Chapter 1

AMBIENT INTELLIGENCE

A Gentle Introduction

P.Remagnino[1], H.Hagras[2], N.Monekosso[1], S.Velastin[1]

[1] *Digital Imaging Research Centre (DIRC), Kingston University, UK*
{p.remagnino, n.monekosso,s.velastin}@kingston.ac.uk

[2] *Department of Computer Science, University of Essex, UK*
hani@essex.ac.uk
Keywords: Ambient intelligence, people detection, machine learning, smart environments.

1. Introduction

This introductory chapter describes Ambient Intelligence *(AmI)* from the perspectives of researchers working in the field of Artificial Intelligence and Computer Vision. It is for the reader to get acquainted with some of the ideas that will be explored in greater detail in the following chapters.

Ambient Intelligence is a term that was introduced by the European community (see [9, 12]) to identify a paradigm to equip environments with advanced technology and computing to create an ergonomic space for the occupant user. Here the term ergonomic is used in a broad sense, encompassing both better living environment, secure space, but also an active, almost *living* space around us; capable of aiding us with daily chores and professional duties. Later on in the book the reader will be able to see examples of enhanced homes for the elderly, intelligent buildings, devices built for education and entertainment and conventional visual surveillance systems, easily portable to other domains of application, such as the training of professionals.

The AmI paradigm can be realised only through a number of technologies, all involving modern computing hardware and software. In particular, an AmI system requires the use of distributed sensors and actuators to create a pervasive technological layer, able to interact transparently with a user, either passively by observing and trying to interpret what the user actions and intentions are, but also actively, by learning the preferences of the user and adapting the system parameters (applied to sensors and actuators, for instance) to improve the quality of life and work of the occupant.

It must be born in mind that the AmI paradigm is not restricted to any type of environment. The idea of an *augmented* space surrounding a user could be an open or a close environment, constrained in a physical location, or spread across a large space. The most important concept is that the pervasive network is able to track the user preferences through space and time, improving the human-machine *relationship* that, in the AmI paradigm becomes very much anthropomorphic.

Section 2 presents the Essex approach to AmI, Section 3 introduces motion detection as a technique commonly used in Computer Vision and Video Surveillance and describes how it could be used as a presence detector. More advanced Computer Vision techniques will be introduced in later chapters, all aimed at identification, classification and tracking of individuals in a more or less cluttered scene. Concluding remarks appear in Section 4.

2. The Essex approach

The Department of Computer Science at Essex University carries out research in the field of AmI. Their approach is focused on the implementation of AmI as indoors smart environments. In particular, state of the art Artificial Intelligence techniques are employed in the implementation of a futuristic *Intelligent Dormitory* (iDorm). The following sections will describe their approach and some of the employed technology. The main idea here is to illustrate a concrete example of AmI put into practice with success. Later on in this chapter the reader will be able to understand how Computer Vision could enhance the iDorm and typical AmI *enabled* smart environments.

2.1 The iDorm - A Testbed for Ubiquitous Computing and Ambient Intelligence

The Essex intelligent Dormitory (iDorm), pictured in Figure 1.1 (left), is a demonstrator and test-bed for Ambient Intelligence and ubiquitous computing environments. Being an intelligent dormitory it is a multi-use space (i.e. contains areas with differing activities such as sleeping, working, entertaining, etc) and can be compared in function to other living or work environments, such as a one-room apartment for elderly or disabled people, an intelligent hotel room or an office. The iDorm contains the normal mix of furniture found in a study/bedroom allowing the user to live comfortably. The furniture (most of which are fitted with embedded sensors that provide data to the network for further processing) includes a bed, a work desk, a bedside cabinet, a wardrobe and a PC-based work and multimedia entertainment system. The PC contains most office type programs to support work and the entertainment support includes audio entertainment (e.g. playing music CDs and radio stations using Dolby 5.1 surround sound) as well as video services (e.g. television, DVDs, etc). In order to make the iDorm as responsive as possible to the needs of the occupant it is fitted with an array of embedded sensors (e.g. temperature, occupancy, humidity, light level sensors, etc) and effectors (e.g. door actuators, heaters, blinds, etc). Amongst the many interfaces, we have produced a virtual reality system (VRML) shown in Figure 1.1 (right) that marries the Virtual Reality Modeling Language with a Java interface controlling the iDorm. It provides the user with a visualization tool showing the current state of the iDorm and allows direct control of the various effectors in the room. Although the iDorm

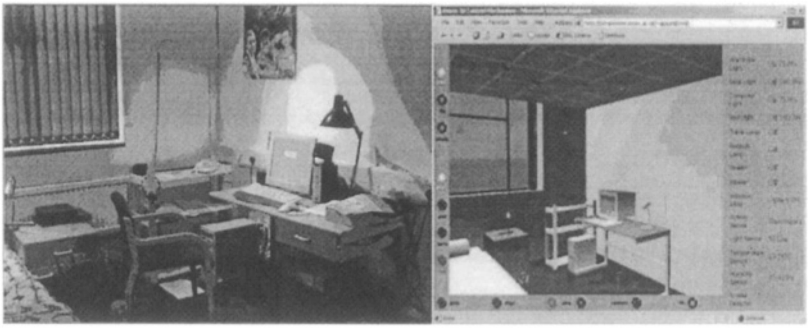

Figure 1.1. left) Photo of iDorm. right) The iDorm VRML interface.

looks like any other room, above the ceiling and behind the walls hide a multitude of different networks and networked embedded devices. In building the iDorm, we have installed devices that reside on several different types of networks. As such access to the devices needs to be managed, gateways between the different networks can be regarded as critical components in such systems, combining appropriate granularity with security. Currently the iDorm is based around three networks: Lonworks, 1-Wire (TINI) and IP, providing the diverse infrastructure present in ubiquitous computing environments and allowing the development of network independent solutions.

Lonworks is Echelon's proprietary network and encompasses a protocol for building automation. There are many commercially available sensors and actuators for this system. The physical network installed in the iDorm is the Lonworks TP/FP10 network. The gateway to the IP network is provided by Echelon's iLON 1000 web server. This allows the states and values of sensors and actuators to be read or altered via a standard web browser using HTML forms. The majority of the sensors and effectors inside the iDorm are connected via a Lonworks network.

The 1-Wire network, developed by Dallas semiconductor was designed for simple devices to be connected over short distances. It offers a wide range of commercial devices including small temperature sensors, weather stations, ID buttons and switches. The 1-Wire network is connected to a Tiny Internet Interface board (TINI board), which runs an embedded web server serving the status of the networked devices using a Java servlet. The servlet collects data from the devices on the network and responds to HTTP requests.

The IP network forms a backbone to interconnect all networks and other devices like the Multi-media PC (MMPC). The MMPC will be the main focus for work and entertainment in the iDorm. Again the MMPC uses the HTTP protocol to display its information as a web page.

The iDorm's gateway server is a practical implementation of a HTTP server acting as a gateway to each of the room's sub networks. This illustrates the concept that by using a hierarchy of gateways it would be possible to create a scalable architecture across such heterogeneous networks in ubiquitous computing environments. The iDorm gateway server allows a standard interface to all of the room's sub networks by exchanging XML formatted queries with the entire principal computing components, which overcomes many of the practical problems of mixing networks. This gateway server will allow the system to operate over any standard network such as EIBus, Bluetooth and Lonworks and could readily be developed to include 'Plug N Play' allowing devices to be automatically discovered and configured using intelligent mechanisms. In addition, it is clear such a gateway is an ideal point to implement security and data mining associated with the sub network. Figure 1.2 shows a logical network infrastructure in the iDorm.

2.2 The iDorm Embedded Computational Artifacts

The iDorm has three types of embedded computational artifacts connected to the network infrastructure. Some of these devices contain agents.

The first type is a physically static computational artifact closely associated with the building. In our case this artifact contains an agent and thus is termed the iDorm Embedded Agent. The iDorm agent receives the iDorm sensor values through the network infrastructure and contains intelligent learning mechanisms to learn the user's behaviour and compute the appropriate control actions and send them to iDorm effectors across the network. The iDorm Embedded Agent is shown in Figure 1.3(left) and is based on 68000 Motorola processor with 4 Mbytes of RAM, an Ethernet network connection and runs VxWorks Real Time Operating System (RTOS). The sensors and actuators available to the iDorm Agent are as follows: The agent accesses eleven environmental parameters (some, such as entertainment, being parameters on multi-function appliances):

- Time of the day measured by a clock connected to the 1 - Wire network

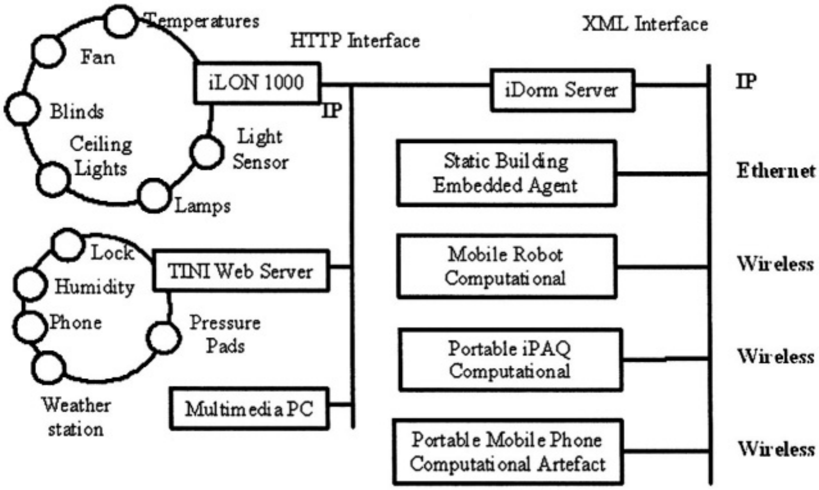

Figure 1.2. The logical network infrastructure in the iDorm.

- Inside room light level measured by indoor light sensor connected to the Lonworks network
- Outside outdoor lighting level measured by an external weather station connected to the 1-Wire network
- Inside room temperature measured by sensors connected to the Lonworks and the 1-Wire networks
- Outside outdoor room temperature measured by external weather station connected to the 1- wire network
- Whether the user is using his audio entertainment on the computer - sensed by custom code publishing the activity on the IP network
- Whether the user is lying or sitting on the bed or not, measured by pressure pads connected to the 1-Wire network
- Whether the user is sitting on the desk chair or not, measured by a pressure pad connected via a low power wireless connection to the 1-Wire network
- Whether the window is opened or closed measured by a reed switch connected to the 1-Wire network

- Whether the user is working or not, sensed by custom code publishing the activity on the IP network
- Whether the user is using video entertainment on the computer - either a TV program (via WinTV) or a DVD using the Winamp program sensed by custom code publishing the activity on the IP network

The agent controls nine effectors, which are attached to the Lonworks network:

- Fan Heater
- Fan Cooler
- A dimmable spot light above the Door
- A dimmable spot light above the Wardrobe
- A dimmable spot light above the Computer
- A dimmable spot light above the Bed
- A Desk Lamp
- A Bedside Lamp
- Automatic blind status (i.e. open/closed and angle)

The room is also equipped with other sensors such as a smoke detector, a humidity sensor, activity sensors and telephone sensor (to sense whether the phone is on or off the hook) as well as a camera to be able to monitor what happens inside the iDorm. It is possible to follow (and control) the activities inside the iDorm, via a live video link over the Internet.

The second type of the embedded computational artifacts is a physically mobile computational artifact. This takes the form of a service robot and contains an agent and thus termed robotic agent. The robotic agent can learn and adapt, online, the robot navigation behaviors (which is different to iDorm embedded agent that seeks to realize ambient intelligence). The robot prototype used in the iDorm is shown in Figure 1.3(middle left). The robot can be regarded as a servant-gadget with the aim of delivering various objects of interest to the user of the iDorm such as food, drink and medicine. The mobile robotic agent has a rich set of sensors (9 ultrasound, 2 bumpers and an IR beacon receiver) and actuators (wheels). It uses 68040 Motorola processors and runs VxWorks Real Time Operating System (RTOS). The robot is equipped with essential behaviors for navigation, such as obstacle avoidance, goal-seeking and edge-following. These behaviors are combined and co-ordinated with a fuzzy coordination module so the robot can reach a desired location whilst avoiding obstacles. The robot's location is passed to and processed as an additional input by the static iDorm embedded agent that controls the iDorm. In the experimental set up we use a simplified system in which the robot can go to two locations identified by infrared beacons to pick up objects. After picking up an object the robot can deliver it to the user and then go to its charging station, which is identified by another infrared beacon. The robotic agent sends information about its location to the iDorm agent and it takes destination instructions from that agent depending on the user's previous behavior. For example the robot might have learned to go and fetch a newspaper from a specific location whenever it is delivered in the morning.

The communication between the static iDorm embedded agent and the mobile robotic agent is implemented via a wireless link. Communication is established by initiating a request from the iDorm embedded agent to the mobile robotic agent server. Once the request has been sent the server passes it to the robotic agent to carry out the task and informs the iDorm embedded agent of the robot's current status. If the task is in progress or not completely finished then the server sends a message indicating that the job is not complete. Every time the iDorm embedded agent wants to send out a new request, it waits until the previously requested job has been successfully completed.

The third type of the embedded computational artifacts is a physically portable computational artifact. Typically these take the form of wearable technology that can monitor and control the iDorm wirelessly. The handheld iPAQ shown in Figure 1.3(middle right) contains a standard Java process that can access and control the iDorm directly, this forms a type of "remote control" interface that would be particularly suitable to elderly and disabled users. Because the iPAQ supports Bluetooth wireless networking, it was possible to adjust the environment from anywhere inside and nearby outside the room. It is also possible to interact with the iDorm through mobile phones as the iDorm central server can also support the WML language. Figure 1.3(right) shows the mobile phone WAP interface which is a simple extension of the web interface. It is possible for such portable devices to contain agents but this remains one of our longer-term aims.

The learning mechanisms within the iDorm embedded agent are designed to learn behaviors relating to different individuals. In order to achieve this the embedded agent needs to be able to distinguish between users of the environment. This is achieved by using an active lock, designed and built within the University of Essex, based on Dallas Semiconductors 1-Wire protocol. Each user of the environment is given an electronic key, about the size of a penny. This is mounted onto a key fob and contains a unique identification number inside its 2-kilobyte memory. The Unique ID Number of the user is passed to the iDorm embedded agent so that it may retrieve and update previous rules learnt about that user. We have tried various learning mechanisms to realize

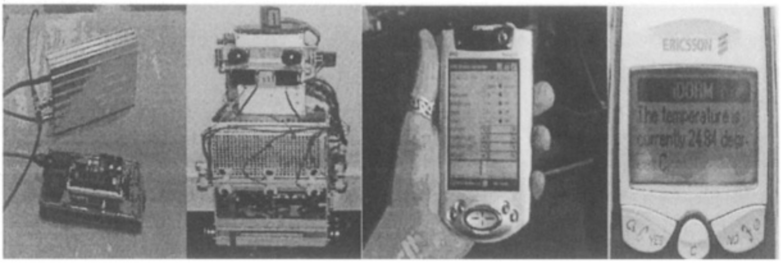

Figure 1.3. The iDorm embedded computational artefacts left) Static iDorm Embedded Agent, middle left) Mobile Service Robot, middle right) Portable iPAQ interface, right) Portable mobile phone interface.

the ambient intelligence vision in ubiquitous computing environments. More details can be found in [6], [4], [5], [2], [3].

3. Integrating Computer Vision

Computer Vision used to exist as one of the many incarnations of Artificial Intelligence [11][15]. At present and for a few decades now, Computer Vision has become an independent research field in its own right. The main goal of Computer Vision research is to interpret images and sequences of images, by learning models of stationary and moving people and objects captured by a more or less large network of sensors. Sensors employed in Computer Vision are 2D cameras. In general a Computer Vision system deploys heterogeneous networks of cameras, including fixed cameras mounted with casing or on tripods, motorized cameras (pan-tilt-zoom, also abbreviated as PTZ) and omni directional cameras, capable of a 360 degree field of view, typically using a semi-hemispherical mirror lens.

Describing Computer Vision is not possible in a Chapter, the interested reader is referred to [13][1] as textbooks on the subject. The aim of this chapter is to illustrate one of the many techniques and describe how it can be integrated in an AmI system. Vision is what we would call a *fragile* technology, because, even though it has been around for quite sometime its results are mainly at academic level and only a small portion of Computer Vision techniques can be deemed sufficiently robust to be marketed and employed in real-life applications robustly and reliably. Typically, what is usually called low level vision - sometimes called image processing - is more robust and it forms the springboard for camera based machine vision applications. The next section illustrates a well known technique used to detect moving objects in the scene. The idea is very simple, a visual process learns the background of the scene - that is whatever is stationary

- and a statistical model is built. Moving objects in the scene - represented by those pixel not conforming to the statistical model - are then extracted and subsequently merged to form $2D$ models of objects and people. Such models are then usually further analyzed, extracting their main characteristics, including chromatic and geometric information.

Computer Vision is essential for an AmI system: it gives the possibility to monitor an environment and report on visual information, which is commonly the most straightforward and human-like way of describing an event, a person, an object, interactions and actions of achieving robustness the scene. What is not straightforward is to make robust a suite of techniques that are intrinsically fragile. One way is by learning from raw data and information, rather than creating precompiled models of ideal objects, people or events. The technique chosen as one of the most representative in Computer Vision, does exactly this; it builds a model of what is stationary in the scene by learning from raw visual data.

3.1 User Detection

An AmI system should be able to detect the presence of a user, and machine vision processes can do this. One could imagine building the expected model of the user given the current field of view of the monitoring camera, but this would most certainly fail to address the simple problem of identifying whether someone has entered their office, they have decided to sit down or they are now at the keyboard typing away a document or coding a piece of software.

Visual detection in this chapter is performed by employing a learning rule on a pixel basis. Consider the intensity of a pixel as a stochastic process, clearly non stationary because of continuous variations in ambient light, either due to natural or artificial lighting conditions. The idea here follows one of the many background modeling techniques where for each pixel a number of modes is learnt (peaks on the pixel probability density function) in terms of a mixture of Gaussian models. In mathematical terms a pixel p_i will be modeled in terms of a set of Normal distributions $N(\mu_i, \sigma_i)$. Both μ and σ can be learnt by using the algorithm in Figure 1.4. The variable β plays the role of the learning coefficient,

At time $t+1$, for pixel p_{t+1}

$$\begin{aligned} \mu_{t+1} &= \mu_t + \beta_t * (\mu_t - p_{t+1}) \\ \sigma^2_{t+1} &= \sigma^2_t + \beta_t * (\sigma^2_t - (\mu_{t+1} - p_{t+1}) * (\mu_{t+1} - p_{t+1})) \end{aligned}$$

where $\beta_{t+1} = \beta_t - \epsilon$

Figure 1.4. Background updating equations.

usually initialized to a value close to 1 and slowly decremented. This means that at the beginning of the learning phase the current pixel values are considered greatly and their contribution slowly decays as β reaches its lower threshold, set to a small value, typically 0.2, to allow a contribution from newly read pixel values.

A user can then be detected as a compact ensemble of foreground pixels by estimating whether a newly read pixel value statistically falls within acceptable boundaries. The described method is a slight variation of the method proposed by Stauffer [16]. A pixels is classified using the Mahalanobis distance, which is tested for all modes of the pixel distribution $\frac{(\mu_t - p_t)^2}{\sigma^2} < th.$

The following series of frames (Figure 1.4 and 1.5) illustrate two examples of how Computer Vision could be used to detect the presence of a user and maintain it.

Figure 1.5. User reaching a working station: (first row) original, (second row) extracted foreground.

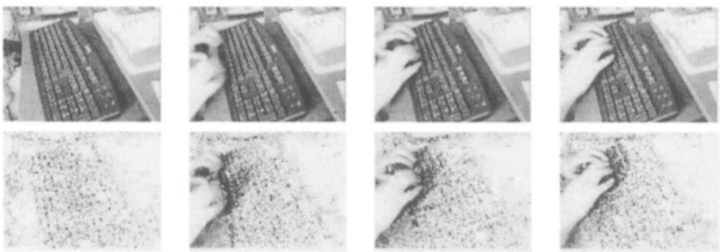

Figure 1.6. User at working station: (first row) original, (second row) extracted foreground.

Both series of frames illustrates the identification phase, which follows a training phase during which the background statistical model is built. This technique has its advantages and disadvantages. The training phase can be relatively short, no more than 100 frames are required to build a stable background model. The main problem is that the faster the background models adapts to changes in the scene, the quicker it does adapt to the presence of a person, if this person stops moving, as they do when sitting at a working station or reading a book. There are therefore two basic solutions to this problem: on one hand the visual processes could keep a log of a person having entered the environment, for instance, and then log out the person when she moves out again (detected by the motion process). On the other hand one can think of not being bothered of whether the person actually stands still and accept that he or she will in fact not move at times, counting on the fact that every once in a while the person will indeed move and be detected again. This is what happens when a person is at the keyboard.

3.2 Estimating reliability of detection

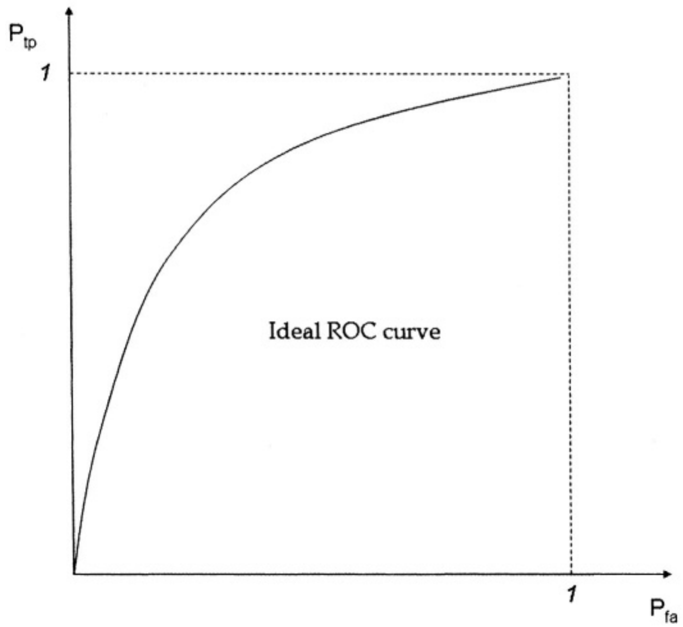

Figure 1.7. An ideal ROC curve.

In order to estimate the reliability of the presence detection process a series of experiments were run. A webcam was fixed looking down a computer keyboard, acquiring video at frame rate; a visual process was learning the model of the background and then extracting foreground pixels, used to estimate the presence. Another process, in parallel, was storing information on the keystrokes of the user. Both processes being time-stamped provided a way of comparing ground truth data - the keyboard strokes - and the estimated presence - the foreground detection. The reliability of the visual process can

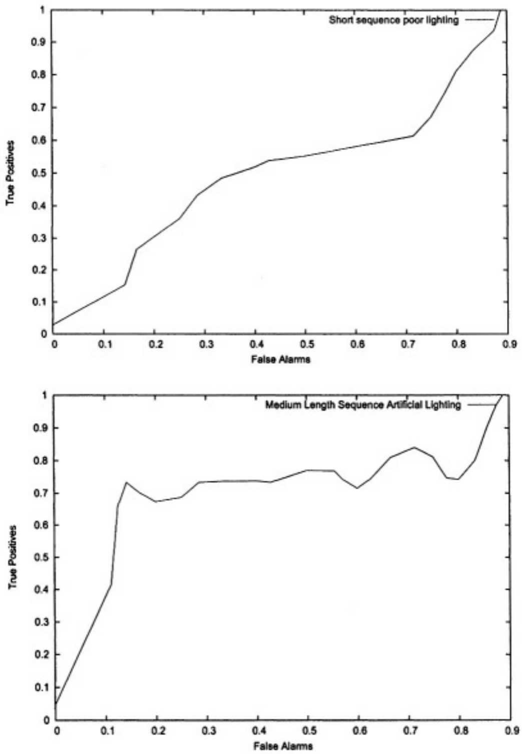

Figure 1.8. ROC curves: (top) Short experiment with poor lighting, (bottom) Medium length experiment with artificial lighting.

be demonstrated using the *receiver operating characteristic* (ROC) [14]. A ROC curve plots couples of points (P_{fa}, P_{tp}), representing respectively the probability of false alarms and the probability of true positives. ROC curves have been widely used in statistics and more recently in Computer Vision, to illustrate the robustness of Vision algorithms [10]. Figure 1.7 shows an ideal ROC curve. In essence, the probability of true positives must always be high. A pair (P_{fa}, P_{tp}) is estimated by counting, over a time window the number of *true positives* (N_{tp}), *true negatives* (N_{tn}), *false positives*(N_{fp}) and *false negatives* (N_{fn}), using the following two formulas:

$$P_{fa} = 1 - \frac{N_{tn}}{N_{tn} + N_{fp}} \quad P_{tp} = \frac{N_{tp}}{N_{tp} + N_{fn}} \tag{1.1}$$

The two graphs of Figure 1.8 illustrate the ROC curves for two experiments. Figure 1.8(top) shows the result of a short experiment run in the evening with poor lighting conditions, while Figure 1.8(bottom) shows the result of a medium sequence run with artificial lighting. By looking at the two graphs one can easily notice that the longer the experiment, the more robust the system becomes.

3.3 Vision in the iDorm

Computer Vision can be easily integrated in the iDorm. A cluster of cameras could be installed in the environment, monitoring the activity of the occupant. A part from making use of the described presence detection algorithm, it is possible to create a temporal model of the scene, keeping track of where the person has been and when storing spatial and temporal information as a Markov chain as shown in [17][8][7] and also described in Chapters 7 and 6 of the book.

4. Conclusions

This chapter has introduced AmI as a novel paradigm able to create new synergies between human and machine. An AmI system is meant to work transparently, to proactively aid the user. All chapters in this collection illustrate examples of techniques that could be employed in a fully integrated AmI environment. This Chapter has also introduced Computer Vision as an essential component for an AmI system. The specific technique of motion detection has been described as one of the most popular and useful in Computer Vision and, most likely, one of the most important for an AmI compliant system.

References

[1] D. A.Forsyth and J.Ponce. *Computer Vision A Modern Approach.* Prentice Hall, 2003.

[2] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent for ubiquitous computing environments. In *Proceedings of the IEEE International Conference on Fuzzy Systems,* 2004.

[3] F. Doctor, H.Hagras, and V. Callaghan. An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A,* 2005.

[4] H. Hagras, V. Callaghan, M. Colley, G. Clarke, and H. Duman. *Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems,* chapter Online Learning and Adaptation for Intelligent Embedded Agents Operating in Domestic Environments book, pages 293–323. Studies in Fuzziness and Soft Computing Series , Physica-Verlag, 2002.

[5] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems,* 2004.

[6] H. Hagras, M. Colley, V. Callaghan, G. Clarke, H. Duman, and A. Holmes. A fuzzy incremental synchronous learning technique for embedded agents learning and control in intelligent inhabited environments. In *Proceedings of the IEEE International Conference on Fuzzy systems,* pages 139–145, 2002.

[7] M.Brand. An entropic estimator for structure discovery. Technical report, MERL - A Mitsubishi Electric Research Laboratory, 1998.

[8] M.Brand, N.Oliver, and A.Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of Computer Vision and Pattern Recognition conference,* pages 994–999, 1997.

[9] N.Shadbolt. Ambient intelligence. *IEEE Intelligent Systems,* pages 2–3, 2003.

[10] F. Oberti, A. Teschioni, and C. S. Regazzoni. Roc curves for performance evaluation of video sequences processing systems for surveillance applications. In *Proceedings of the International on Image Processing,* pages 949–953, 1999.

[11] P.Winston. *Artificial Intelligence.* Addison-Wesley, 1992.

[12] G. Riva, P. Loreti, M. Lunghi, F. Vatalaro, and F. Davide. *Being There: Concepts, effects and measurement of user presence in synthetic environments,* chapter Presence 2010: The Emergence of Ambient Intelligence, pages 59–82. IOS Press, 2003.

[13] R.Jain, R.Kasturi, and Brian G. Schunck. *Machine Vision.* McGraw-Hill, 1995.

[14] R.O.Duda, P.E.Hart, and D.G.Stork. *Pattern Classification.* Wiley, 2001.

[15] S.Russell and P.Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, 2003.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition,* pages 246–252, 1999.

[17] V.Kettnaker and M.Brand. Minimum-entropy models of scene activity. In *Proceedings of Computer Vision and Pattern Recognition conference,* pages 281–286, 1999.