelligence environments. The resulting system characteristics (i.e. performance trends, intrinsic robustness, benefits of power-performance management) are likely to be identical, although exact performance numbers will be application specific.

6.4 Activity recognition accuracy-power trade-off by dynamic sensor selection

In an activity recognition system, high classification accuracy is usually desired. This implies the use of a large number of sensors distributed over the body, depending on the activities to detect. At the same time a wearable system must be unobtrusive and operate during long periods of time. This implies minimizing sensor size, and especially energy consumption since battery technology tends to be a limiting factor in miniaturization.

In this section we investigate how to extend network life in an activity

recognition system, while maintaining a desired accuracy, by capitalizing on an redundant number of small (possibly unreliable) sensors placed randomly over the user arms.

We exploit the activity recognition system presented in section 6.3 and we extend it using a *dynamic sensor selection* algorithm in order to modulate the number of sensors that contribute to activity recognition at runtime. Most sensor nodes are kept in low power state. They are activated when their contribution is needed to keep the desired classification accuracy, such as when active nodes fail or turn off due to lack of energy. This approach copes with dynamically changing networks without the need for retraining and allows activity recognition even in the presence of unexpected faults, thus reducing the frequency of user maintenance. The algorithm can be easily parallelized to best use the computational power of a sensor network. We show how this approach fits the Titan framework that we are developing for the execution of distributed context recognition algorithms in dynamic and heterogeneous wireless sensor networks.

Our technique has been validated using the same dataset described in 6.3.2.

6.4.1 Dynamic sensor selection

Multiple sensors allow for network-level power-performance management. A dynamic sensor selection (DSS) scheme is used to modulate the number of sensors contributing to activity recognition in order to manage power-performance tradeoffs at run-time according to the application needs.

The DSS procedure works continuously. Thus, when the system is at risk of not meeting its performance goal (e.g. when an active sensor runs out of energy or fails), the dynamic sensor selection scheme reconfigures the network to restore the desired performance. This contributes to improved robustness against faults and longer network lifetime, and also reduces maintenance burden as there is no need to immediately replace defective parts of the system as long as enough resources are available.

At any given instant t , the status of the nodes of the network (s_i) can be one of the following:

Power Off The node is not active but waiting to participate to the gesture recognition.

Active classification The node is active and classifies gestures.

Active fusion The node is active and fuses the decisions of individual node classifiers.

Faulty The node is either faulty or out of energy.

The nodes that are active at a certain time t form the *Active Cluster*, $C(t)$. The active cluster is dynamically adapted by the DSS scheme according to the status of the network.

Whenever one active node stops operating the DSS algorithm is executed to adjust the Active Cluster. The resulting cluster must fulfill the following requirements:

1. achieve a minimum application defined performance in presence of m faulty nodes;
2. use the smallest possible number of nodes.

Since the DSS algorithm *starts when one of the nodes fails*, m defines how many faults can be tolerated between the time the first sensor fails and the time the DSS algorithm produces the replacement cluster. A higher value m results in a larger number of active nodes, but provides a higher fault tolerance.

Figure 6.15 illustrates the DSS algorithm. Formally, the algorithm looks for a new cluster $C(t+1)$ as a function of the cluster at time t ($C(t)$), the minimum accuracy fixed by the application (A), the number of failures that can be tolerated (m), the status of the nodes of the network ($\{s_i\}$), and a set of gesture instances used to evaluate the cluster performance (I).

$$C(t+1) = f(C(t), A, m, \{s_i\}, I) \quad (6.4)$$

The DSS algorithm follow the following steps:

1. evaluate if the remaining remaining active nodes $c_i = C(t)$ with dimension $d_i = D(t)$, where $i = 0$ fulfill the requirement above (all subclusters Sc_k with dimension $d_{Sc_k} = d_0 - m$ present a correct classification ratio equal or greater than A on a reference set of gesture instance I). If so, this cluster is the new Active Cluster ($C(t+1) = C(t) = c_0$, i.e. no nodes need to be turned on).
2. If not, select a new node to add to the active cluster and form the new cluster c_{i+1} with dimension $d_{i+1} = d_i + 1$.
3. evaluate if cluster c_{i+1} fulfill the requirement above. If so, this cluster is the new Active Cluster $C(t+1) = c_i$, otherwise repeat the previous step until either one suitable cluster have been found or all nodes are used.

Three heuristics are used to select which node is added to form the new cluster c_{out} :

Best Build all possible clusters of dimension $d_i + 1$ by adding one Power Off node to c_i and test them. Return the cluster that shows the best accuracy.

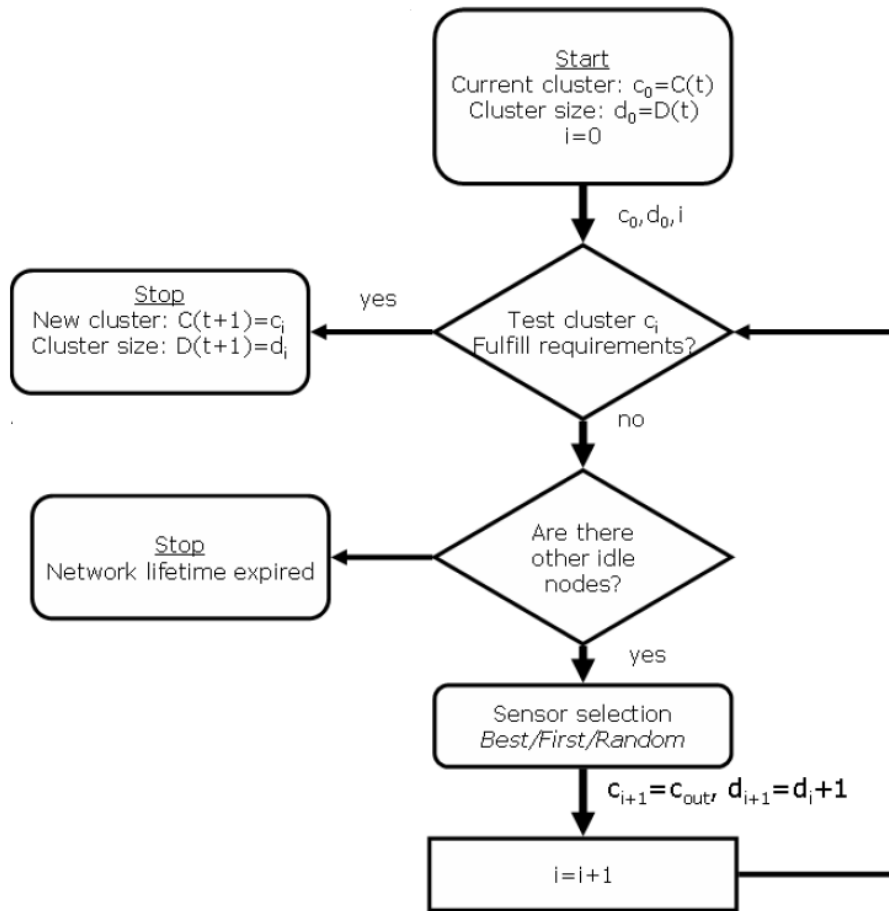


Figure 6.15: The dynamic sensor selection algorithm starts by evaluating the existing cluster of nodes. If it does not fulfill the performance requirement, it looks for a new cluster of nodes by iteratively evaluating clusters of increasing size until the performance requirements are fulfilled.

First Build a cluster of dimension $d_i + 1$ by adding one Power Off to c_i and test it. If it fulfills the requirements return this cluster, otherwise test another cluster of dimension $d_i + 1$ built by adding another Power Off node and repeat until a suitable cluster is found. If no suitable clusters are found then return the one with the best accuracy.

Random Returns a cluster of dimension $d_i + 1$ by adding one random Power Off node to c_i .

The performance of the active cluster is defined by its accuracy at classifying a set of representative reference gesture instances (I , in equation 6.4).

The outcome of the node level classification is fixed given a trained classifier (here an HMM model) and a sequence of observation. Thus, the result of the classification of the instances in set I by sensor i can be pre-computed (i.e.

c_{out}^i or \tilde{c}_{out}^i in section 6.3.1). The DSS algorithm thus only implements the sensor fusion step on the basis of these pre-computed node classification results. Concretely, each node only needs to store the result of the classification of the sequences in the reference set I , and does not need the whole acceleration sequences. This reduces the amount of memory required and the complexity of the evaluation of the cluster performance. In fact, the memory required is $N \times N_{ref}$ (where N_{ref} is the number of reference instances). Moreover, the complexity is related only to the fusion step of the algorithm, which is $O(N^2)$ for the Best and First schemes (this latter considered in the worst case), where N is the number of sensors, and $O(N)$ for the Random scheme.

This approach enables a generic modeling of the performance of the system for any gesture. Given a representative set of reference gestures it enables an accurate prediction of the performance of a set of nodes, since the system is directly used as its own model. This approach is only limited by how representative of the problem at hand the instances in the set I are.

6.4.2 Accuracy of cluster performance estimation

The performance of one cluster is predicted on the basis of a set of reference instances (see equation 6.4). This approach assumes that the performance of the system can be generalized from a limited number of reference instances.

We assess the accuracy of this approach by selecting a set of 20 instances as reference set and the remaining 50 as validation set. On the basis of the reference and the validation instances we determine the performance of clusters of increasing size composed of randomly picked nodes. This is repeated 150 times using three different reference sets (50 trials for each size with the same reference set). The probability density function (PDF) of the evaluation error is built using the Parzen window method with Gaussian windows. The PDF in case of naïve Bayes fusion and single axis sequences is presented in figure 6.16. From this figure we can see that the performance prediction is more accurate for larger clusters. For the larger cluster the absolute value of the error is smaller than $\pm 3\%$, while with smaller cluster it can increase to about $\pm 10\%$.

The effect of the number of reference instances on the performance prediction accuracy is assessed in the same way by varying the number of instances in the reference set. Increasing the reference set initially increases performance prediction up to a plateau (see figure 6.16). This behavior stems from the fact that a small set of reference instances is less likely to be representative of the activity classes.

These results illustrate the accuracy of the performance prediction method described in section 6.4.1. It does not guarantee that the effective performance

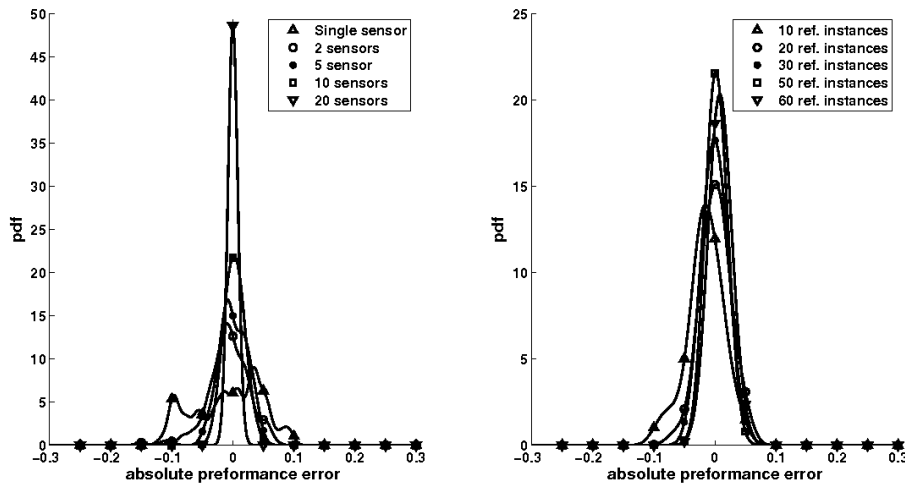


Figure 6.16: Probability density function of the performance prediction error as a function of: the number of nodes in the cluster(a), the reference set size (b)

of the system is systematically higher than the predicted performance. In an application the performance prediction (and thus indirectly gesture and user variability) must be characterized beforehand according to the method described here. On this basis the appropriate reference set size can be selected according to the desired likelihood that the system effectively meets - or is above - the estimated performance.

Another approach is to add a margin to the minimum classification accuracy requested by the application. This margin is related to the number of active nodes, size of the reference set, and gesture variability (e.g. for bigger clusters or more reference instances a smaller margin is selected).

By placing a fault tolerance (in our tests $m = 1$, that is all sub cluster of the active node set of dimension $D - 1$ should achieve the minimum performance required by the application) we can assume that the probability of violating the minimum accuracy is higher after a fault occurs. However at this point the DSS algorithm starts and, in a short time, returns the new active cluster. Thus we can define two level of violation: *soft violation*, *critical violation*. The former indicates when a sub cluster violates the application constraints, the latter when the cluster out of the DSS algorithm does it. According to the application, a soft violation can be tolerated while a critical one not.

6.4.3 Power-performance tradeoff characterization

We compare the system lifetime and performance obtained when the DSS algorithm selects the active nodes to the reference case where all sensors are simultaneously active. The system lifetime is defined as the time point from the

start of operation when the system breaks the classification accuracy threshold defined by the application. In this section we assume to have perfect cluster performance estimation. Thus we do not consider soft or critical violations due to incorrect cluster performance estimation.

We perform this characterization using the three fusing method (majority voting, naïve Bayes or Borda count), different minimum required performances (80%, 85% or 90%) and different sensor selection method (random, first, best or no sensor selection i.e. all sensor active at the same time). During each simulation the sensors are selected in a "node wise" manner, i.e. when using single axis sequences we activate all three axes from the same accelerometer simultaneously. Moreover all three axes fail at the same time. The motivation is that when we activate a sensor node we turn on all three axes of its accelerometer and when the energy of the node is depleted all three acceleration axes are lost simultaneously. In each case, 50 trials are performed and the results that are presented are averages.

In figure 6.17 we show the average network lifetime as a function of the minimum performance target, for different sensor selection criteria and input features. As expected, when all the nodes are used at the same time the average network lifetime is mostly independent from the performance target. In that case, all the nodes are likely to run out of energy at about the same time μ and the performance drops (given the lifetime model variance).

With the DSS algorithm, however, the system lifetime can be significantly extended by better managing the number of nodes participating in activity recognition. In this case, there is a strong dependency between the performance target and the network lifetime. With a higher performance target more nodes need to contribute to activity recognition, therefore depleting the available energy sooner than with a lower performance target.

The sensor selection criterion has limited effect on the power-performance trade-off. The random selection criterion is slightly less effective than the *first* and *best* selection criteria. However, the random selection criterion requires less computational effort since the number of cluster to evaluate is smaller. This is discussed in more details later in this section.

Borda count fusion typically results in the longest average network lifetime, while majority voting fusion in the shortest lifetime (see figure 6.18). This is linked to the performance after sensor fusion (see section 6.3.3): if a fusion method provides higher performance with a smaller number of nodes, then a larger number of sensors can be left into *idle* state for later use.

The use of single accelerometer axis input sequences results in longer network lifetime in comparison to magnitude sequences (see figure 6.19). The reasons are the same as the one presented for figure 6.18: higher performance

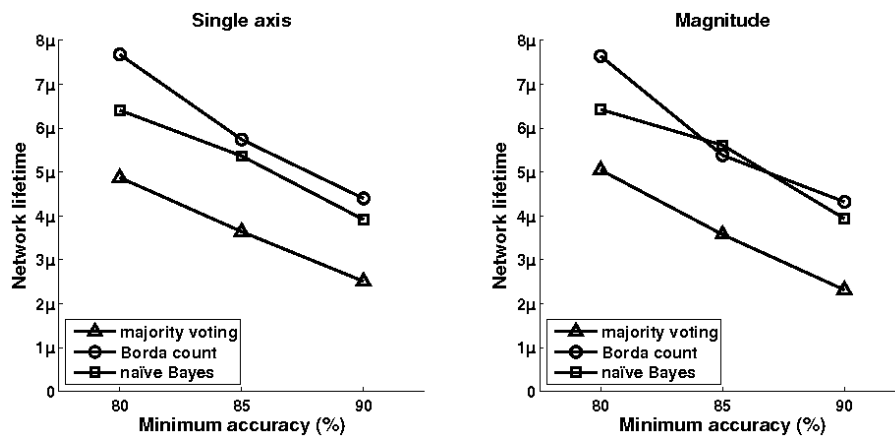


Figure 6.17: Average network lifetime in multiple of μ as a function of the target performance with Borda count as fusion method. The DSS sensor selection algorithm *best*, *first* and *random* are compared with the reference case where *all* the nodes are active together (no DSS algorithm). The DSS algorithm enables longer network lifetime thanks to the better management of available resources.

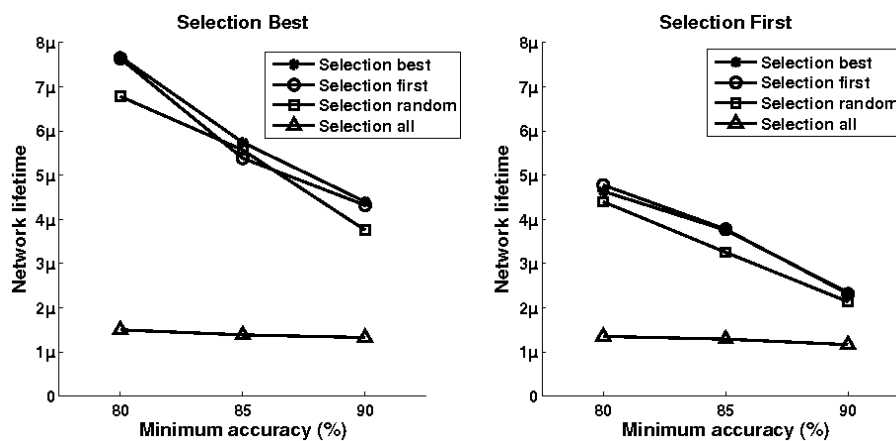


Figure 6.18: Average network lifetime in multiple of μ as a function of the target performance with single accelerometer axis features. The majority voting, Borda count and naïve Bayes fusion methods are compared. The use of the Borda count method results in longer network lifetime in comparison to the other fusion methods.

results in smaller active cluster thus a larger number of nodes can be used to replace active ones.

The performance evolution with the DSS algorithm selecting the sensors that participate in the gesture classification is compared to the reference case where all the nodes are used simultaneously (see figure 6.20). In all three examples the Borda count fusion method, single accelerometer axis sequences and the *first* sensor selection criteria are used.

When all the nodes are active simultaneously, the starting performance is

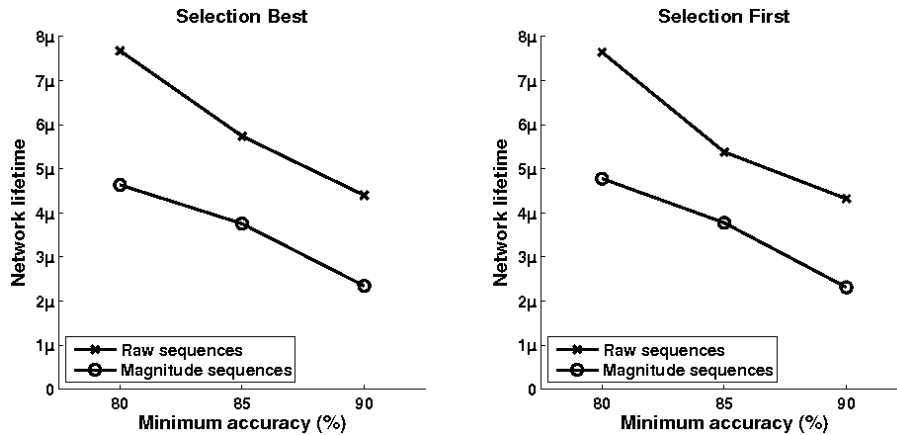


Figure 6.19: Average network lifetime in multiple of μ as a function of target performance for single acceleration axis and acceleration magnitude features. Borda count fusion method is used. The use of single accelerometer axis features results in longer lifetime.

| Accuracy (μ) | Majority voting | | | Borda count | | | Naïve Bayes | | |
|--------------------|-----------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|
| | <i>Best</i> | <i>First</i> | <i>Rand</i> | <i>Best</i> | <i>First</i> | <i>Rand</i> | <i>Best</i> | <i>First</i> | <i>Rand</i> |
| 80% | 4.8747 | 5.0465 | 4.6045 | 7.679 | 7.6425 | 6.7808 | 6.4091 | 6.4252 | 6.0012 |
| 85% | 3.6406 | 3.5753 | 3.4497 | 5.7394 | 5.3818 | 5.5703 | 5.3649 | 5.6105 | 4.7944 |
| 90% | 2.5085 | 2.3142 | 2.2006 | 4.3989 | 4.3204 | 3.7569 | 3.9144 | 3.9357 | 3.4354 |

Table 6.4: Average network lifetime (single accelerometer axis features).

higher than when the DSS algorithm is used. However as time approaches the average node life μ the nodes start to run out of energy and the performance quickly drops below the target. When the DSS algorithm is used, the system performance at start is usually lower than in the previous case yet still above target. The sharp changes in performance result from node failure and their replacement. Since the cluster performance estimator is assumed to be perfect the performance never drops below the threshold until the system reaches end of life.

In figure 6.21 we illustrate the sequence of activation of the nodes for the example presented in figure 6.20. Dark spots indicate when a node is participating in the gesture recognition. The figure illustrates the fact that with a higher target performance the number of nodes that are active at the same time tends to be higher. This results in a higher system power consumption and smaller system lifetime.

Tables 6.4 and 6.5 summarize the results.

The computational cost of the DSS algorithm is a function of the sensor selection criteria and of the fusion method. The computational cost of the fusion method is presented in table 6.3. Thus, only the effect of the sensor selection

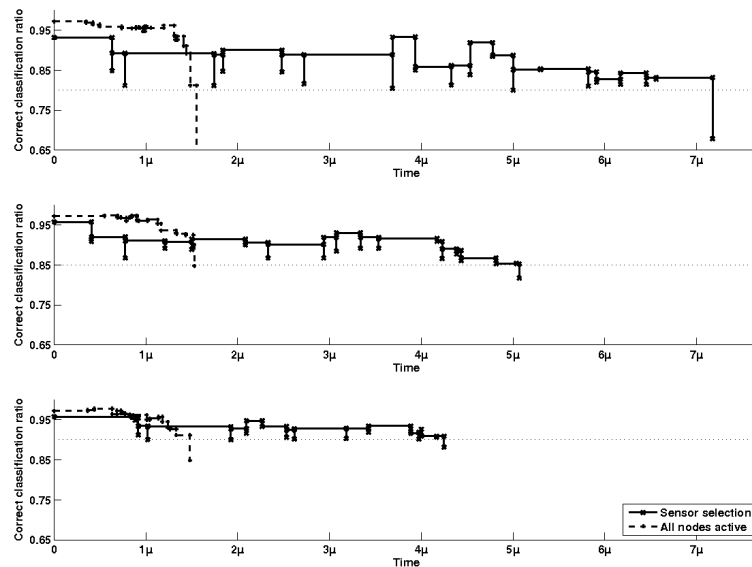


Figure 6.20: Evolution of the classification accuracy over time, as a function of the target performance. The Borda count fusion method and single accelerometer axis features with *first* sensor selection criteria are used. The continuous line indicates the performance with the DSS algorithm selecting the nodes participating to activity recognition. The dashed line indicates the performance of the system where all the nodes are active simultaneously. Vertical segments indicate when a sensor is turned off and replaced. The dotted line indicates the target performance. When all the nodes are simultaneously used, the performance of the system quickly drops under the target threshold once time approaches the average node lifetime. With the DSS algorithm a better management of the active nodes allows to lengthen the system operation time.

| Accuracy (μ) | Majority voting | | | Borda count | | | Naïve Bayes | | |
|-----------------------|-----------------|--------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|
| | <i>Best</i> | <i>First</i> | <i>Rand</i> | <i>Best</i> | <i>First</i> | <i>Rand</i> | <i>Best</i> | <i>First</i> | <i>Rand</i> |
| 80% | 3.4941 | 3.6135 | 3.4228 | 4.6341 | 4.7779 | 4.3987 | 3.4870 | 3.4236 | 3.2841 |
| 85% | 2.586 | 2.5454 | 2.3231 | 3.7537 | 3.7765 | 3.2538 | 2.4037 | 2.5310 | 2.1175 |
| 90% | 1.5459 | 1.5425 | 1.2213 | 2.3421 | 2.3108 | 2.1357 | 1.1019 | 1.1044 | 1.0244 |

Table 6.5: Average network lifetime (acceleration magnitude features).

criteria is discussed here. We evaluate the complexity of the DSS algorithm by the number of clusters that need to be evaluated until a cluster satisfying the performance target is found.

Larger clusters require more effort to be evaluated (see table 6.6). However, table 6.3 indicates that the computational cost to classify an instance using any fusion method is linear with the number of sensors fused. Thus, the computational cost to evaluate a cluster of N nodes will be N times the one to evaluate

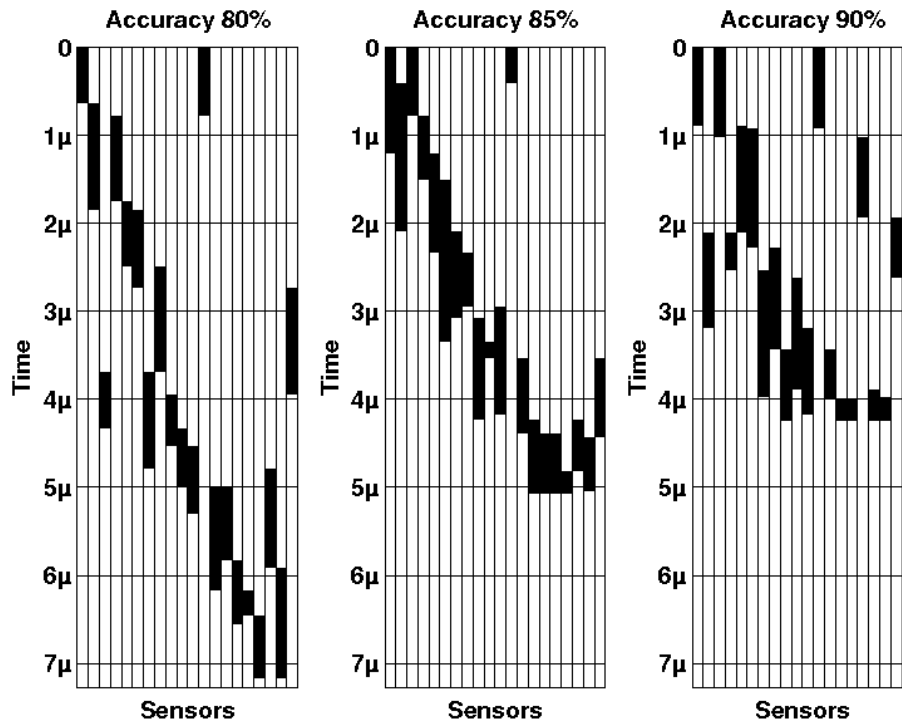


Figure 6.21: Node activation sequence corresponding to figure 6.20. The vertical axes represents the time (increasing toward the bottom). Each column corresponds to a node. Dark spots indicate when a node is active. Whenever a node is turned off it is replaced by one or more idle nodes until all nodes of the network have been used.

a single node, for a given fusion method and for a fixed number of reference instances.

| | (a) Single accelerometer axis sequences | | | (b) Magnitude sequences | | | |
|----------|---|--------------|---------------|-------------------------|-------------|--------------|---------------|
| | <i>Best</i> | <i>First</i> | <i>Random</i> | | <i>Best</i> | <i>First</i> | <i>Random</i> |
| Majority | 3105 | 1487 | 854 | Majority | 5826 | 3716 | 1568 |
| Borda | 1786 | 675 | 389 | Borda | 3664 | 1952 | 923 |
| Bayes | 1949 | 697 | 485 | Bayes | 5622 | 3853 | 1737 |

Table 6.6: Dynamic sensor selection algorithm computational effort when 85% minimum classification accuracy is needed. The values in the table represent the average number of equivalent clusters of size 1 that are evaluated during the network evolution.

The processing cost of the *best* sensor selection criteria is almost twice that of the *first* selection criteria. Yet, tables 6.4 and 6.5 indicate that the *best* and *first* sensor selection criteria result in similar network lifetime. The processing cost of the *random* criteria is half that of the *first* criteria, while resulting only in slightly reduced the network lifetime.

The reason why the more exhaustive search within the space of the possible

cluster does not produce noticeable improvements may come from the fact that all three methods perform an optimization at a single time point, i.e. given the actual cluster and actual state of the network they try to find the next best one. A better approach may consider the optimization of the *sequence* of active nodes as a whole, such as to keep the performance of the system above the desired threshold while reducing the number of active nodes at the same time.

6.4.4 Implementation using Tiny TAsk Networks (Titan)

The algorithm described above needs to be mapped on a wireless sensor network. The *Titan* framework that we are developing for context recognition in heterogeneous and dynamic wireless sensor networks can be used for this purpose [126]. We develop Titan as part of the ongoing e-SENSE project as a tool to enable and explore how context awareness can emerge in a dynamic sensor network. Titan simplifies the algorithm description, automates data exchange between selected sensor nodes, and adapts execution to dynamic network topologies. It thus qualifies for the implementation of the algorithm presented before.

Most context recognition algorithms can be described as a data flow from sensors, where data is collected, followed by feature extraction and a classification algorithm, which produces the context information. Within Titan, context recognition systems are represented as *Task Graphs*. It offers for each processing step (sampling, feature extraction, and classification) a set of predefined tasks. A task is usually a simple signal processing function, such as a filter, but may also be a more complex algorithm such as a classifier. A context recognition algorithm can be composed from those modular building blocks, which are provided by the nodes participating in the network.

A set of tasks are programmed into the sensor network nodes as a *Task Pool*. These tasks are instantiated when they are needed (i.e. they use RAM and CPU cycles only when they are used by a *Task Graph*). In a heterogeneous network, node processing power may vary, and nodes with higher processing power can provide more complex *Task Pools* than simpler nodes.

Figure 6.22 shows the Titan architecture and illustrates how a classification task graph is distributed on the sensor network; the *Task Graph Database* contains the classification algorithm description containing sensor tasks S_i , feature tasks F_i , a classification task C , and an actuator A_1 receiving the end result. Upon request to execute the algorithm, the *Network Manager* inspects the currently available nodes in the network, and decides on which node to instantiate what tasks, such as to minimize processing load, overall power consumption, or maximise network lifetime. The Network Manager then sends a configura-

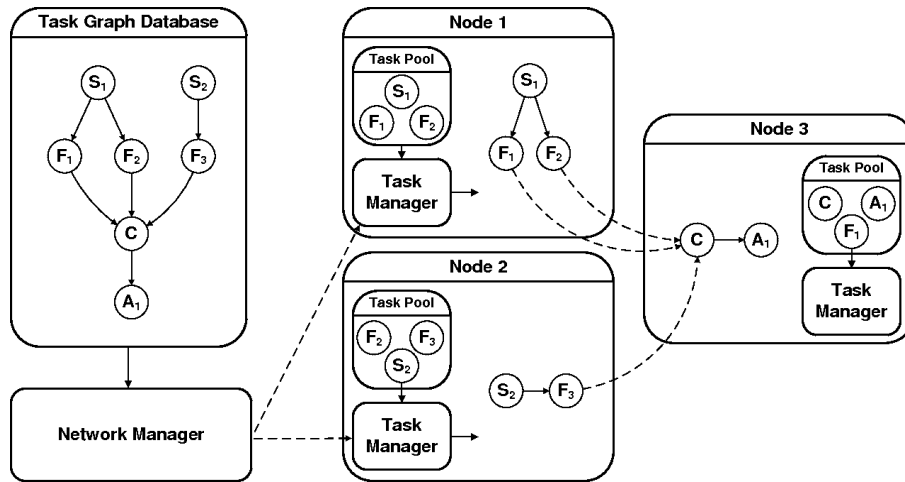


Figure 6.22: Titan configures an application task graph by assigning parts of the graph to participating sensor nodes depending on their processing capabilities

tion message to the *Task Managers* on the sensor nodes, which instantiate the tasks on the local node. The *Task Manager* assigns a share of dynamic memory to the tasks for their state information and configures the connections between tasks, including transmitting data to other nodes.

During execution of the task graph, the *Network Manager* receives error messages from tasks or sensor nodes, and checks whether all participating sensor nodes are still alive. If changes to the current configuration are required, it adapts the distribution of the task graph on the network.

Titan provides several advantages. Ease of use, since a designer can describe his context recognition algorithm simply by interconnecting different tasks and selecting a few configuration parameters for those tasks. Portability, because it is based on TinyOS [85] which has been ported to a range of sensor network hardware and due to the abstraction of tasks, it is able to run on heterogeneous networks. Flexibility and speed, since it can reconfigure nodes in less than 1ms in order to quickly react to changes in dynamic sensor networks.

The meta classifier with dynamic sensor selection can be incorporated into Titan by dividing it into a set of tasks that can be instantiated on different nodes. In particular, we define three new tasks: 1) a "gesture classification" task, which implements the HMM algorithm, 2) a "meta classification" task that performs Bayesian inference and decides the gesture class, 3) a "dynamic sensor selection" task that defines the set of sensors contributing to the meta classification task.

The initial cluster of nodes is created by the dynamic sensor selection task. The *Network Manager* instantiates on each of the nodes within this cluster the gesture classification task. The system runs as-is until a node fails (i.e. runs

out of power). When the meta classification task senses that a node fails to send data it sends an error message to the Network Manager. The Network Manager instantiates the dynamic sensor selection task on a device with sufficient computational power (PDA, mobile phone), and then adapts the configuration of the nodes as needed. Since the cluster can tolerate the failure of any one of its nodes and guarantee the desired classification performance, the system can work continuously even when the dynamic sensor selection task is running. This relaxes the time constraint on this task and allows relatively complex clustering algorithms for the dynamic sensor selection task.

The task of the Network Manager for running the presented distributed gesture recognition algorithm is light-weight. To remember the current configuration of the participating nodes, it has to store just 1 byte for the node ID, 1 byte for their status (active, failed, not used, meta classifier), and a single byte for the current cluster size. This amounts to 39 bytes of storage for running the gesture recognition algorithm on our example of 20 nodes. The processing time is limited as well, as it just has to generate a small number of configuration messages at every update of the network. We are thus confident that the algorithm presented here is able to run on sensor network nodes, with the exception of the non-optimized dynamic sensor selection task which runs on a PDA or mobile phone.

6.4.5 Discussion

Several power-performance management approaches were presented in section 6.2. Most of the ones specific to human activity recognition in wearable system and ambient intelligence environments tend to reduce the set of used sensors [149] or adjust processing parameters to achieve energy savings.

In the field of (wireless) sensor networks, energy considerations led to optimized routing protocol, radio transceivers, and communication protocols, or data reduction techniques. These approaches are however disconnected from a particular application goal, such as activity recognition accuracy. The characteristic of this work is to manage power aspects at high level by taking into account the application performance requirements. This complements approaches operating on signal processing parameter adjustment. This work also benefits from and complements energy saving techniques typical of (wireless) sensor network technologies by reflecting the application performance target at the networking level.

The performance evaluation method proposed here is computationally efficient since only the fusion of the decision of the nodes belonging to the cluster needs to be computed at runtime, while the classification by individual nodes

of the set of stored reference gesture instances is pre-computed during training.

It is also generic and applicable to any gesture set, and we showed how to assess and control its accuracy. Given a more constrained target application, an alternative approach may be to build a model linking system performance to the nodes contributing to activity recognition, as well as the parameters of these nodes (e.g. sampling rate). Such a model may be built from analysis carried out during system training (e.g. the confusion matrix of a sensor, parameter-performance linkage) [16, 189]. However, a model-based approach needs to be devised for a specific application (gesture set) and decision fusion algorithm. Although this may lead to more accurate performance prediction or lower computational costs, it is at the expense of a loss of generality.

The DSS algorithm may be further enhanced by using class-specific performance models instead of a global performance model as now as proposed in [149]. This may allow to select the sensors more appropriate for specific activity classes. It also allows to provide individual performance target for each activity class. This provides more flexibility to define the performance target and may translate in further energy savings. For instance wearable sensing may be combined with ambient sensing. Presence sensors or smart cameras in the environment may detect the location of the user. According to the user location, the dynamic sensor selection algorithm may turn off the body-worn sensors when the user is far from the car. This behavior may be obtained with the appropriate system performance model without fundamental changes in the DSS algorithm.

Systems relying on energy scavenging may also benefit from the approach presented here. In fact, the constraints on the amount of energy to harvest are more relaxed since nodes can be replaced with one or more others while replenishing their energy buffer. A generalized performance model may allow the DSS algorithm to select appropriate sensors according to the performance target and available energy, and potentially also the likelihood of specific activity classes.

To improve the analysis of the network evolution, the proposed nodes' energy model can be improved. We assumed a Gaussian node lifetime model and no power use for inactive nodes. In a real scenario the nodes use energy even in sleep modes. Periodic wake-up is required to synchronize with the other nodes of the network and, if required, change the node state. A more realistic energy model may capture these characteristics. It may also include system reactivity considerations (i.e. how often nodes turn on to assess whether they should start contributing to activity recognition). This however does not affect our main result, which is to underscore the benefits of runtime power-performance management through dynamic sensor selection.

6.4.6 Conclusion

Wearable computing seeks to empower users by providing them context-aware support. Context is determined from miniature sensors integrated into garments or accessories. In a general setting the sensor network characteristics may change in unpredictable ways due to sensor degradation, interconnection failures, and jitter in the sensor placement. The use of a dense mesh of sensors distributed on the body may allow to overcome these challenges through sensor fusion techniques. Since such systems must remain unobtrusive, the reduction of node dimension and node interconnection is of high importance. Wireless sensor networks help achieving this unobtrusiveness since they do not require any wire connection. However, this implies that each sensor node must be self powered. In order to reduce obtrusiveness, the battery dimension must be kept at minimum, which results in low power availability.

Energy aware design aims to extend sensor nodes life by using low power devices and power aware applications. Power aware applications typically rely on duty cycling: they reduce the amount of time when the radio is active, and they increase the amount of time when the node can be placed in a low power state. In wearable computing, unpredictable duty cycles are proscribed. We described a different approach to extend network life while achieving desired accuracy. We capitalized on the availability of large number of nodes to implement a dynamic sensor selection scheme together with a metaclassifier that performs sensor fusion and activity recognition. This technique copes with dynamically changing number of sensor without need to retrain the system.

The method minimizes the number of nodes necessary to achieve a given classification ratio. Active nodes recognize locally gestures with hidden Markov models. The output of active nodes is fused by a naive Bayes metaclassifier. Inactive nodes are kept in a low power state. Once an active node fails the system activates one or more additional nodes to recover the initial performance.

The effectiveness of the DSS algorithm has been evaluated. Assuming a Gaussian node lifetime model $N(\mu, \sigma)$ reflecting a generic battery size and power consumption (this model can be tailored to specific hardware) we reported the network lifetime as a multiple of the node lifetime μ . The average network lifetime when all the nodes are simultaneously active is slightly more than μ . We compared three cluster search heuristics. We showed that exhaustive cluster evaluation returning the best performing cluster brings the same benefits than a simpler algorithm returning the first cluster found that satisfies the performance goal. For instance, in both cases with Borda count fusion method and a target of 90% classification accuracy, the system lifetime can be increased respectively to 4.40μ and 4.32μ . Even with a primitive heuristic (here one that adds randomly selected nodes to the cluster until a suitable cluster is

found) significant improvements in network operation time can be reached (3.76μ with the same conditions as above).

When accounting for the computational complexity of the DSS algorithm the heuristic selecting the first cluster satisfying the performance goal shows a good trade off between computational cost and network lifetime.

We described how this method fits within the Titan framework that we develop to support context-aware applications in dynamic and heterogeneous sensor networks. Titan allows fast network configuration and is well suited for our technique as it allows to easily exploit network resources dynamically.

6.5 Tangerine SMCube: a smart device for human computer interaction

With the objective of enhancing the interaction with smart environment, smart objects can be used as tangible interfaces and play a fundamental role in improving human experience within interactive spaces for entertainment and education (see section 6.2.2).

In this section we present the Tangerine Smart Micrel Cube (SMCube) a smart device for Human Computer Interaction (HCI). The SMCube is a tangible smart object equipped with sensors (digital tri-axes accelerometer) and actuators (infrared LEDs, vibro-motors) embedded in a cube. The SMCube is a tangible interface developed for the *TANGerINE* framework, a tangible tabletop environment where users manipulate smart objects in order to perform actions on the contents of a *digital media table* [24].

Data from accelerometer is used to locally detect the active face (the one directed upward) and a set of gesture performed by the user. These information are wirelessly sent to a base station for processing. Furthermore, through the LEDs the node can interact with a vision based system in a multi modal activity detection scenario.

6.5.1 The TANGerINE framework

In the *digital media table* scenario, smart object based tangible interface (tangibles) have to be considered both in relation to the shape of the surface around which users stand and the digital contents visualized on it. From the physical point of view, the suitable tangibles that can be easily used for the interaction on a tabletop are those that assume a stable steady state when left on a horizontal plane.

The tabletop scenario is characterized by different context according to the area where the interaction occurs (see figure 6.23):

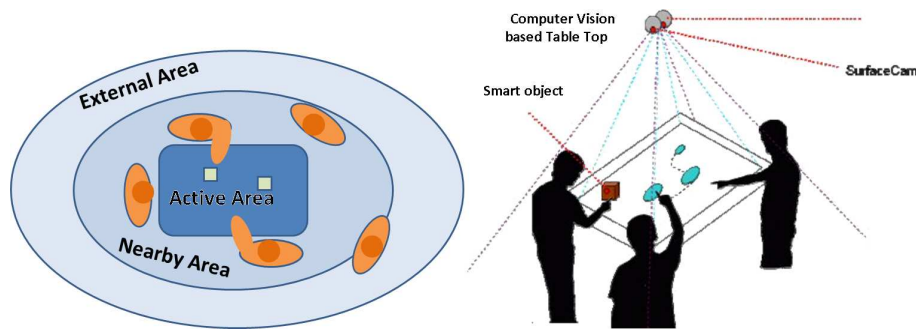


Figure 6.23: Setup for the Tangerine framework and identification of the three contexts.

- *Active Context (AC)*: it is the horizontal visualization surface, typically the scene where users interact with tangibles (recognized by the system) as well as digital elements. In this area there is a direct mapping between the position and orientation of tangible objects and the digital ones.
- *Nearby Context (NC)*: it is the area right around the tabletop where both intentional and non intentional actions can be performed. The body of the user can be tracked and this information can be used to study his behavior. The position of the user can be useful also for attributing the ownership of actions performed in the AC. In this context tangibles position could not be precisely tracked, but can still be manipulated and provide information about their orientation in space and some action performed.
- *External Context (EC)*: it is the outer area, unrelated with the first two contexts. In this area no position tracking occurs, but the user can still interact with the tangible object and carry it with him across different tabletops. The object therefore becomes a bridge between different interactive artifacts. The user could perform some actions on a tabletop and use the same tangible on other artifacts, in this case the physical object can become a container of different kind of information (e.g. session data or user profile).

The current TANGerINE system layout consists of a ceiling mounted case that embeds all of the required elements: computer, projector, camera and illuminator, targeting the horizontal surface of a normal table that is positioned under the case, where also the interface is visualized [9].

Users interact with the system manipulating a physical object. We chose a cube shape for the availability of six steady states, as well as its clear affordance.

- The user intuitively considers the uppermost face “active”, as if reading

the face of a die, and therefore identifies the object as being able to embody six different actions or roles.

- The cube faces can be used as visualization areas for symbols related to the current object role (augmentation) and also to provide the space for markers needed to object detection and tracking..

In the active context the relation between the cube steady state and the “active face” is the most important: users can place the object on the surface and move it sliding over the table while keeping the same upper face, or grab it and rotate it to choose another face. The cube can also be taken by the user and manipulated out of the active area, in the nearby context. In this case the manipulation has more degrees of freedom. The variety of these actions allows for a more expressive interaction language and provides to the application designer an environment with richer modes of operation that depend on the context in which the user acts. In the latter situation the possibility by the cube to recognize gesture performed by the user can greatly enhance the expressiveness of the interaction.

6.5.2 SMCube overview

The SMCube is a smart object equipped with sensors (a digital tri-axes accelerometer from STM, LIS3LV02DQ, and 6 photo transistors) and actuators (infrared LEDs and vibro motors) (see figure 6.24). It embeds an ATmega 168 low-power, low-cost microcontroller to sample and process data from its sensors, and a Bluetooth 2.0 transceiver from BlueGiga (WT12) to wirelessly communicate with a PC. The ATmega 168 features a RISC architecture that can operate up to 24MHz and offers 16 KB of Flash memory, 1 KB of RAM and 512 Bytes of EEPROM. The microcontroller includes a multiplier and several peripherals (ADC, timers, SPI and UART serial interfaces etc.). The firmware has been implemented in C using the Atmel AVR Studio 4 IDE that provides all the APIs necessary to exploit the peripherals and perform operations with 8, 16, and 32 bit variables.

Originally, the SMCube was used as a tilt-aware artifact, with intelligence on board to perform sensing, actuation, storage and processing of data. The cube is identified by an id number, which helps disambiguating when more than one cube is present at a time. Thanks to its wireless communication capabilities it can receive queries and controls and exchange bidirectional information with the environment in which is placed. Therefore, the use of the SMCube in a multi sensory enhanced context enables the use of redundancy to improve recognition abilities of the overall system. The cube can provide both

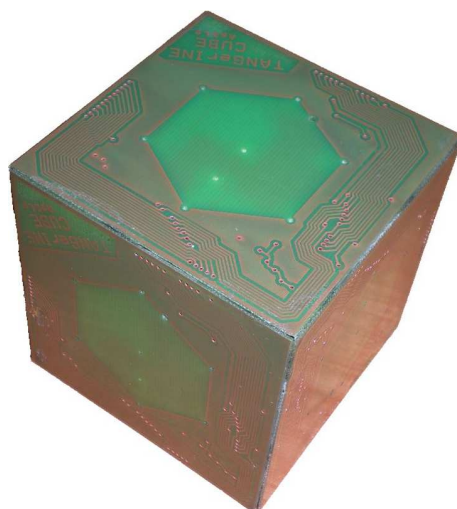


Figure 6.24: The Tangerine SMCube and its use within the tabletop environment

direct feedback on its state via wireless communication and visual, or tactile feedback by use of its actuation capabilities.

The tri-axial accelerometer embedded in the cube measures static and dynamic acceleration. The former is used to extract the tilt of the cube with respect to gravity acceleration vector by use of simple trigonometric consideration on the acceleration collected along the three axes. Therefore, the SMCube is able to derive which of the six faces is the top or the bottom face at a certain instant. The result is both stored on the cube and translated in visual feedback, i.e. the led matrix on the top face is turned on for the CV subsystem to track and identify the object. In a similar way, the tilt can be translated in vibration.

The cube tilt, and more generally all the information stored in it (e.g. its id number, other attributes relating to its state, etc.) can be sent wirelessly to any device enabled with Bluetooth communication capabilities. At present the packet effectively sent contains different fields, in particular the raw accelerometers data, the code corresponding to the cube face currently lighted, the *id* of the cube. The cube provides two operational modalities: *Inquiry mode* or *Continuous mode*

Inquiry mode When a transition from a face to another is detected, a single transmission is carried out. We have introduced a configurable latency (or reaction time), consisting of the number of data frames after which the cube is considered halted at a certain tilt. This functionality is added to hide transitional states. In this mode, the cube is responsive to inquiry commands, e.g. requests of further transmissions of the packet containing the state of the cube.

Continuous mode The data-packet is sent continuously at a configurable frame-rate. The overall system is then informed frame by frame of the actual cube state and can monitor its movements. This mode enables, for example, signal processing for gesture recognition and cube motion tracking on the fixed, and more powerful, PC.

Furthermore, in both operational modes, additional inquiry commands can be performed, as turning off and on the LEDs, changing the frame rate, switching between modes or modifying the configurable latency for face detection.

6.5.3 Gesture detection algorithm

We augmented the intelligence of the SMcube implementing an on board gesture recognition algorithm based on a decision tree. The algorithm is able to distinguish between three gestures performed by the user: cube *placed on the table*, cube *held*, cube *shake* and *tap*. The latter one, in particular, happens when the cube is placed on the table and the user gently hits the upper (active) face. The gestures recognized provoke reactions defined by the system or application the cube is interacting with.

Data from the three axes accelerometer is sampled at 40Hz. To detect the first three gestures we classify partially overlapped windows of 16 consecutive samples. The variance of the data within each window is used to classify the gesture performed using a C4.5 decision tree classifier [167] previously trained with the WEKA [230] toolkit using a set of pre-stored instances. The choice of this algorithm is motivated by its easiness of implementation and the limited amount of resources needed.

A valid gesture is detected when the classifier returns the same class for N ($N = 3$) consecutive windows. N defines a trade-off between robustness of classification and reactivity.

To detect the *tap* event we implemented an ad-hoc technique. This technique is based on a Finite State Machine (FSM) that checks if the actual status is *placed on the table* and then analyzes the accelerometer output waveform. Figure 6.25 shows the accelerometer output when a *tap* event happens.

From this picture it is easy to understand that a *tap* can be recognized when the waveform presents a rapid spike then the signal returns quickly to its quiet state.

Figure 6.26 presents a diagram of the application implemented on the SM-Cube. As can be seen 2 tasks operate in parallel. The first implements the C4.5 tree classifier (see figure 6.27(a)), the second the FSM for the *tap* recognition (see figure 6.27(b)).

Preliminary tests showed that the performance (calculated as correct classi-

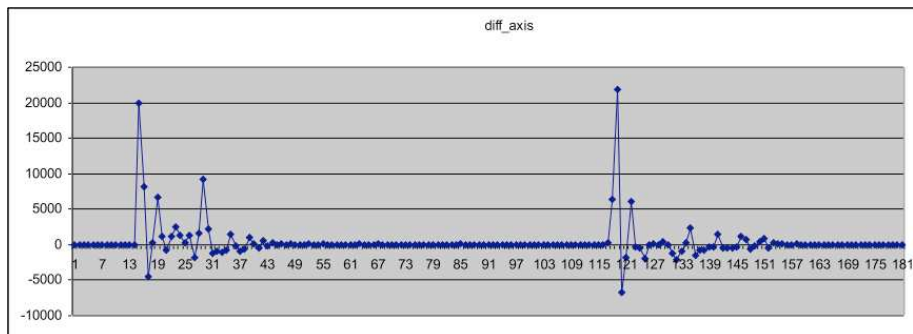


Figure 6.25: Accelerometer output waveform when a *tap* event happens.

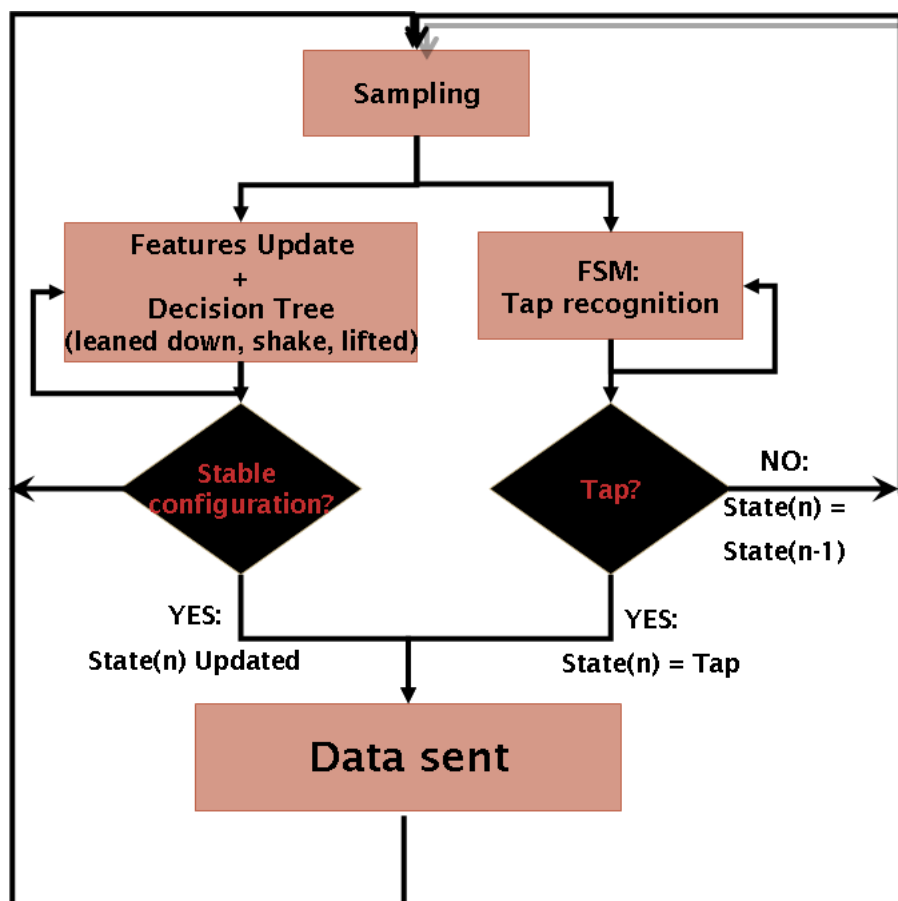
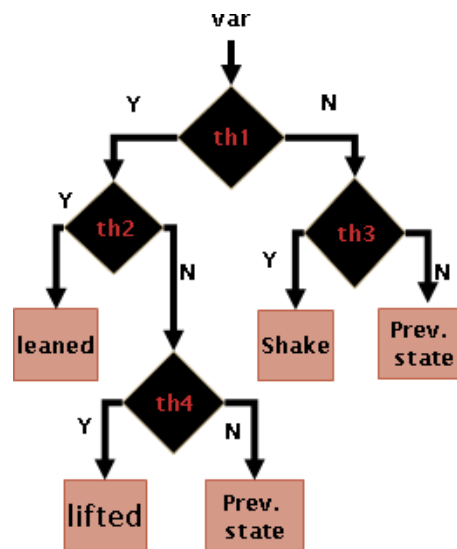


Figure 6.26: Micrel SMCube application.



(a) C4.5 tree classifier.

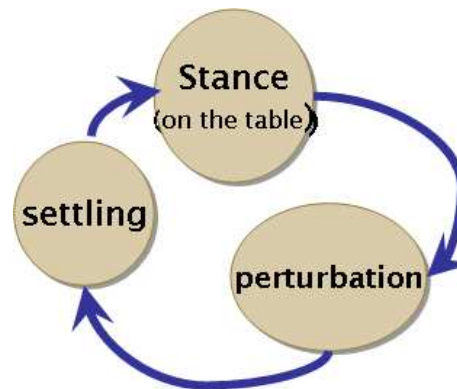
(b) *tap* FSM.

Figure 6.27: Micrel SMCube tasks.

fication ratio) of our approach tested on a small group of students which were not involved in the development of this application is above 98% for the three actions cube *placed on the table*, cube *held* and cube *shaked*.

Tap recognition is 96,25% and increase up to 98,75% if we increase the accelerometer sampling frequency from 40 to 100Hz. This behaviour is simply due to the best identification of the the accelerometer waveform.

The latency of detection in the case of sampling rate equal to 100Hz is presented in table 6.7

| Latency | | Next State (sec) | | | |
|----------------|-----------|------------------|--------|-------|------|
| | | Leaned dw | Lifted | Shake | Tap |
| Previous state | Leaned dw | | 0.47 | 0.42 | 0.35 |
| | Lifted | 0.48 | | 0.47 | |
| | Shake | 0.25 | 0.36 | | |
| | Tap | 0.25 | | | |

Table 6.7: Latency in gesture recognition.

6.5.4 HMM for SMCube, a feasibility study

Hidden Markov Models (HMMs) allow to handle temporal dynamics and classify more complex gestures than the ones described in the previous section. Typically, classification with HMM is performed using a recursive algorithm called *forward* algorithm. In section 4.5.1 we shown how a recursive approach that requires few memory resources can be applied for this algorithm. Although this process is a lightweight task, several issues must be considered in order to implement it on a low-power, low-cost microcontroller such as the one embedded on the SMCube.

In this section we evaluate the fixed point implementation of the forward algorithm. Discussion of ad-hoc solution to solve numerical problems while keeping low overall computational complexity are presented. Consideration about the complexity of the algorithm, both in terms of computational and memory cost, and its performance are discussed.

Lets recall here the steps of the forward algorithm presented in 4.5.1.

1. Initialization: $\alpha_1(i) = \pi_i(O_1)b_i(O_1), 1 \leq i \leq N$
2. Induction: $\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i)a_{ij}]b_j(O_{t+1}), 1 \leq j \leq N$ and $1 \leq t \leq T - 1$
3. Termination: $P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$

Normalization

According to the induction step of the forward algorithm we can see that the $\alpha_t(j)$ are sum of a large number of terms in the form $(\prod_{s=1}^{t-1} a_{q_s, q_{s+1}} \prod_{s=1}^t b_{q_s}(O_s))$. Since both the a_{ij} and the $b_i(k)$ are smaller than 1 as t become large $\alpha_t(j)$ tends to zero exponentially and soon it will exceed the range of any machine.

In order to avoid underflow, the $\alpha_t(j)$ are normalized at every step using the scaling factor $c_t = \frac{1}{\sum_{i=1}^N \alpha_t(i)}$. The scaled $\hat{\alpha}_t(j)$ are used in place of the $\alpha_t(j)$.

This normalization procedure is not suitable for low-power microcontrollers since it requires to perform N division each time a new sample is processed.

Thus we propose an alternative scaling approach:

1. check if all $\alpha_t(j)$ are smaller than $\frac{1}{2}$, otherwise scaling is not needed;
2. calculate the number of shift to the left l needed to render the highest $\alpha_t(j)$ greater than $\frac{1}{2}$;
3. shift all $\alpha_t(j)$ to the left of l bits.

This procedure requires only shifts and can be efficiently implemented on a microcontroller.

Likelihood

To compute the final sequence probability we can not use the scaled $\hat{\alpha}_t(j)$. The motivation can be easily understood if we think to the classical normalization where $\sum_{i=1}^N \hat{\alpha}_T(i) = 1$.

However we can notice that:

$$\sum_{i=1}^N \hat{\alpha}_T(i) = \prod_{t=1}^T 2^{l_t} \cdot \sum_{i=1}^N \alpha_T(i) = \prod_{t=1}^T 2^{l_t} \cdot P(O|\lambda) = r \longrightarrow P(O|\lambda) = \frac{r}{\prod_{t=1}^T 2^{l_t}} \quad (6.5)$$

Since P can be very small, we compute $\log P(O|\lambda) = \log(r) - \sum_{t=1}^T \log 2^{l_t}$.

If we decide to use \log_2 we already have the value of $\sum_{t=1}^T \log 2^{l_t}$ by keeping track of how many shift we computed for scaling. Furthermore, we do not need to compute $\log(r)$ since logarithm is a monotonically increasing function. Thus, to compare 2 models, we simply check for the one that required less shifts for scaling, in case of tie the one with higher r is the most probable model.

Evaluation

To evaluate this implementation we used a dataset made up of 10 complex gestures collected on a car assembly scenario (see section 6.3.2). The dataset has been extended since its first use and now includes 70 repetition for each gesture. These gestures can be compared to the ones that may be used within role-playing games.

To recognize these gestures we used a discrete HMM with 4 states ($N = 4$). Accelerometer' streams have been quantized to 3 symbols ($M = 3$).

To assess the complexity of the forward algorithm we assumed the values presented in tables 6.8 and 6.9, where N is the number of HMM states and C is the number of gestures we want to recognize (here $C = 10$). The memory cost is given by $\frac{\text{data size}}{8} \cdot C \cdot (N^2 + N \cdot M + 2 \cdot N)$. Where M is the number of symbols in the accelerometer stream.

| Operation | Cost |
|------------------------|------|
| Shift | 1 |
| Variables comparison | 1 |
| Sum 8 bits | 1 |
| Sum 16 bits | 2 |
| Sum 32 bits | 4 |
| Multiplication 8 bits | 2 |
| Multiplication 16 bits | 4 |
| Multiplication 32 bits | 6 |

Table 6.8: Computational complexity

| Algorithm | Cost |
|-------------------------------|---|
| $\alpha_{t+1}(i)$ Calculation | $(N + 1)$ mul. + N sum. |
| Normalization | $2 \cdot N + 1 + 2 \cdot \text{data size}$ |
| Single step (8-bit) | $C \cdot [N \cdot (3 \cdot N + 2) + 2 \cdot N + 17]$ |
| Single step (16-bit) | $C \cdot [N \cdot (6 \cdot N + 4) + 2 \cdot N + 33]$ |
| Single step (32-bit) | $C \cdot [N \cdot (10 \cdot N + 6) + 2 \cdot N + 65]$ |

Table 6.9: Algorithm complexity

To evaluate the performance loss due to the use of fixed point data representation, we classified the dataset using a floating point representation of the data and the traditional normalization algorithm (optimal performance), and using a fixed point representation and the shift scaling algorithm.

Performances are evaluated using the following indexes (see table 6.10):

- *Correct Classification Ratio*: $CCR = \frac{\text{number of correctly classified instances}}{\text{total number of instances}}$; is a global indication of the performance of the classifier.
- *Precision*: $PR_i = \frac{\text{number of instances correctly classified for class } i}{\text{number of instances classified as class } i}$; is an indication of the exactness of the classifier.

- *Recall*: $RC_i = \frac{\text{number of instances correctly classified for class } i}{\text{total number of instances from class } i}$;
is an indication of the performances of the classifier over a specific class

| Class | PR 8b | PR 16b | PR 32b | PR fl | RC 8b | RC 16b | RC 32b | RC fl |
|-------------------|-------|--------|--------|-------|---------------|---------------|---------------|---------------|
| Gesture 1 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 0.99 |
| Gesture 2 | 0.50 | 0.66 | 0.66 | 0.66 | 0.01 | 0.64 | 0.64 | 0.64 |
| Gesture 3 | 0.38 | 0.54 | 0.54 | 0.54 | 0.41 | 0.56 | 0.56 | 0.56 |
| Gesture 4 | 0.54 | 0.60 | 0.61 | 0.61 | 0.64 | 0.67 | 0.69 | 0.69 |
| Gesture 5 | 0.29 | 0.67 | 0.69 | 0.69 | 0.36 | 0.50 | 0.50 | 0.50 |
| Gesture 6 | 0.36 | 0.53 | 0.53 | 0.53 | 0.43 | 0.36 | 0.36 | 0.36 |
| Gesture 7 | 0.53 | 0.65 | 0.65 | 0.65 | 0.59 | 0.86 | 0.86 | 0.86 |
| Gesture 8 | 0.47 | 0.56 | 0.56 | 0.56 | 0.52 | 0.63 | 0.63 | 0.63 |
| Gesture 9 | 0.77 | 0.87 | 0.87 | 0.87 | 0.81 | 0.89 | 0.89 | 0.89 |
| Gesture 10 | 0.93 | 0.96 | 0.96 | 0.96 | 0.90 | 0.99 | 0.99 | 0.99 |
| CCR | | | | | 56.71% | 70.71% | 70.86% | 70.86% |

Table 6.10: Classification performances

Table 6.10 and 6.11 presents PR, RC, CCR, computational and memory cost for our implementations. The implementations that use 16 and 32 bits fixed

| Variables Size (bits) | CCR (%) | Memory cost (bytes) | Computational cost |
|-----------------------|---------|---------------------|--------------------|
| 8 | 56.71 | 360 | 810 |
| 16 | 70.71 | 720 | 1370 |
| 32 | 70.86 | 1440 | 2090 |
| Floating point | 70.86 | | |

Table 6.11: Performance and cost comparison

point data representation achieve similar or even equal CCR than the floating point solution. On the other hand the 8 bits fixed point implementation worsen the CCR by 14.15 %. However, the 32 bit solution can not be implemented on the ATmega168 since it requires more RAM than available, therefore the 16 bits solution is the optimal choice for the SMCube.

6.5.5 Conclusion

In this section we introduced the TANGerINE project, a natural interactive framework that exploits both wireless sensors electronics and video techniques in order to enrich tabletop interaction.

Within the framework two application scenarios have been developed.

TANGerINE Theater This project concerns a new kind of long form improvisation performance in which the audience is able to change the multimedia contributions (used as scenographies) manipulating the SMCube; the actors consequently improvise adapting the stories to the different changing settings.

A new tangible dialogue device has been introduced to enable the communication between audience and actors. By manipulating the SMCube, the audience is able to switch through six different multimedia scenographies projected in a large screen used as a frame for the stage. Every scenography is associated to a particular story improvised by the actors so that the audience can switch the story by switching the scenography, becoming de facto a “director-audience”.

A first version of the performance has been presented to the *Creativity Festival 2007* in Florence where a numerous audience have enjoyed and joined the show.

TANGerINE Tales Another application scenario focuses on supporting childrens face-to-face collaborative story-making.

One of the main concerns when designing for collaboration is that of supporting distributed participation: in a synchronous setting such as a face-to-face collaborative story-making activity, a system should enable multiple users simultaneous interaction. SMCubes can support this by allowing each child to use his own cube to interact with the system at the same time as other children. By switching the cube functionalities, the roles can also be fluidly re-assigned to different children according to emerging or predefined activity scripts. The spatial configuration of sensitive areas can also encourage different levels of engagement: a child in the Nearby Area can be involved in the activity peripherally, while a child in the Active Area is taking the leading role.

Finally, lessons from the design of collaborative systems have stressed the importance of accounting for authoring identity, especially when the supported activity is an open ended, creative one. The TANGerINE framework supports this, because it delivers a clear history of the actions performed by each individual, the system provides not only a strong motivational aspect for childrens participation, but also a useful tool for educators to assess each child's level of participation both in quantitative and qualitative terms.

We extended the capabilities of the SMCube in order to locally classify a set of four simple gesture performed by the user through a simple but efficient C4.5 decision tree and an ad-hoc algorithm. This enhancement allow multi-user, multi-context human computer interaction.

HMM is a common approach in gesture recognition, thus the possibility to implement this algorithm on a smart object and use it as a tangible interface greatly enhances potential for using the smart object as an effective HCI device.

Here we presented our evaluation of a fixed point implementation of the forward algorithm for HMM and our solutions to the peculiar numerical problems of this classification algorithm.

The 16-bit implementation is the best solution that can be implemented on our target microcontroller (ATMega168). This solution shows performance only slightly worse than the optimal ones of the floating point implementation (70.71% CCR, 16 bit fixed point; 70.68% floating point) and makes this implementation suitable for smart object equipped with low-power, low-cost microcontrollers such as the SMCube.

6.6 Pervasive datasets

Ambient intelligence envisions a world where the environment is able to sense its own physical state, the presence of people, their state and current activity. With this information, the environment itself can provide context-aware services to support its inhabitants. In such an environment, a dense mesh of sensors is integrated into stationary objects, artifacts, clothing. An important aspect of context information is the activity of people within the smart environment. Consequently the development of activity recognition algorithms is a very active field of research. One of the main challenges is that both the design and the validation of activity recognition techniques require large datasets that must be obtained through time-consuming and expensive test sessions.

In this section, we describe our considerations and experiences with collecting data from an sensorized environment with end goal of producing a high-quality, freely available reference dataset for benchmarking activity recognition algorithms. Our experiments include 5 different sensing modalities and up to 12 wireless sensors communicating at the same time. The dataset is constructed out of 8 different scenarios of everyday life, which include 17 activities composed of 64 micro-activities. The activities have been performed by two test subjects 10 times each. During the time a subject performed the activities, the experiment supervisor recorded time markers to identify the start and duration of each activity.

The dataset will be available for research purposes and is intended to be a common benchmark for design, evaluation and comparison of different activity recognition approaches. To the best of our knowledge, it is the first that includes such detailed labeling of activities recorded from body worn and environmental sensors and smart objects. In this paper we describe our recording setup and recommendations and hope that others will contribute to the dataset to make it grow to a commonly useful resource.

6.6.1 Experiment setup

The 8 scenarios of the dataset involve: cooking a soup in the kitchen (Kitchen), assembling a shelf with three boards (Shelf assembly), then attaching a metal crossbar to it (Crossbar assembly), three sets where the subject is working on its desk reading, writing, and using the computer (Relaxing 1+2, Working), a set where two subjects collaboratively assemble the shelf (Collaboration), and a last set where the subjects perform activities which are not to be recognized, but may cause false positives in the recognition algorithms, such as scratching the head or using a mobile phone (Distractions).

The activities were recorded by body-worn sensors featuring accelerometers at both wrists and on the left leg right above the knee and bend sensors monitoring the extension of the fingers of the right hand. Further accelerometers were placed on 12 objects and tools the subjects interacted with and on a shelf leg, a shelf board, and a chair. Additional 8 light sensors were placed into drawers and cupboards to monitor whether they have been opened by the test subject. Work on the computer was sensed by recording the number of key presses and mouse movements. A pyroelectric infrared (PIR) motion sensor recorded when the subject entered and left the room after each recording (see table 6.13). Finally, a camera filmed the room during the experiment.

The raw data samples from the sensors have been collected through wireless communication to a laptop PC where a supervisor labeled the beginning and the end of each activity using custom software developed for this project. Only the PIR was connected by a serial cable. For synchronization, a timestamp was added at the reception of every message on the recording PC.

Due to the complexity of the recording setup we have considered several points to successfully conduct the experiment. We report here on our experiences with labeling the activities and with ensuring a good performance of the wireless communication.

Activities and labeling

Our goal was to have a detailed record of micro activities performed during our experiments. We thus defined 64 atomic activities, such as picking up a screwdriver or turning a screw, which should allow identifying which of the 17 composite activities the subject has been performing at that time, such as fixing a crossbar on the shelf. Table 6.12 lists the different scenarios and the number of different composite and atomic activities they include. Some atomic and composite activities occur multiple times during a recording, and some distracting activities, such as scratching the head, were occasionally inserted. The average number of labels to be set during a recording is indicated in the

last column of table 6.12.

| Scenario | Composite activities | Atomic Activities | Average occurrences |
|-------------------|----------------------|-------------------|---------------------|
| Relaxing 1 | 3 | 9 | 27 |
| Relaxing 2 | 2 | 9 | 18 |
| Crossbar Assembly | 4 | 12 | 27 |
| Kitchen | 8 | 37 | 68 |
| Shelf Assembly | 5 | 14 | 99 |
| Working | 5 | 17 | 31 |
| Collaboration | 6 | 17 | 87 |
| Distraction | 6 | 10 | 60 |
| Totals | 17 | 64 | 400 |

Table 6.12: Number of composite and atomic activities and total number of activity occurrences.

One option to annotate the recordings would have been to record all data without labels and to add the activity information after the experiments by inspecting the recorded films. However, we expected this to take considerably more time than labeling the activities online and manually check the labels later on. The drawback of this approach is that due to the large number of different activities, the sequence of the activities needs to be fixed, such that the experiment supervisor can find them in a list in useful time.

We have therefore designed a simple user interface which displayed the sequence of activities to be performed by the subject (see figure 6.28). The experiment supervisor could select the activity to be performed next and conveniently start and stop the time during which it was performed. After a short training session, a user is able to efficiently annotate the activities. The software added the time-stamped event to the recorded data flow. An important addition is a sensor health indication showing the active sensors. The sensor identifiers are colored green when data has been received during the last seconds. When data was missing, the corresponding identifier turns red, such that the experiment supervisor is alerted and can decide whether or not to stop the recording. We experienced several times during the experiment that sensors failed to deliver data only for short time, than quickly recovered the transmission. By monitoring the network status we were able to stop the experiment if critical sensors were not responding for an extended amount of time.

In a post-processing step, the labels were inspected and corrected manually by cross-checking against the video recording of the experiment. The accuracy of the online labeling by the experiment supervisors was evaluated, such that it can be compared to automatic context recognition algorithms. For the evaluation, we accepted a human-set label as true positive if it intersected at least on one sample with the ground truth label. This definition does not allow reporting events early or late. Labels that had no match on ground truth were

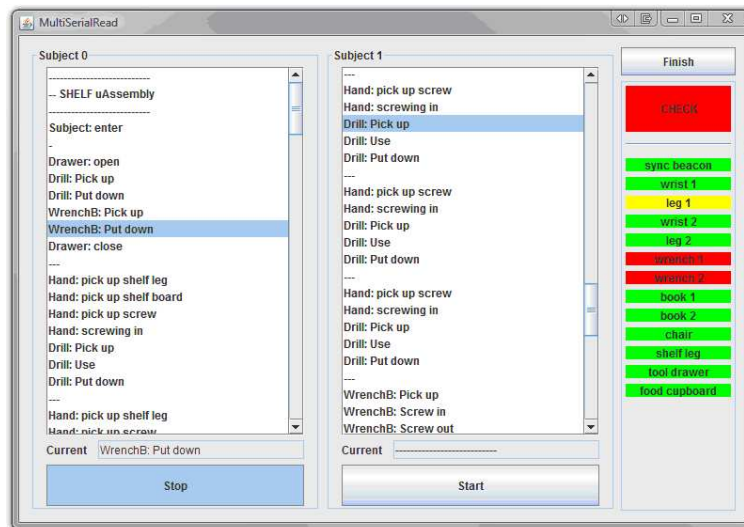


Figure 6.28: Labeling software with a) atomic label sequence, b) composite label sequence c) start/stop button, and d) sensor health status.

reported as false positives while ground truth labels without matching event as false negatives. Multiple matches between labels and ground truth were all counted as true positives.

It is interesting to note that even the “gold reference” activity recognition performed by the human brain is not perfect. The accuracy of the human labeling was determined to be 96.75%, with a precision of 97.83% and a recall of 98.87%. Most errors came from mixing up two different activities or setting the label too late.

Wireless communication

Data from the sensors has been collected through different media by a single laptop PC. The PIR sensor readings were gathered using a serial cable, data from the right wrist and the bend sensors was sent using a Bluetooth radio, all other sensor nodes were based on Tmote platforms and use the TinyOS wireless stack based on IEEE 802.15.4. The accelerometers on the sensor nodes were sampled at 50 Hz on three axes, requiring a sensor data throughput of 2.4 kbit/s. As expected (Shnayder, 2005), early tests showed that at this data-rate we suffer for high message loss when more than 3 nodes are streaming on a single channel, thus we decided to use multiple parallel channels and assign only 2 sensors to each.

During the experiment we still experienced nodes failures or communication loss which reduced the quality of the acquired streams. Messages losses vary from 0% up to 38.0% (PIR sensor) with an average of 8.7% (see table 6.13).

Higher packet losses are due to sensor that failed at the beginning of an experiment and whose failure was not recognized until the end of the test, while lower data losses are mainly due to temporary occlusion due to movements of the user or to interference between sensors streaming on the same channel. The high data loss on the PIR sensor is not a real loss of packets, but it is simply due to the fact that this sensor was not available during all experiments. This analysis shows that we successfully reduced the data loss by using different channels. However, wireless sensor nodes communication is still unreliable and depends on body positions and movements, as shown also in [116].

| Category | Position | Message Loss (%) |
|----------------|----------------|------------------|
| Infrastructure | PIR | 38.0 |
| | Computer | 0.0 |
| Tools | Hammer | 0.9 |
| | Screw Driver | 2.1 |
| | Scissors | 0.0 |
| | Knife | 10.9 |
| | Book 1 | 12.7 |
| | Book 2 | 9.8 |
| | Phone | 13.8 |
| | Stirring Spoon | 3.7 |
| | Drill | 0.6 |
| | Wrench Small | 23.7 |
| | Wrench Big | 15.3 |
| | Pen | 0.7 |
| | Furniture | Shelf Board |
| Chair | | 4.3 |
| Food Cupboard | | 2.0 |
| Dish Cupboard | | 2.1 |
| Cutlery Drawer | | 2.6 |
| Garbage | | 0.6 |
| Pot Drawer | | 1.8 |
| Shelf Leg | | 5.9 |
| Tool Drawer | | 0.1 |
| Desk Drawer | 2.6 | |
| Body Worn | Glove | 0.5 |
| | Wrist | 0.6 |
| | Left Leg | 10.2 |
| Average | | 8.7 |

Table 6.13: Overall experiment message loss.

6.6.2 Benchmark for context recognition

The dataset we built is suitable for the development and the comparison of several activity recognition approaches and techniques. Possible usages include [96]:

- Comparison of Approaches. Typical activity recognition techniques rely on sensors that are either placed on user body, or on objects, or in the environment. As a consequence, different algorithms are tested on different datasets and their comparison is almost impossible. We believe

that the possibility to test different approaches on a common benchmark will allow a better understanding of the benefit provided by different techniques. Furthermore, since sensor networks are dynamic systems, a researcher can compare different scenarios where the user (or different users) may or may not be equipped with smart garments or performs her/his activity with or without smart objects and moves within or outside a smart environment.

- **Distributed vs. centralized recognition.** The authors of (Amft, 2007) have shown how distributed recognition of activities can be performed. An evaluation of a centralized recognition algorithm, which has all the data available, vs. a distributed recognition algorithm, which recognizes activities locally on the individual sensors while a centralized node only fuses their results, can be performed. There are as well some intermediate solutions where sensor data on different sensors may be correlated.
- **Hierarchical Activity Recognition.** In a multilevel hierarchical approach to activity recognition, researchers can develop activity recognition techniques specifically for each level. By feeding back their recognition results to the dataset, higher-level or lower level algorithm designers can investigate the influences of different approaches on other levels on the overall performance. Furthermore they can test the benefit of cross-level information exchange.
- **Context-aware activity recognition.** The knowledge of the higher level activities may be used to restrict the search space of the lower-level activities to improve their recognition accuracy. As classifiers have then to only discriminate a limited subset, an improvement of their performance can be expected. For example, micro activity detection can benefit from the knowledge that the user is currently mounting a shelf board by restricting its detection to relevant micro activities and omitting the detection of activities related to cooking.

6.6.3 Conclusion

This paper summarizes our experiences setting up and running a diverse set of scenarios within an ambient intelligence environment. The experiments included multiple sensing modalities on sensors mounted on the body, embedded within tools used by the subject, and sensors in the environment. With respect to the datasets presented in 6.2.3 we produced a dataset with a high number of repetition of each activity performed by multiple user and moni-

tored through a heterogeneous set of wore, embedded in tools and environmental sensors.

In contrast to the other dataset presented in section 6.2.3 here we focus on activity recognition through a set of inertial sensors. The addition of our work are:

- Combination of body worn sensors, smart objects, environmental sensor.
- Several repetition of the same activity.
- Multiple user.
- Hierarchical organization of gestures.
- Fully labeled activities.

We described how to perform efficient labeling and ensure a good performance of the wireless channel. The resulting dataset will be publicly available and we hope that it may support different researchers to engage into the research challenges we have outlined.

Chapter 7

Conclusion

Ambient Intelligence (AmI) is a vision on the future of the information society where smart, electronic environment are aware and responsive to their context. In such smart environment, technology is invisible and embedded into the surrounding. People moving into this settings engage many computational devices and systems simultaneously even if they are not aware of their presence. AmI stems from the convergence of three key technologies: *ubiquitous computing*, *ubiquitous communication* and *natural interfaces*.

The dependence on a large amount of fixed and mobile sensors embedded into the environment makes of *Wireless Sensor Network (WSN)* one of the most relevant enabling technologies for AmI. WSN are complex systems made up of a number of sensor nodes, simple devices that typically embed a low power computational unit (microcontrollers, FPGAs etc.), a wireless communication unit, one or more sensors and a some form of energy supply (either batteries or energy scavenger modules). Low-cost, low-computational power, low energy consumption and small size are characteristics that must be taken into consideration when designing and dealing with WSN.

In order to handle the large amount of data generated by a WSN several multi sensor data fusion techniques have been developed. The aim of multi-sensor data fusion is to combine data to achieve better accuracy and inferences than could be achieved by the use of a single sensor alone.

In this dissertation we presented our results in building several AmI applications suitable for a WSN implementation. All the presented applications exploit data collected from either a homogeneous or heterogeneous sensor network and fuses such information to gather high level inferences about the context.

The work can be divided into two main areas: *Multimodal Surveillance* and *Activity Recognition*.

Within the field of multimodal surveillance we developed new techniques to track people movement through the use of a dense mesh of Pyroelectric InfraRed (PIR) sensors. PIR sensors are low-power, low-cost devices able to transduce changes in incident infrared radiation into an electric signal. Such devices typically are used as simple presence-absence detectors. In section 5.3 and 5.4 we shown how we can extract more information than simple presence by fusing simple features from a mesh of PIR sensor. In particular we presented 2 system able to detect number and direction of movement of people and distance of movement.

In section 5.5 we shown how this PIR sensor network can be integrated with a video surveillance system in order to overcome to several limits of the vision system such as changes in direction of movement behind occlusion and reflections.

Section 5.6 describes how we can extend the lifetime of a wireless video node powered by a solar scavenger using a PIR sensor and a tunable wake-up threshold.

Within the field of activity recognition we studied how we could perform gesture recognition in a future scenario where a redundant number of low-cost, low-power sensors are deployed in the environment. In particular in section 6.3 we proposed a novel approach where a large number of nodes placed on the user arm classify his gesture. Through the use of a meta classifier we exploit the redundant information in order to increase single sensor accuracy while increase noise and fault tolerance. Furthermore in section 6.4 we shown how redundancy can be used to perform an application driven power-performance trade off. Such technique allow the whole network lifetime extension while achieving a minimum, application defined, quality of recognition.

Activity recognition techniques can be used to build natural, tangible interfaces with the smart environment. In section 6.5 we presented our Smart Micrel Cube (SMCube), a tangible interface used within the TANGerINE framework. In this section we presented our work in increasing the power of this smart object by developing gesture recognition algorithm for this low-power, low-cost device.

The development of activity recognition algorithm often is slowed down due to the lacking of available datasets to build and compare new techniques. Aware of this limit we built a dataset that will be available for research purposes and is intended to be a common benchmark for design, evaluation and comparison of different activity recognition approaches. Our experience is presented in section 6.6.

Bibliography

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002.
- [2] Alan Daniels. Ubiquitous computing. <http://www.cc.gatech.edu/classes/cs6751.97.fall/projects/gacha/daniels.essay.html>, 1997.
- [3] E. G. Araujo, Y. Yang, R. A. Grupen, P. A. Deegan, B. S. Lerner, E. M. Riseman, and Z. Zhu. Software mode changes for continuous motion tracking. In *In Int. Workshop on Self Adaptive Soft*, 2000.
- [4] R. W. Astheimer and F. Schwarz. Thermal imaging using pyroelectric detectors. *Appl. Opt.*, 7(9):1687–1695, 1968.
- [5] A. Asuncion and D. Newman. UCI machine learning repository, 2007.
- [6] Y.-W. Bai and H. Teng. Enhancement of the sensing distance of an embedded surveillance system with video streaming recording triggered by an infrared sensor circuit. *SICE Annual Conference, 2008*, pages 1657–1662, Aug. 2008.
- [7] D. Bannach, O. Amft, K. Kunze, E. Heinz, G. Tröster, and P. Lukowicz. Waving real hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game. *IEEE Symposium on Computational Intelligence and Games, 2007. CIG 2007.*, pages 32–39, 1-5 April 2007.
- [8] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing: Proc. of the 2nd International Conference*, pages 1–17, Apr. 2004.
- [9] S. Baraldi, A. Bimbo, L. Landucci, and A. Valli. wikipable: finger driven interaction for collaborative knowledge-building workspaces. *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, June 2006.
- [10] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and F. Y. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655–1695, May 2007.
- [11] S. Bauer, S. Bauer-Gogonea, and B. Ploss. The physics of pyroelectric infrared devices. *Applied Physics B: Lasers and Optics*, 54:544–551, June 1992.
- [12] A. Y. Benbasat and J. A. Paradiso. Groggy wakeup - automated generation of power-efficient detection hierarchies for wearable sensors. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007) March 26 28, 2007 RWTH Aachen University, Germany*, 2007.
- [13] L. Benini, E. Farella, and C. Guiducci. Wireless sensor networks: Enabling technology for ambient intelligence. *Microelectron. J.*, 37(12):1639–1649, 2006.

- [14] K. P. Bennett. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, 2:2000, 2000.
- [15] Berkley. <http://robotics.eecs.berkeley.edu/pister/SmartDust/>.
- [16] N. Bharatula, P. Lukowicz, and G. Tröster. Functionality-power-packaging considerations in context aware wearable systems. *Personal and Ubiquitous Computing*, 12(2):123–141, 2008.
- [17] D. Bhatia, L. Estevez, and S. Rao. Energy efficient contextual sensing for elderly care. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4052–4055, Aug. 2007.
- [18] E. P. Blasch and S. Plano. Jdl level 5 fusion model: user refinement issues and applications in group tracking. *Signal Processing, Sensor Fusion, and Target Recognition XI*, 4729(1):270–279, 2002.
- [19] R. Bodor, B. Jackson, and N. Papanikolopoulos. Vision-based human tracking and activity recognition. In *Proc. of the 11th Mediterranean Conf. on Control and Automation*, pages 18–20. Kostrzewa Joseph, 2003.
- [20] D. Brunelli, L. Benini, C. Moser, and L. Thiele. An efficient solar energy harvester for wireless sensor nodes. In *DATE '08: Proceedings of the conference on Design, automation and test in Europe*, pages 104–109, New York, NY, USA, 2008. ACM.
- [21] D. Brunelli, E. Farella, L. Rocchi, M. Dozza, L. Chiari, and L. Benini. Bio-feedback system for rehabilitation based on a wireless body area network. *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5 pp.–531, March 2006.
- [22] P. Bryant and H. W. Braun. Some applications of a motion detecting camera in remote environments. Technical report, University of California San Diego, HPWREN, 2003.
- [23] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [24] O. Cafini, E. Farella, L. Benini, S. Baraldi, N. Torpei, L. Landucci, and A. Del Bimbo. Evolving tuis with smart objects for multi-context interaction. In *Proc. of International Conference on Computer-Human Interaction (CHI)*, Florence, Italy, April 2008. ACM, ACM Press.
- [25] S. Calderara, A. Prati, R. Vezzani, and R. Cucchiara. Consistent labeling for multi-camera object tracking. In *13th International Conference on Image Analysis and Processing (ICIAP 2005)*, pages 1206–1214, Cagliari, Italy, Sept. 2005.
- [26] Carnegie Mellon University. Aura project overview. <http://www-2.cs.cmu.edu/aura/>.
- [27] G. Chambers, S. Venkatesh, G. West, and H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *Proc. 16th IEEE Conference on Pattern Recognition*, pages 1082–1085, 2002.
- [28] H. Chen, H. Wu, and N.-F. Tzeng. Grid-based approach for working node selection in wireless sensor networks. In *IEEE International Conference on Communications*, volume 6, pages 3673– 3678, 20–24 June 2004.
- [29] X. Chen, X. Zhang, Z.-Y. Zhao, J.-H. Yang, V. Lantz, and K.-Q. Wang. Hand gesture recognition research based on surface emg sensors and 2d-accelerometers.

- 11th IEEE International Symposium on Wearable Computers, 2007, pages 11–14, 11–13 Oct. 2007.
- [30] C.-Y. Chong and S. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.
- [31] S. Commuri and V. Tadigotla. Dynamic data aggregation in wireless sensor networks. *IEEE 22nd International Symposium on Intelligent Control, 2007. ISIC 2007.*, pages 1–6, 1–3 Oct. 2007.
- [32] B. Cornel. Localization system using video and pir information fusion. *Soft Computing Applications, 2007. SOFA 2007. 2nd International Workshop on*, pages 21–23, Aug. 2007.
- [33] B. Cornel. Recognition system using video and infrared information fusion. *Computational Intelligence and Intelligent Informatics, 2007. ISCIII '07. International Symposium on*, pages 281–283, March 2007.
- [34] CrossBow. <http://www.xbow.com/Products/productdetails.aspx?sid=174>.
- [35] CrossBow. <http://www.xbow.com/Products/productdetails.aspx?sid=164>.
- [36] CrossBow. <http://www.xbow.com/index.aspx>.
- [37] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337–1342, Oct. 2003.
- [38] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human-behavior analysis. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 35(1):42–54, Jan. 2005.
- [39] R. Cucchiara, A. Prati, L. Benini, and E. Farella. T-park: ambient intelligence for security in public parks. *Intelligent Environments, 2005. The IEE International Workshop on (Ref. No. 2005/11059)*, pages 243–251, June 2005.
- [40] R. Cucchiara, A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi. Using a wireless sensor network to enhance video surveillance. *Journal of Ubiquitous Computing and Intelligence (JUCI)*, 1:1–11, 2006.
- [41] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.
- [42] L. Dai and P. Basu. Energy and delivery capacity of wireless sensor networks with random duty-cycles. In *to appear in 2006 IEEE International Conference on Communications*, pages 3503 – 3510, June 2006.
- [43] N. Davies, D. P. Siewiorek, and R. Sukthankar. Special issue: Activity-based computing. *IEEE Pervasive Computing*, 7(2):20–21, 2008.
- [44] D. L. de Ipina, J. I. Vazquez, D. Garcia, J. Fernandez, I. Garcia, D. Sainz, and A. Alineida. A platform to build smart spaces controllable from mobile devices. *Intelligent Environments, 2006. IE 06. 2nd IET International Conference on*, 1:31–40, July 2006.
- [45] S. de Vlaam. Object tracking in a multi sensor network, Aug. 2004.
- [46] I. Demirkol, C. Ersoy, and F. Alagoz. Mac protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, April 2006.
- [47] J. Deng and H. Tsui. An hmm-based approach for gesture segmentation and recognition. In *15th International Conference on Pattern Recognition*, volume 3, pages 679–682. 2000.

- [48] S. Dengler, A. Awad, and F. Dressier. Sensor/actuator networks in smart homes for supporting elderly and handicapped people. *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, 2:863–868, May 2007.
- [49] Department of Electronic Informatic and Systems (DEIS). <http://www.deis.unibo.it/>.
- [50] J. Dias, N. Barata, P. Santos, A. Correia, P. Nande, and R. Bastos. In your hand computing: tangible interfaces for mixed reality. *Augmented Reality Toolkit Workshop, 2003. IEEE International*, pages 29–31, Oct. 2003.
- [51] T. G. Dietterich. *Machine Learning for Sequential Data: A Review*. 2008.
- [52] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J. Burgelman. Thats what friends are for ambient intelligence (ami) and the is in 2010. In *Innovations for an e-society*, 2001.
- [53] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J. C. Burgelman. Scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, February 2001.
- [54] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [55] R. Dugad and U. B. Desai. A tutorial on hidden markov models. Technical report, Signal Processing and Artificial Neural Networks Laboratory Department of Electrical Engineering Indian Institute of Technology Bombay Powai, Bombay 400 076, India, 1996.
- [56] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005.*, pages 497–502, 15 April 2005.
- [57] Ember. <http://www.ember.com/index.html>.
- [58] Ember. http://www.ember.com/pdf/120-3021-000_App.Dev.Ref.Manual.pdf.
- [59] T. Eswam and P. Chapman. Comparison of photovoltaic array maximum power point tracking techniques. *Energy Conversion, IEEE Transaction on*, 2:439–449, 2007.
- [60] ETH Zurich. <http://www.btnode.ethz.ch/>.
- [61] L. Evers, M. Bijl, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. Wireless sensor networks and beyond: A case study on transport and logistics, 2005.
- [62] J.-S. Fang, Q. Hao, D. J. Brady, B. D. Guenther, and K. Y. Hsu. Real-time human identification using a pyroelectric infrared detector array and hidden markov models. *Opt. Express*, 14(15):6643–6658, 2006.
- [63] J.-S. Fang, Q. Hao, D. J. Brady, M. Shankar, B. D. Guenther, N. P. Pitsianis, and K. Y. Hsu. Path-dependent human identification using a pyroelectric infrared sensor and fresnel lens arrays. *Opt. Express*, 14(2):609–624, 2006.
- [64] W.-C. Fang, S. Kedar, S. Owen, G.-Y. Wei, D. Brooks, and J. Lees. System-on-chip architecture design for intelligent sensor networks. In *IIH-MSP '06: Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 579–582, Washington, DC, USA, 2006. IEEE Computer Society.

- [65] E. Farella, A. Pieracci, and A. Acquaviva. Design and implementation of wimoca node for a body area wireless sensor network. *Systems Communications, 2005. Proceedings*, pages 342–347, Aug. 2005.
- [66] E. Ferrein, L. Hermanns, and G. Lakemeyer. Comparing sensor fusion techniques for ball position estimation. In *In Proc. of the RoboCup 2005 Symposium*, 2005.
- [67] K. P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal Ubiquitous Comput.*, 8(5):347–358, 2004.
- [68] Freescale. http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13224V&nodeId=0106B9869925657103.
- [69] Freescale. 13192 developers starter kit, application note 2762. Technical report, Freescale, 2007.
- [70] A. Garg, S. Balakrishnan, and S. Vaithyanathan. Asynchronous hmm with applications to speech recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, 1:I-1009–12 vol.1, 17-21 May 2004.
- [71] S. K. Goldenstein. A gentle introduction to predictive filters. *Revista de Informtica Terica e Aplicada*, 11(1), 2004.
- [72] G. Gonzalez, M. Organero, and C. Kloos. Early infrastructure of an internet of things in spaces for learning. *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 381–383, July 2008.
- [73] U. Gopinathan, D. Brady, and N. Pitsianis. Coded apertures for efficient pyroelectric motion tracking. *Opt. Express*, 11(18):2142–2152, 2003.
- [74] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 129–143, New York, NY, USA, 2004. ACM.
- [75] S. Gupta and C. Jaynes. The universal media book: tracking and augmenting moving surfaces with projected information. *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 177–180, Oct. 2006.
- [76] D. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, Jan 1997.
- [77] C. Hardenberg and F. Brard. Bare-hand human-computer interaction. In *Proceedings of the ACM Workshop on Perceptive User Interfaces*, pages 1–8, 2001.
- [78] H. Harms, O. Amft, and D. R. G. Tröster. Smash: A distributed sensing and processing garment for the classification of upper body postures. In *Third international conference on body area networks*, 2008.
- [79] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda. People count system using multi-sensing application. *Solid State Sensors and Actuators, 1997. TRANSDUCERS '97 Chicago., 1997 International Conference on*, 2, Jun 1997.
- [80] T. He, S. Krishnamurthy, L. Luo, T. Y. and Lin Gu and Radu Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. H. Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):1–38, 2006.
- [81] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An

- integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2(1):1–38, 2006.
- [82] E. Heinz, K. Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In *Proc. IEEE Symposium on Computational Intelligence and Games (CIG)*. 2006.
- [83] J. L. Hernandez-Rebollar. Gesture-driven american sign language phraselator. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 288–292, New York, NY, USA, 2005. ACM Press.
- [84] Hill. Mica: a wireless platform for deeply embedded networks. *Micro, IEEE*, 22(6):12–24, Nov/Dec 2002.
- [85] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *Architectural Support for Programming Languages and Operating Systems*, Nov. 2000.
- [86] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:66–75, 1994.
- [87] Honeywell security and custom electronic. Is-215t datasheet, 2008.
- [88] IBM. Planet blue project overview.
<http://www.research.ibm.com/compsci/planetblue.html>.
- [89] IEEE 802.15.4 Task Group 4. <http://www.ieee802.org/15/pub/TG4.html>.
- [90] M. Ilyas, I. Mahgoub, and L. Kelly. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, Inc., Boca Raton, FL, USA, 2004.
- [91] Image Lab. <http://imagelab.ing.unimore.it/>.
- [92] Information Society Technology. Istag mission.
<http://cordis.europa.eu/ist/istag.htm>, 1999.
- [93] InfraTech. <http://www.infratec.de/index.php?id=134&L=>.
- [94] S. S. Intille, K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1941–1944, 2005.
- [95] H. Ishii. The tangible user interface and its evolution. *Commun. ACM*, 51(6):32–36, 2008.
- [96] A. Jaimes and N. Sebe. Multimodal human-computer interaction: A survey. *Comput. Vis. Image Underst.*, 108(1-2):116–134, 2007.
- [97] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reactable. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.
- [98] H. Junker, O. Amft, P. Lukowicz, and G. Tröster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recogn.*, 41(6), 2008.
- [99] H. Junker, P. Lukowicz, and G. Tröster. Sampling frequency, signal resolution and the accuracy of wearable context recognition systems. *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on*, 1:176–177, 31 Oct.-3 Nov. 2004.
- [100] H. Junker, M. Stäger, G. Tröster, D. Blättler, and O. Salama. Wireless networks in context aware wearable systems. In *Proceedings of the 1st European Workshop on Wireless Sensor Networks (EWSN 2004), Work-in-Progress Session*, pages 37–40, Jan. 2004.

- [101] M. M. K. Yoshinari and. A human motion estimation method using 3-successive video frames. In *Proc. of Int. Conf. on Virtual Systems and Multimedia GIFU*, pages 135–140, 1996.
- [102] S. Kallio, J. Kela, P. Korpipää, and J. Mäntyjärvi. User independent gesture interaction for small handheld devices. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(4):505–524, 2006.
- [103] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [104] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the IEEE International Conference on Data Mining*, pages 289–96, 2001.
- [105] A. Kerhet, M. Hu, F. Leonardi, A. Boni, and D. Petri. Svm-like algorithms and architectures for embedded computational intelligence. 2007.
- [106] A. Kerhet, F. Leonardi, A. Boni, P. Lombardo, M. Magno, and L. Benini. Distributed video surveillance using hardware-friendly sparse large margin classifiers. In *AVSS 2007: Proceedings of the 2007 IEEE International Conference on Advanced Video and Signal based Surveillance*, 2007.
- [107] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1355–1360, Oct. 2003.
- [108] J. Kittler and al. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [109] R. Kling. Intel motes: advanced sensor network platforms and applications. *Microwave Symposium Digest, 2005 IEEE MTT-S International*, pages 4 pp.–, June 2005.
- [110] G. Klir and T. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, New Jersey, 1988.
- [111] M. Ko, G. West, S. Venkatesh, and M. Kumar. Online context recognition in multisensor systems using dynamic time warping. pages 283–288. 2005.
- [112] A. Kotranza and B. Lok. Virtual human + tangible interface = mixed reality human an initial exploration with a virtual breast exam patient. *Virtual Reality Conference, 2008. VR '08. IEEE*, pages 99–106, March 2008.
- [113] M. Kranz, D. Schmidt, P. Holleis, and A. Schmidt. A display cube as tangible user interface. In *In Adjunct Proceedings of the Seventh International Conference on Ubiquitous Computing (Demo 22)*, September 2005.
- [114] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. P. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proc. of the 9th International Symposium on Wearable Computers*, pages 20–26, Los Alamitos, CA, 2005. IEEE Computer Society Press.
- [115] S. Kulkarni, G. Lugosi, and S. Venkatesh. Learning pattern classification—a survey. *Information Theory, IEEE Transactions on*, 44(6):2178–2206, Oct 1998.
- [116] M. Kusserow, O. Amft, and G. Tröster. Bodyant: Miniature wireless sensors for naturalistic monitoring of daily activity. In *Bodynets 2009: Proceedings of the 4th International Conference on Body Area Networks*. ACM press, 2009.

- [117] LAICA. Laica project overview.
<http://www.laica.re.it/index.html>.
- [118] K. Lam, H. Tung, and K. Tsang. A zigbee reminder system for mobile data transfer in airports. *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 807–808, Jan. 2008.
- [119] L. Lam and C. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *27(5):553–568*, 1997.
- [120] D.-S. Lee and S. N. Srihari. Handprinted digit recognition: a comparison of algorithms. In *Proc. 3rd Int. Workshop Frontiers Handwriting Recognition*, pages 153–162, May 1993.
- [121] J. Y. Lee, D. Seo, G. W. Rhee, S. H. Hong, and J.-S. Nam. Virtual and pervasive smart home services using tangible mixed reality. *Parallel and Distributed Processing with Applications, 2008. ISPA '08. International Symposium on*, Dec. 2008.
- [122] J. Lifton, D. Seetharam, M. Broxton, and J. Paradiso. Pushpin computing system overview: a platform for distributed, embedded, ubiquitous sensor networks. In *in F. Mattern and M. Naghshineh (eds): Pervasive 2002, Proceedings of the Pervasive Computing Conference*, pages 139–151. Springer Verlag, 2002.
- [123] J. Liu, J. Sun, and S. Wang. Pattern recognition: An overview. *IJCSNS International Journal of Computer Science and Network Security*, 6(6):57–61, June 2006.
- [124] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, and F. White. Revisiting the jdl data fusion model ii. In *In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004*, pages 1218–1230, 2004.
- [125] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *UbiComp*, pages 483–500, 2007.
- [126] C. Lombriser, M. Stäger, D. Roggen, and G. Tröster. Titan: A tiny task network for dynamically reconfigurable heterogeneous sensor networks. In *15. Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, pages 127–138, 2007.
- [127] M. Magno, D. Brunelli, P. Zappi, and L. Benini. A self-powered video node triggered by pir sensors. In *5th European Conference on Wireless Sensor Networks (EWSN)*, Ganuary 2008.
- [128] M. Magno, D. Brunelli, P. Zappi, and L. Benini. A solar-powered video sensor node for energy efficient multimodal surveillance. In *DSD '08: Proceedings of the 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*, pages 512–519, Washington, DC, USA, 2008. IEEE Computer Society.
- [129] D. P. Mandic, D. Obradovic, A. Kuh, T. Adali, U. Trutschel, M. Golz, P. De Wilde, J. A. Barria, A. Constantinides, and J. A. Chambers. Data fusion for modern engineering applications: An overview. In *ICANN (2)*, pages 715–721, 2005.
- [130] M. Marin-Perianu, C. Lombriser, O. Amft, P. Havinga, and G. Tröster. Distributed activity recognition with fuzzy-enabled wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2008.
- [131] M. Maroti, G. Simon, A. Ledeczi, and J. Sztipanovits. Shooter localization in urban terrain. *Computer*, 37(8):60–61, Aug. 2004.

- [132] K. Martinez, J. K. Hart, and R. Ong. Environmental sensor networks. *Computer*, 37(8):50–56, 2004.
- [133] C. Mattmann, O. Amft, H. Harms, G. Tröster, and F. Clemens. Recognizing upper body postures using textile strain sensors. In *Proc. of the 11th IEEE International Symposium on Wearable Computers (ISWC)*, 2007.
- [134] I. Mautua, P. T. Kirisci, T. Stiefmeier, M. L. Sbodio, and H. Witt. A wearable computing prototype for supporting training activities in automotive production. In *4th International Forum on Applied Wearable Computing (IFAWC)*, 2007.
- [135] P. S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.
- [136] A. Mazalek, M. Reynolds, and G. Davenport. Tviews: An extensible architecture for multiuser digital media tables. *Computer Graphics and Applications, IEEE*, 26(5):47–55, Sept.-Oct. 2006.
- [137] Clare, Inc. Clare solar cells - cpc series data sheet <http://www.clare.com/products/solarcell.htm>. August, 2005.
- [138] Y. Meng, M. Dunham, F. Marchetti, and J. Huang. Rare event detection in a spatiotemporal environment. *IEEE International Conference on Granular Computing, 2006.*, pages 629–634, 10-12 May 2006.
- [139] Micrel Lab. <http://www-micrel.deis.unibo.it/>.
- [140] Microsoft Corporation. Microsoft surface. <http://www.microsoft.com/SURFACE/index.html>.
- [141] K. Ming Hsiao, G. West, S. Vedatesh, and K. M. Online context recognition in multisensor system using dynamic time warping. In *Proc. of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 283–288, 2005.
- [142] MIT. Oxygen project overview. <http://www.oxygen.lcs.mit.edu/>.
- [143] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, 37(3):311–324, May 2007.
- [144] A. Mittal and L. Davis. Unified multi-camera detection and tracking using region-matching. In *WOMOT '01: Proceedings of the IEEE Workshop on Multi-Object Tracking (WOMOT'01)*, page 3, Washington, DC, USA, 2001. IEEE Computer Society.
- [145] M. Moghavvemi and L. C. Seng. Pyroelectric infrared sensor for intruder detection. *TENCON 2004. IEEE Region 10 Conference*, D:656–659 Vol. 4, Nov. 2004.
- [146] E. Monton, J. Hernandez, J. Blasco, T. Herve, J. Micallef, I. Grech, A. Brincat, and V. Traver. Body area network for wireless patient monitoring. *Communications, IET*, 2(2):215–222, February 2008.
- [147] S. Moon and J.-N. Hwang. Robust speech recognition based on joint model and feature space optimization of hidden markov models. *IEEE Transactions on Neural Networks.*, 8(2):194–204, Mar 1997.
- [148] C. Moser, L. Thiele, L. Benini, and D. Brunelli. Real-time scheduling with regenerative energy. In *ECRTS '06: Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 261–270, Washington, DC, USA, 2006. IEEE Computer Society.

- [149] K. Murao, T. Terada, Y. Takegawa, and S. Nishio. A context-aware system that changes sensor combinations considering energy consumption. In J. Indulska, D. J. Patterson, T. Rodden, and M. Ott, editors, *Pervasive*, volume 5013 of *Lecture Notes in Computer Science*, pages 197–212. Springer, 2008.
- [150] Murata manufacturing. Pyroelectric infrared sensors, 2005.
- [151] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel. The intel®mote platform: a bluetooth-based sensor network for industrial monitoring. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 61, Piscataway, NJ, USA, 2005. IEEE Press.
- [152] NASA. <http://sensorwebs.jpl.nasa.gov/>.
- [153] T. Nicosevici, R. Garcia, M. Carreras, and M. Villanueva. A review of sensor fusion techniques for underwater vehicle navigation. *OCEANS '04. MTS/IEEE TECHNO-OCEAN '04*, 3:1600–1605 Vol.3, Nov. 2004.
- [154] Ognjen Obucina. Ambient intelligence: The vision of information society. http://www.dke.univie.ac.at/extern/bi_ws20012002/ss2002/Ambient-Intelligence.pdf, 2002.
- [155] C. Otto, E. Jovanov, and A. Milenkovic. A wban-based system for health monitoring at home. *Medical Devices and Biosensors, 2006. 3rd IEEE/EMBS International Summer School on*, pages 20–23, Sept. 2006.
- [156] M. P. Micromachined infrared detectors based on pyroelectric thin films. *Reports on Progress in Physics*, 64:1339–1388, 2001.
- [157] J. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, 4(1):18–27, Jan.-March 2005.
- [158] V. I. Pavlović, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:677–695, 1997.
- [159] M. Perillo and W. Heinzelman. *Fundamental Algorithms and Protocols for Wireless and Mobile Networks*. CRC Hall, 2005.
- [160] Perkin Elmer. Frequency range for pyroelectric detectors. <http://www.perkinelmer.com>.
- [161] J. Perng, B. Fisher, S. Hollar, and K. Pister. Acceleration sensing glove (asg). *The Third International Symposium on Wearable Computers, 1999. Digest of Papers.*, pages 178–180, 1999.
- [162] Pesch, Dirk and Barrett, John and O'Reilly, Fergus. Ubiquitous communication through adaptive wireless systems. <http://www.irishscientist.ie/2001/01toc1.htm>, 2001.
- [163] Philips. <http://www.research.philips.com/technologies/projects/ambintel.html>.
- [164] M. M. Pradhan and R. K. Garg. Pyroelectric null detector for absolute radiometry. *Appl. Opt.*, 21(24):4456–4458, 1982.
- [165] A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi. An integrated multi-modal sensor network for video surveillance. In *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 95–102, New York, NY, USA, 2005. ACM.

- [166] D. Puccinelli and M. Haenggi. Wireless sensor networks: applications and challenges of ubiquitous sensing. *Circuits and Systems Magazine, IEEE*, 5(3):19–31, 2005.
- [167] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [168] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.
- [169] B. Radunovic and J.-Y. Le Boudec. Optimal power control, scheduling, and routing in uwb networks. *IEEE Journal on Selected Areas in Communications*, 22(7):1252–1270, Sept. 2004.
- [170] A. Rajgarhia, F. Stann, and J. Heidemann. Privacy-sensitive monitoring with a mix of IR sensors and cameras. In *Proceedings of the Second International Workshop on Sensor and Actor Network Protocols and Applications*, pages 21–29, Boston, Massachusetts, USA, August 2004.
- [171] N. S. V. Rao. On fusers that perform better than best sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), 2001.
- [172] I. Rish. An empirical study of the naive bayes classifier. In *International Joint Conference on Artificial Intelligence*, pages 41–46, 2001.
- [173] I. Rish, J. Hellerstein, and J. Thathachar. An analysis of data characteristics that affect naive bayes performance. In *ICML-01*, 2001.
- [174] K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(6):54–61, Dec. 2004.
- [175] D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information Systems*, 7:1–10, 2000.
- [176] S. Santini and K. Römer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006)*, pages 29–36, Chicago, IL, USA, June 2006.
- [177] J. Z. Sasiadek. Sensor fusion. *Annual Reviews in Control*, 26(2):203 – 228, 2002.
- [178] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.
- [179] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8:22–32, 1994.
- [180] B. Schölkopf and A. Smola. *Learning with Kernels*. The MIT Press, 2002.
- [181] A. Sekmen, M. Wilkes, and K. Kawamura. An application of passive human-robot interaction: human tracking based on attention distraction. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 32(2):248–259, Mar 2002.
- [182] <http://www.sensorsportal.com/>.
- [183] Sentilla. <http://www.sentilla.com/moteiv-transition.html>.
- [184] R. C. Shah, L. Nachman, and C.-y. Wan. On the performance of bluetooth and ieee 802.15.4 radios in a body area network. In *BodyNets '08: Proceedings of the ICST 3rd international conference on Body area networks*, pages 1–9, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

- [185] M. Shankar, J. B. Burchett, Q. Hao, B. D. Guenther, and D. J. Brady. Human-tracking systems using pyroelectric infrared detectors. *Optical Engineering*, 45(10):106401 1–10, 2006.
- [186] J. G. Sheridan, B. Short, G. Kortuem, K. Van Laerhoven, and N. Villar. Exploring cube affordance: Towards a classification of non-verbal dynamics of physical interfaces for wearable computing. In *In proceedings of the IEE Eurowearable 2003*, pages 113–118. IEE Press, 2003.
- [187] F. Simjee, D. Sharma, and P. H. Chou. Everlast: long-life, supercapacitor-operated wireless sensor node. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 315–315, New York, NY, USA, 2005. ACM.
- [188] B. Song, H. Choi, and H. S. Lee. Surveillance tracking system using passive infrared motion sensors in wireless sensor network. *Information Networking, 2008. ICOIN 2008. International Conference on*, pages 1–5, Jan. 2008.
- [189] M. Stäger, P. Lukowicz, and G. Tröster. Power and accuracy trade-offs in sound-based context recognition systems. *Pervasive and Mobile Computing*, 3:300–327, 2007.
- [190] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. pages 265–270, Nov 1995.
- [191] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.
- [192] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 292–299. ACM, 2000.
- [193] A. N. Steinberg, C. L. Bowman, and F. E. White. Revisions to the jdl data fusion model. *Sensor Fusion: Architectures, Algorithms, and Applications III*, 3719(1):430–441, 1999.
- [194] T. Stiefmeier, C. Lombriser, D. Roggen, and G. Tröster. Event-based activity tracking in work environments. In *Third International Forum on Applied Wearable Computing*, 2006.
- [195] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *10th IEEE International Symposium on Wearable Computers*, 2006.
- [196] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2):42–50, 2008.
- [197] T. Stiefmeier, D. Roggen, and G. Tröster. Fusion of string-matched templates for continuous activity recognition. In *11th IEEE International Symposium on Wearable Computers*, pages 41–44, October 2007.
- [198] I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, Nov 2001.
- [199] D. Surie, O. Laguionie, and T. Pederson. Wireless sensor networking of everyday objects in a smart home environment. *Intelligent Sensors, Sensor Networks and*

- Information Processing, 2008. ISSNIP 2008. International Conference on*, pages 189–194, Dec. 2008.
- [200] R. Szwedczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *EWSN '00: Proceedings of the 1st European Workshop on Wireless Sensor Networks*, pages 307–322, 2004.
- [201] L. Terrenghi, M. Kranz, P. Holleis, and A. Schmidt. A cube to learn: a tangible user interface for the design of a learning appliance. *Personal Ubiquitous Comput.*, 10(2-3):153–158, 2006.
- [202] C. Thompson. Smart devices and soft controllers. *Internet Computing, IEEE*, 9(1):82–85, Jan.-Feb. 2005.
- [203] C. Tsai and M. Young. Pyroelectric infrared sensor-based thermometer for monitoring indoor objects. *Review of Scientific Instruments*, 74:5267–5273, Dec. 2003.
- [204] University of Bologna. <http://www.unibo.it/>.
- [205] University of Modena and Reggio Emilia. <http://www.unimore.it/>.
- [206] O. Urfaliglu, E. Soyer, B. Toreyin, and A. Cetin. Pir-sensor based human motion event classification. *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, pages 1–4, April 2008.
- [207] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings -*, 152(2):192–204, April 2005.
- [208] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM Press.
- [209] K. Van Laerhoven and H.-W. Gellersen. Spine versus porcupine: a study in distributed wearable activity recognition. In *Proceedings of Eighth International Symposium on Wearable Computers (ISWC)*, pages 142–149, 2004.
- [210] V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.
- [211] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [212] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [213] M. Vieira, J. Coelho, C.N., J. da Silva, D.C., and J. da Mata. Survey on wireless sensor network devices. *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, 1:537–544 vol.1, Sept. 2003.
- [214] M. A. M. Vieira, A. B. da Cunha, and D. C. J. da Silva. Designing wireless sensor nodes. In *SAMOS*, pages 99–108, 2006.
- [215] C. M. Vigorito, D. Ganesan, and A. G. Barto. Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07.*, pages 21–30, 18-21 June 2007.
- [216] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part i and ii. *Proceedings of the IEEE*, 85(1):54–79, 1997.
- [217] D. Wang. A power-balancing scheme for clustered wireless sensor networks. *Future generation communication and networking (fgcn 2007)*, 1:231–236, Dec. 2007.

- [218] T. Wang, Y. Han, P. Varshney, and P. Chen. Distributed fault-tolerant classification in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 3(4):724–734, 2005.
- [219] J. Ward, P. Lukowicz, G. Tröster, and T. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, 2006.
- [220] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.
- [221] R. Watson. A survey of gesture recognition techniques. Technical Report TCD-CS-93-11, Department of Computer Science, Trinity College, Dublin, 1993.
- [222] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, September 1991.
- [223] M. Weiser. Hot Topics: Ubiquitous Computing. *IEEE Computer*, Oct. 1993.
- [224] M. Weiser, R. Gold, and J. S. Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Syst. J.*, 38(4):693–696, 1999.
- [225] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.
- [226] C. Weyrich. Orientations for workprogramme 2000 and beyond. Technical report, IST Advisory Group, September 1999.
- [227] R. Whatmore. Pyroelectric devices and materials. *Reports on Progress in Physics*, 49(12):1335–1386, 1986.
- [228] F. White. A model for data fusion. In *Proc. 1st National Symposium on Sensor Fusion*, 1988.
- [229] J. S. Wilson. *Sensor Technology Handbook*. Elsevier, 2005.
- [230] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.
- [231] K. Woods, J. W. P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):405–410, 1997.
- [232] D. Wright, S. Gutwirth, M. Friedewald, E. Vildjiounaite, and Y. Punic. *Safeguards in a World of Ambient Intelligence*. Springer, 2008.
- [233] L. Xu, A. Krzyzak, and C. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435, May/June 1992.
- [234] Q. Yang, S. J. Pan, and V. W. Zheng. Estimating location using wi-fi. 23(1):8–13, 2008.
- [235] Z. Yao, X. Yang, and S. Zhu. Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks. In *Proc. of 6th Int'l Conf on EMMCVPR*, Aug. 2007.
- [236] P. Zappi, E. Farella, and L. Benini. A pir based wireless sensor node prototype for surveillance applications. In *3th European Conference on Wireless Sensor Networks (EWSN)*, February 2006.
- [237] P. Zappi, E. Farella, and L. Benini. Pyroelectric infrared sensors based distance estimation. *to be published in Sensor 2008*, 2008.

- [238] P. Zappi, E. Farella, and L. Benini. Enhancing the spatial resolution of presence detection in a pir based wireless surveillance network. *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 295–300, Sept. 2007.
- [239] P. Zappi, C. Lombriser, E. Farella, D. Roggen, L. Benini, and G. Tröster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *5th European Conference on Wireless Sensor Networks (EWSN)*, pages 17–33, January 2008.
- [240] P. Zappi, C. Lombriser, E. Farella, D. Roggen, L. Benini, and G. Tröster. Experiences with experiments in ambient intelligence environments. In *To appear in Proc. of IADIS International Conference on Wireless Applications and Computing (WAC), 2009*, June 2009.
- [241] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach (The Morgan Kaufmann Series in Networking)*. Morgan Kaufmann, July 2004.
- [242] X. Zhu, S. Ying, and L. Ling. Multimedia sensor networks design for smart home surveillance. *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pages 431–435, July 2008.
- [243] Zigbee Alliance. <http://www.zigbee.org/>.