ROUTINE LEARNING: FROM REACTIVE TO PROACTIVE ENVIRONMENTS

SUSANNA PIRTTIKANGAS

Faculty of Technology,
Department of Electrical and
Information Engineering,
University of Oulu

OULU 2004

SUSANNA PIRTTIKANGAS

ROUTINE LEARNING: FROM REACTIVE TO PROACTIVE ENVIRONMENTS

Academic Dissertation to be presented with the assent of the Faculty of Technology, University of Oulu, for public discussion in Auditorium TS 101, Linnanmaa, on November 19th, 2004, at 12 noon

OULUN YLIOPISTO, OULU 2004

Supervised by
Professor Juha Röning

Reviewed by
Doctor Pasi Koikkalainen
Professor Jouko Lampinen

ISBN 951-42-7564-0 (nid.)
ISBN 951-42-7565-9 (PDF) http://herkules.oulu.fi/isbn9514275659/
ISSN 0355-3213 http://herkules.oulu.fi/issn03553213/

OULU UNIVERSITY PRESS
OULU 2004

Pirttikangas, Susanna, Routine Learning: from Reactive to Proactive Environments
Faculty of Technology, University of Oulu, P.O.Box 4000, FIN-90014 University of Oulu, Finland,
Department of Electrical and Information Engineering, University of Oulu, P.O.Box 4500,
FIN-90014 University of Oulu, Finland
2004
Oulu, Finland

Abstract

Technological development and various information services becoming common has had the effect that data from everyday situations is available. Utilizing this technology and the data it produces in an efficient manner is called context-aware or ubiquitous computing. The research includes the specifications of each application, the requirements of the communication systems, issues of privacy, and human - computer interaction, for example. The environment should learn from the user's behaviour and communicate with the user. The communication should not be only reactive, but proactive as well.

This thesis is divided into two parts, both representing methodology for enabling intelligence in our everyday surroundings. In part one, three different applications are defined for studying context-recognition and routine learning: a health monitoring system, a context-aware health club application, and automatic device configuration in an office space.

The path for routine learning is straight forward and it is closely related to pattern recognition research. Sensory data is collected from users in various different situations, the signals are pre-processed, and the contexts recognized from this sensory data. Furthermore, routine learning is realized through association rules. The routine learning paradigm developed here can utilize already recognized contexts despite their meaning in the real world. The user makes the final decision on whether the routine is important or not, and has authority over every action of the system.

The second part of the thesis is built on experiments on identifying a person walking on a pressure-sensitive floor. Resolving the characteristics of the special sensor producing the measurements, which lies under the normal flooring, is one of the tasks of this research. The identification is tested with Hidden Markov models and Learning Vector Quantization.

The methodology developed in this thesis offers a step along the long road towards functional and calm intelligent environments.

Keywords: context-recognition, data mining, human-computer interaction, pattern recognition, pervasive computing, proactive computing, ubiquitous computing

Acknowledgments

This study has been carried out in the Intelligent Systems Group at the Department of Electrical and Information Engineering of the University of Oulu, Finland, during the years 1999-2004. I am most grateful to my supervisor Professor Juha Röning, who has given me time and the opportunity for my research.

My collegues in the Computer Engineering and Information Processing laboratories have ensured a comfortable working place. Professor Jukka Riekki has influenced me greatly through his enthusiasm and ideas. My friends and collegues Satu Tamminen, Ulla Elsilä, Maija Metso, Professor Janne Heikkilä, and Perttu Laurinen I appreciate for the discussions about life, universe, and everything. I want to thank the personnel of the Neural Network Research Group, the co-authors, and research assistants that have participated in this research.

Professor Jouko Lampinen and Doctor Pasi Koikkalainen reviewed this thesis. I thank them for their good comments and criticism.

This work was financially supported by the Infotech Oulu Graduate School, the National Technology Agency of Finland, the Academy of Finland, the Tauno Tönning Foundation, Tekniikan edistämissäätiö, and the Oulu University Scholarship Foundation. Their support is acknowledged.

My dearest friends Jouko and Sanna have kept me sane with their, sometimes black, humour. My friends at the local go club deserve a vote of thanks too, although, I am not sure whether this particular hobby speeded up or delayed the finishing of this thesis. Furthermore, I thank all the people who have been around.

My father and mother have always loved and cared for me despite anything. I grew up with my brother Petri, sister Nina, and her companion Tuomas. The foundation for everything good in my life was built there. And my loving companion Paavo, well, who wants a boring life anyway.

Oulu, October 2004 Susanna Pirttikangas

Symbols and abbreviations

Mathematical notations

$C(X \Rightarrow Y)$	confidence of an association rule
$P(\cdot\|\cdot)$	conditional probability
$S(X)$	support value for an itemset X
$\overline{x}$	sample mean
$X \Rightarrow Y$	an association rule
$\| \cdot \|$	cardinality

Latin letters

A	transition probability matrix
a_{ij}	transition probability from state i to state j
B	observation probability distribution x
b_j	observation probability in state j
c	the nearest prototype vector
D	database of transactions
M	number of observation symbols
m_i	codebook vector i
N	number of states
O	observation sequence vector
S_j	state j
T	transaction
v_k	observation from state k
x	input variable
X, Y	itemsets
w	weight value

Greek letters

$\alpha(t)$	learning rate

λ	Hidden Markov Model
π	initial state probability matrix
σ	standard deviation
σ^2	variance
σ_a	reject threshold
Ψ	reliability estimator

Abbreviations

AD	Analog to Digital Converter
ASL	American Sign Languange
DBP	Diastolic blood pressure
ECG	Electrocardiograph
EM	Expectation-Maximization
EMFi	Electromechanical film
FIR	Finite Impulse Response
FUP	Fast Update
GPS	Global positioning system
HMM	Hidden Markov Model
ICA	Independent component analysis
kNN	k-Nearest Neighbour
LAN	local area network
LVQ	Learning vector quantization
MIT	Massachusetts Insitute of Technology
MLP	MultiLayer Perceptron
OLVQ	Optimal Learning Vector Quantization
PC	Personal computer
PCA	Principal component analysis
PDA	Personal Digital Assistant
QS	Quality of sleep
RN	Recurrent Networks
R-R	Time interval between two succesive R-waves in electrocardiograph
SBP	Systolic blood pressure
SRN	Simple Recurrent Networks
SOM	The Self-organizing Map
TDNN	Time-Delay Neural Networks

Original publications

PART I, Context-Recognition and Routine Learning

I Tamminen S., Pirttikangas S., Röning J. (2000): *The Self-Organizing Maps in Adaptive Health Monitoring.* Proc. International Joint Conference of Neural Networks (IJCNN2000), July 24-27, Como, Italy, pp. 259-266.

II Pirttikangas S., Riekki J., Kaartinen J., Miettinen J., Nissilä S., Röning J. (2001): *Genie of the Net: A New Approach for a Context-Aware Health Club.* Proc. ECML/PKDD-2001 Workshop on Ubiquitous Data Mining for Mobile and Distributed Environments, September 7, Freiburg, Germany, pp. 1-15.

III Pirttikangas S., Riekki J., Kaartinen J., Röning J. (2003): *Context-Recognition and Data Mining Methods for a Health Club Application.* Proc. ISCA 12th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE 2003), July 9-11, San Francisco, CA, USA, pp. 79-84.

IV Pirttikangas S., Riekki J., Porspakka S., Röning J. (2004): *Know Your Whereabouts.* Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04), January 19-22, San Diego, CA, USA.

V Pirttikangas S., Riekki J., Röning J. (2004): *Routine Learning: Analyzing Your Whereabouts.* Proc. International Conference on Information Technology (ITCC 2004), April 5-7, Las Vegas, NV, USA, pp. 208-212.

PART II, User Identification

VI Pirttikangas S., Suutala J., Riekki J., Röning J. (2003): *Footstep Identification from Pressure Signals Using Hidden Markov Models.* Proc. Finnish Signal Processing Symposium (FINSIG'03), May, Tampere, Finland, pp. 124-128.

VII Pirttikangas S., Suutala J., Riekki J., Röning J.(2003): *Learning Vector Quantization in Footstep Identification.* Proc. 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 2003), September 8-10, Benalmadena, Spain, pp. 413-417. IASTED, ACTA Press.

VIII Suutala J., Pirttikangas S., Riekki J., Röning J. (2004): *Reject-Optional LVQ-*

Based Two-Level Classifier to Improve Reliability in Footstep Identification. Proc. Second International Conference on Pervasive Computing (PERVASIVE 2004), April 21-23, Linz/Vienna, Austria, pp. 182-187.

Contents

1 Introduction

In the beginning of 1990, in the research on artificial intelligence, a new branch emerged as people started talking about ubiquitous computing (Weiser 1991), (Weiser 1993), context-awareness (Schilit & Theimer 1994), (Salber *et al.* 1999), (Pascoe 1999), and pervasive computing (Huang *et al.* 1999), (Mark 1999). The purpose is to create applications that react to the situation at hand, learn the user's habits, utilize other intelligent applications and help in everyday tasks. In robotics, this kind of methodology has been studied for over 20 years. Technological development has enabled machines to reach intelligent behaviour, but it is still mainly about physical rather than algorithmic development. Think about Deep Blue, the first chess machine beating the world chess champion in a six-game match. The intelligence in Deep Blue was about its enormous calculating power, database, and fast computer chips (Hsu 2002). Robots can walk human like, but this is achieved with mechanics, and not because they learn to walk through effort and failure. In context-aware computing, the aim is to utilize sensory data from the environment, the user, and the equipment to enable various services. The keywords here are mobility, communication networks, and learning. It is not impossible to build databases for covering different kinds of situations, but it would be ideal if the applications and environments would identify the situations automatically, in order to communicate with the user.

Developing a context-aware application starts by specifying the application (equipment, services to be offered etc.). Then, the data is collected and analyzed to find out what kind of contexts can be recognized, and which of them are useful in the application. The data must be preprocessed and finally, methods are implemented to recognize the selected contexts. After these stages, the contexts have to be utilized in order to provide useful services for the users. The context information is used to find interesting rules on the user's behaviour. This is called routine learning. The development is clearly an iterative process, since every step consolidates the expertise of the research field, and one might want to return to an earlier step as knowledge increases. Actually, this is the basic approach to data mining problems (Pyle 1999). Hence, this approach for developing context-aware applications might be called context-aware data mining, or, more generally, ubiquitous data mining.

In Figure 1, the process of ubiquitous data mining is described. It also presents the main themes of this thesis. The development of a context-aware application is done in several phases. The difficulties arise in the utilization of existing signal processing and modelling methods, and in the formation of a functional and useful application from different kinds of building blocks. After determining the requirements of the application, sensory data is collected (lowest box), and the sensor data is preprocessed. It has been considered that at least 90 % of the time spent in modelling problems goes on the preprocessing stage (Pyle 1999). After these stages, context recognition can be done. These recognized contexts are then employed in routine learning. In Figure 1, user identification is parallel to context recognition, since it can only be done after a preprocessing stage, and the methdology is quite similar to context recognition in many cases.

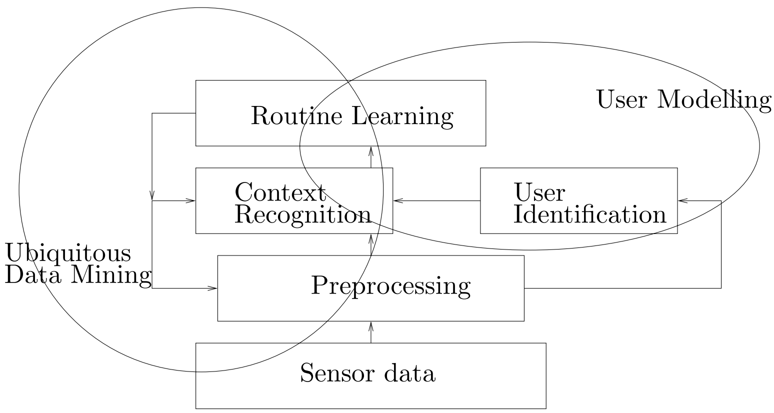

Figure 1. Building Intelligent Environments - Ubiquitous Data Mining.

The first part of the original publications introduces a path from signal preprocessing and context recognition to novel ideas on routine learning. The lowest box in Figure 1 is for sensor data. In this thesis, different kinds of sensory data have been processed, including various measurements of a human being. Physiological signals (heart rate, blood pressure, etc.) have been processed in determining the physiological context and the well being of a human. Also other measurements (location, speed, cadence) have been processed to determine the context of not just a human being, but the devices around her.

As seen in Figure 1, context recognition is needed before routine learning. Depending on the application, the method for recognizing the context must be determined. In this thesis different context recognition methods were utilized. The methods are mainly pattern recognition or statistical modelling methods, and the same problems that occur in those research fields apply here.

The behaviour of a human being has been studied widely in many research fields. In ubiquitous data mining, the characterization of human behaviour can be called routine learning (the topmost box in Figure 1). In our research, the definition of a routine has been restricted to that of a (temporal) context sequence or associating different contexts together. Therefore, routine learning is a procedure where the user is "under surveillance" - different sensory data are collected from the user and the users environment, the contexts of the user and the environment are recognized, and these contexts are exploited in determining the habits of the user.

Here, the algorithms for routine learning have initially been designed for the purposes of data mining and pattern recognition, and in this thesis, the suitability of those algorithms are tested for ubiquitous environments. Routine learning is considered as associating different contexts to each other. These associations are derived as IF-THEN rules, called association rules. The routine learning algorithm is selected so that the control of the system is with the user. The system learns the user's habits, and then communicates with the user in order to utilize the learned information.

The second part of the thesis consists of experiments on a pressure sensitive floor. In this research, the occupants of a smart living room are recognized based on their footsteps. The further development in this research can be extended to also cover routine learning. Once the user is recognized, she can be monitored, and routine learning based on her contexts can be performed. The goal is to perform all functionality calmly, so that the user will not be annoyed by the environment, but will benefit from the automatisations of the environment.

1.1 The author's contributions

In this thesis, a data driven approach for enabling proactiveness in our everyday surrounding has been performed. The work is development work aiming at useful applications. There are four applications for evaluating the difficulties arising from using distinct and very diffent kinds of sensors, most already available for ordinary users. The design of two context-aware systems, a health monitoring system and a context-aware health club were accomplished. Also, automatic device configuration, and user identification in a smart room were tested. The design procedure was to specify the application areas, collect the data, select suitable methods for each application, and implement the systems into demonstrations.

Furthermore, the concept of routine learning was proposed, and association rule algorithms were tested for routine learning. In these tests, the environment was an office space, where the users' devices were configured automatically based on the users' actions. The results show that association rules can be utilized to increase the proactiveness of the systems.

In order to enable personal profiling for a ubiquitous system, user identification was studied. Two different pattern recognition methods were utilized in determining walkers on a pressure sensitive floor, and their suitability for user identification was analysed. The final goal is to develop adaptive pattern recognition methods. In this thesis, a step towards adaptiveness was found in the form of rejection parameters.

The research shows that building proactive applications is possible in certain applications, but when human physiological signals are considered, the task is far more difficult. There are many possibilities to present the information learned from the user's behaviour for the user herself, but the main difficulties arise in verifying the learned information and in the deficiency of data. These methdologies still cannot be used without certain guidance on the user's behalf.

This thesis consists of eight publications, and they can be grouped to two different parts, as follows:

Part I consists of five publications. Publication I describes the preprocessing of several physiological sensor signals and subjective diary markings. The data consisted of over eight weeks of measurements recorded in a home environment for four test subjects. The sensory signals were syncronized, and dynamic control limits were constructed for each continuously measured signal. The preprocessing was performed by the author of this thesis. A health monitoring system was built together with S. Tamminen who utilized self-organizing maps to combine the information from the dynamic control limits. The system was developed into a demonstration, and the author of the thesis made a user interface together with S. Tamminen. This modelling device can be used for monitoring one's health. Professor J. Röning gave guidance on the utilization of neural networks and gave comments on the outline of the article.

Context-recognition was studied in publications II and III. In these articles, the idea of a context-aware health club was developed by J. Riekki, J. Kaartinen, J. Röning, and S. Pirttikangas. In publication II, the main contributions are organizing the data collection, and the preprocessing of sensor data. The author's work also included consulting different experts on sports, and utilizing this information for the application. The focus of the publication is on data mining for a context-aware application. The author designed a user interface, together with J. Riekki, in order to visualize the data collected during exercises. In publication III, the focus is on context recognition methods. The author of this thesis is responsible for context-recognition with neural networks, as well as with the method of least-squares. J. Kaartinen was also involved in planning the neural network modelling. The user interface designed in publication II was extended to cover more details on each particular exercise. Professors J. Riekki and J. Röning gave guidance on the outline of the articles.

Routine learning is introduced in publications IV and V. The author created and designed the framework, in which association rules can be utilized in a ubiquitous environment. In publication IV, data was collected in an office environment, important locations were derived from location data, and the profile information of a PDA were associated with these locations. The design was intended to be simple, since a prototype for context aware middleware architecture was to be built. The implementation of the prototype was the work by S. Porspakka. In publication V, a larger set of contexts was utilized. The author organized the data collection. Here, the constraints and problems related to adding more context variables than location and profile information were studied. The author tested the Apriori- algorithm separately from the prototype in forming interesting rules for the data sets collected.

Part II of the thesis consists of three papers, VI, VII, VIII. The author of the present thesis produced the idea of using footsteps for identifying a walker on a pressure-sensitive floor. The ideas of utilizing Hidden Markov models, and the learning vector quantization (LVQ) were also developed by the author. The author organized, and accomplished the data collection. The author, together with J. Suutala, selected the features for the study. The modelling with Hidden Markov Models and LVQ was carried out by J. Suutala. In publication VIII, J. Suutala developed a method for recognizing the user, based on three successive footsteps. This paper is a natural continuation for the earlier publications. The author of this thesis contributed to the publication by adding the idea of the adaptivity of the system utilizing the rejection parameters. The outline of the publication was guided by the author.

2 Review of the literature

2.1 Context-Recognition

"The definition for context depends on the context."

The main theme in this thesis is related to utilizing pattern recognition and data mining methods in building intelligent environments and devices. The most difficult task for ubiquitous computing is to combine all the various technologies developed throughout the world. There have been different experiments trying to create a global, easy to use framework in order to handle all the information sources, networks and terminals into a seamless service entity. One of these efforts is the Context Toolkit (Dey & Abowd 2000), (Salber *et al.* 1999) developed in the Georgia Institute of Technology. The Context toolkit uses context widgets to communicate between the environment and the applications. Among other widgets, they have implemented a *Meeting widget* to identify several persons within the neighborhood of the user. The information for the toolkit is produced by, for example, speech recognition software, machine vision machinery, microfones, Active Badges (Want & Hopper 1992) etc. The sensor data is filtered and interpreted, but for more sophisticated applications, machine learning and context history have to be catered for.

Genie of the net (Riekki *et al.* 2003) uses agent technology for the communication. They introduce a term "information overload" to describe the difficulties in context-aware computing and the utilization of various services. They suggest an architecture that remembers the different services available, requests them on behalf of the user, locates the ones needed in a particular location or context, and guides the user in using them. Other architecture specifications and infrastructures for context-awareness can be found in (Román *et al.* 2002), (Tandler 2004), and (Sun *et al.* 2003), for example.

Schilit et al. (Schilit *et al.* 1994), (Adams *et al.* 1993) have studied context-aware applications and divide these applications into four different categories: *proximate selection, reconfiguration, contextual information and commands*, and *context-triggered actions.* The proximate selection concerns user interface issues and reconfiguration is for the components and connections of the system. Context-triggered actions are the most interesting part from the point of view of this thesis, since it consists of information about the user, that is, identification, event types and the location of the user. When an event is detected, the application carries out a predefined action. The problem is how to create useful actions, maintain the important, and delete futile and old information.

Classification of the data to different contexts has been studied thoroughly by Schmidt et al. (Schmidt 2000), (Laerhoven 1999). They had special equipment, a sensor box (Mäntyjärvi 2003), used also in this thesis, for data gathering. The measurement sensors include illumination, 3 dimensional acceleration, temperature, and skin conductivity. Several simple actions were performed and the data was stored and analyzed off-line. They calculated different statistics from the data and tried to find features that discriminate different situations, such as “device in the hand”, “device on the table”, and “device in a suitcase”. In their analysis, they used self-organizing maps for context-recognition, and, based on this analysis, a rule database was formed for context-recognition.

The sensor box has been utilized for determining the location of the user, where the device is in relation to the user, and what the user is doing (Tuulari 2000). Independent component analysis (ICA) and principal component analysis (PCA) have been used to both fuse the multidimensional sensor box data, and classify it to different situations (Himberg *et al.* 2001b). Other research groups have utilized neural networks and acceleration sensors for identifying different stages of walking (Aminian *et al.* 1993), (Aminian *et al.* 1995). Wavelets have also been used (Sekine *et al.* 1998) to process acceleration data.

In wearable computing, the user carries all sensor equipment on herself rather than uses environmental sensor information, but this does not exclude the utilization of both the affective state of the user and the context of the environment (Rhodes *et al.* 1999). Picard and Healey argue about the importance of labeling the situations accurately for analysis, and have tried to create physiological fingerprints for the user’s affective states (Picard & Healey 1997), (Healey & Picard. 1998). They have noticed the differences in physiological signals between different days and used Fisher linear discriminant projection (Duda & Hart 1973), and leave one out test for discriminating different states.

The most studied context is the location of the users and devices. Location sensing has been studied thoroughly in ubiquitous computing. Since there are many different kinds of problems to be solved in the ubiquitous world, it is natural that the approaches for location sensing vary in many different aspects, such as, the sensing apparatus, power requirements, infrastructure, and resolution of time and space. A survey on these aspects has been done by Intel Research (Hightower & Boriello 2001). In this thesis, these issues are not considered more deeply, but it is assumed that the location of the user is available.

2.2 Routine Learning

Routine learning has been studied in the framework of building intelligent or smart homes. It would be convenient if our homes would know our routines, predict what we are about to do, and do our tasks automatically. But, one must be very careful in offering services for the user, since everybody knows how annoying a false reaction from a machine is. The Georgia Institute of Technology is building an Aware Home (Abowd *et al.* 2000), (Sanders 2000), Brunel University is working on a Millenium home (Millenium 2004), and MIT is developing an intelligent office environment in Oxygen (Oxygen 2004), (Brown 2001). In these settings, the utilization of speech recognition, image processing techniques, and signal processing make the environments intelligent. Routine learning is by far not the most interesting area of research in these projects, but rather the communication between the tenant and the environment. All in all, a great deal of research is going on, but fully functional systems are still unavailable.

The Aware Home uses computer vision and audio processing techniques to locate and identify the environment's occupants, and to recognize their activities. They also have a Smart Floor (Orr & Abowd 2000) containing ten strategically sized and located force-sensitive load tiles throughout the Aware Home to gather footstep data from the occupants. They identify users on the floor with two techniques: Hidden Markov Models (HMMs) (Rabiner 1989), and simple feature-vector averaging. Identification with Hidden Markov models succeeded with accuracy of 91 %, and with feature-vector averaging with 93 %.

The Millennium Home uses a variety of environmental sensors, connected to a computer, to monitor the state and activity of elderly people. Typically, they detect occupancy of the bed and other furniture, use of the lavatory, state of locks and doors, gas and water taps, movement in the rooms, doors, and bed temperature. If the software detects a potentially dangerous situation, the system will first attempt to remove the danger by speaking to the tenant. If this negotiation fails, it will call for volunteer support by telephone.

The University of Washington is trying to develop methods that support and enhance the independence and quality of Alzheimer's patients in Assisted Cognition project (Kautz *et al.* 2002). The assisted cognition systems are supposed to sense characteristics from the patient's location and environment, relying on GPS, active badges, motion detectors, and other ubiquitous infrastructure. Moments of distress, disorientation, or confusion are interpreted from the patterns of everyday behavior, using techniques from state estimation, plan recognition (Kautz 1991), and machine learning. The first step is their prototype for a wireless handheld device, the Activity Compass (Patterson *et al.* 2002), which learns a model of the traveler's current mode and transportation as well as his likely route from a GPS sensor stream (Patterson *et al.* 2003). The functionality for the activity compass is based on a Bayesian model (Druzdzel & van der Gaag 2000). A further goal is to memorize the Alzheimer patient's daily routines and to offer him directions when he gets lost or becomes confused. They state that programming the Activity Compass may take up to five years.

Routine learning can be associated to wearable computing also, and to the work on making computers that understand people (Pentland 1995), (Pentland & Liu 1995). Researchers at MIT are not only trying to recognize faces, sounds, and gestures, but they try to integrate these basic perceptual functions with higher-level models of human behavior, in order to understand what the person is doing.

They have modeled the human as a Markov device with a number of mental states, each with its own particular behavior, and inter-state transition probabilities. The internal states can be actions or intentions, and the input for these models are hand or leg movement, for example. They have built two systems for interpreting human action. The first system reads American Sign Language (ASL) using hand measurements provided by Pfinder (hand position, orientation, and width/height ratio). This real-time system performed near perfect classification of a forty-word subset of ASL. The second system interpreted people's actions while driving a car (e.g., passing, turning, stopping, car following, lane change, or speeding up), and it identified 86 % of actions within 0.5 seconds of the beginning an action, and 97 % within 2 seconds.

Microsoft has studied user's routines while using a PC (Horvitz *et al.* 1998). They also utilize Bayesian networks to automate interfaces based on user modelling in Lumiere, which includes a model of user activity and needs, and continuous online modelling of user actions to infer butlering tasks. The origin of Microsoft research is in inferring the intention or leading goal of a pilot through interpreting available information about flight status, airplane configuration, and pilot activity (Cooper *et al.* 1998). The first application from Microsoft in routine learning was the Office '97 product suite, containing the Office Assistant. The Office Assistant has thousands of user goals connected in Bayesian models, and the assistant captures information about the current view and document. However, no profile information or events over time are considered in these models.

An interesting application in routine or rather location modelling is MIT's comMotion (Marmasse 1999), (Marmasse & Schmandt 2000). This location model is for a set of learned places and destinations, which coincide to a latitude and a longitude (from GPS), that the user has categorized. In the system, a location learning agent monitors the user's travel patterns and learns her frequented locations. It is considered that if a user frequents a location often enough, it must be of some importance in her life. Once a new location has been identified, the user is prompted to name it (such as "work", "home").

The system takes time to learn locations which are not often frequented. Therefore the user can always actively teach the system a location by pressing a button and naming the place. Shadowing of the GPS signal by tall buildings is a problem, and they consider modelling these dead zones in order to predict them. Marmasse and Schmandt also tried to classify different routes in order to predict where the user is going and estimate time to destination. They tested Bayes Classifiers, Histogram modelling, and Hidden Markov Models for route tracking. Histogram modelling was considered to be the best technique for the task, although they state that if there was more data available, the HMM would probably be more suitable.

During the development of comMotion, several observations have been made. The system incrementally learns the user's frequented places, so initial configuration is not needed. As the routines change, the system will adapt and incorporate new places. A location is inferred as a building if the GPS signal is lost, and later reappears within the same radius. After the identification of a building, the user gives a name to the location. The trajectory between labeled destinations is defined as a route (made up of latitude, longitude, and time). Problems arise when separate locations are close to one another (different stores in a mall), or when the area is very large (the mall parking lot, for example). The system also takes time to learn locations which are not often frequented.

A fairly new term describing routine learning, is *rhythm modelling* (Begole *et al.* 2003). In their research, Begole et al. process data from computers, email and telephone activity, presence sensors, online calendars, and other sources. They developed a model for temporal patterns for the user activity. The model can be augmented by the user's own perceptions of their rhythms. They identify candidate transitions in the activity data by simple thresholding, they compare these candidates based on the start, end and duration times, and refine the estimates based on the variances of these three properties. They also describe the visualization of these activity levels for the user.

Context history has been studied in (Byun & Cheverst 2001), (Byun & Cheverst 2002), and (Byun & Cheverst 2003). In their research, Byun *et al.* suggest that intelligent, proactive behaviour of systems can be achieved with noticing patterns from the user's context history, and utilizing machine learning algorithms for this data. This kind of thinking is very close to that of this thesis. They utilize decision tree algorithms for inducing the rules from the context history. An advantage of using decision trees for routine learning is that numerical context values (such as 26 degrees) can be converted to symbolic ones (e.g. hot). They advocate the fuzzy representation of contexts (Mäntyjärvi & Seppänen 2003) to avoid too sensitive reactions from the system. Naturally, if the context history is small, then there is less certainty and coverage in the rules, and the computational cost for learning is also small.

Work on developing methods for routine learning is going on, but still there are several issues to be solved. The path is straight forward, signal preprocessing and context recognition methods are needed before routine learning can be performed. Different sensory equipment set different requirements for context recognition, and one cannot at the moment build a global context recognizing artificial intelligence. After routine learning, the level of uncertainty of the rules and the adaptations of the system must be determined. It is also important to notify the user of these uncertainties, or only suggest actions and let the user choose her own configuration. Everybody wants to turn off the Office Assistant when they start using Microsoft programs, although there is a clear need for an assistant like that. To make an unannoying assistant or environment, the system must not interfere, but help the user in her tasks. This requires a lot of training time for the models, but this affects another problem. Nobody wants to train a computer for five years and it is not even possible, but we want to gain results from the machine fast.

In the above studies, the simplest models seem to be the most beneficial, since the training time is short, and the methods work even when there is very little data available. The most popular methods nowadays are naturally probabilistic ones, such as Bayesian methods or Hidden Markov models. However, this is natural since it is only possible to try to determine *if* the user is going to do something, not to know for sure. In the future, people will get their Personal Assistants when they are born, and it will not matter if the machine learns in the background and reports this knowledge to the user after a long training period.

2.3 Techniques for Context Recognition

In this thesis, several different kinds of data sets have been processed in order to perform context recognition. Human physiological data (publication I) is very different from the data collected with simulation studies (publication V). However, if a researcher collects the data, this is different compared to data collected by an ordinary person (testee). In the next section, a short review of the methods used for signal preprocessing is presented. The purpose in not to copy signal processing textbooks, but to offer a viewpoint to the subject from the context aware research. Furthermore, as seen in section 2.1, neural networks have been applied to many different applications in context recognition research. Neural networks have been applied in this thesis also, and a glance at time dependent neural networks is given in section 2.3.2.

2.3.1 Signal Preprocessing

As in other pattern recognition and machine learning problems, the data sets must be preprocessed in order to effectively model the different contexts. In context recognition research, this subject has been somewhat vaguely described and it is assumed to be a natural part of the systems. However, preprocessing and selecting the best features is the most important part of any modelling problem. In addition to the selected context recognition (modelling) techniques, the various information sources, such as acceleration signals, illumination sensors, location sensors, active badges, cameras, mobile phones etc. set the requirements for preprocessing.

Modelling sensory data in ubiquitous environments requires the handling of a continuous flow of incoming data. There have been attempts to segment signals online for mobile applications in (Himberg *et al.* 2001a), and (Kargupta *et al.* 2002). Segmenting events from data streams has been studied in (Koho *et al.* 2004). As in this thesis also, offline segmentation and analysis have been used in most applications, however.

A well-known term to describe a feature, specifically in context-aware research, is a *cue* (Laerhoven & Cakmakci 2000), (Laerhoven 1999). A cue is derived from sensor information to enable context recognition. This is called feature extraction in pattern recognition research. For example, when processing a 3-dimensional acceleration signal, the context cues can be standard deviations or frequency domain features from the Fourier transform (Laerhoven & Cakmakci 2000) or wavelets (Sekine *et al.* 1998), (Mäntyjärvi *et al.* 2001). Chosen a context modelled with an illumination board, the context cue can be the level of signal or frequency. The cues from a camera or a microphone can be moments for luminance and chrominance channels, volume measurements, and densities (Clarkson *et al.* 2000) or histograms (Aoki *et al.* 1999).

In feature selection, it must be determined what subset of the features derived from the sensor data is best for successful modelling. The features selected need to contain information about the common as well as the discriminative properties of the different patterns. Usually, some systematic procedure is used for searching the feature candidates. A common way is to derive a large number of features and test subsets by classifying them to find the optimal set of features using a selected quality factor. It would, however, be time consuming and expensive to try all the feature combinations. Therefore it is rational to reduce the search space. The branch-and-bound (Jain *et al.* 2000) finds an optimal subset of features, but it may also be expensive due to its tree-like search. Forward-backward- search (Fukunaga 1990) is a method that reduces the search space in an efficient manner. Single features are tested one by one, and the one giving the best classification result is selected to the subset (forward-phase). When two features have been selected, all features are used for classification, and one feature selected in the forward-phase giving a poorer result is removed from the feature subset (backward-phase). The procedure continues by adding two features to the subset and removing one at each phase.

Throughout the thesis, a commonly used method before modelling human physiological contexts or environmental contexts is scaling (Bishop 1995). Scaling is a linear transformation which helps in providing fairly similar (in magnitude) values for the features. Some estimation and modelling techniques are not scale invariant, so the features need to be set commensurable. The most common method for scaling is to calculate the feature average $\overline{x_k}$ and variance σ_k^2 in the training set, and substract the average from each feature value and divide the difference with the variance. After normalization, the feature values have a zero mean. Other feature scaling techniques (to scales $[0, 1]$, $[-1, -1]$) can be found in (Theodoridis & Koutroumbas 1998).

2.3.2 Neural Networks

Neural networks are familiar tools in context-aware computing, as was mentioned in section 2.1. In this area, what has been studied less is the development of time dependent modelling. It has been considered that the contexts can be segmented from the data stream, and classification of the context can be performed offline with different models. Hidden Markov models provide a means to time dependent modelling, but also neural networks can be used. In this section, a short review of neural networks for modelling time dependent stationary data is presented.

Multilayer neural networks (Haykin 1994) can be used for modelling stationary time dependent data. In this case, the input $x(n)$ is the previous values for the time series $x(n) = [x(n-1), x(n-2), ..., x(n-p)]^T$, where p is the length of history that is considered significant, n is the number of iterations, and T is the transpose. If the time series data is non stationary, an ordinary multilayer neural network will fail in the modelling due to catastrophic interference (Sharkey & Sharkey 1995). Furthermore, trends in a time series will cause problems, because the network will try to learn this trend information. This also leads to failure, since the network will forget about other important variability in the series, although it is quite probable that there is important information in the seasonal trends. In this thesis, multilayer perceptrons with time-delayed input were used.

Recurrent networks (RN) (Williams & Zipser 1989) have at least one feedback loop and they learn the memory depth required for modelling automatically from the training data. Real-time recurrent networks use a temporal supervised learning algorithm which tries to match the outputs of certain neurons in the processing layer to the desired values at specific instants of time. In predicting time series, a real-time recurrent network is arranged in a pipelined manner in order to reduce the computational complexity. Simple recurrent networks (SRN) deal with time-varying input by using both a copy of the previous hidden representation $(t-1)$ and the current input (t) as input to the network. The training of recurrent networks is time consuming.

The time-delay neural network (TDNN) (Waibel *et al.* 1989) bases its decision at time-instant t on the inputs at times $t-n, t-n+1, \ldots, t$, where n is the maximum time-delay. This way, every time instant up to the maximum time-delay is equally valuable for the present determination of weights (in contrast to recurrent networks where the distant past has less influence than the recent values) and it is not possible to take information into account that occurred before $(t-n)$. Temporal back-propagation enables parallel distributed processing for TDNN:s, and thus they may also be useful.

FIR multilayer perceptrons (Haykin 1996) (or FIR with adaptive weights) are suitable for dynamic system identification and modelling of non stationary time series. The network can be trained by a temporal back-propagation algorithm. Hebbian networks can also be used for time processing if delays are added as input.

The adjustment of a static neural network model to non stationary environments can be done using structurally adaptive solutions, that is by reforming the model every time a new kind of situation occurs in the system under research (Ramamurti & Ghosh 1999). Pruning and growing (Reed 1993) in the neuron, as well as on the structural level of the network are important aspects to consider in forming an operative model.

2.4 Techniques for Routine Learning

Routine is a temporal (context) sequence that occurs often or frequent association of different contexts. In this section, two possible methods for routine learning are presented. Hidden Markov models have been utilized in many routine learning experiments (section 2.2). In this thesis, they have been utilized for user identification (section 4), but they are presented here due to their vast applicability in many areas of research. In section 2.4.2, the methodology of association rules is described. In publications IV and V, the framework for association rules in routine learning is developed. In sections 4.2.2 and 3.3, the utilisation of these methods in the development work of this thesis are described.

2.4.1 Hidden Markov Models

Hidden Markov Models (HMM) (Rabiner 1989) provide a natural way of modelling time-dependent signals, and they have been used for speech recognition (Levinson *et al.* 1983), context recognition (Mäntylä *et al.* 2000), and robotics (Shatkay & Kaelbling 2002), in addition to routine learning.

In HMM-based classification, it is assumed that the observation sequence $O = O_1 O_2 \cdots O_T$ is generated by a Markov model. A Markov model is a finite-state machine which changes its state once every time unit. Each time t a state S_j is entered, an observation symbol O_t is generated from a certain observation probability distribution B. The transition probabilities between the states and the observation symbol probabilities determine the joint probability that O is generated by the model. In practice, only the observation sequence is known, while the underlying state sequence is hidden, which is why they are called Hidden Markov Models.

There are different types of HMMs specified by the possible connections between states. In an ergodic (fully connected) HMM, every state of the model can be reached from every other state. In many speech recognition applications, the left-right model (Bakis 1976), (Jelinek 1976), depicted in Figure 2, is used. In a left-right HMM, the state index increases or remains unchanged as time increases. This property leads to a natural choice of left-right models for modelling signals that change over time. Left-right HMMs are also used in this thesis.

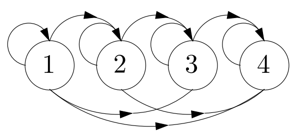

Figure 2. A 4-state left-right HMM.

An HMM is thus a probabilistic model, and it can be fully described by two model parameters (N and M), specification of observation symbols, and three probability measures, A, B, and π. Generally, the notation $\lambda = (A, B, \pi)$ is used for an HMM. It should be noted that if the modelled parameters are continuous, the observation distribution B within the HMM is continuous. In this thesis, a discrete observation symbol density distribution B is used.

The model parameter N refers to the number of states in the model; $S = \{S_1, S_2, \ldots, S_N\}$. The number of distinct observation symbols per state is M. The individual observation symbols are $V = \{v_1, v_2, \ldots, v_M\}$.

The probability distribution $A = \{a_{ij}\}$ is the state transition probability distribution:

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N,$$

where q_t is the state at time t. In a left-right HMM, the state transition coefficients have the property

$$a_{ij} = 0, \quad j < i$$

which indicates that no transitions are permitted for states whose indices are lower than the current state.

The observation symbol probability distribution in state j is $B = \{b_j(k)\}$, where

$$b_j(k) = P[v_k \text{ at } t|q_t = S_j], \qquad 1 \leq j \leq N, 1 \leq k \leq M.$$

The initial state distribution is $\pi = \{\pi_i\}$, where

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N.$$

When using a left-right HMM, the initial state probabilities have the property

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases}$$

Therefore the state sequence is determined to start in state 1. To apply HMMs in classification, the determination of the model parameters $\lambda = (A, B, \pi)$ utilizing training sequences is required. This can be done via Baum-Welch estimation (Baum 1966) (equal to the EM method (Dempster *et al.* 1977)) or by using gradient methods (Levinson *et al.* 1983). The best possible state sequence to explain the observation sequence must also be determined. This can be done via the Viterbi-algorithm (Viterbi 1967). Furthermore, the maximum likelihood $P(\mathbf{O}|\lambda)$ is calculated for each observation sequence to be classified and for each model trained. The number of Hidden Markov Models needed equals the number of known classes in the classification problem.

2.4.2 Association Rules

In this thesis, the possibilities of learning a user's routines by discovering frequent patterns is studied. It is suggested that the methodology of association rule mining can be utilized in context aware computing. The idea is to identify all combinations of items that are found in a sufficient number of examples. Association rules were first developed in analyzing market-basket data, where rules like "A customer who buys beer and sausage will also buy diapers with a probability of 0.85" can be calculated from a large amount of transaction data. The applications of association rules in a data mining framework have been in mining telecommunication alarm data (Dehaspe & Toivonen 1999), predicting chemical carcinogenicity (Dehaspe *et al.* 1998), and finding similar web pages (Broder 1997), among other things.

An association rule is an expression $X \Rightarrow Y$, where X and Y are sets of items. Given a database D of transactions, where $T \in D$ is a set of items, $X \Rightarrow Y$ expresses that whenever a transaction T contains X, then T probably contains Y also. The probability is called the *rule confidence*, and it is defined as the percentage of transactions containing Y in addition to X with regard to the overall number of transactions containing X. In other words, the rule confidence is the conditional probability $P(Y \subseteq T | X \subseteq T)$. A more formal definition is given below.

Let $I = \{x_1, \ldots, x_n\}$ be a set of distinct items (over binary domain $\{0, 1\}$). A set $X \subseteq I$ with $k = |X|$ is called a k-itemset. Let database D be a multiset of subsets of I. Each $T \in D$ is called a transaction. A transaction $T \in D$ supports an itemset $X \subseteq I$ if $X \subseteq T$ holds. An association rule is an expression $X \Rightarrow Y$, where X and Y are itemsets, and $X \cap Y = \emptyset$ holds. The fraction of transactions T supporting an itemset X with respect to database D is called the *support* of X, $S(X) = |\{T \in D | X \subseteq T\}|/|D|$. The support of a rule $X \Rightarrow Y$ is defined as $S(X \Rightarrow Y) = S(X \cup Y)$. The *confidence* of this rule is defined as $C(X \Rightarrow Y) = S(X \cup Y)/S(X)$.

The Apriori algorithm. The most common algorithm in discovering association rules is the *Apriori* algorithm developed by Agrawal et al. (Agrawal & Srikant 1995), (Hipp *et al.* 2000). Other association rule algorithms can be found in (Srikant & Agrawal 1995), (Srikant & Agrawal 1996). The basic goal is to find all combinations of items that have certain user-specified minimum support (*minsup*) and confidence (*minconf*). The combinations that have transaction support above the minimum support are called *large* itemsets, and all other combinations *small* itemsets. The large itemsets are used to generate desired rules. That is if, say $ABCD$ and AB are large itemset, then we can determine if the rule $AB \Rightarrow CD$ holds by computing the ratio $r = S(ABCD)/S(AB)$. Only if $r >$*minconf*, then the rule holds. The rule will have minimum support because $ABCD$ is large.

The support threshold is for determining the percentage occurrence of an itemset before it qualifies as a frequent pattern.

2.4.3 Discussion

The methods presented here are not the only ones that can be applied to routine learning. The Hidden Markov models have been utilized in many such applications, but other probabilistic models have also been used (section 2.2). Association rules have not been utilized in routine learning before.

Routine learning with the Apriori algorithm requires that context recognition has been done with the sensory data. Then, the context data can be associated with each other. The Apriori- algorithm is only effective when the rules of interest occur very frequently. However, it can expected that in a ubiquitous environment frequency is not the key point, but confidence is. It is more important to find the contexts that always or almost always occur together. In the case of a small database, the Apriori algorithm can be run with very low support and high confidence, and high confidence rules can be found. But if the database is very large, other algorithms have to be considered since Apriori is too slow. In (Cohen *et al.* 2000), an algorithm is developed where low support and high confidence are key issues.

Another weakness of the Apriori algorithm is that the user has to define the minimum support and confidence values. An algorithm that bypasses this property has been developed in (Udechukwu *et al.* 2004), for example. Furthermore, the whole data set has to be processed every time when new data arrives. Incremental association rules can be utilized in these situations The first algorithm for the incremental mining of association rules was FUP (Fast Update) in (Cheung *et al.* 1996). After that, many algorithms have been proposed ((Cheung *et al.* 1997), (Ayan *et al.* 1999), for example).

Association rules can also be used for the prediction of user behaviour. The data log must be viewed as a collection of events of sequential structure. The problem is to find the collections of events occurring frequently close to each other. These collections are called episodes. The algorithm for finding these frequent episodes (Mannila *et al.* 1995) is given a class of episodes, an input sequence of events, a window width, and a frequency threshold. All episodes that occur frequently enough are determined. The incremental updating of sequential patterns is a difficult problem, since the search space is extremely large. The SPADE algorithm (Pudi & Haritsa 1994) uses a database format, where each sequence is associated with a list of objects in which it occurs, along with the time stamps.

3 Experiments on Context recognition and Routine Learning

A significant part of each application is the handling of different sensory information from various sources. In Table 1, sensory equipment that has been utilized to provide the data sets for this thesis is presented. The column for publication refers to the original publications.

Table 1. Sensory equipment throughout the thesis.

Name	unit	publication
R-R interval recorder	msec	I
activity monitor	pulse/min	I
scales	kg	I
sphygmomanometer	mmHg	I
velocimeter	10*km/h	II, III
cadence meter	rounds/min	II, III
heart rate monitor	beats/min	II, III
PDA	coordinate	IV, V
sensor box	-	V
EMFi- floor	voltage	VI, VII, VIII

In this chapter, the methodology and applications on context-recognition and routine learning are described. The chapter covers the first part of the original publications.

3.1 Signal Preprocessing: Health Indicators

In publication I, the focus (from the author's part) is on signal preprocessing. A method for preprocessing human physiological signals was presented. The aim was to develop a health monitoring system for ordinary people to help in making assumptions about the effect of their lifestyles on their physiological signals and well-being. In Figure 3, the health-monitoring system built is described. The person's physical signals are measured, dynamic features are extracted and entered into a self-organizing map, which analyzes the measurement state (Tamminen *et al.* 2004).

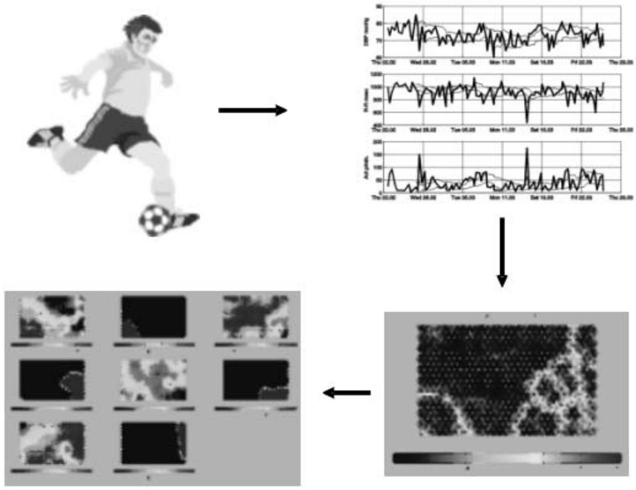

Figure 3. The process of health-monitoring.

3.1.1 Data

The data for this research was obtained during the spring of 1996. Fourteen healthy middle-aged male volunteers collected their physiological data daily for eight to ten weeks. Four of them were selected for the research. In Table 2, the data utilized in publication I is described.

The testees' R-R interval and activity were measured continuously with an R-R interval recorder and an activity monitor during the daytime. An R-R series consists of the time spans measured between two R peaks in an ECG signal. Diastolic and systolic blood pressure, along with body temperature were recorded three times a day by the subjects. The first measurements were made in the morning after waking up. The second measurements were made between 2:00 and 8:00 PM and the third in the evening before going to bed. The quality of sleep was evaluated at night. During the measurement period, all the subjects were living normal lives. The variation in measurement times was due to unsupervised self-measurements.

The subjects further filled in a diary, indicating their daily emotional states, such as fatigue, happiness, pain etc. The amounts of coffee, tea, cigarettes and alcohol consumed were also reported. The notes on the emotions during the daytime were made between 2:00-8:00 PM and those on the rest of the day before going to bed. The physical exercise and meal times were also recorded.

Table 2. The measurements for the health monitoring system.

name	description	source	unit/sampling time
R-R	R-R interval data	R-R interval and activity monitor	-/msec
ACT	activity exceeding 0.1*g	R-R interval and activity monitor	mpm/min
DBP	diastolic blood pressure	blood pressure monitor	mmHG/3 times a day
SBP	systolic blood pressure	blood pressure monitor	mmHG/3 times a day
temp	body temperature	thermometer	$\circ$C 3 times a day
w	body weight	scale	kg/once a day (morning)
QS	sleep phase:quiet sleep	biomatt	-/once per min
Wake up	wake up assesment	diary	-
Sleep	sleep assesment	diary	-
MS	mental striving	diary	-
PS	physical striving	diary	-
Manage	management	diary	-
C/T	coffee/tea cups	diary	portions
cig	cigarettes	diary	portions
alc	alcohol consumption	diary	portions

The measurement time of over eight weeks produced a very large data set. To make the signals compatible, the continuously measured variables were discretized by taking averages for one-hour spans simultaneous to the discretely measured variables. Thus, three values for each day resulted in data vectors of 170 observations or longer.

The data was analyzed using different statistical methods. This analysis and discussions with experts led to the selection of eight variables with high quality, stability and descriptivity. The others were discarded because of the long missing periods (several days or even weeks) and noisy or clearly erroneous measurements. The variables were *diastolic* and *systolic blood pressure, the mean* and *the standard deviation of R-R intervals, activity, body temperature, weight* and *quality of sleep.*

3.1.2 Dynamic Control Limits

The first step in process modelling was to define the normal and abnormal states of the system. In this thesis, a method for analyzing the measured signals of the system was developed. The dynamic control limits monitor the variability of different signals in order to recognize alarming situations. When abnormalities are detected, a new binary variable can be composed to identify hazardous observations. These binary variables are fed to the self-organizing maps. The work and analysis on self-organizing maps will be reported by S. Tamminen.

The idea is borrowed from process control, and it is a variation of the well-known 3-σ-rule (Kume 1989). When the process is measured continuously, a sliding technique can be used. If there are l variables, let x_{ij}, where $i = 1, 2, \ldots$ and $j = 1, 2, \ldots, l$, stand for the measured value at time i for the j:th variable. The limits for the value entering the system at time k ($k > 4$) are calculated from $x_{1j}, x_{2j}, \ldots, x_{(k-1)j}$. The second limits for the value at time $k+1$ are calculated from $x_{2j}, x_{3j}, \ldots, x_{kj}$ and so on. In this way, the limits are computed every time a new value enters the system and they evolve along with the signal, in other words they are dynamic. In Figure 4, the dynamic control limits for diastolic blood pressure, the minimum value of the R-R interval and activity are shown. The Figure presents the measurements for one testee, and the limits are calculated for the measurement period of one month.

In cases that are not predictable and smooth, e.g. certain industrial processes, the present value of the signal might be strongly dependent on previous values. Furthermore, if certain values of the process show better correlation than others, a weighting procedure for the signal history is a way of approaching the problem.

If w_{ij} stand for the weights, the expected mean $(\bar{x}_w)_j$ for the weighted signal is

$$(\bar{x}_w)_j = \sum_{i=1}^{k} w_{ij} x_{ij} / \sum_{i=1}^{k} w_{ij} \tag{3.1}$$

and the standard deviation $\sigma_{(x_w)_j}$

$$\sigma_{(x_w)_j} = \sqrt{\sum_{i=1}^{k} w_{ij}(x_{ij} - (\bar{x}_w)_j)^2 / \sum_{i=1}^{k} w_{ij}}. \tag{3.2}$$

The confidence limits can be formed as follows

$$UL_{(x_w)_j} \quad = \quad (\bar{x}_w)_j + T\sigma_{(x_w)_j} / \sqrt{\textstyle\sum_{i=1}^{k} w_{ij}}$$

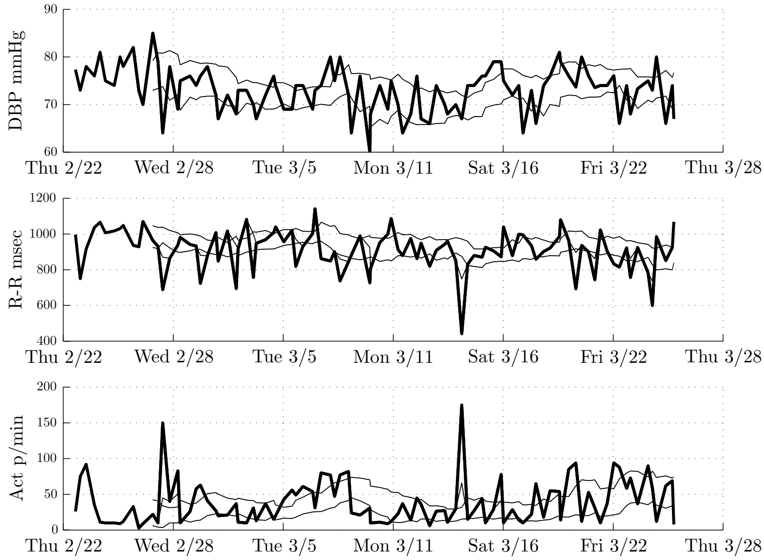

Figure 4. Class identification of diastolic blood pressure, the minimum value of R-R interval and activity.

$$LL_{(x_w)_j} \quad = \quad (\bar{x}_w)_j - T\sigma_{(x_w)_j} / \sqrt{\textstyle\sum_{i=1}^{k} w_{ij}}, \tag{3.3}$$

where T is a constant defining the width of the limits, $UL_{(x_w)_j}$ is the upper limit, and $LL_{(x_w)_j}$ is the lower limit.

If the signals contain a lot of irregularities, a suitable filtering structure is needed. The dynamic control limits are not defined based on the original signal, but on a particular flattened signal. The present value of the flattened signal is obtained from the history of the dynamic control limits by defining the percentage between the original signal and the previous upper or lower limit. If the original signal exceeds these limits by this percentage, the signal is flattened. In this way, the erroneous and irregular values of signals do not affect the adaptation of the dynamic limits too much. On the other hand, if there are known limits the signal has to conform to, this *a priori* information can be noted in defining the dynamic limits.

When the dynamic control limits are ready, there is an upper and a lower boundary value for each signal. Every time a signal is beyond these values, there might be something wrong with the system. This overdrafting will be referred to as an alarm.

An appropriate T-value and sliding history were chosen to be three and 12 respectively. This means that four days of measurements were taken into account in building the present value of the dynamic control limit and all the previous values were ignored.

The weighting procedure for DBP was defined to follow the person's diurnal blood pressure rhythm (Staessen *et al.* 1992). The weights were chosen to strengthen the previous value ($x_{(k-1)j}$) and the measurement made at the same time on the previous day ($x_{(k-3)j}$). The value between these two was also weighted ($x_{(k-2)j}$), but with a smaller amount. A similar weighting procedure was used for all the variables.

With this approach, two (0,1) indicators are established; one for a lower alarm and one for an upper alarm of the signal. There are thus two more variables for each signal, and they can be used as input for the self-organizing maps.

3.1.3 User interface

A user interface was built to illustrate the data sets. In Figure 5, the variable listing, and the different menus are shown. In the Figure, the variable listing corresponds to the variables selected in (Tamminen *et al.* 2004). The monitoring device was designed during the research, and it was implemented to a Unix environment with Matlab 5.3.1. The system calculates all the necessary features, but for monitoring, the user can select variables based on personal interests (e.g. blood pressure and weight). It is also possible for the user to select the observed time period on the screen.

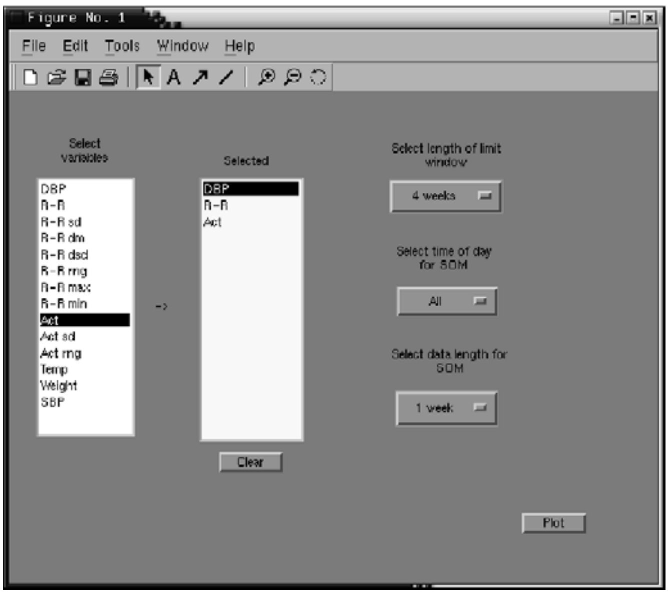

Figure 5. A user interface for the health-monitoring system.

3.1.4 Discussion

Determining one's health status with a machine is a very difficult task. The users of this kind of monitoring system can observe their measurements visually in a user interface, and get an idea how their own lifestyles affect their signal levels.

The dynamic control limits were created with several medical experts. The signal noises in question are not normally distributed, as should be assumed when using control limits. Therefore a heuristic approach was taken to formulate the limits. The signals needed to be flattened in order to avoid too much oscillation. The weighting procedure for the signals was selected to follow the dirunal blood pressure rhythm.

One can find similarities in this procedure to not only process control, but also to routine learning. The human physiological signals (heart rate, blood pressure, etc.) are processed to obtain the context of a human being. If the signal is under control, the context is considered "normal", and both the upper and lower exceeds of limits are considered "unusual". Routine learning is defined as associating different contexts, and the self-organizing maps combine (associate) all the context information to a higher level context, shown visually on a map.

3.2 Context recognition: Health Club Application

The Health Club developed as a part of this thesis, is a context-aware service concept. It was build on service architecture, Genie of the Net (Riekki *et al.* 2003), and in publications II and III, context recognition methods for the Health Club were developed. Publication II mainly considers the preprocessing of the sensory data, and in publication III, contexts are recognized.

In Health Club, the system guides the user in a cycling exercise. A typical scenario is as follows: Before exercising, the user plans at her terminal an exercise schedule and possibly also outlines more detailed instructions for each exercise. The system presents the calendar containing the exercise schedule to the user automatically in the specified context and reminds about the forthcoming exercise. Before the exercise, the user checks from her terminal what kind of exercise she has planned to perform. During the exercise, a heart rate monitor records her heart rate, cadence, and the bike's speed. After the exercise, she goes to her terminal and loads the data to the system. The system recognizes the context history of the exercise based on the collected data and other available information (e.g. the height profile of the route). The system analyzes the exercise and presents the collected data, the context history and the results of the analysis to the user. Additional information about, for example, the route cycled, enables different methods for analyzing the exercise. Sharing exercise information with other users enables comparisons. Furthermore, sharing calendars makes it possible to plan group exercises.

3.2.1 Data

The data for developing the context aware Health Club was collected in the years 2000 and 2001. A strict route was defined where three cyclists exercised. The route was 11.5 km long and consisted of about four kilometers of uphill and about four kilometers of downhill, while the rest was classified as level ground. All in all, 14 exercises were done in one direction along the route and 7 exercises in the opposite direction with two different bikes. One bike had 10 gears and the other 24 gears. Two exercises were made on level ground, where the cyclist rode for one minute per gear, going through all 24 gears of the bike.

The measurements were made with a heart rate monitor, Polar XTrainer Plus[1]. The measurements are in Table 3. The cyclist's heart rate, cadence and speed were measured during the training period with a sample frequency of 5 seconds. The cyclist pushed an interval time button at certain points of the route. An example of an exercise is presented as a time series in Figure 6.

Table 3. The measurements for the health club application.

Name	Description	unit
HR	heart rate	beats/min
Cadence	cycling frequency	rounds/min
Speed	speed of the bike	10*km/h

The most important contexts of the Health Club application were divided into three parent classes. Every parent class contains child classes. The parent classes are *bike*, *cyclist* and *environment*. The environment class consists of *weather*, *route*, *time*, and *quality of air* child classes. The cyclist has contexts labeled *movement*, *physiology*, and *anatomy*. The bike contains contexts labeled *movement*, *functionality*, and *structure*.

In Table 4, the division of the different contexts studied are shown.

Table 4. Different contexts in the Health Club application.

Parent Class	Bike	Cyclist	Environment
Child class	movement	movement	weather
	functionality	physiology	route
	structure	anatomy	time

[1]This product is a trademark of Polar Electro Oy.

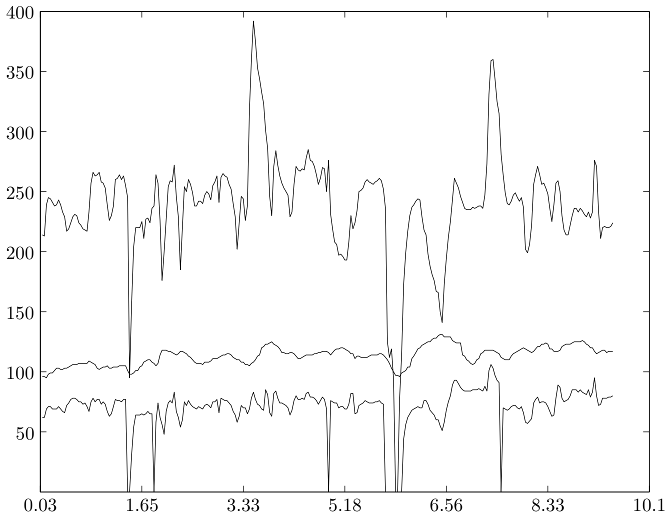

Figure 6. Data collected during a cycling exercise. The curves show, from top down: velocity, heart rate and cadence, and the corresponding units are 10*km/h, beats per minute and rounds per minute.

3.2.2 Data Preprocessing

Clearly erroneous values were removed, and the data was prepared for further analysis. All the exercises were done with a light moderate intensity zone, meaning that the heart rate of the cyclist was between 60-70 % percent of the person's maximum heart rate most of the time.

A profile and coordinates of the route were used in the classification of the measurements to different geographical attributes. It was noticed that when the distance was calculated from the speed, which had been measured only once every 5 seconds, there was a cumulative error in the location of the cyclist. Therefore the measurements had to be synchronized with the map coordinates according to the cyclist's own interval markings and the known positions on the map.

The route was classified as downhill, uphill and level ground according to the angles of the z-axes on the map in monotonous acclivity and declination.

In publication II, self-organizing maps were used as a data mining tool, and to get familiar with the data. In these experiments, the absolute values and the fuzzified values of the sensory data were used as input for the networks. Later on, the fuzzification of context sensory data has been utilized in (Mäntyjärvi & Seppänen 2003), for example.

3.2.3 Recognizing Environmental Contexts

The classification aimed to distinguish different ground declinations for each observation. The target classes were obtained in the preprocessing stage: downhill, uphill, and level ground.

Multilayer perceptrons (Haykin 1994) were used for classification. The inputs for the perceptron were the current values of the variables (speed, cadence, and heart rate), the differences of these variables, and the observations (or differences) within a certain history window. Several different lengths of history windows were tested.

The variables were normalized to a scale of [-1,1]. The data was divided into a training set and a test set. The training sets consisted six- to 12-dimensional variable vectors (depending on the length of the history window) of length 1000, and the length of the training set was 800. Several multilayer perceptrons were trained with different sizes and with different variable combinations using Levenberg-Marquardt optimization (Bishop 1995).

The results for the best experiment are shown in Table 5. With a three-layer perceptron and a history window of three observations, the results for the test set are below 50% of correct target classes. The variables did not contain enough information for adequate classification.

Table 5. Confusion matrices of classification results for training and test sets.

Training set				Test set			
	downhill	level	uphill		downhill	level	uphill
downhill	64%	16%	19%	downhill	46%	37%	18%
level	17%	54%	29%	level	21%	46%	32%
uphill	11%	16%	73%	uphill	23%	38%	39%

3.2.4 Recognizing Gear Shift

Cyclists use several different tactics, depending on their individual strength, but the use of gears differentiates between a beginner and a professional. If it is possible to detect the moment at which a gear has been shifted, the effect of different shifting tactics on the overall performance can then be analyzed.

The estimation of the points in time (and location) where the gear has been shifted was done by a recursive least-squares algorithm. One could apply a batch-mode classification algorithm, but here the goal was to recognize the gear shifts in real time. One could also mark down the cogs of the gear wheel into the application, but this is a contradiciton to the ubiquitous world. Every configuration task needed on the user's behalf requires attention and time. Recursive least squares was tested to enable automatic learning of the equipment structure based on the measurements. The method is a simple case of structurally adaptive models (Ramamurti & Ghosh 1999). The case in hand is simple and easy to understand for testing such a model in real the world.

The bike had 24 different gears, three gearwheels at the front (24, 34, and 42 cogs, respectively) and eight gearwheels at the back (11, 13, 15, 17, 20, 23, 26, and 30 cogs, respectively). A switching ratio was calculated for each gear based on the number of the cogs; the ratios are presented in Table 6. In Table 6, the gears are organized based on the magnitude of this ratio from the smallest to the largest gear. Also, the difference between consecutive gears is shown. In this case, the diameter of the front tyre is 28 inches.

Table 6. Gears in an ascending order, based on the switching ratio. G refers to the gear, and R to the ratio, respectively.

G	R	Diff	G	R	Diff	G	R	Diff	G	R	Diff
1.	0,80		17.	1,40	0,09	19.	1,83	0,13	21.	2,47	0,20
2.	0,92	0,12	5.	1,41	0,01	7.	1,85	0,02	15.	2,62	0,15
3.	1,04	0,12	11.	1,48	0,07	13.	2,00	0,15	22.	2,80	0,19
9.	1,13	0,09	6.	1,60	0,12	20.	2,10	0,10	16.	3,09	0,29
4.	1,20	0,07	18.	1,62	0,02	8.	2,18	0,08	23.	3,23	0,14
10.	1,31	0,11	12.	1,70	0,09	14.	2,27	0,09	24.	3,82	0,59

The method developed for the identification of gear shifts has the following properties: it can update its parameters on-line, it can adapt to local changes in the input-output space by modifying only a few of its parameters, and it adapts to the complexity of the situation by growing its own structure.

Every gear is represented by a line. At time $t = 2$, the first gear line $l1$ is estimated based on the first two measurements. The gear line represents the ratio between speed and cadence when using a specific gear. The parameters estimated are the slope and the intercept of the line. When new measurements (cadence, speed) arrive, the distance between this observation and the existing gear lines is calculated. If a predefined threshold is exceeded, a new gear line is estimated. In this case, it is assumed that an unknown gear is being used. If the observation is within the threshold, that is, close to an earlier estimated gear line, the parameters of this gear line are updated with a recursive least-squares algorithm. In this case, the observation adjusts the gear line into a more accurate estimate of the gear. The threshold is not always the same, but it grows as cadence and speed grow.

The method was implemented with Matlab. The estimated gear lines for one exercise are shown in Figure 7. The number of gear lines is 17, which means that not all gears could be identified. In one test exercise, the gear was shifted once a minute to a higher gear, going through all the gears. The reliability of the results could be evaluated because the actual gear used was known. The method was also tested with a 10-gear bike.

The moment a gear has been switched can be determined. This information can be used to evaluate the effect of switching tactics to overall performance. The information of the precise gear is irrelevant, since the gears can be divided into small, medium, and high gear, for example.

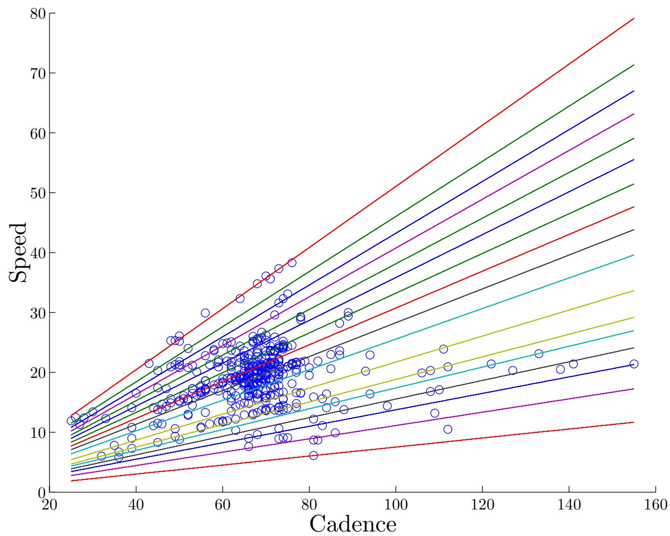

Figure 7. The estimated gear lines in an exercise.

3.2.5 Discussion

Context recognition was studied using cycling data in order to plan a more efficient exercise schedule. An analysis of the context history results in comments on the exercise, pointing out, for example, that the user started at too high a speed, or that the gear shift was improperly timed. Furthermore, the success in keeping the heart rate, cadence and speed within the intended intervals can be judged. Numerous simple tools can be offered to the user for studying the minimum and maximum values of the collected data, comparing two exercises, matching the collected data against the height profile, calculating the time that elapsed between two locations, etc.

The season for a cyclist consists of four different phases; improving and maintaining basic condition, building up for competition, the competition period and a transitional recovery phase. Within all these periods, there are certain types of exercises to do: those that improve explosive quickness, that is, interval and uphill exercises; those that improve aerobic fitness; those that help the body to recover, etc. Clearly, the Health Club offers different analyzing services for different types of exercise.

The classification of the observations into different route declinations was not successful, and the contexts *cyclist going uphill*, *cyclist going downhill* and *cyclist on level ground* could not be recognized. Other resistant forces, including wind, ground material, rain, brakes, etc., confuse the classification. The low sample frequency was also a problem in the classification, since the location of the cyclist had to be approximated. The ground declination contexts could be recognized straightforwardly using additional measurements, for example, a tri-axial acceleration sensor or an altimeter, but the cost of the system was to be low.

In a competition, the gear shifting tactics affect the overall performance considerably. When riding uphill, the gear must not be shifted, and one has to follow the route's declinations by changing the gears in order to minimize extra burden. The point in time where the gear was shifted can be recognized quite reliably. There are points where the method assumes that a shift has occurred, even if this is not so (the cyclist is not pedaling at full efficiency). This situation can be handled by saving the classification results into a buffer and filtering away single shifts (the gear differs from the last and the next gear in the buffer). In other words, the shift can be identified very reliably after a delay of 15 seconds.

3.3 Routine Learning: Know Your Whereabouts

In publications IV and V, experiments on routine learning are described. The use case *Know Your Whereabouts*, introduced in publication IV, describes the different aspects on routine learning. In the use case, an ordinary user buys a mobile device, and uses it as an ordinary phone. After a certain learning period, the device asks the user whether she wants to get familiar with all the possibilities the device has to offer. The device has learned her frequently visited locations from the location data, and shows them on a map. Furthermore, if there are certain actions she has performed at those locations, the actions can be automated.

Clearly, there are several research topics in the use case, the first being the learning aspect. The contexts (important locations, contexts) have to be recognized from raw sensor data. Routine learning is the second phase. Also, the bandwidth requirements, data storage issues and user interface design are important, although, they are not covered here. In this thesis, association rule algorithms are proposed for associating different contexts and actions (or contexts) together. Therefore the device will need a certain training period for learning those rules. The second point is that the device must not interfere, and the user has to have an option to decline the services. Automatic configuration of the system is also prohibited. The user must accept all the actions the device is able to do.

3.3.1 Data

The first phase of the routine learning experiments was to build a prototype for middleware architecture (Davidyuk *et al.* 2004), and test simple routine learning tasks with it. This work is described in publication IV. In both publications IV and V, location data and profile changes of a PDA were collected in the research group's office space. A Compaq iPAQ with a Wireless LAN card was used for the data collection. A Wireless LAN covers the whole office space and the location of the user is established using Ekahau (Ekahau 2002), which tracks the location of the Wireless LAN card (therefore, the location of the iPAQ). The data consisted of the location and profile status information (the PDA can be set to *General, Silent, Meeting*, and *Outdoors* modes) of three users. The sampling frequency in Ekahau is 0.5Hz. All in all, data was collected for over 60 hours. Additionally, profile statuses and locations were simulated for testing purposes.

In publication IV, a sensor box containing measurement devices for acceleration (x,y,z), humidity, temperature, skin conductivity, and illumination was attached to the PDA. In Figure 8, a picture of the measurement device is shown. A more

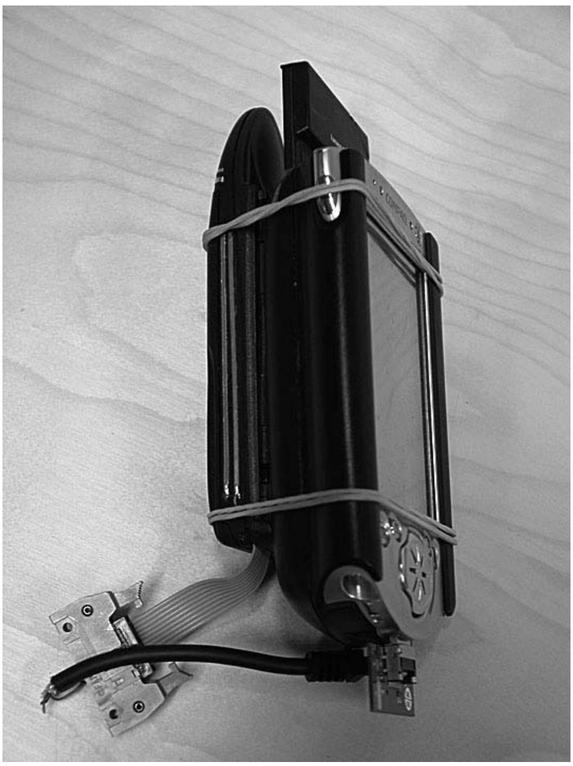

Figure 8. **The sensor box attached to a PDA.**

detailed description of the sensor box can be found from (Mäntyjärvi 2003).

The sensorbox data collected consisted of three different scenarios. One scenario was about 20 minutes long. The first scenario was repeated 10 times, the second and the third 5 times. The sampling frequency for the sensor box is 10Hz and for the Ekahau 0.5Hz. Data was collected by going to different locations in the research group's office space. One of the three scenarios is described in Table 7.

Table 7. A scenario for collecting data in a routine learning experiment.

Step	Action
1.	The testee sits by a desk in location 1 for about six minutes. The device is on the table (illumination sensor upwards).
2.	The testee gets up and walks to the elevators (location 4), and orders the elevator.
3.	When the elevator arrives, the testee steps into the elevator, and goes down to the first floor in the building.
4.	When arriving downstairs, the testee gets out of the elevator and walks to the main doors.
5.	At the doors, the testee steps outside, and walks to the door beside a corridor.
6.	The testee walks back to location 1.

The variables for testing the Apriori algorithm are in Table 8. The data was tagged every time a context changed. This way, the recognition of different contexts can be verified. The contexts were labeled by hand for further processing with the APriori algorithm, however. This means, that *a priori* information from the tags were used in determining the contexts, and no actual context recognition occurred.

Table 8. The contexts related to different sensory measurements.

Measurement	Contexts
acceleration signals	walking, walking downstairs, walking upstairs, sitting in an elevator
illumination	inside, outside
Ekahau	in the office, in a library, in a lobby, in a stairway in a restaurant
Profiles	general, silent, outdoors, meeting

A context variable describing the profile status of a PDA (mobile phone) was integrated to the data matrix. It was assumed that at each location, a suitable profile was used. The profiles are general, silent, outdoors, and meeting.

A binary data matrix was generated for the user behavior and contexts. The context variables were arranged according to the sensorbox data, so the transaction database consisted of over 6000 lines. In other words, this transaction file has 16 columns for the contexts, and for each context there is either 0 or 1 indicating whether the user is in this particular context. The Apriori algorithm was implemented with java. The algorithm goes through the binary matrix and results in a set of IF-THEN rules for important actions.

3.3.2 Results

The testing of the algorithms was done in two phases. In the first phase (publication IV), the recognition of important locations and associating different PDA profiles with those locations was implemented into a prototype application. The important location is considered as a circle, the radius covering an area of one room in our research laboratory. If the user stays in one location for more than five minutes, and on two different occasions during the measurement period, then it is considered to be important. The second phase (publication V) included sensor box data, and in this case also, the important locations were derived. The Apriori algorithm was tested in the second phase.

Figure 9 shows the results of one test. The duration of the test was about 26 hours. The system correctly recognized eight important locations from the location data: Meeting Room TS335, Post Lobby, Lab X, TS380, Coffee Room, Stairway, Medialab and Robolab. The names were prompted from the user after the test.

During the tests, the user changed the profile as would happen in real life. The profile was changed into a silent mode when entering a meeting room and into a general mode when entering the office, for example. Seven routines were identified easily, and they are presented in Figure 9. The routines were:

Location:TS380 $\Rightarrow$ Mode:General
Location:TS335 $\Rightarrow$ Mode:Meeting
Location:LabX $\Rightarrow$ Mode:Silent
Location:Robolab $\Rightarrow$ Mode:Silent
Location:CoffeeRoom $\Rightarrow$ Mode:General
Location:Stairway $\Rightarrow$ Mode:General
Location:Medialab $\Rightarrow$ Mode:Meeting

It is not relevant whether the profile is changed before or after entering the room, as the routine can be identified either way.

In publication V, the Apriori- algorithm described in section 2.4.2 was utilized in determining the user's routines. The known contexts labeled (and the sensor measuring the situation) are presented in Table 8.

The binary matrix containing the context variables was processed with the Apriori algorithm, and different support and confidence values were tested. The processing time of our data set was less than one second. All the combinations for support values of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 and confidence values of 0.6, 0.7,

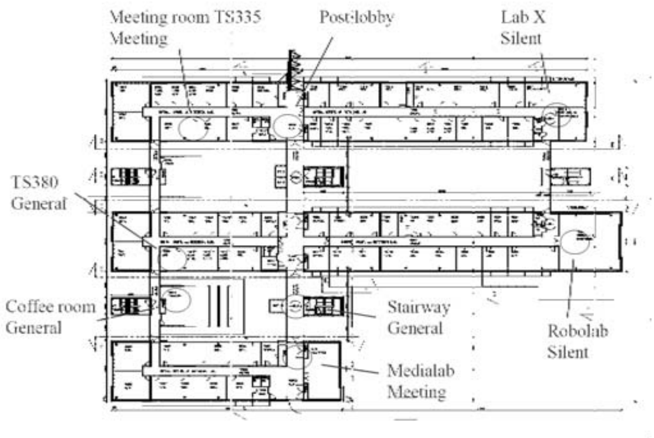

Figure 9. **The ground plan of the office space and the locations and routines identified.**

0.8, 0.9 and 1 were tested. First, the algorithm produces the frequent k-itemsets, and then the rules. The most interesting rules were generated with low support value and high confidence. This is natural, since it is important to know if some contexts occur together in most of the cases; if the user visits a meeting room only twice during the measurement period and always switches the profile into a silent mode, for example. Below, the rules for the support value of 0.1 and confidence value of 0.8 are shown.

Walking ⇒ general, inside
In the Office ⇒ general
In the Office ⇒ sitting, inside, general
In the library ⇒ sitting, inside, silent
In the lobby ⇒ sitting, inside, meeting
Silent ⇒ sitting, inside, in library

The number of rules was 97, but the rules shown above are selected so that the location or the profile context is the main factor for the rule. The other rules are basically combinations of these rules. The complete listing of rules is presented in Appendix A.

3.3.3 Discussion

The implementation of routine learning starts with identifying the contexts of the user. Here, the important locations were derived, and several other contexts were labeled by hand, based on the actual whereabouts of the users. After this, the user gives a meaning for the important contexts, and the user might define for example areas that have gone unnoticed by the machine. Then, association rules can be calculated.

It is a fact that there will be many rules which are not interesting for the user. But if the user is allowed to communicate with the system, then the user already has defined the contexts of interest. Therefore the system can represent only those rules which are attached to these interesting contexts. In this research, representing the rules for the user has not been covered very deeply. One possibility is to predefine certain sentences, and ask the user whether she considers them important. Also visualizing the association rules would give the user a chance to evaluate proper values for the parameters of the algorithm. Furthermore, in a suitable user interface, the user could modify the time span of measurements given the algorithm.

There are many free association rule mining programs available, and it is easy to try how this methodology is suitable for pervasive computing. There are challenges in developing these algorithms further since in the ubiquitous, dynamic world it is likely that not only more data will arise but new variables also for processing, and this too has to be handled adaptively and automatically.

The most difficult problem in routine learning is to decide the amount of context history needed for reliable routine learning. Future work in routine learning is to test this property by selecting different time windows for the rule generation. These time windows can be overlapping. The earlier rules can be saved, since there might be some cyclic behaviour in the actions of the users, and they can be utilized in another setting based on the context of the user. One possibility is to utilize incremental association rules (section 2.4.2). Different settings of contexts can also be studied in different experiments.

4 User Identification

4.1 Introduction

A good extension for designing intelligent environments is the automatic recognition of the occupants. In this thesis, a pressure sensitive floor has been utilized to achieve this task. Automatic recognition leads to personal profiling and enables smooth interaction between the environment and the occupant. Utilizing a person's behaviour or characteristics for identification is called biometric identification (Jain *et al.* 1999). Biometric identification methods are based on the person's speech, signature, gait or face, for example.

Biometric identification systems are pattern recognition systems where an individual is identified from her distinguishing physiological or behavioural characteristics. The problems related to this kind of identification are due to the fact that a person's behaviour or characteristics are affected by her physical or emotional state, environmental noise, and the effect of time. For example, in face recognition the alternative facial expressions, different lighting conditions, and the effect of aging have to be considered in the identification (Ahonen *et al.* 2004). In our research, different footware, utilization of the left or right foot, and the pace of steps affect the signal characteristic.

In face and speech recognition, the research has sharpened into the comparison of different techniques. Considering footstep identification, only a few experiments have been made, and the best results have been achieved with combining different biometric identifiers (Suutala 2004). Hidden Markov Models and Nearest-Neighbor classification have been used in recognizing walkers, and applied in (Addlesee *et al.* 1997) and (Orr & Abowd 2000), respectively. The difference compared to our research is the utilization of dissimilar sensors that measure the vertical component of the *ground reaction force* caused by the weight and inertial forces of the body. The other studies have had only small areas covered with sensors throughout the floor which is capable of measuring the steps, while we have the whole floor area capable of measurement.

Our research environment includes a home theater, two degree-of-freedom active cameras, four mobile robots and one manipulator, a wireless LAN network, and various mobile devices (PDAs, a tablet PC, Symbian mobile phones). Wireless LAN positioning covers a large part of the campus (including the laboratory), and a home automation network is being installed. The aim is to gradually build a versatile infrastructure that offers various generic services for pervasive applications. Naturally, this kind of environment enables realistic experiments that lead to a better understanding of such applications.

In this thesis, pattern recognition methods were applied to footstep identification. In publication VI, initial experiments with Hidden Markov models (introduced in section 2.4.1) are described. The results are not generalizable, since only three testees were discriminated from each other. This work did, however, give important information and experience on the sensory equipment used, and the signal processing needed to enable identification. In publication VII, Learning Vector Quantization was selected for identification, and also a more comprehensive data set was collected. A natural continuation for the earlier experiments on this subject is to utilize the floor more effectively, since it is possible to extract several footsteps from one person for the classification. In publication VIII, J. Suutala developed a method for utilizing three successive footsteps in the identification.

In section 4.1.1, the EMFi-material is described, and the footstep data is presented in section 4.1.2. The two different techniques used for footstep identification are described in section 4.2. To make comparisons between the two techniques, results on experiments with Hidden Markov Models using the same data set as in LVQ experiments are described in section 4.2.2.

4.1.1 EMFi-Material

Electro Mechanical Film (Paajanen *et al.* 2000) (EMFi) is a thin, flexible, low-price electret material, which consists of cellular, biaxially oriented polypropylene film coated with metal electrodes. In the EMFi manufacturing process, a special voided internal structure is created in the polypropylene layer, which makes it possible to store a large permanent charge in the film using the corona method, with electric fields that exceed the dielectric strength of EMFi. An external force affecting the EMFi surface causes a change in the film's thickness, resulting in a change in the charge between the conductive metal layers. This charge can then be detected as a voltage.

EMFi material has been used for many commercial applications, such as keyboards, microphones in stringed musical instruments and small and large area sensors. A Finnish company, Screentec Ltd, has developed vandal-proof keyboards and keypads using EMFi foil protected by a steel or plastic plate. EMF Acoustics Ltd has produced EMFi-based microphones for different stringed instruments, such as bass guitars, acoustic guitars and violins.

EMFi material has been installed in the Intelligent Systems Group's (ISG) research laboratory at the University of Oulu. The covered area is 100 square meters. The EMFi floor in the ISG laboratory is constructed of 30 vertical and 34 horizontal EMFi sensor strips, 30 cm wide each, that are placed under the normal flooring (see Figure 10). The strips make up a 30x34 matrix with a cell size of

Figure 10. The setting for EMFi sensor strips under the laboratory's normal flooring.

30x30 cm. Instead of simply installing squares of EMFi material under the flooring, strips were used, because this layout requires clearly less wiring. If squares were installed, the number of wires would be over a thousand. If a smaller room were to be covered with EMFi material, squares could be used. This would make it easier to determine the locations of the occupants in the room.

Each of the 64 strips produces a continuous signal that is sampled at a rate of 100Hz and streamed into a PC, from where the data can be analyzed in order to detect and recognize the pressure events, such as footsteps affecting the floor. The analogous signal is processed with a National Instruments AD card, PCI-6033E, which contains an amplifier. It would be possible to increase the sampling frequency up to 1.56kHz.

4.1.2 Data

In the autumn 2002, the characteristics of the pressure sensitive floor were examined. The first data set was collected and it consisted of the measurements of 3 persons walking casually on the pressure-sensitive floor. Each person walked alone around the room for 30 seconds. The setting was made as natural as possible. All the testees weighed 66 kg ± 2 and wore shoes. This data was used in publication

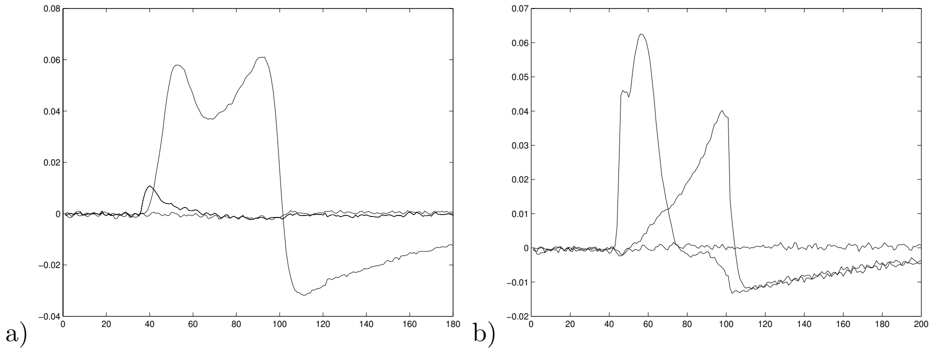

Figure 11. a) A footstep that hits mainly on one strip, but a small fraction of the step affects the measurements of an adjacent channel. b) A footstep in the crossing of two strips.

VI.

More data were collected in the spring 2003 and it consisted of the measurements of 11 persons walking on the pressure-sensitive floor. The testees stepped on one particular strip, and wore shoes. Footsteps targeted into the crossing of two strips were collected as well. Furthermore, stepping with the right foot and the left foot were separated. Also, in one test, footsteps without shoes were collected. All in all, over 60 footsteps were collected from each testee. This data was used in publications VII and VIII.

The footsteps were identified and segmentated from noisy channel data. The segmentation problem has been studied in (Koho *et al.* 2004). Here, raw segmentation was made with hybrid-median filters (Heinonen & Neuvo 1987), and the best footsteps were selected manually.

In Figure 13, one footstep including the important phases of the signal in time and amplitude are presented. The beginning of the footstep is in coordinate $[x_{start}, 0]$. The local maximum $[x_{max1}, y_{max1}]$ is the moment when the heel has hit the floor, and the local minimum $[x_{min}, ymin]$ describes the point between the heel push and the push affected from the ball of the foot. The local maximum $[x_{max2}, y_{max2}]$ is for the hit from the ball of the foot. The actual foostep ends at point $[x_{mid}, 0]$. The negative ending is a property of the EMFi material.

Different problems arise in finding "good-quality" steps for modelling. If a person steps on the crossing of two strips, the amplitude of the step is lower than if he stepped on the center of one strip. This is natural, because only a small part of the step hits on the particular strip. The footstep in Figure 11 a) has not completely hit only one strip, but to a small extent it reaches the adjacent strip as well. The footstep in Figure 11 b) has hit on the crossing of two strips. The measurements from adjacent channels were summed to achieve the whole footstep.

4.2 Techniques for Footstep Identification

In this section, two techniques selected for the footstep classification are introduced. The first idea with footstep identification was based on the fact that a person can recognize other persons based on the sound of their movement. In speech recognition, the Hidden Markov models has been utilized successfully in word recognition, and it was tested here as well. In section 4.2.1, Learning Vector Quantization is described. Another statistical classification method could have been selected, also.

4.2.1 Learning Vector Quantization

Learning Vector Quantization (LVQ) (Kohonen 1997) is a well known tool in various applications where statistical classification is needed, such as texture analysis (Livens *et al.* 1995), speech recognition (Duchon & Katagiri 1993), and image analysis (Cheng *et al.* 2000), to name a few. LVQ algorithms classify the data based on piecewise linear class boundaries, which are determined by supervised learning.

Consider the samples c derived from a finite set of classes $\{C_k\}$. In LVQ, a subset of codebook vectors are assigned to each class C_k. Then, each c is set to belong to the same class as the closest (in Euclidean sense) codebook vector m_i. Let $j = \arg\min_i\{||c - m_i||\}$ define the index of the nearest m_i to c. The equations below define the basic LVQ1 algorithm.

$$\begin{aligned} m_j(t+1) &= m_j(t) + \alpha(t)[c(t) - m_j(t)] && (4.1)\\ m_j(t+1) &= m_j(t) - \alpha(t)[c(t) - m_j(t)] && (4.2)\\ m_i(t+1) &= m_i(t), \text{for } i \neq j, && (4.3)\end{aligned}$$

where $0 < \alpha(t) < 1$, and $\alpha(t)$ is the learning rate, which decreases with time. The algorithm minimizes the rate of misclassification error by iteratively updating the codebook vectors at times $t = 0, 1, 2, \ldots$. Eq. (4.1) is used if c and m_j belong to the same class, and Eq. (4.2) if c and m_j belong to different classes.

In optimized LVQ1, attention is paid to the learning rate $\alpha(t)$, and it is determined optimally for the fastest convergence. To achieve this, the equations above are expressed as

$$m_j(t+1) = [1 - s(t)\alpha_j(t)]m_j(t) + s(t)\alpha_j(t)c(t), \tag{4.4}$$

where $s(t) = +1$ if the classification is correct, and $s(t) = -1$ if the classification is wrong. Then it can be shown that the optimal learning rates are determined by the recursion

$$\alpha_j(t) = \frac{\alpha_j(t-1)}{1 + s(t)\alpha_j(t-1)}. \tag{4.5}$$

In this work, the LVQ_PAK (Kohonen *et al.* 1996) developed at the Faculty of Information Technology at the Helsinki University of Technology, was used for creating the codebook for classification.

4.2.2 LVQ-Hidden Markov Model combination

In section 2.4.1, Hidden Markov models were introduced. In this thesis, the combination of learning vector quantization and HMMs is used. The method has been succesfully utilized in speech recognition (Iwamida *et al.* 1990). In the classification, an LVQ codebook is first trained with feature vectors from the training set, and each prototype vector in the codebook is given an index. After the indices have been determined, the feature vectors are replaced with the index of the nearest prototype vector. One HMM is trained with the index sequences of each class.

In Figure 12, footstep identification for a special case of three persons with HMMs is shown. First, the observation sequences acquired from each person's steps are used for training the three models M_1, M_2, and M_3. Then, the unknown observation sequence is identified by defining the maximum likelihood for each model.

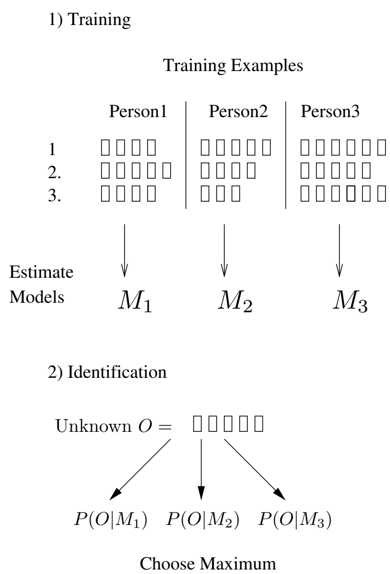

Figure 12. Using HMMs for walker identification.

4.3 Experimental Results

4.3.1 Studies with Hidden Markov Models

In this work, the HTK Toolkit version 3.2 (Young 1993), developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department, was used for creating the LVQ codebook and the Hidden Markov Models.

In the first phase, the identification was studied with three persons' footstep data. The segmented footsteps were divided into ovelapping time windows (window width n and overlapping m). Different window widths and overlapping segments were tested. The best results were gained with a window width $n = 15$ and overlapping 5. The features obtained from each window were the *mean*, *standard deviation*, *minimum* and *maximum* of the amplitude of the pressure signal. A codebook was formed with these feature vectors with vector quantization. Each feature vector was then assigned an integer index from the codebook. The observation sequences were then modelled with Hidden Markov Models containing six states. Different numbers of states and codebook sizes were tested as well. The results are presented in publication VI.

It was noticed that the footsteps of one person were most distinguishable. There was notable confusion in the footsteps of the other two persons, and identification was not very reliable. The features chosen for this experiment (mean, standard deviation, minimum and maximum) did not capture the characteristics of the signals adequately well.

Another experiment included the data of 11 persons' footsteps. The data is similar to that in publication VII, and it consists of about 40 footsteps from each person. A similar procedure as in phase one was used for feature extraction, codebook generation and HMM modelling. The results gave a overall recognition rate of under 40 %.

A different approach for describing the footstep signals into observation sequences was therefore adopted and the footsteps were divided into three parts (see Figure 13). The first part is from the beginning of the footstep x_{start} to a local minimum x_{min}. The second part from the local minimum to the end of the footstep x_{mid}. The third part from the end of the footstep to the point where the negative part has been restored to zero level. The features calculated from these divisions are described in Appendix B.

The footstep data was divided into a training set (about 25 steps from each class) and to a test set (about 15 steps from each class), and the tests were performed by ten randomly selected divisions of the data. Each segmented footstep was featurized according to the description above, and with the features in Appendix B. All feature vectors were normalized to scale $[0, 1]$. The normalized feature vectors were indexed into integers with an LVQ codebook. For each person to be identified, an HMM was trained. The structures were left-right models with three states. A typical confusion matrix from the experiments is presented in Table B.2, in Appendix B. The overall identification rate was 52.09 % (+- 3.67) for the ten experiments. The best results were obtained with a codebook size of 128.

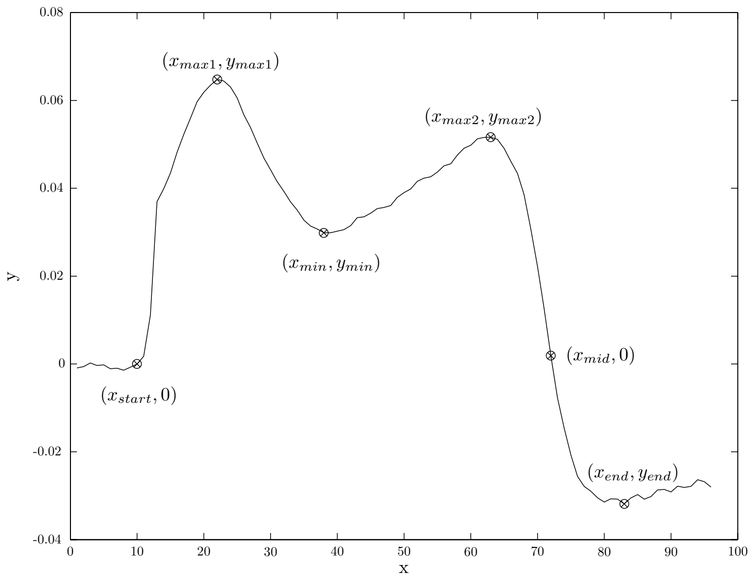

Figure 13. **The features derived from each footstep.**

4.3.2 Studies with Learning Vector Quantization

In utilizing the Learning Vector Quantization, the data was divided into a training set and a test set as in the previous section. In this classification experiment, the footsteps from both the left and right foot were used. The training set consisted of 272 footsteps (around 20-26 examples for each class), and the test set consisted of 131 footsteps.

For feature selection, the footsteps were divided into two sections: the ball of the footstep and the heel of the footstep. The partition was made according to the local minimum (x_{min}, y_{min}), as shown in Figure 13. A software package LNKnet (Kukolich & Lippmann 1999) developed at MIT Lincoln Laboratory was used in feature selection. Several features were derived from footstep data (including spatial and frequency domain features), and the most descriptive ones were selected with forward-backward search described in section 2.3.1 using a kNN- classifier in LNKnet (Kukolich & Lippmann 1999). The selected features are described in Table B.3 in Appendix B.

The codebook sizes from 55 prototype vectors (5 vectors/class) to 200 prototype vectors (18 vectors/class) were tested. The classification results did not change much between the tests. The OLVQ1 was run for 100 iterations, and the LVQ1 for 1000 iterations. The overall recognition accuracy was 66%. A confusion matrix of the results obtained with a codebook size of 60 are presented in publication VII.

There was notable confusion in the footsteps of three testees. In Figure 11 a), the footstep of one such testee is shown. In studying the results, it was noticed

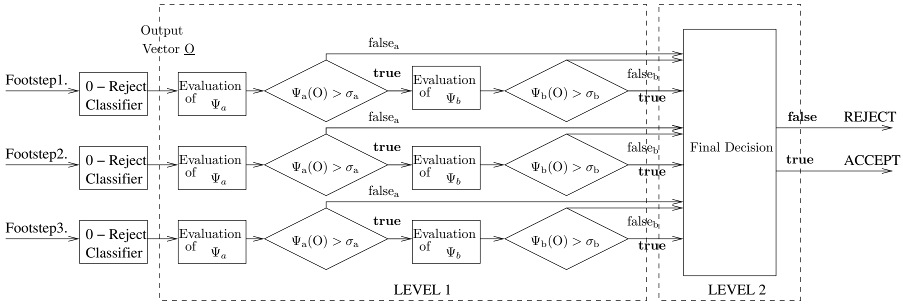

Figure 14. Two-level identification system, which consists of two different reliability evaluators, Ψ_a and Ψ_b. Level 1 rejects footsteps if Ψ_a is below σ_a, or if Ψ_b is below σ_b. Level 2 rejects or accepts three steps pre-classified at the first level. At the second level, samples can be accepted if a majority of them belong to the same class or if two of them are rejected by σ_b (from the overlapping region) at the level 1.

that almost all footsteps from the poorly identified walkers were of the same kind. Classification was successful for the footsteps as for that shown in Figure 11 b).

The reason for the badly identified walkers may be in the summing process made before classification. The measurements from adjacent channels were summed to achieve the whole footstep. The summing works if the step is hit in the center of the two strips. If only a small part of the step hits on the other strip, the summing affects the coordinate points of the maximum to change into another location, and in this case, the classification is not reliable. If these situations are ignored, the overall recognition accuracy is 78%.

4.3.3 The Rejection Parameters

If the user identification were to fail, it would be unbearable if the room would react to a person based on another person's profile. Therefore the results of the classification must be verified.

In publication VIII, the two-level classifier depicted in Figure 14 was developed by J. Suutala. The first step in identification is to classify a single footstep with a predefined LVQ- codebook. At this point, the step is either accepted or rejected, based on a reject option (Stefano *et al.* 2000). On the second level, a decision is made based on three footsteps. A footstep is accepted if a majority of the samples are classified to the same class or if two rejected samples are in an overlapping (within trained class boundaries) region and one sample is classified to a class. Three footsteps can be collected immediately when a person steps into the room (within three seconds at best).

In this method, the failure in a single footstep identification does not necessarily effect the overall recognition result. The rejection is defined to cover the highest possible percentage of samples that would otherwise be missclassified. There are two reasons for the rejection of an input sample: the sample is far from the trained class boundaries or the sample lies in an overlapping region. Clearly, the first rejection criteria corresponds to the situation where an unknown person is walking on the floor (or to noisy measurements). The second rejection criterion is more complicated. The reason for the rejection can correspond to an unknown person, to two persons having similar features in their footsteps or to noisy measurements.

The rejection criteria are calculated from the training data. To create adaptiveness to the system, the criteria can be utilized. If an input sample is rejected based on the first criteria (the input sample is far from the trained class boundaries), the system can assume that an unknown person has entered the room. Then, it is possible to start retraining the classifier automatically, based on these new footsteps. A deeper analysis of the system will be reported by J. Suutala.

4.4 Discussion

The identification was tested using two different methods, a hybrid LVQ-HMM and with Learning Vector Quantization. A data set consisting of the footsteps of 11 persons was used in comparing the two methods. Two different feature sets were tested with the LVQ-HMM, a windowing of the footstep into segments and a division into three segments based on the local minimums and maximums of the signal. In both experiments, the overall recognition rates were below 50 %. With LVQ, the overall recognition rate was 78 %, if poor measurements were ignored. LVQ performed better in considering individual recognition of persons. In some cases, individual classes were falsely classified with 60 % of the test samples when using HMMs. When using the two-level system, 90 % of the footstep data was correctly classified and 20 % was rejected.

In these tests, the number of samples per class is quite low. Especially Hidden Markov Models would require more data to be able to estimate the state and transition distributions of the known classes. If there were more training examples, the recognition could possibly be increased. However, in developing a calm environment, the user cannot be asked to walk on the floor for ten minutes to achieve a broader data set. Learning vector quantization is a simple method, and it does not need so much data to formate the decision boundaries. LVQ is the more suitable method for this kind of environment.

5 Summary and Conclusions

In this thesis, a set of service concepts, the methodology and the signal processing to enable the creation of intelligent environments was presented. The goal was to study what kind of methodologies are suitable for enabling proactiveness in ubiquitous systems. Several service concepts were designed, and proactiveness was made possible through the application of association rules in routine learning. Furthermore, user identification for a smart room was studied.

Context recognition was studied within a Health Club application. The concept of a Health Club in the World Wide Web was also developed. A good example of such a club can be found from (PolarElectro 2004), for example. Neural networks and the method of least squares were used in deriving different contexts. At this point, the research on specifying the cyclists location based on the speed is somewhat out-of-date. Several companies have implemented a GPS into a watch which is easy to carry with the user. However, the research was justified since the cost of the system was to be low. Locationing techniques have progressed greatly during the past two years. These locationing techniques have been used in the routine learning experiments. Routine learning was defined in publications IV and V, and the Apriori algorithm, which is originally a data mining algorithm, was tested in routine learning. The results show that the methodology is applicable in ubiquitous environments.

The second part of the thesis is built on experiments for identifying a person walking on a pressure-sensitive floor. Resolving the characteristics of the special sensor producing the measurements, and which lies under the normal flooring, was one of the tasks of this research. The identification was tested with hybrid LVQ-Hidden Markov models and Learning Vector Quantization. Learning vector quantization outperformed the Hidden Markov Models when using a data set of 11 persons. The future goal of the research is to make an automatic and adaptive identification system which can recognize an unknown (to the system) person and start training itself. This work has begun in publication VIII, and continued in (Suutala & Röning 2004b). The overall recognition accuracy reached a level of 90 % in publication VIII, and even 95 % without rejection (Suutala & Röning 2004a), which can be considered reliable for some ubiquitous applications, but not for authentification.

Representing all the various information from different analyses is a challenging task. The questions of privacy, scalability, bandwidth requirements, etc. need to be considered before introduction. In this thesis, user interfaces were built to visualize the contexts and routines of the user. In (Pirttikangas & Riekki 2004), the visualization of time variant data sets has been developed. The work is a continuation on the experiments in this thesis, and it builds a theoretical framework on certain visualization studies.

The ubiquitous world is a great challenge for pattern recognition. The signals need to be segmented from a continuous data flow and processed in real time. The system must not annoy the person but assist, so minimal initial configuration should be required. The users want to gain fast benefit from the devices they use and this may cause the data sets to be not extensive enough to make, for example, statistically valid desicions. The confidence of the analysis of different situations can therefore be quite low. The user will need to be informed how accurately each situation has been determined, or be allowed to make the decision on her own.

References

Abowd, G. D., Atkeson, C. G., Bobick, A. F., Essa, I. A., MacIntyre, B., Mynatt, E. D. & Starner, T. E. (2000) Living laboratories: The future computing environments group at the Georgia Institute of Technology. In: Extended Abstracts of the ACM Conference on Human Factors in Computing Systems. The Hague, The Netherlands, pp. 215–216.

Adams, N., Gold, R., Schilit, B. N., Tso, M. M. & Want, R. (1993) An infrared network for mobile computers. In: USENIX Symposium on Mobile & Location-Independent Computing. USENIX Association, Cambridge, Massachusetts, USA, pp. 41–52.

Addlesee, M. D., Jones, A., Livesey, F. & Samaria, F. (1997) ORL active floor. IEEE Pers Commun 4(5):35–41.

Agrawal, R. & Srikant, R. (1995) Mining sequential patterns. In: Proc. International Conference on Data Engineering (ICDE). Taipei, Taiwan.

Ahonen, T., Hadid, A. & Pietikäinen, M. (2004) Face recognition with local binary patterns. In: Proc. Computer Vision, (ECCV 2004). Springer, Lecture Notes in Computer Science 3021, pp. 469–481.

Aminian, K., Robert, P., Jequier, E. & Schutz, Y. (1993) Level, downhill and uphill walking identification using neural networks. Electron Lett 29(17):1563–1566.

Aminian, K., Robert, P., Jequier, E. & Schutz, Y. (1995) Estimation of speed and incline of walking using neural networks. In: Proc. Instrumentation & Measurement Technology Conference, IMTC 5/95. Japan, pp. 160–162.

Aoki, H., Schiele, B. & Pentland, A. (1999) Realtime personal positioning system for wearable computers. In: Proc. 3rd International Symposium on Wearable Computers. San Francisco, CA, USA, pp. 37–43.

Ayan, N. F., Tansel, A. U. & Arkun, E. (1999) An efficient algorithm to update large itemsets with early pruning. In: Proc. 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA, USA, pp. 287–291.

Bakis, R. (1976) Continuous speech word recognition via centisecond acoustic states. In: Proc. ASA Meeting. Washington, DC.

Baum, L. E. (1966) Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37:1554–1563.

Begole, J., Tang, J. C. & Hill, R. (2003) Rhythm modeling, visualizations and applications. In: Proc. 16th Annual ACM Symposium on User Interface Software and Technology, UIST 2003. Vancouver, BC, Canada, pp. 11–20.

Bishop, C. M. (1995) Neural Networks for Pattern Recognition. Oxford University Press Inc., New York.

Broder, A. (1997) On the resemblance and containment of documents. In: Proc. Compression and Complexity of Sequences (SEQUENCES'97). Positano, Italy, pp. 21–29.

Brown, E. S. (2001) Project oxygen's new wind. Technology Review .

Byun, H. E. & Cheverst, K. (2001) Exploiting user models and context-awareness to support personal daily activities. In: Proc. Workshop on User Modelling for Context-Aware Applications. Sonthofen, Germany.

Byun, H. E. & Cheverst, K. (2002) Harnessing context to support proactive behaviours. In: Proc. Workshop in ECAI2002 on AI in Mobile Systems (AIMS2002). Lyon, France.

Byun, H. E. & Cheverst, K. (2003) Supporting proactive "intelligent" behaviour: the problem of uncertainty. In: Proc. Workshop on User Modelling for Ubiquitous Computing in User Modeling 2003. Pittsburgh, Pennsylvania, USA.

Cheng, K. S., Sun, R. & Chow, N. H. (2000) Cancerous liver tissue differentiation using LVQ. In: Proc. 1st International Conference on Artificial Neural Networks in Medicine and Biology, (ANNIMAB-1). Göteborg, Sweden.

Cheung, D. W., Han, J., Ng, V. T. & Y.Wong, C. (1996) Maintenance of discovered association rules in large databases: An incremental update technique. In: Proc. 12th International Conference on Data Engineering (ICDE'96). New Orleans, Louisiana, USA, pp. 106–114.

Cheung, D. W., Lee, S. D. & Kao, B. (1997) An incremental technique for maintaining discovered association rules. In: Proc. 5th International Conference on Database Systems for Advance Applications (DASFAA'97). Melbourne, Australia, pp. 185–194.

Clarkson, B., Mase, K. & Pentland, A. (2000) Recognizing user context via wearable sensors. In: Proc. Fourth International Symposium on Wearable Computers, (ISW C2000). Sheraton Colony Square, Atlanta, Georgia, USA, pp. 69–75.

Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullmann, J. D. & Yang, C. (2000) Finding interesting associations without support pruning. In: Proc. 16th International Confenrence on Data Engineering (ICDE 2000). San Diego, California, USA.

Cooper, G., Horvitz, E. & Curry, R. (1998) Conceptual design of goal understanding systems: Investigation of temporal reasoning under uncertainty. Tech. Rep. NAS2-12381, NASA-Ames Research Center, Mountain View, California, USA.

Davidyuk, O., Riekki, J., Rautio, V.-M. & Sun, J. (2004) A context-aware middleware for mobile multimedia applications. In: Proc. 3rd International Conference on Mobile and Ubiquitous Multimedia. College Park, Maryland, USA.

Dehaspe, L. & Toivonen, H. (1999) Discovery of frequent `datalog` patterns. Data Min Knowl Disc 3:7–36.

Dehaspe, L., Toivonen, H. & King, R. D. (1998) Finding frequent substructures in chemical compounds. In: Proc. 4th International Conference on Knowledge Discovery and Data Mining. AAAI Press, New York, New York, USA, pp. 30–36.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc 39(1):1–22.

Dey, A. K. & Abowd, G. D. (2000) The context toolkit: Aiding the development of context-aware applications. In: Proc. Workshop on Software Engineering for Wearable and Pervasive Computing. Limerick, Ireland.

Druzdzel, M. J. & van der Gaag, L. C. (2000) Building probabilistic networks: Where do the numbers come from? IEEE Trans Knowledge Data Eng 12(4):481–486. Guest Editor's Introduction.

Duchon, A. & Katagiri, S. (1993) A minimum-distortion segmentation/LVQ hybrid algorithm for speech segmentation. J Acoust Soc Japan 14(1):37–42.

Duda, R. O. & Hart, P. E. (1973) Pattern Classification and Scene Analysis. Wiley-Interscience.

Ekahau, I. (2002) Ekahau positioning engine 2.0. Tech. rep., Ekahau, Inc. Technology White Paper.

Fukunaga, K. (1990) Introduction to Statistical Pattern Recognition. Academic Press.

Haykin, S. (1994) Neural Networks, A Comprehensive Foundation. Macmillan Publishing Company.

Haykin, S. (1996) Adaptive Filter Theory. Prentice Hall Information and System Sciences Series.

Healey, J. & Picard., R. (1998) Digital processing of affective signals. In: Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Seattle, Washington.

Heinonen, P. & Neuvo, Y. (1987) FIR-median hybrid filters. IEEE Trans Acoust Speech Signal Processing ASSAP-35(6):832–838.

Hightower, J. & Boriello, G. (2001) Location systems for ubiquitous computing. IEEE Computer pp. 57–66.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J. & Toivonen, H. T. T. (2001a) Time series segmentation for context recognition in mobile devices. In: Proc. IEEE International Conference on Data Mining (ICDM'01). San Jose, California, USA, pp. 203–210.

Himberg, J., Mäntyjärvi, J., Korpipää, P. & Mannila, H. (2001b) Using PCA and ICA for exploratory data analysis in situation awareness. In: Proc. IEEE Conference on multisensor fusion and integration in intelligent systems. Baden-Baden, Germany, pp. 127–131.

Hipp, J., Güntzer, U. & Nakhaeizadeh, G. (2000) Algorithms for association rule mining – a general survey and comparison. Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining 2(1):58–64.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D. & Rommelse, K. (1998) The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In: Proc. 14th Conference on Uncertainty in Artificial Intelligence. Madison, Wisconsin, USA, pp. 256–265.

Hsu, F. H. (2002) Behind Deep Blue : Building the Computer that Defeated the World Chess Champion. Princeton Univ Press.

Huang, A. C., Ling, B. C. & Ponnekanti, S. (1999) Pervasive computing: What is it good for? In: Proc. ACM International Workshop on Data Engineering for Wireless Mobile Access (MobiDE 99). Seattle, Washington, USA, pp. 84–91.

Iwamida, H., Katagiri, S., McDermott, E. & Tohkura, Y. (1990) A hybrid speech recognition system using HMMs with an LVQ-trained codebook. J Acoust Soc Japan 11(5):277–286.

Jain, A. K., Bolle, R. & Pankanti, S. (eds.) (1999) BIOMETRICS: Personal Identification in Networked society. Kluwer Academic Publishers.

Jain, A. K., Duin, R. P. W. & Mao, J. (2000) Statistical pattern recognition: A review. IEEE Trans Pattern Anal Machine Intell pp. 4–31.

Jelinek, F. (1976) Continuous speech recognition by statistical methods. In: Proc. IEEE. pp. 532–536.

Kargupta, H., Park, B. H., Pittie, S., Liu, L., Kushraj, D. & Sarkar, K. (2002) Contributed articles on online, interactive, and anytime data mining: MobiMine: monitoring the stock market from a PDA. ACM SIGKDD Explorations Newsletter 3(2):37–46.

Kautz, H. (1991) Reasoning About Plans, Morgan Kaufmann Publishers, chap. A Formal Theory of Plan Recognition and Its Implementation, pp. 69–126.

Kautz, H., Arnstein, L., Borriello, G., Etzioni, O. & Fox, D. (2002) An overview of the Assisted Cognition project. In: Proc. AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care. AAAI Press, Menlo Park, California, USA.

Koho, K., Suutala, J., Seppänen, T. & Röning, J. (2004) Footstep pattern matching from pressure signals using segmental semi-markov models. In: Proc. European Signal Processing Conference, (EUSIPCO 2004). Vienna, Austria.

Kohonen, T. (1997) Self-Organizing Maps. Springer-Verlag Berlin Heidelberg New York.

Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. & Torkkola, K. (1996) LVQ_PAK: The Learning Vector Quantization program package. Tech. rep., Helsinki University of Technology. Report A30.

Kukolich, L. & Lippmann, R. (1999) LNKnet user's guide. http://www.ll.mit.edu/IST/lnknet/ (available on 4.10.2004).

Kume, H. (1989) Laadun parantamisen tilastolliset menetelmät. Metalliteollisuuden keskusliitto, Helsinki. in Finnish.

Laerhoven, K. V. (1999) On-line adaptive context awareness starting from low-level sensors. Ph.D. thesis, University of Brussels.
Laerhoven, K. V. & Cakmakci, O. (2000) What shall we teach our pants? In: Proc. 4th International Symposium on Wearable Computers (ISWC'00). Atlanta, Georgia, pp. 77–84.
Levinson, S. E., Rabiner, L. R. & Sondhi, M. M. (1983) An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst Tech J 62(4):1035–1074.
Livens, S., Scheunders, P., de Wouwer, G. V., Dyck, D. V., Smets, H., Winkelmans, J. & Bogaerts, W. (1995) LVQ classification of corrosion images from wavelet features. In: Proc. Trinocular Joint Meeting on Electron Microscopies. Lausanne, Switzerland, pp. 26–30.
Mannila, H., Toivonen, H. T. T. & Verkamo, A. I. (1995) Discovering frequent episodes in sequences. In: Proc. First International Conference on Knowledge Discovery and Data Mining, KDD-95. Montreal, Canada, pp. 210–215.
Mäntyjärvi, J. (2003) Sensor-based context recognition for mobile applications. Ph.D. thesis, University of Oulu. VTT Publications.
Mäntyjärvi, J. & Seppänen, T. (2003) Adapting applications in handheld devices using fuzzy context information. Interact Comp 15(4):521–538.
Mäntyjärvi, J., Himberg, J. & Seppänen, T. (2001) Recognizing human motion with multiple acceleration sensors. In: Proc. IEEE International Conference on Systems, Man, and Cybernetics. Tucson, Arizona, vol. 2, pp. 747–752.
Mäntylä, V. M., Mäntyjärvi, J., Seppänen, T. & Tuulari, E. (2000) Hand gesture recognition of a mobile device user. In: Proc. IEEE International Conference on Multimedia and Expo 2000 (ICME 2000). New York, New York, USA, vol. 1, pp. 281 – 284.
Mark, W. (1999) Turning pervasive computing into mediated spaces. IBM Syst 38(4):677–692.
Marmasse, N. (1999) comMotion. In: Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99). Pittsburgh, Pennsylvania, USA, pp. 320–321. Extended abstract.
Marmasse, N. & Schmandt, C. (2000) Location-aware information delivery with comMotion. In: Proc. Second International Symposium on Handheld and Ubiquitous Computing (HUC2k). Bristol, United Kingdom, pp. 157–171.
Millenium (2004) Millennium homes: A technology support domestic environment for frail and elderly people. http://disc.brunel.ac.uk/research/index.asp?val=summary19. Available 4.10.2004.
Orr, R. & Abowd, G. (2000) The smart floor: A mechanism for natural user identification and tracking. In: Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2000). ACM Press, New York, New York, USA.
Oxygen (2004) MIT's oxygen project. http://oxygen.lcs.mit.edu/. Available 4.10.2004.
Paajanen, M., Lekkala, J. & Kirjavainen, K. (2000) Electromechanical film (EMFi) - a new multipurpose electret material. Sens Actuators A 84(1-2).
Pascoe, J. (1999) Issues in developing context-aware computing. In: Proc. International Symposium on Handheld and Ubiquitous Computing (HUC'99). Karlsruhe, Germany, pp. 208–221.
Patterson, D., Etzioni, O. & Kautz, H. (2002) The activity compass. In: Proc. First International Workshop on Ubiquitous Computing for Cognitive Aids. Gothenberg, Sweden.
Patterson, D. J., Liao, L., Fox, D. & Kautz, H. (2003) Inferring high-level behavior from low-level sensors. In: Proc. 5th International Conference on Ubiquitous Computing (UbiComp 2003). Seattle, Washington.
Pentland, A. (1995) Machine understanding of human action. In: Proc. 7th International Forum on Frontier of Telecommunication Technology. Tokyo, Japan.
Pentland, A. & Liu, A. (1995) Toward augmented control systems. In: Proc. IEEE Intelligent Vechile Symposium. Detroit, Michigan, USA, pp. 25–26.

Picard, R. W. & Healey, J. (1997) Affective wearables. In: Proc. 1st IEEE International Symposium on Wearable Computers. Cambridge, Massachusetts, USA, vol. 1, pp. 90–97.

Pirttikangas, S. & Riekki, J. (2004) Visual data mining for a timevariant data set. In: Proc. International Workshop on Processing Sensory Information for Proactive Systems (PSIPS 2004). Oulu, Finland, pp. 43–48.

PolarElectro (2004) Polar personal trainer. https://www.polarpersonaltrainer.com. (available 4.10.2004).

Pudi, V. & Haritsa, J. (1994) Quantifying the utility of the past in mining large databases. Inf Syst 19(4):1–54.

Pyle, D. (1999) Data Preparation for Data Mining. Morgan Kaufmann Publishers.

Rabiner, L. R. (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. In: Proc. IEEE. vol. 77, pp. 257–286.

Ramamurti, V. & Ghosh, J. (1999) Structurally adaptive modular networks for nonstationary environments. IEEE Trans Neural Networks 10:152–160.

Reed, R. (1993) Pruning algorithms-A survey. IEEE Trans Neural Networks 4(5):740–747.

Rhodes, B. J., Minar, N. & Weaver, J. (1999) Wearable computing meets ubiquitous computing: reaping the best of both worlds. In: Proc. Third International Symposium on Wearable Computers. Digest of Papers. San Francisco, California, USA, pp. 141–149.

Riekki, J., Huhtinen, J., Ala-Siuru, P., Alahuhta, P., Kaartinen, J. & Röning, J. (2003) Genie of the net, an agent platform for managing services on behalf of the user. Comp Comm 26:1188–1198. Special Issue on Ubiquitous Computing.

Rom*á*n, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H. & Nahrstedt, K. (2002) A middleware infrastructure for active spaces. IEEE Pervasive Comput 1:74–82.

Salber, D., Dey, A. K. & Abowd, G. D. (1999) The Context Toolkit: Aiding the development of context-enabled applications. In: Proc. Conference on Human Factors in Computing Systems (CHI'99). Pittsburgh, Pennsylvania, USA, pp. 434–441.

Sanders, J. M. (2000) Sensing the subtleties of everyday life. Research Horizons, Georgia Tech.

Schilit, B., Adams, N. & Want, R. (1994) Context-aware computing applications. In: Proc. Workshop on Mobile Computing Systems and Applications. Santa Cruz, California, USA, pp. 85–90.

Schilit, B. N. & Theimer, M. M. (1994) Disseminating active map information to mobile hosts. IEEE Network 8(5):22–32.

Schmidt, A. (2000) Implicit human computer interaction through context. Personal Technologies 4:191–199.

Sekine, M., Tamura, T., Ogawa, M., Togawa, T. & Fukui, Y. (1998) Classification of acceleration waveform in a continuous walking record. In: Proc. IEEE 20th International Conference on Engineering in Medicine and Biology Society. IEEE, Hong Kong, pp. 1523–1526.

Sharkey, N. E. & Sharkey, A. J. C. (1995) An analysis of catastrophic interference. Connect Sci 7(3/4):313–341.

Shatkay, H. & Kaelbling, L. P. (2002) Learning geometrically-constrained Hidden Markov Models for robot navigation: Bridging the topological-geometrical gap. J Artif Intell Res 16:167–207.

Srikant, R. & Agrawal, R. (1995) Mining generalized association rules. In: Proc. 21st International Conference on Very Large Databases. Zurich, Switzerland.

Srikant, R. & Agrawal, R. (1996) Mining quantitative association rules in large relational tables. In: Proc. ACM SIGMOD Conference on Management of Data. Montreal, Canada.

Staessen, J., Bulpitt, C. J., O'Brien, E., Cox, J., Fagard, R., Stanton, A., Thijs, L., Hulle, S. V., Vyncke, G. & Amery, A. (1992) The diurnal blood pressure profile. a population study. Am J Hypertens 5(6):386–392.

Stefano, C. D., Sansone, C. & Vento, M. (2000) To reject or not to reject: That is the question. an answer in case of neural classifiers. IEEE Trans Syst Man Cybern C 30(1):84–94.

Sun, J., Tenhunen, J. & Sauvola, J. (2003) A middleware architecture for network-aware adaptive applications. In: Proc. 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Beijing, China, vol. 1, pp. 839–843.

Suutala, J. (2004) Methods for person identification from the pressure signal of walking step. Master's thesis, University of Oulu. (in Finnish).

Suutala, J. & Röning, J. (2004a) Combining classifiers with different footstep feature sets and multiple samples for person identification. In: Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004). (submitted).

Suutala, J. & Röning, J. (2004b) Towards the adaptive identification of walkers: Automated feature selection of footsteps using distinction sensitive LVQ. In: Proc. International Workshop on Processing Sensory Information for Proactive Systems (PSIPS 2004). Oulu, Finland, pp. 61–67.

Tamminen, S., Pirttikangas, S. & Röning, J. (2004) Adaptive signal classification and self-organizing maps in health monitoring. Comp Prog Biomed (submitted).

Tandler, P. (2004) The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments. J Syst Software 69(3):267–296.

Theodoridis, S. & Koutroumbas, K. (1998) Pattern Recognition. Academic Press, New York.

Tuulari, E. (2000) Context aware hand-held devices. VTT publications:412, VTT Electronics, Espoo.

Udechukwu, A., Barker, K. & Alhajj, R. (2004) Mining user navigational patterns in dynamically changing environments. In: Proc. International Conference on Information Technology (ITCC 2004). Las Vegas, Nevada, USA, pp. 372–377.

Viterbi, A. J. (1967) Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans Inform Theory IT-13:260–269.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K. (1989) Phoneme recognition using time-delay neural networks. IEEE Trans Acoust Speech Signal Processing 37(3):328–339.

Want, R. & Hopper, A. (1992) Active badges and personal interactive computing objects. IEEE Trans Consumer Electron 38(1):10–20.

Weiser, M. (1991) The computer for the 21st century. Sci Am pp. 66–75.

Weiser, M. (1993) Some computer science issues in ubiquitous computing. Commun ACM 36(7):75–84.

Williams, R. J. & Zipser, D. (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comp 1:270–280.

Young, S. J. (1993) The HTK Hidden Markov Model Toolkit: Design and philosophy. Tech. rep., University of Cambridge Eng. Dept. TR.153.

Appendix A: The listing of rules with the Apiori-algorithm.

The purpose of Appendix A is to present a complete listing of rules generated by the Apriori- algorithm with support 0.1 and confidence 0.8 for one test case. The Apriori- algorithm is introduced in section 3.3.

```
The rules generated are:

walking,->inside,
walking,->General,
walking,->inside,General,
sitting,->inside,
in office,->sitting,
in office,->inside,
in office,->General,
in office,->sitting,inside,
in office,->sitting,General,
in office,->inside,General,
in office,->sitting,inside,General,
in library,->sitting,
in library,->inside,
in library,->Silent,
in library,->sitting,inside,
in library,->sitting,Silent,
in library,->inside,Silent,
in library,->sitting,inside,Silent,
in lobby,->sitting,
in lobby,->inside,
in lobby,->Meeting,
in lobby,->sitting,inside,
in lobby,->sitting,Meeting,
in lobby,->inside,Meeting,
in lobby,->sitting,inside,Meeting,
inside,->sitting,
General,->inside,
Silent,->sitting,
Silent,->in library,
Silent,->inside,
Silent,->sitting,in library,
sitting,in lobby,->inside,
sitting,in lobby,->Meeting,
sitting,in lobby,->inside,Meeting,
sitting,General,->in office,
sitting,General,->inside,
sitting,General,->in office,inside,
sitting,Silent,->in library,
sitting,Silent,->inside,
sitting,Silent,->in library,inside,
sitting,Meeting,->in lobby,
sitting,Meeting,->inside,
sitting,Meeting,->in lobby,inside,
in office,inside,->sitting,
in office,inside,->General,
in office,inside,->sitting,General,
in office,General,->sitting,
in office,General,->inside,
in office,General,->sitting,inside,
in library,inside,->sitting,
in library,inside,->Silent,
in library,inside,->sitting,Silent,
in library,Silent,->sitting,
in library,Silent,->inside,
in library,Silent,->sitting,inside,
in lobby,inside,->sitting,
in lobby,inside,->Meeting,
in lobby,inside,->sitting,Meeting,
in lobby,Meeting,->sitting,
in lobby,Meeting,->inside,
in lobby,Meeting,->sitting,inside,
inside,Silent,->sitting,
```

```
Silent,->sitting,inside,
Silent,->in library,inside,
Silent,->sitting,in library,inside,
Meeting,->sitting,
Meeting,->in lobby,
Meeting,->inside,
Meeting,->sitting,in lobby,
Meeting,->sitting,inside,
Meeting,->in lobby,inside,
Meeting,->sitting,in lobby,inside,
walking,inside,->General,
walking,General,->inside,
sitting,in office,->inside,
sitting,in office,->General,
sitting,in office,->inside,General,
sitting,in library,->inside,
sitting,in library,->Silent,
sitting,in library,->inside,Silent,
inside,Silent,->in library,
inside,Silent,->sitting,in library,
inside,Meeting,->sitting,
inside,Meeting,->in lobby,
inside,Meeting,->sitting,in lobby,
sitting,in office,inside,->General,
sitting,in office,General,->inside,
sitting,in library,inside,->Silent,
sitting,in library,Silent,->inside,
sitting,in lobby,inside,->Meeting,
sitting,in lobby,Meeting,->inside,
sitting,inside,General,->in office,
sitting,inside,Silent,->in library,
sitting,inside,Meeting,->in lobby,
in office,inside,General,->sitting,
in library,inside,Silent,->sitting,
in lobby,inside,Meeting,->sitting,
```

Appendix B: Details on user identification.

The purpose of Appendix B is to provide details of walker identification which was initially presented in Chapter 4. In Table B.1, the features selected for Hidden Markov Models are described. In Table B.2, a confusion matrix for the results of the hybrid LVQ-HMM method is presented. In Table B.3, the features selected for LVQ are described.

Table B.1. Footstep features for HMM. The first column describes the segment of the footstep that the features have been calculated from. The sequence for HMMs consists of three different parts.

Segment	number	name	description
beginning	1.	$mean_1$	average from (x_{start}, x_{min})
	2.	std_1	standard deviation from (x_{start}, x_{min})
	3.	$area_1$	square area (amplitude) from (x_{start}, x_{min})
	4.	$length_1$	the length of the first segment
	5.	x_{max1}	a local maximum (time, heel push)
	6.	y_{max1}	a local maximum in (amplitude, heel push)
	7.	x_{heel}	heel push starting point (amplitude over y_{min})
	8.	shape_{heel}	$((y_{max1} - y_{min})/(x_{min} - x_{heel}))$
middle	9.	$mean_2$	average from (x_{min}, x_{mid})
	10.	std_2	standard deviation from (x_{min}, x_{mid})
	11.	$area_2$	square area (amplitude) from (x_{min}, x_{mid})
	12.	$length_1$	the length of the middle segment
	13.	x_{max2}	a local maximum (time, toe push)
	14.	x_{toe}	toe push ending point (amplitude under y_{min})
	15.	shape_{toe}	$((y_{max2} - y_{min})/(x_{toe} - x_{mid}))$
	16.	x_{mid}	the ending point of the actual footstep
end	17.	$mean_3$	average from (x_{mid}, x_{end})
	18.	std_3	standard deviation from (x_{mid}, x_{end})
	19.	$area_3$	square area (amplitude) from (x_{mid}, x_{end})
	20.	$length_3$	the length of the last segment
	21.	x_{neg}	a mimimum point (time)
	22.	y_{neg}	a mimimum point (amplitude)
	23.	shape_{end}	$length_3/y_{neg}$
	24.	rel_{end}	y_{max2}/x_{end}

Table B.2. Confusion matrix for the eleven persons' footsteps using the hybrid LVQ-HMM method.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
P1	62.5	37.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P2	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P3	0.0	50.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P4	0.0	0.0	0.0	87.5	0.0	0.0	0.0	0.0	0.0	12.5	0.0
P5	0.0	0.0	0.0	12.5	25.0	12.5	0.0	0.0	0.0	0.0	50.0
P6	0.0	0.0	0.0	0.0	0.0	88.9	0.0	0.0	0.0	0.0	11.1
P7	0.0	0.0	0.0	0.0	0.0	14.3	28.6	42.9	0.0	0.0	14.3
P8	0.0	0.0	0.0	12.5	0.0	0.0	25.0	50.0	12.5	0.0	0.0
P9	0.0	0.0	14.3	0.0	14.3	0.0	0.0	14.3	57.1	0.0	0.0
P10	0.0	0.0	0.0	0.0	0.0	14.3	28.6	0.0	14.3	42.9	0.0
P11	0.0	0.0	0.0	0.0	0.0	25.0	12.5	0.0	0.0	0.0	62.5

Table B.3. Footstep features for LVQ- modelling.

number	name	description
1.	x_{max1}	a local maximum in time (heel push)
2.	y_{max1}	a local maximum in amplitude (heel push)
3 .	x_{min}	a local minimum (time, between heel and toe)
4 .	y_{min}	a local minimum (amplitude, between heel and toe)
5 .	x_{max2}	a local maximum (time, toe push)
6.	y_{max2}	a local maximum (amplitude, toe push)
7 .	x_{neg}	a mimimum point (time)
8 .	x_{neg}	a mimimum point (amplitude)
9.	$mean_1$	average from (x_{start}, x_{min})
10.	std_1	standard deviation from (x_{start}, x_{min})
11.	$mean_2$	average from (x_{min}, x_{mid})
12.	std_2	standard deviation from (x_{min}, x_{mid})
13.	$mean_{max}$	average for the differences of y_{max_1}, y_{max2}, y_m