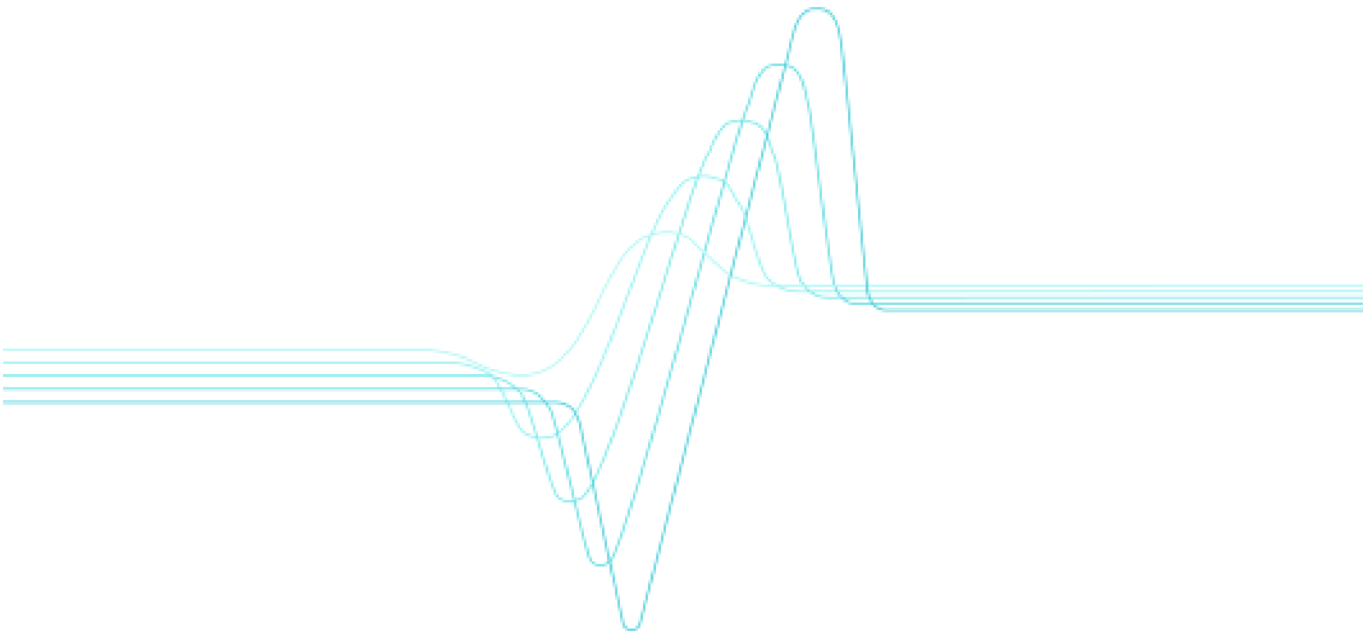
ESPOO 2003

VTT PUBLICATIONS 511

Jani Mäntyjärvi

Sensor-based context recognition for mobile applications



VTT PUBLICATIONS 511

Sensor-based context recognition for mobile applications

Jani Mäntyjärvi

VTT Electronics

Academic dissertation to be presented, with the assent of the Faculty of Technology, University of Oulu, for public discussion in Raahensali (Auditorium L10), Linnanmaa, on December 18th, 2003, at 12 noon.

ISBN 951–38–6253–4 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6254–2 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

VTT, Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Maini Manninen

Otamedia Oy, Espoo 2003

Mäntyjärvi, Jani. Sensor-based context recognition for mobile applications. Ph.D. Thesis. Espoo 2003. VTT Publications 511. 118 p. + app. 60 p.

Keywords Ubiquitous computing, pervasive computing, mobile computing, pattern recognition, data exploration, human-computer interaction

Abstract

Context-aware computing is proposed as an enabling technology for adaptating different functions in computer-aided devices. The development of context-awareness for mobile devices requires recognition and extraction of implicit context information from the usage situations and environment of a device. Context information is provided for applications and services, which adapt their appearance and functions accordingly. Mobile devices contain several potential context data sources such as, location, time and applications. In this thesis, sensors integrated into a mobile device are utilised as sources for context information.

The main challenge in sensor-based context-aware computing for mobile devices is how to define and carry out context recognition from sensor signals to facilitate use of context information in mobile applications. In this thesis we have divided this into specific research problems: How should low-level context information be extracted from sensor signals to obtain a rich and usable representation? How should low-level context information be processed and examined to obtain higher-level contexts? How to utilise context representation in applications? An empiric and data centric approach including signal processing, feature extraction and explorative data analysis methods is used in examining and defining a procedure for sensor-based context recognition.

The main result of this work is a procedure for sensor-based context recognition that is demonstrated with experiments and with the applications developed. The main technical solutions developed include methods for extracting context information and converting it into a suitable context representation, solution for collaborative recognition of the context of a group of mobile devices, an approach for controlling mobile applications and a solution for enhancing remote communication with context information. The thesis includes a review of context data processing and the utilisation of context information in mobile devices.

Academic Dissertation

Faculty of Technology, Department of Electrical and Information Engineering, Information Processing Laboratory, University of Oulu, Finland

Custos & Supervisor

Professor Tapio Seppänen

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

Reviewers

Professor Heikki Kälviäinen

Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing

Professor Martti Mäntylä

Helsinki University of Technology, Information Technology

Opponents

Professor Heikki Kälviäinen

Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing

Docent Samuel Kaski

Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Computer and Information Science

Preface

The work reported in this thesis was carried out during the years 1999–2002, in Nokia Mobile Phones, Oulu, Finland and in the Nokia Research Center, Helsinki, Finland.

I would like to express my deepest gratitude to Dr. Pertti Huuskonen and Mr. Urpo Tuomela for making this work possible.

Special thanks go to Dr. Adrian Flanagan for commenting and reviewing several versions of thesis.

I am grateful to my supervisor Professor Tapio Seppänen for guiding my dissertation efforts. I am grateful to Professor Martti Mäntylä and Professor Heikki Kälviäinen the official reviewers of the thesis, for their comments and suggestions to the manuscript. Many thanks go to Professor Heikki Mannila, Mr. Johan Himberg, Mr. Panu Korpipää, Mr. Antti Takaluoma, and to other co-authors and colleagues in Nokia and VTT for their support.

I wish to thank my friends Mr. Tapio Turunen, Mr. Kimmo Heikkilä, Dr. Tommi Vaskivuo, Mr. Mika Herrala and Mr. Jarkko Tihinen for occasional brainstorming sessions.

I thank my parents Hilkka and Hannu and my sister Piritta for their support during this work.

Finally, I would like to thank my wife Marjo for her support and patience during these years.

The financial support granted by Seppo Säynäjäkangas Science Foundation, Nokia Foundation, and University of Oulu is gratefully acknowledged.

Oulu, October 2003 Jani Mäntyjärvi

List of original publications

The thesis consists of the following papers, which will be referred to in the text by the corresponding Roman numerals (I–VIII). Publications I–II and VII describe methods for extracting low-level information about the context of a mobile handheld device and its user. Publications III–VI discuss methods for examining and extracting higher-level contexts. Publications VII and VIII discuss the utilisation of context information in potential applications.

I. Mäntylä, V-M., Mäntyjärvi, J., Seppänen, T. & Tuulari, E. 2000. Hand gesture recognition of a mobile device user. In Proceedings of the International IEEE Conference on Multimedia and Expo (ICME), New York, USA. Pp. 281–284.

II. Mäntyjärvi, J., Himberg, J. & Seppänen, T. 2001. Recognizing human motion with multiple acceleration sensors. In Proceedings of the International IEEE Conference on Systems, Man and Cybernetics (SMC), Tucson, USA. Pp. 747–752.

III. Mäntyjärvi, J., Himberg, J., Korpipää, P. & Mannila, H. 2001. Extracting the context of a mobile device user. In Proceedings of the International Symposium on Human-Machine Systems (HMS), Kassel, Germany. Pp. 445–450.

IV. Himberg, J., Mäntyjärvi, J. & Korpipää, P. 2001. Using PCA and ICA for exploratory data analysis in situation awareness. In Proceedings of the International IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden-Baden, Germany. Pp. 127–131.

V. Flanagan, J.A., Mäntyjärvi, J., & Himberg, J. 2002. Unsupervised clustering of symbol strings and context recognition. In Proceedings of the International IEEE Conference on Data Mining (ICDM), Maebashi, Japan. Pp. 171–178.

VI. Mäntyjärvi, J., Himberg, J. & Huuskonen, P. 2003. Collaborative context recognition for handheld devices. In the Proceedings of the International IEEE Conference on Pervasive Computing and Communications (PerCom), Dallas, USA. Pp.161–168.

VII. Mäntyjärvi, J. & Seppänen, T. 2003. Adapting applications in handheld devices using fuzzy context information. Interacting with Computers Journal, Vol. 15(4), pp. 521–538.

VIII. Schmidt, A., Takaluoma, A. & Mäntyjärvi, J. 2000. Context-aware telephony over WAP. Personal Technologies, Vol. 4(4), pp. 225–229.

Publication I describes hand gesture recognition experiments with signals from accelerometers integrated into a mobile handheld device. Hand gestures form a set of implicit information that can be utilised in human-computer interaction (HCI). The self-organising map (SOM) is used to cluster six types of static gestures while Hidden Markov Models (HMMs) are used to construct time-series models of dynamic gestures. Experiments are performed with three-dimensional acceleration signals collected from real usage situations. In the experiments, the performances of both clustering and dynamic models are examined with test data. Results suggest that with the methods used it is possible to extract hand gesture information to be utilised in the development of implicit methods for HCI. The author of this thesis is responsible for suggesting the study motivated by the development of implicit HCI methods. He presented the idea of processing accelerometer signals into components describing static and dynamic gestures and process gestures separately. He also designed and performed hand gesture recognition experiments with the SOM. Lic. Tech. V-M Mäntylä carried out dynamic hand gesture recognition experiments with HMM. Lic. Tech. E. Tuulari and the author of this thesis jointly designed a general-purpose sensor box. The role of Prof. T. Seppänen was to give professional guidance related to the methods used.

Publication II presents human walking pattern recognition experiments with signals from accelerometers integrated into a mobile handheld device. Six-dimensional signals are collected from two sensor boxes attached to a user. Two sensor boxes simulated the mobile handheld device with its accessory. The primary motivation for using two sensor boxes is to apply methods for multidimensional data analysis such as, principal component analysis (PCA) and independent component analysis (ICA) to examine “interesting directions” of human activity data. Principal components (PC) and independent components (IC) are examined visually and with classification experiments, which are performed using a multilayer perceptron classifier and feature generation with a wavelet transform. The results of this study suggest that common human walking patterns can be classified using analysis methods for multidimensional data and wavelet transform in feature extraction phase. The author of this thesis is responsible for suggesting the study. He presented the idea of using PCA and ICA to obtain an insight into "interesting directions" of the walking patterns. He designed preprocessing and classification tasks and performed all experiments and analysis. M.Sc. J. Himberg gave guidance related to analysis methods for multidimensional data, to classification and to the outline of the publication. Prof. T. Seppänen gave guidance related to wavelets and to the outline of the publication.

Publication III describes experiments for exploring methods for extracting higher-level context information from multidimensional low-level context data. The focus is in the analysis of the clustering and segmentation performance of low-level context information in a crisp and fuzzy representation. Clustering and segmentation performances obtained with the crisp and fuzzy representation are compared. The motivation is to find a low-level context representation that gives consistent results for higher-level context extraction. The clustering is performed with k-means clustering while minimum-variance segmentation is used for time series segmentation. The clustering and segmentation are examined qualitatively with video recordings from measurements. The results show that data preprocessed with fuzzy quantisation gives more consistent clustering while the clustering with crisp data includes diverse sets of variables into clusters. Furthermore, segmentation is able to detect higher-level context changes from time series data that correspond to changes in real context. The author of this thesis suggested the study to select the representation for low-level context information by comparing crisp and fuzzy data representations. He also suggested the qualitative examination of clustering and segmentation with video recordings. This is motivated by the fact that it is difficult to obtain faultless context measurements with a mobile handheld device when the user labels data himself. This is why video recordings are used. The author of this thesis also suggested the processing of sensor signals into a low-level context representation. He designed and applied preprocessing and feature extraction methods with co-author M.Sc. P. Korpipää. These methods are used in Publications III, IV, V, VI, and VII. M.Sc. J. Himberg performed clustering and segmentation experiments. Prof. H. Mannila is responsible for the idea of the minimum-variance segmentation algorithm for multidimensional context data.

Publication IV describes an approach for examining multidimensional low-level context information with analysis methods for multidimensional data, PCA and ICA. PCA is used to fuse and compress multidimensional context data into a more compact representation to be used in the visual examination of higher-level contexts, and to examine the variability of individual context variables in higher-level contexts. ICA is applied to extract patterns containing independent low-level information in higher-level contexts. A few principal components are used to visualize multidimensional context data with one- or two dimensional data projections. The author of this thesis and M.Sc. J. Himberg composed together the study concerning the compression and the visualisation of compressed low-level context information to analyse higher-level contexts, and to analyse the independence of context variables. They also performed these experiments together. The author of this thesis also suggested processing of sensor signals into a low-level context representation, and also designed and applied preprocessing algorithms with co-author M.Sc. P. Korpipää.

Publication V presents a general framework for processing context information. In the framework context information is represented as individual symbols and as strings of symbols. The paper describes a novel method for extracting higher-level contexts from lower-level context information by using an unsupervised clustering approach. The main result of this paper is that the novel context recognition framework is presented and verified with experiments with a context data set. The results of this publication suggest also that various levels of context can be composed. The algorithm is designed and implemented by co-author Dr. J.A. Flanagan. One of the main results of this publication was the formulation of a framework for processing context information by describing the context recognition problem as one of generating higher level contexts. This framework came as a result of the collaboration between the author's of the publication over a period of time. Furthermore, they jointly developed a symbolic representation for multilevel context data. The author of the thesis is responsible for designing and applying methods for processing low-level sensor information into a symbolic representation. M.Sc. J. Himberg produced segmented data for these experiments.

Publication VI describes a novel method for collaborative context recognition using low-level context information. The idea of the method is that several devices within a certain area are able to negotiate the current context together and control applications accordingly. The results of this study suggest that a method is capable of providing more reliable context information of the current context than recognised by an individual device. The idea of the method is conducted from examining the behaviour of people in different contexts. The author of this thesis is responsible for the main idea and the operating principle of the method. He composed the framework for collaborative context recognition together with co-author M.Sc. J. Himberg. The author of this thesis designed algorithms and performed all experiments. Analysis and critical evaluation of implications of the method are performed jointly by all authors.

Publication VII explains the adaptation of user interface (UI) applications in mobile handheld devices according to fuzzy low-level context information. The approach utilises fuzzy logic controllers, which are designed to control the representation of information presented on a screen as well as the loudness of operating tunes according to the current context. The results of this study show that fuzzy context information can be used in controlling UI applications enhancing the applications' capabilities for representing information according to context. Moreover, the results indicate that end users accept application adaptation in many situations while insisting on retaining most control over their device functionality. The author of the thesis composed the study together with co-author Prof. T. Seppänen. The architecture for controlling applications was also composed jointly. The author of this thesis designed and applied the algorithms for processing sensor signals into context information. He also designed and implemented rule bases, fuzzy logic controllers, performed all experiments, analysis and studies with end users. The role of Prof. T. Seppänen was to give guidance in these issues.

In Publication VIII context-aware call application for mobile phones is presented. The paper includes the analysis of behavior of people when making a call with a mobile phone. Analysis reveals that the lack of knowledge about context at the other end leads to initiation of calls, which are not appropriate to the current situation. As a solution to this particular problem the paper presents an application for exchanging context information before initiating the call. The application is implemented using the wireless application protocol (WAP). This study shows that mutual context of people plays an important role when establishing communication, particularly with a mobile phone. The approach described in this paper provides a way to deliver mutual context information when establishing remote communication, thus making it more natural. The author of the thesis viewed people's behavior when communicating with context-aware mobile phones. He also composed the idea of a context-call application together with co-authors Dr. A. Schmidt, who implemented the context-call application, and Mr. A. Takaluoma, who provided guidance for the implementation of the system.

Contents

Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.vtt.fi/inf/pdf/)

Symbols and abbreviations

Abbreviations

A/D	Analog-Digital
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
IC	Independent Component
ICA	Independent Component Analysis
IIR	Infinite Impulse Response Filter
IrDa	Infrared Data Association
PC	Principal Component
PCA	Principal Component Analysis
PDA	Personal Digital Assistant
RF	Radio Frequency
RMS	Root Mean Square
SCM	Symbolic Clustering Map
SOM	Self-Organising Map
STD	Standard Deviation
UI	User Interface
WAP	Wireless Application Protocol

Variables and constants

i,j,k	General purpose index
a	Filter coefficient, also scaling variable
b	Filter coefficient, also shifting variable
c	Wavelet coefficient
$C_{i,j}$	Context information on a level i from source j
d_i	Desired output of neuron i, also wavelet coefficient

e_i	Error signal of neuron i
f	Frequency
$i(\mathbf{x})$	Winning neuron for input pattern $\mathbf{x}$
m	Time delay
n	Iteration index, also dimension of vector
o_i	Actual output of neuron i
Q	Vector quantizer
$s(\cdot)$	Similarity measure
t	Time index
v_i	Local field of a neuron i, also Gaussian variable
x	Input variable
x_m	Magnitude of signal
y	Output variable
z	Zero mean random variable
α	Momentum constant
δ_i	Local gradient of neuron i
η	Learning rate
λ_i	Eigenvalue
θ	Kernel width
σ	Width of the topological neighborhood

All the vectors are printed in boldface using lowcase letters

$\mathbf{s}$	n-dimensional source vector
$\mathbf{s}'$	Vector of independent components
$\mathbf{u}_i$	Eigenvector
$\mathbf{v}$	Whitened data vector
$\mathbf{w}_i$	Weight vector of a neuron i
$\mathbf{x}$	n-dimensional data vector, also observation vector
$\mathbf{x}'$	Principal component vector

All the matrices are printed in boldface using capital letters

$\mathbf{A}$	Unknown mixing matrix in the ICA model
$\mathbf{B}$	Demixing matrix in the ICA model

$\mathbf{C}$	Covariance matrix
$\mathbf{U}$	Matrix containing eigenvectors as columns
$\mathbf{V}$	Whitening matrix
$\mathbf{W}$	Separating matrix in the ICA algorithm
$\mathbf{\Lambda}$	Matrix containing eigenvalues

Functions:

$Cx(\cdot)$	Wavelet transform function
$c_{xx}(\cdot)$	Autocovariance function
$E\{\cdot\}$	Mathematical expectation
G	Guadratic function
$H(\cdot)$	Differential Entropy
$I(\cdot)$	Mutual information
$J(\cdot)$	Negentropy
$K(\cdot)$	Gaussian kernel function
$kurt(\cdot)$	Kurtosis, fourth order kumulant
L	Cost function
$n_{j,i(\cdot)}$	Neighborhood function of winning neuron
$P(f)$	Power spectral density function
$\varphi_j(\cdot)$	Scaling function of a neuron j
$\Phi(\cdot)$	Scaling function
$\mu(\cdot)$	Fuzzy membership function
$\Psi(\cdot)$	Wavelet function

1. Introduction

Recent technical advances in electronics, digital signal processing and wireless communication have enabled the fast market growth of mobile handheld devices such as, mobile phones and personal digital assistants (PDAs). To fulfil end user needs these devices contain an increasing number of applications using personal information, personal communication, information browsing, and entertainment. These devices have ”high” computational capability and they support both long and short-range wireless communication standards such as, the Global System for Mobile Communications (GSM), the General Packet Radio Service (GPRS), and Bluetooth.

The role of computer-aided mobile handheld devices is changing along with technical advances and consumer needs. These devices enable remote communication between humans and machines and they are ubiquitous access points to information located in the Internet. Furthermore, these devices are becoming stores of confidential information, for example, contact information, and our everyday money transactions. Interconnected mobile computing devices with an increasing numbers of applications and services set challenges for human-computer interaction (HCI) development. HCI methods in these devices are based on the traditional means of explicit computer interaction and the ways to input and output information are basically keyboard, display and audio. Due to the small size of these devices the possibilities for utilising traditional HCI methods are limited, for example, the number of buttons is small, which may lead to complex key combinations to execute functions and to complex menu structures.

Although other interaction methods such as, voice control, tactile feedback, and touch screens have been developed they are still used explicitly to execute certain predefined functions. To find solutions for emerging needs in alternative HCI methods for small computer-based devices, the attention has recently turned to the development of methods to utilise the implicit interaction information concerning the usage environment and the usage history of a device and applications. In particular, this means analysing information from situations and the environment in which the device, services and applications are used, and adapting the HCI accordingly. This is called context-aware computing and here the implicit information is referred as context information.

The analysis of context information requires monitoring of internal and external processes of a device such as, used applications, wireless services, cellular and local area network connections and time. The important part of context data of a mobile handheld device is its usage environment consisting of location of usage, social environment, physical quantities of the environment like loudness and illumination, user activity like movements and gestures, orientation of a device, and determining whether the device is in the user's hand or not. Gathering and analysing context data from these quantities requires selection and integration of suitable sensing elements into a device, extraction of relevant features from various information sources, and composition of relevant information from extracted features to a more expressive description of the current situation. Characteristic of context information in mobile devices is the dynamic and overlapping nature of contexts. This is due to the high mobility of their users. Particularly, for sensor-based context information the characteristic feature is unreliability since extracted context information describes approximations of real contexts at best.

1.1 Problem statement

In this thesis, the methods for processing context information provided by sensors to enable the utilisation of relevant information in mobile handheld device applications are studied. As a short summary, we claim that the element slowing down the development of mobile context-aware systems is the lack of understanding of the dynamic characteristics and the structure of context information. The extracted information should be represented in a form that would enable easier utilisation in applications.

1.1.1 Research problem

When drawing up the research plan for this study the research of sensor-based context-aware mobile handheld devices was in its early stages: First trials for system architectures for context-aware computing existed (Schilit 1995, Salber *et al*. 1999). Mobile handheld devices with integrated sensors as well as first trials to develop context recognition methods appeared, see e.g. (Schmidt *et al*. 1999). The research of context recognition is evolving. The understanding of underlying data is required before online recognition systems can be developed. However, there exists little information related to the examination of the context data of a mobile handheld device and its user. Thus the general research problem (RP) for this thesis is:

RP: How should sensor-based context recognition be defined and performed to facilitate the utilisation of context representation in mobile applications?

The general research problem is divided into specific subproblems that are studied using example scenarios. To clarify the subproblems a few concepts are first explained; utilisation of context information in mobile applications refers to new application ideas and novel ideas for adapting applications according to context. Sensors provide signals that convey context information. Low-level context information refers to data extracted directly form sensor signals while higher-level context information refers to data obtained by combining low-level context information. Altogether, by context information we denote both lower and higher-level context information.

The specific subproblems (SP) are:

SP_1: How should low-level context information be extracted from sensor signals to obtain rich and usable context representation?

SP_2: How should low-level context information be processed and examined to obtain higher-level contexts?

SP_3: How to utilise context representation in applications?

1.1.2 Research hypothesis

The division of the research problem (RP) into subproblems ($SP_{1\text{-}3}$) allows us to state the main hypothesis: Low-level context information should be extracted from sensor signals using signal processing methods while the examination and *extraction of higher-level context information should be approached by means of data centric analysis methods.*

We state another hypothesis where the signal processing methods for processing low-level context data should be selected so that extracted features describe concepts of the real world, and representation for low-level context data including categorisation and labelling of context information should be composed according to these concepts.

1.1.3 Research assumptions

The general assumption in the research of sensor-based context-awareness for mobile handheld devices is that carefully chosen sensors integrated into a device enable detection of important aspects of context. The basic assumption for creating context-awareness by processing sensor signals is that the information content of signals recorded in similar contexts consistently resemble each other (Schmidt & Van Laerhoven 2001). A consequence of this assumption is that contexts can be recognised by examining sensor signals. To examine context recognition by applying feature extraction methods for sensor signals it is reasonable to further assume that features extracted from sensor signals measured from similar contexts are similar to each other.

To be able to develop the representation for context information it is reasonable to assume that the categorisation and labelling of sensor-based low-level context information is feasible. We further assume that the selection and the application of feature extraction methods are such that extracted context data describes aspects of real world situations.

A central assumption for the processing of context information is that the simultaneous examination of several features helps us to understand the characteristics and the structure of context data better. Moreover, the examination of data requires an assumption that suitable data analysis methods can be selected or developed for examining dynamic characteristics of data and context representation. As claimed by Schmidt (Schmidt 2002), context can be regarded as a pattern. Therefore we assume that the features extracted from signals describe the contextual patterns.

1.1.4 Research methods

In this thesis, a constructive approach for solving the research problem is applied. The structure of context information in dynamic environments is still unknown or at least blurred. For this reason we have chosen an empiric and data centric approach to both examine context recognition and demonstrate the utilisation of context information in mobile applications. It is acknowledged that the creation of context-awareness in these devices enables the creation of novel types of applications. Examining people’s behavior in different contexts enables us to find novel solutions for context recognition and applications. Moreover, a literature review is used as a research tool.

1.2 Scope of research

1.2.1 Application areas

We have restricted the application area of this research to mobile handheld devices a user carries with her/him such as mobile phones and PDAs. The results obtained from this research concerning context recognition and the utilisation of context information are applicable to other application areas of context-aware computing. Perhaps the closest application area is context-aware wearable computers that are not exactly mobile handheld devices but are worn by a user and thus with the user all the time. Moreover, potential application areas include intelligent environments that consist of static computer-aided devices and sensing facilities embedded in rooms and buildings. From now on, mobile handheld devices are denoted as mobile devices.

1.2.2 Research areas

Context-aware computing for mobile devices is an interdisciplinary research topic including research aspects from computer science, information processing, wireless communications and HCI.

In this research, context information processing has a dominant role. To extract low-level context information from sensor signals several carefully chosen feature extraction methods are applied. In the examination of context information a data analysis approach is chosen since there exists little knowledge of the characteristics and structure of context data. Analysis methods for multidimensional data provide methods to examine context information. To examine the data and to utilise it in applications a suitable representation for extracted information is required. In this work, low-level context information is represented using an ad-hoc ontology for sensor-based context data. There exist a variety of methods in artificial intelligence for describing the ontology of information, for example, different description languages. Our approach is data centric and we have excluded this kind of approach.

Computer science and software engineering are research areas related closely to context-aware systems. They provide methods for developing and studying architectures and algorithms from the viewpoint of the constraints set by the operating environment, for example, processing capacity requirements. However, in this study context recognition methods and the utilisation of context information are examined from the data point of view. It is not our aim to speculate on the constraints of future devices' operating environments.

Wireless communications is an important research area for mobile devices since it covers the research for enabling short range and remote communication. We claim that communication of context is an essential part of context-aware computing. We will discuss how devices could benefit from context communication, rather than how context communication should be carried out to fit in certain communication protocols.

The research area of human-computer interaction considers the evaluation and development of the interaction methods between human and computer-aided devices. Context-aware computing has been proposed to improve HCI. Thus, HCI is relevant when the user interaction with context-aware applications is studied.

1.2.3 Outline of this thesis

This thesis summarises the research into signal processing and pattern recognition methods from sensor signals for context-aware mobile device applications. The research was carried out at Nokia Mobile Phones, Oulu, 1999–2000 and Nokia Research Center, Helsinki, 2000–2002.

The organisation of the thesis is the following: Chapter 2 reviews context-aware computing for mobile devices with definitions and related work. Chapter 3 presents an approach for the design and development of context recognition procedure and utilisation of context information in applications. The basic methods used are introduced and how they have been applied is explained. The results obtained and discussion are presented in Chapter 4. Chapter 5 presents summary and conclusions.

2. Context-aware computing

Communication between humans is contextual, i.e., related to a current situation, where the situation consists of the social settings of the environment, common history of the communicating parties about the topic and each other, among others. Humans have a need to perceive context that can be noticed, e.g., in face-to-face communication where humans perceive context unconsciously and they react to context changes according to learned behaviour models. The objective of context-aware computing is to enhance man-machine interaction by transferring some of these mechanisms to machines.

2.1 Introduction

This chapter introduces the research theme of context-aware computing. The introduction is focused on mobile devices. The model of the concept we have sketched consists of five main themes as shown in Figure 1. Information sources cover a heterogeneous set of information sources relevant for context-aware computing. The focus of context recognition is to provide a common representation for information and to carry out information fusion. Utilisation of context information finds answers to questions like; how can context information be utilized, covering development issues of context-aware systems, novel applications, services and features for them from the viewpoint of data representation. Interaction deals with user interaction with context-aware systems, particularly, research for design and evaluation of the interaction methods for context-aware applications. System architectures covers research and the design of architecture solutions to ensure the operation of context-aware systems.

This thesis deals with the themes; information sources for context data focusing on sensors integrated into a mobile device, context recognition methods and utilisation of context information. Themes; system architectures and interaction are not in the focus of this thesis, but they are briefly introduced in this chapter. Since system architecture issues affect all other themes they are discussed throughout this chapter.

This chapter presents fundamental research work in the field of context-aware computing. The terminology is introduced by explaining the definitions. The main information sources for mobile devices and methods used in the development of context recognition are presented. The introduction is consummated by giving a review of the utilisation of information and interaction with mobile context-based systems. The concept is summarised and important issues from the viewpoint of this thesis are presented.

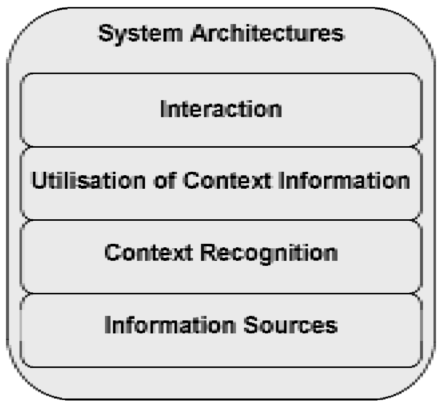


Figure 1. Overview of context-aware computing research.

2.1.1 Emergence of the field

Context-aware computing is proposed as an enabling technology for adaptating different functions in computer-aided devices. The adaptation is proposed to be carried out by recognising and composing implicit information from the usage situations and environment of a device and its user and to provide this information, e.g., for various types of applications and services, which can adapt their appearance and functions accordingly. Context-aware computing is also defined as situated computing (Hull *et al*. 1997).

Originally, context-aware computing was considered to be a part of a wider concept called ubiquitous computing first introduced by Weiser (Weiser 1991). He introduced ubiquitous computing as a paradigm of computing devices disappearing and embedding seamlessly into an environment to automatically perform together tasks to assist their users.

The first context-aware computing experiments were carried out at early 90's: The active badge location system by Olivetti Research (Want *et al*. 1992), and ParcTab location system by Xerox Laboratories (Want *et al*. 1995). Both experiments demonstrated the use of small mobile devices operating together with an infrastructure. Although context information sources were limited to location of people and devices it enabled a set of novel applications (Want *et al*. 1992, Want *et al*. 1995) such as finding individuals and groups of people, location histories of people, location based notifications and reminders and call and message delivery.

In the work of Want *et al.* (Want *et al*. 1995) the basic requirements for context-aware computing are identified; both mobile and static computing devices must be able to formulate their context information, transfer their context and system status information to other devices and applications distributed over the environment. Schilit (Schilit 1995) developed the first software architecture for context-aware computing at Xerox Laboratories during their ubiquitous computing experiment. It was designed to support mobile application adaptation in the office environment triggered by basic types of context information such as, location of use, social environment, presence of accessible devices and changes of these things over time. All types of context information used in that work were derived from location information.

Research by Lamming *et al*. (Lamming *et al.* 1994) is closely related to context-aware computing. Particularly, their research relates to sensing and recognising activities of a mobile device user. The objective of their research is to develop a personal computing system to extend the user's ability to recall things that are specific to his or her life (Lamming *et al*. 1994). The system consists of a set of sensing and recording devices distributed in the environment such as, video cameras, sensors, workstation activity monitoring systems, etc., to provide a comprehensive set of information of what a user is doing.

In 1997 Rhodes (Rhodes 1997) presented ideas and examples of how to utilise context-awareness in wearable computing, particularly in a wearable memory prosthesis, which is similar to the concept of Lamming. Rhodes also discusses that wearable computers can be equipped with capabilities for gathering and processing context information obtained via wearable sensors such as, camera, microphone, accelerometer and user body functions.

In the same year Pascoe (Pascoe 1997) discussed the general needs for context-aware computing. He proposed that in the design of architecture for context-aware computing, contexts should be modelled in context classes and context information should be organised in a hierarchical manner. In the same year Brown (Brown *et al*. 1997) discussed the commercialisation of context-aware devices, how to make easier to create context-aware applications, and suggested common architectural modules such as, triggering module, notification module, and sensing module. Brown (Brown 1998) proposed common triggering mechanisms for executing context-aware applications and services.

Considerable effort in context-aware computing is expended on the research carried out at Georgia Institute of Technology. The study by Abowd *et al*. (Abowd *et al*. 1997) extended the application domain. They introduced a mobile context-aware tour guide, which provided location aware tourist applications and services for both indoor and outdoor environments. The tour guide is an attractive research topic for context-aware computing for mobile devices since it connects important context sources, the location and user's mobility into various types of applications. Similar studies concerning tour guides using location information are carried out, (Cheverst *et al*. 1998, Oppermann & Specht 2000).

Salber and Abowd (Salber & Abowd 1998) presented an infrastructure for handling context information. In the same year Dey *et al*. (Dey *et al*. 1998) presented a software framework for context-aware computing. The framework was developed to ease the tasks of programmers and users when programming and interacting with a context-aware system. In 1999 Salber *et al*. (Salber *et al*. 1999) introduced a general architecture to support the development of context-aware applications and services. They addressed four issues concerning context-aware computing: Context infrastructure should accommodate distribution of applications, context sources, and their components. Components handling context information should compose and provide more abstract context information. There should be communication across heterogeneous components. There should be mechanisms to handle highly dynamic context information. These are critical issues for architectures that must be solved before context-aware computing for mobile devices can be developed.

The research of Schmidt (Schmidt 2002) deals with development and prototyping of context-aware computing for ubiquitous computing environments. His approach is also bottom-up, as we have here, but his research focuses more on examination of low cost sensors as context sources, gathering of low-level context information obtained from the sensing devices and prototyping mobile and static context-aware systems for ubiquitous computing environments. His contribution to sensor-based context-awareness research is notable. His work model for context and guidelines for developing sensor-based context-aware systems are presented and discussed in this chapter.

The rest of this chapter first presents definitions for context and context-awareness and then focuses on context recognition and utilisation of context in mobile applications.

2.1.2 Definitions

Context

The use of the term is confusing since the term has several meanings depending in which field of research it has been used. The term has been widely used in linguistics, psychology, computer science, etc. The general definition of the term according to the dictionary (Merriam-Webster's Collegiate Dictionary) is:

'The interrelated conditions in which something exists or occurs'.

Only in the field of computer science has the term been widely used, in illustrating different topics, for example, 'context sensitive help', 'contextual search', 'multitasking context switch', to mention a few (Chen & Kotz 2000).

The use of commonsense in designing user interfaces for computer based devices has been recognised (Minsky 2000). The concept of context has been exploited in developing the knowledge base of human commonsense and common knowledge (Lenat *et al*. 1990). That particular work is the most extensive approach for developing the concept. The concept of context in the knowledge base of human commonsense is explained via 12 mostly-independent dimensions along which contexts vary, as Table 1 presents. Four of the dimensions are related to the time and place while the other dimensions are more abstract concepts.

Table 1. Context dimensions of commonsense knowledge base (Lenat 1998).

Context Type	**Explanation**
Absolute Time	A particular time interval in which events occur
Type of Time	A non-absolute type of time period
Absolute place	A particular location where events occur
Type of Place	A non-absolute type of place
Culture	Linguistic, religious, ethnic, age-group, wealth, etc. of typical actors
Sophistication/ Security	Who already knows this, who could learn it, etc.
Topic/Usage	Drilling down into aspects and applications – not subsets
Granularity	Phenomena and details which are (and are not) ignored
Epistemology	Who wants/believes this content to be true?
Argument-Preference	Local rules for how to resolve pro-con argument disputes
Justification	Are things in this context generally proven, observed, on faith
Let’s	Local bindings of variables etc. that hold true in that context

The utilisation of context information is not straightforward since the composition of context information from low-level information describing the usage situation of a device is difficult. However, context-aware computing for mobile applications might gain from the issues presented in Table 1. Dividing context space into independent dimensions and use of relative representation of context information might facilitate the deployment of context information. Time and location can be considered as potential information sources and it may be useful to tie users’ tasks into those attributes. Additionally, time information already exists in mobile devices and location information is finding its way into them (FCC2000, Drane *et al*. 1998).

Schilit and Theimer (Schilit & Theimer 1994) formulated the concept of context for context-aware computing for the first time in 1994. They defined it to consist of location, identities of nearby people and objects, and changes to those. A more detailed definition for context is given by Schilit *et al*. (Schilit *et al*. 1994). It consists of categories; computing context, user context, physical context. Later definitions follow the definition by Schilit *et al*. (Schilit *et al*. 1994), but Brown (Brown *et al*. 1997) added time information, Dey (Dey *et al*. 1998) added the user's emotional state and focus of attention, and Pascoe (Pascoe 1997) added blood pressure information.

Dey and Abowd (Dey & Abowd 2000b) provide a general definition of context for context-aware computing:

'Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves'.

This practical definition however remains quite abstract as the abstraction is shifted from the definition of context to the definition of information and of knowledge. The definition does not indicate how context can be processed. Neither does it indicate how the relevancy of context can be acquired.

The work of Schmidt (Schmidt 2002) relates closely to the focus of this thesis since it focuses on prototyping mobile and static context-aware devices by integrating suitable sensors with signal processing methods into these devices. In the work model for context by Schmidt (Schmidt 2002) there is a set of relevant features for each context, and for each relevant feature a range of determined values. The context feature space is assumed to be categorical and hierarchically organised as presented in Table 2. There are two main categories with three subcategories each. Subcategories may include their own subcategories, conditions may consist of quantities, which can be sensed by sensors. Moreover, he proposes that measured sensor values can be processed further into features and thus their model provides some structure for context. Also, the importance of the history of the information is addressed. Schmidt (Schmidt 2002) proposes that it is simpler to implement context-aware systems using contexts on entity level. This can be considered as an approach to categorise and label context information.

Table 2. Context feature space (Schmidt 2002).

Main category	Subcategory
Human Factors	User
	Social environment
	Task
Physical Environment	Conditions
	Infrastructure
	Location

The above mentioned definitions are gathered from the seminal papers in the field of context-aware computing. The definitions show that there are number of relevant dimensions or factors in useful context information for mobility. In addition, approaches for modeling context information exist. Depending on the developer's viewpoint various categories and context sources are emphasised.

In this thesis we adopt the general definition for context by Dey & Abowd (Dey & Abowd 2000b). We consider the work model for context by Schmidt (Schmidt 2000) as a good approach to categorising and forming hierarchy for extracted context information.

Context-awareness

The term context-aware has been used regularly in this research field. However, we have used a noun context-awareness since the term context-aware is adjectival and in our opinion an adjective is not a suitable form for the definition.

Along with the first definition for context, Schilit and Theimer (Schilit & Theimer 1994) also formulated the definition for context-awareness. They state that:

'Applications are context-aware when they adapt themselves to context'.

The definition formulated by Pascoe (Pascoe 1998) explains context-aware computing as the ability of devices to detect, sense, interpret and respond to changes in the user's environment and computing devices themselves. Dey *et al*. (Dey *et al*. 1999) define context-awareness as the automation of a software system based on knowledge of the user's context. There exist several similar definitions that treat it as applications' ability to adapt or change dynamically their operability according to the operation state of the application and the user, see e.g. (Schilit *et al*.1994, Kortuem *et al*. 1998, Brown *et al*. 1997). The terms context-aware and context-awareness are used to define various types of adaptability. Common to these definitions is that they are closely tied to the user's context. Dey and Abowd (Dey & Abowd 2000b) provide a general definition for the term. They state that:

'The system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's tasks'.

Definitions for context-awareness presented in the literature are quite similar. There is no need to revise the definition for our purposes, and we adopt this general definition. However, this definition also arouses the question how to define the relevancy of information.

2.2 Information sources

Context information with methods for composing and representing it are crucial parts of context-aware computing. As Table 1 shows the content of context information is on a high abstraction level. Thus various types of sources, demanding tasks for extracting correct information and use of some background information to interpret extracted data semantically are required. This section presents a review of the sources for mobile context-aware devices. Figure 2 presents the main types of these sources. Location and time are obvious sources as explained in the previous section. Device processes represent mobile devices' internal information, for example, operating applications. Sources located over networks can provide information using, for example, the Internet. This thesis concentrates on a novel type of source, sensing elements integrated into mobile device.

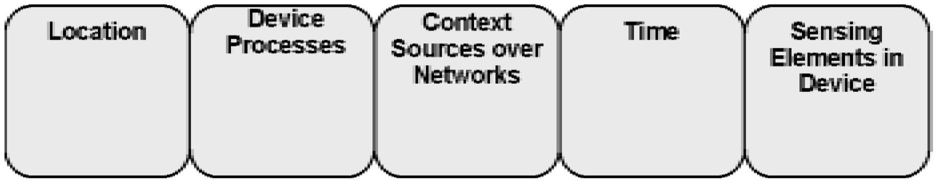


Figure 2. Main types of context information sources for mobile devices.

Location information is one of the most important sources of context information, particularly for mobile devices since they are proposed to support global positioning (FCC 2000, Drane *et al*. 1998). The first context recognition systems operated in an indoor environment and they mainly utilised location information, see e.g. (Schilit *et al*. 1994, Want *et al*. 1992, Want *et al*. 1995). In these positioning systems each room is equipped with a communication cell communicating with the mobile devices that the users are carrying. Wireless communication is carried out by exploiting infrared transceivers and a data communication protocol, e.g., Infrared Data Association (IrDa). The cells relay location information and user identification, sent by mobile devices, to a central computer, which further processes data into a suitable representation to be provided for applications.

Another method for indoor positioning is based on the combination of radio frequency (RF) and ultrasonic (Harter *et al*. 1999) systems. In this system a base station located in a building sends RF signals. Mobile devices are equipped with RF transceivers and ultrasonic transmitters while rooms in a building are equipped with a network of ultrasonic receivers located in the ceiling. After a mobile device receives an RF signal it in turn sends ultrasonic pulses. Location information is calculated in a central computer according to time delays of received ultrasonic pulses. Location information is delivered to a particular node via RF. RF based indoor positioning systems utilise, for example, wireless local area networks (Bahl & Padmanabhan 2000). In operational mode the RF link of a mobile device is connected to multiple wireless access points and it continuously compares signal strength and signal to noise ratio of communication signals to them. These signal characteristics are used to estimate the location of a device.

Outdoor positioning systems utilise mainly a global positioning system (GPS), which can also be integrated into context-aware devices (Abowd *et al*. 1997, Brown *et al*. 1997). Utilisation of positioning systems requires transformation from coordinate information to a representation suitable for applications. Particularly if a mobile device is a mobile phone, positioning methods based on communication signals between terminal and base stations can be utilised (Drane *et al*. 1998). Location information is a valuable information source since by further processing, it gives absolute and relative location, information about a device's movements, and nearby devices of a mobile device user.

Monitoring of a mobile device's internal processes provides explicit information on how device applications have been used. Essential information sources are basic communication applications such as, call, messaging and browsing. Other applications providing useful information are the ones that contain explicit information about user's scheduled tasks, preferences, social network and device settings in various situations. These types of applications include, for example, calendar, phonebook and profiles.

Mobile devices support various wireless networks such as, communication networks, and short-range ad-hoc networks. These communication methods enable the utilization of various types of context information located over networks, for instance, in the Internet.

Yet another source of context information is time. Usually contextual events occur repeatedly at certain times, for example, a user goes to work at a certain time during workdays while on weekends he/she does something else. Events may also occur together, that is they are related to each other. For example, usually when the user leaves his/her workplace he/she goes to a grocery store, or makes a phone call to a certain number. Thus, time information gives two types of context information; absolute and relative time (Lenat 1998). By fusing time information with other information the events can be ordered and certain events can be closely linked together.

Essential sensors suitable for observing the environment of a mobile device are thermometers, illumination sensors and microphones (Schmidt *et al*. 1999). Other sensors used to observe the environment include humidity sensor (Schmidt & Van Laerhoven 2001) and wearable camera, which is used, e.g., in (Aoki *et al*. 1999, Clarkson *et al*. 2000, Starner *et al*. 1998). Accelerometers are the main sensors used to monitor the activities of a mobile device user (Schmidt *et al*. 1999). When the user physically interacts with a device it can be detected with touch sensing system (Hinckley *et al*. 1999).

2.3 Context recognition

Context recognition is a process for extracting, fusing and converting relevant data from sources to a representation to be utilised in the applications. Mobility of devices adds dynamic characteristics for data that can be examined in context recognition procedures. Thus context recognition shares common aspects with information processing in particular signal processing, pattern extraction, data analysis and machine learning.

In the data processing, raw data from various sources are first preprocessed. Preprocessing typically includes noise removal, data calibration and reforming of data distributions (Pyle 1999). Secondly, data are further used in the formation of relevant information including various tasks for feature extraction. These tasks may also include utilisation of classifiers and clustering tools. The focus of our work is to develop methods for processing signals obtained from sensors integrated into mobile device to extract relevant context data. Thus we concentrate on introducing methods utilised in context information extraction from sensor signals.

The main objective for feature extraction and selection in context recognition is well-known in the area of pattern recognition: To select extracted features so that they provide a suitable solution for pattern recognition problem at hand and to enable the best classification performance. Features extracted from sensor signals describe small amounts of context information. They are called features (Clarkson *et al*. 2000), cues (Schmidt *et al*. 1999, Schmidt 2002) or context atoms (Himberg *et al*. 2001, Mäntyjärvi *et al.* 2002, Korpipää *et al.* 2003). In this thesis a name context atom is used since extracted features describe the smallest ‘atomary’ quantity of context information with semantic meaning.

Methods for extracting features in context recognition include feature calculation operations in the time domain, frequency domain or in other domains. Commonly used feature extraction methods in the time domain include the calculation of signal characteristics such as, mean and standard deviation (Van

Laerhoven & Cakmakci 2000, Schmidt *et al*. 1999), and moments (Clarkson *et al*. 2000). Moreover, the amount of zero crossings (Van Laerhoven & Cakmakci 2000) and the short time energy (Peltonen *et al*. 2002) are useful time domain features. In user activity recognition signals from two or more accelerometers can be processed into features describing the type and speed of walking by using feature calculation from waveforms (Lee & Mase 2001). The walking pattern recognition is also experimented with wavelet coefficients calculated from acceleration signals (Sekine *et al.* 1998). There exist several methods to calculate features in the frequency domain such as, coarse spectrum estimates (Clarkson *et al*. 2000), spectral centroid, spectral roll-off point and spectral flux (Peltonen *et al*. 2002). These are mainly used for audio signals.

Context recognition from several sources at an instant of time has similar aspects to sensor fusion. About sensor fusion, see e.g. (Iyengar & Brooks 1997). One of the basic methods for carrying out sensor fusion is the use of rule based systems. The first methods for extracting context information and linking information to applications suggested the use of a rule base system (Schilit 1995, Brown *et al*. 1997). Brown *et al*. (Brown *et al*. 1997) proposes the thresholding of the extracted variables i.e., sensors provide a set of variables with certain value ranges and applications are triggered when conditions match with predefined rules. An illustrative example of processing context information is the approach of Schmidt *et al*. (Schmidt *et al*. 1999) that describes the integration of sensing systems into mobile devices and a layered architecture for context-aware adaptation. The layout of their architecture is represented in Figure 3.

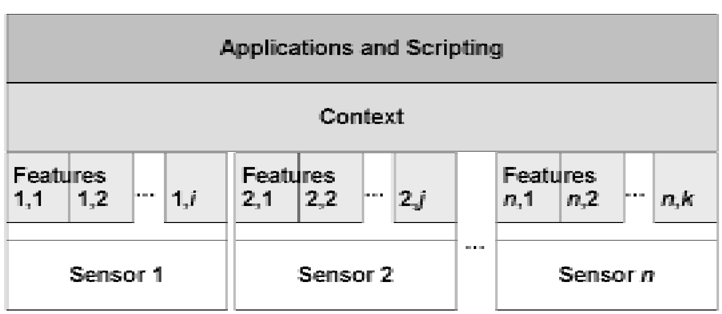


Figure 3. Architecture for processing context information (Schmidt *et al.* 1999).

The architecture shows an overview of the main phases in processing of context information from several sensors. In the lowest level are sensors, from which signals are processed to features that are combined to contexts. Context descriptions are linked to actions with predefined rules defined on the applications and scripting-layer.

One of the commonly used methods in sensor fusion is clustering of feature vectors. Unsupervised clustering of sensor information has been used in studying the extraction of contexts (Schmidt *et al*. 1999). One important aspect of context information is that context is a time varying concept, meaning that preceding contexts affect the following ones. This has been acknowledged and it has motivated several studies for context recognition, see e.g. (Clarkson & Pentland 1999, Aoki *et al*. 1999, Clarkson *et al*. 2000, Van Laerhoven & Cakmakci 2000). In these studies methods suitable for examining time varying processes have been employed.

Clarkson (Clarkson & Pentland 1999, Clarkson *et al*. 2000) applied HMMs for context recognition based on audio-visual information from a wearable camera and microphone. He experimented with several foreground and background HMMs with a varying number of states to find the models giving best context recognition performance. Contexts are labelled online during data recording by a user to obtain correct labels to be used in experiments. The results show that by using HMM models it is possible to generate context models with different time resolutions and obtain good recognition accuracies for various contexts, which in their case are the user’s actions in certain locations. Models generated have a different number of states and time resolutions. This suggests that a method for extracting various levels of contexts should be able to model them through simultaneous states in various time resolutions.

Aoki *et al*. (Aoki *et al*. 1999) experimented with dynamic programming to recognise previously visited locations using feature vectors extracted from a wearable camera. Dynamic programming is applied to compare the trajectories of time dependent feature vector sequences to recognise visited locations. The similarity of two feature vector sequences at a time instant, as well as the observed feature vector sequence and sequence in location in a dictionary are used. The location dictionary is selected to consist of the most distinctive sequences describing locations. The results show that dynamic programming approach enables the recognition of time dependent context information, Extracted sequences have different lengths suggesting that context recognition should be able to handle contexts of temporally different lengths.

The approach of Van Laerhoven & Cakmakci (Van Laerhoven & Cakmakci 2000) presents a context recognition architecture consisting of four layers: sensors, cues, classification and supervision. They propose that features extracted from accelerometer signals are classified using clustering with multiple SOMs, to which the k-nearest neighbor classifiers are applied to form a vector quantiser for representing various contexts. The user labels data during data recording. The time dependency of contexts is created by applying Markov chains that are used to model the state transitions between clusters. Their results show that time varying contexts can be recognised with a hybrid architecture.

Korpipää *et al.* (Korpipää *et al.* 2003) experiment with sensor-based context recognition using a Bayesian classifier. Naïve Bayesian networks are used to classify different contexts Classification is performed on features extracted mainly from audio signals. They propose a layered context recognition model consisting of layers for measurements, feature extration, quantisation, classification and analysing secuences. Their experiments cover layers up to the classification layer and examination of time dependency is omitted. They report that context recognition based on Naïve Bayesian classification gives a good context recognition performance. They discuss that a lot of background knowledge is required in modelling higher-level contexts using a Bayesian classifier. However, over-specific background information may cause the loss of generality in recognition (Korpipää *et al.* 2003).

The context recognition task is challenging since situations under recognition consist of a number of low-level events from different sources at a certain time instant and over time. Some background knowledge is needed when designing systems for obtaining higher-level contexts. This requires, e.g., labelling by the user that leads to a case where the user is in control of the composition of the context knowledge in a system. This has obviously some drawbacks; continuous labelling may become a nuisance. Additionally, a user may forget labelling and training in important situations thus a system may miss contexts. There is obviously a need for mechanisms for automatic labelling.

In addition to information from low cost sensors, audiovisual context information is useful for mobile devices particularly, for wearable computers. For PDAs and mobile phones video is not a suitable source since the recording and interpretation of visual information with these devices is difficult. Visual data must be recorded without the user's intention that may lead to recordings with different orientations of a device, and recordings from strange places, for example, inside a pocket or a bag. Audio information is one of the important context sources for mobile devices. However, placements of a device, for example, in a pocket may change the type of audio environment making the context recognition difficult.

2.4 Utilisation of extracted information

Utilisation of context-aware computing in controlling applications is a way of using implicit information as input in computer-aided devices. Schmidt (Schmidt 2000) has defined a term implicit human computer interaction:

'*An action performed by the user that is not primarily aimed to interact with a computerised system but which the system can understand as input*'.

The definition suggests that context-aware systems should continuously process information from several sources and have the ability to interpret relevant situations such as inputs to executing or controlling applications or services. To enable the effective utilisation of versatile information there should be some generic component in the system to manage information from sources and the recognised higher-level information.

2.4.1 Development issues

There exist some challenges in the development of context-aware systems. Some recognised problems related to the development of the systems have been presented in detail by Dey (Dey 2000). Some of them from the viewpoint of this thesis are briefly presented here. Currently, one problem in the development of context-aware systems is that most of the designed systems and their applications are built in an ad-hoc manner, mainly influenced by sensing technology preventing the extension and reuse of developed components (Dey 2000). The major problem is the lack of support for building and executing applications, and for creating relevant data abstractions. Dey (Dey 2000) has identified three factors slowing down the deployment of context-aware systems:

- A lack of variety of sensors for a given context type, meaning that there is no support for software for abstracting data from several useful sensors into suitable information representation.

- A lack of variety of context types, meaning that there is a lack of support for sources. To obtain rich representation a fusion of several types of context information is required.

- An inability to evolve applications issues from the building of applications in an ad-hoc manner. It prevents the reuse and extension of developed components.

Context may consist of various aspects that are distributed spatially in the nearby environment, similarly the system may have distributed components (Schmidt 2002). For example, sources maybe distributed over networks and in other mobile or static devices in the nearby environment. Thus context-aware systems can be considered as distributed systems. Want *et al*. (Want *et al*. 1995) discusses the requirements for distributed context-aware systems; they should be able to support the operability of distributed components and hide the complexity of the system in order to ease the design of applications, a context-aware system having components distributed over a network may have some drawbacks related to operating delays and communication, operating delays should be kept to a minimum and interruptions should be handled in both the mobile and the fixed part of the system and mobile devices could cache important information to tolerate occasional disconnections.

Mobile systems may have limited resources for processing context information, thus communicating context is essential. Cheverst *et al*. (Cheverst *et al*. 2000) have identified three main reasons why a context-aware system should utilise remote sources and communication in receiving and sensing context information. The reasons are; mobile devices may have insufficient memory to store the information required locally, a device itself can not detect some types of context information and they must be received over a network, and a device may obtain some dynamic information related to the environment and to other devices. When aspects of current context are spatially distributed to environment mobile devices could jointly sense and negotiate the current context (Mäntyjärvi *et al.* 2002). The method for jointly determinate context is presented in more detail in next chapter.

As showed in this section there are several technical concerns in the building blocks of context-aware systems. However, the crucial issue is how to obtain context representation to allow he utilisation of data in mobile applications. In the following we introduce briefly basic types of context-based mobile device applications.

2.4.2 Context-aware mobile applications

Context-aware computing enables the development of various novel applications and adds novel features to the existing ones. Here a summary of the basic types of applications is presented. They are categorised according to the characteristics of their operational features. The categorisation helps in defining the context-aware computing and working system architecture (Dey 2000). Moreover, it allows the examination of the functions required for various types of applications.

There are three successful trials for categorisation of context-aware applications and features. The first one is the classification provided by Schilit (Schilit 1995). His categorisation consists of two orthogonal dimensions; getting information or executing a command, manually or automatically. Using these dimensions he assigns four feature categories. For details see (Schilit 1995).

Pascoe (Pascoe 1998) introduced taxonomy to identify the core features of context-aware systems. The first feature is contextual sensing, that covers detecting, processing, and presenting information to a user. The second feature is contextual adaptation, the ability of an application or a service to adapt automatically. The third feature is contextual resource discovery meaning the locating and utilisation of resources and services. The fourth feature is contextual augmentation meaning the association of digital data to context.

Dey (Dey 2000) performs categorisation of applications by modifying the previous categorisations. The first category is presentation of information and services to a user. This corresponds to Schilit's (Schilit 1995) categories proximate selection and contextual command. Also, Pascoe's (Pascoe 1998) notion of presenting context information is included. The second category is automatic execution of a service, which maps into Schilit's (Schilit 1995) category context triggered actions. The third category is tagging of context information having the same meaning as Pascoe's (Pascoe 1998) category contextual augmentation. The presented models are quite similar; Dey's (Dey 2000) categorisation differs from the others in two main issues: Firstly, he does not consider the utilisation of local resources as a feature. It is called automatic contextual reconfiguration in Schilit's (Schilit 1995) work, and contextual resource discovery appearing in Pascoe's work. Dey (Dey 2000) states that this particular issue is included in the first two categories in his work. Secondly, he does not differentiate between information and services as done in Schilit's (Schilit 1995) work.

When we think about these categorisations from the viewpoint of mobile devices, Dey's (Dey 2000) categorisation seems the most suitable one, since mobile devices have several application types and to obtain reasonable categorisation the main issues; presentation of information and execution of application or feature must be addressed. Moreover, it is worth considering whether the categorisation between manual and automatic execution of applications or features is adequate or not since users may want to control the functionality of their devices. Automatic execution of application or feature may not be desirable. Thus, considering the execution of applications as manual execution may be more appropriate.

In Table 3 an overview and categorisation of typical context-aware applications and features for mobile devices is provided. The categorisation is based on the feature classes provided by Dey (Dey 2000). Applications are selected from the seminal papers in the field focusing on the context-aware computing for mobile devices. As Table 3 shows most of the applications present information to a user i.e., they let a user know the function in hand.

Table 3. Categorisation of typical context-based applications for mobile devices. Presentation is denoted as P., execution as E., tagging as T.

Context-based application	**P**	**E**	**T**
User interface adaptation, e.g., display mode and illumination adaptation, phone profile adaptation, information representation adaptation, operating tune adaptation (Bartlett 2000, Hinckley *et al.* 1999, Hinckley *et al.* 2000, Schmidt *et al.* 1999), Publication VII	X	X	
Triggering functions, e.g., gesture based functions (Publication I), power on/off, keyboard lock, (Hinckley *et al.* 1999, 2000),	X	X	
Access to information in the network (Want *et al.* 1995)	X	X	X
Collaborating on shared documents, e.g., drawings and texts (Want *et al.* 1995).	X		X
Control of nearby devices by using mobile device (Want *et al.* 1995).	X		X
User interfaces follow in adaptive manner (Schilit *et al.* 1994)	X		X
Presentation of context of people (Want *et al.* 1995, Schmidt *et al.* 2001, Keränen *et al.* 2003, Publication VIII)	X		X
Tourist Guide (Abowd *et al.* 1997, Oppermann & Specht 2000, Cheverst *et al.* 1998, 2000)	X	X	X
Electronic assistant, e.g., Conference assistant (Dey *et al.* 1999)	X	X	X
Information delivery and retrieval (Brown & Jones 2001, Marmasse & Schmandt 2000)	X		X
Reminder and notification (Beigl 2000, Dey & Abowd 2000a, Marmasse & Schmandt 2000, Schilit *et al.* 1994, Brown 1996)	X		X
Context-aware phonebook providing context information of the names in the phone book (Schmidt *et al.* 2001)	X		X
Communicate through paging and e-mail (Want *et al.* 1995)	X		X
Context-based phone call and forwarding (Want *et al.* 1995, Publication VIII)	X	X	X
Context-aware instant messaging (Nakanishi *et al.* 2000)	X	X	
Augmentation of meta-information with context (Rhodes 1997, Sumi *et al.* 2001, Suomela & Lehikoinen 2000)	X		X
Monitoring of user's activity and controlling of nearby device accordingly (Farrington *et al.* 1999)	X	X	
Wearable remembrance agent (Rhodes 1997, Suomela & Lehikoinen 2000)	X	X	X

2.5 Interaction with mobile context-aided systems

The one ambitious goal of context-aware computing is to enhance the human-computer interaction of computer aided devices. This is achieved by extracting implicit information from the mobile usage situations of devices and providing it to various types of applications and services that can adapt their appearance and functions accordingly. In this section we briefly discuss the human-computer interaction related to context-aware computing for mobile devices.

The input methods in mobile devices apply explicit input solutions, as keypads, soft and hard buttons, joysticks, rollers and touch screens used with finger or stylus, and one implicit input solution used in an explicit manner - speech input. The input methods are used to perform interaction tasks; navigation in a menu hierarchy, selection, executing and triggering of actions, setting configurations, character input and presenting information. Interaction tasks are used to interact with an application, where the individual input commands constitute actions. It usually takes long sequences of commands to perform an action.

Context-awareness has an impact on the interaction of mobile devices, as showed in Figure 4. The execution and triggering of applications and features can be context dependent (Brown 1998). The reorganization of menu structures and navigation shortcuts shortens the sequences of commands required to perform an action (Shneiderman 1998). Thus, it may be usefull to apply predictive menus and navigation shortcuts in the development of context-based functionalities. Sensors integrated into a device enable the development of novel methods for input and control and for developing multimodal interfaces (Hinckley *et al*. 2000). Configuration settings include, e.g., operation profiles that can change according to context information (Schmidt *et al*. 1999). Representation of information is common for all UI applications including, for example, the adaptation of screen illumination, device audio and font size of text showed on a display.

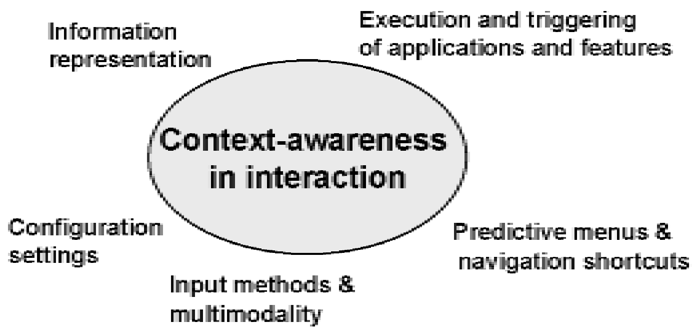


Figure 4. UI aspects advantaging from context-awareness, modified from (Mäntyjärvi et al. 2003).

Mobile devices are often used in multitasking situations and the user's attention is not fully concentrated on the device. The autonomously operating system may modify its functionality in a context dependent manner and adaptively without the need to draw the user's attention from his primary task (Franklin *et al*. 2002). As context dependent systems detect indications from the current overall situation, the number of input dimensions is increased and thus the complexity of the entire interaction system may increase. On the other hand, the context-dependent interaction system may cause additional mental load to a user since, due to adaptability, the interaction logic may not remain consistent. This may lead to confusing situations such as the lack of intuitiveness of the UI for a user. That is against the very basic principles for designing a UI suggesting that: To achieve a successful interaction system, the user's mental model of the system must remain consistent (Shneiderman 1998). Another issue related to the design of context-based systems is that the device's model of the context must be aligned with the user's idea of the context (Erickson 2002). These issues suggest that user interaction related aspects should be studied and evaluated carefully when developing context-dependent systems.

2.6 Summary

Context-aware computing is proposed as an enabling technology for adaptating different functions in computer aided devices. The adaptation is proposed to be carried out by recognising and composing implicit information from the usage situations and environment of a device and its user and to provide this information for various applications and services.

Context-aware computing is interdisciplinary. It shares common aspects for various fields of research, such as, computer science, information processing and wireless communications. It also has similar goals as to HCI research; to provide methods for improving the usability of computer-aided devices. Terminology and definitions related to context-aware computing are on a general level since the research on this field is in the early stage and there exist several types of context information that are not explicitly defined. However, to define context information explicitly is a challenging task since extracted context information is implicit, sometimes even unreliable.

Difficulties in the development of context-aware systems are mainly due to the lack of understanding of the dynamic characteristics and the structure of context information. Other issues slowing down the deployment of context-aware computing include the lack of sensing methods for gathering information from various sources, the lack of methods for fusing data and the lack of common representation. Various “real world” problems plague the monitoring of context sources, particularly sensors: noise, faulty connections, drift, miscalibration, wear and tear, and humidity, to name but a few. Particularly, in the case of mobile devices, reliable sensor data may be hard to obtain. Even if the acquired data is usable, many signal processing and recognition methods may consume significant amounts of memory, energy, and processing time. Still, the results obtained will often be ambiguous.

The context recognition of a mobile device from several context information sources located in a device is a challenging task since contexts are fuzzy, overlapping, and changing with time. When low-level contextual information describe some notions of context they will be just approximations at best. The recognition algorithms must adapt to the changing environment. For systems with supervised context learning, the users must label the meaning of the situation. This may require constant effort from the user, becoming a nuisance. On the other hand, unsupervised learning produces context clusters the meaning of which can be hard to interpret. Thus, context recognition based on learning also adds some difficulties.

3. Context recognition and utilisation

A fundamental problem associated with the development of context-aware systems is the composition of context information from various sources into a useful and rich representation. This is called a context recognition problem. This chapter presents an approach for the design and development of a context recognition procedure. Moreover, the utilisation of composed context information in studying novel applications and user interface adaptation of applications is presented. The objectives are to extract low-level data from several sensors to provide information representation, to examine the dynamic characteristics and structure of data and to extract information with several context levels enabling diverse use of data. Our approach is bottom-up from the data processing viewpoint since we compose higher-level contexts from low-level ones, rather than the top-down approach in which the objective is to first sketch a high-level entity and divide it into smaller subsets until primitives are recognised. About bottom-up/ top-down approaches, see e.g. (Engelmore & Morgan 1988). On the other hand, in data analysis some background information of the phenomena under examination is required. We have selected the bottom-up approach since we explore sensors as context information sources and examine composition and utilisation of context information from the source viewpoint.

The organisation of this chapter is as follows: Section 3.1 presents information flow in the procedure and the measurement setup. Our foundation for context representation is provided in Section 3.2. Section 3.3 introduces methods utilised while Section 3.4 shows how the representation is composed. Section 3.5 presents a method for collaborative recognition of the context of a group of mobile devices. The information fusion for composing higher-level contexts is provided in Section 3.6. Section 3.7 explains how context information is used in applications, and Section 3.8 summarises the approach.

3.1 Information flow and measurement setup

This section will give an overview of an approach to design and develop a context recognition procedure by introducing components of the procedure and information flow. Moreover, it will present the measurement setup for logging data used in experiments. The approach for context recognition comprises the following elements, as presented in Figure 5a; measurements-phase covering raw data measurements from the sources and preprocessing, representation-phase including feature extraction to compose low-level context information, context atoms. Thirdly, the collaborative context recognition-phase is described as a method for exchanging information between interconnected devices to provide more reliable descriptions of context. Finally, in the information fusion-phase the examination of data and extraction of higher-level contexts are carried out.

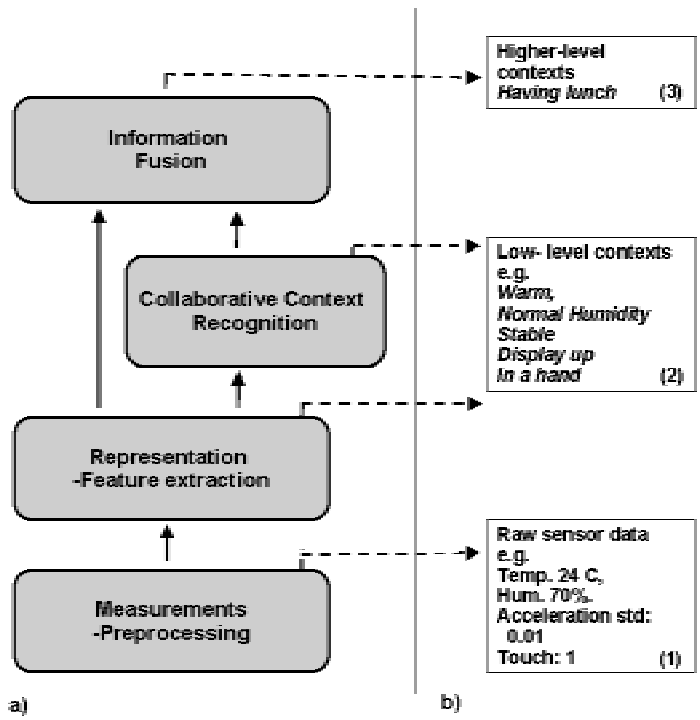


Figure 5. a) Context information flow. b) Interpretations of data on various context levels.

Various types of applications utilise information on different context levels. One objective of the context recognition procedure is to provide several context levels of data enabling versatile use of information. A schematic of information flow in the procedure and examples of data interpretations at various stages in the procedure are showed in Figure 5b. Data interpretations do not represent exact forms of data representation.

Information on the first context level is produced in the measurements-phase. Data consists of measurements that are represented as numerical measurement values describing, for example, temperature level, humidity level and amount of acceleration in a device. Characteristics of numerical measurement values are that they cannot represent semantics. Examples of the interpretation of sensor data are presented in box (1) in Figure 5b. The second context level consists of the low-level context representation, context atoms. The representation on this level is tied to the ad-hoc ontology developed for sensor-based context data for mobile devices. An example of the interpretation of the representation is presented in box (2) in Figure 5b. The principle of collaborative context recognition is demonstrated using the representation on the context atom level. Section 3.5 enlightens this issue. Higher-level contexts, presented in box (3) in Figure 5b, are produced in the information fusion-phase in which states at a certain time and over a certain period of time are combined. Representation used in higher-level contexts is processed from the context atom representation. Higher-level contexts are processed in symbolic representation and it is explained in detail in Section 3.6 and in Section 3.7.3.

Figure 5 shows that several levels of context data are provided in the procedure. This ensures the effective utilisation of various sources, the processing of several context types and the development of various types of applications. We have empirically selected a set of independent low cost sensors, which measure quantities relevant to sensor-based context-awareness (Mäntyjärvi 1999). Context information sources and the descriptions that they measure are presented in Table 4.

We have built the sensor box for examining context-awareness of mobile devices and it is used in context extraction experiments for this thesis. Sensors are placed into a small sensor box, which can be attached into a mobile device, Figure 6 (Mäntyjärvi 1999, Tuulari 2000). Two two-axes accelerometers are placed inside a sensor box and connected to measure accelerations of the device in three orthogonal directions. All other sensors measuring environmental conditions and skin conductivity are placed into the cover of the box. In the experiments, signals from sensors are A/D-converted and sampled at 256 Hz, 12-bit using DaqCard 1200® measurement board manufactured by National Instruments that is connected to a laptop computer. The audio signal is A/D-converted and sampled at 22.05 kHz, 16-bit, using a standard audio card of a laptop computer. Time labels are attached to measurement values. Data is stored on files, which are further processed to context information in an offline manner in the MATLAB® scientific computing environment.

Figure 6. Sensor box for examining context-awareness of mobile devices.

Table 4. Context information sources and their descriptions.

Information source (sensor)	**Description**
Accelerometer x-, y-, z- axis (type ADXL202JQC)	Measures accelerations of the device in orthogonal directions
Illumination (type IPL10530D)	Measures the level of the illumination in immediate environment of a device
Thermometer (type TMP36F)	Measure the level of the temperature in immediate environment of a device.
Humidity sensor (HIH-3605-B)	Measures the level of the air humidity in immediate environment of a device.
Skin conductivity sensor (self-made)	Detects a contact between a device and the hand of a user.
Microphone (customized)	Measures audio from immediate environment of a device.

3.2 On representation

According to the definition adopted in this work, context can be defined as any information that describes the relevant elements of a given situation (Dey & Abowd 2000b). This definition leaves open the exact sources, how to find a set of relevant elements of a current situation and methods for extracting relevant information. When considering the problem of context recognition for mobile applications the main sources include location, time, sensors integrated in a device, internal processes of a device and sources located over networks. In our approach we have used low cost sensors as sources: Illumination, temperature, humidity, audio, touch and accelerometers.

Extraction of relevant information from source signals includes several processing phases, such as, utilisation of signal processing methods to extract carefully chosen features and the composition of features to a low-level representation. The representation results in multidimensional data and explorative data analysis methods can be used in finding a set of relevant data elements of particular situations. What follows is an argument developed in Publication V. We may consider that by extracting relevant features from signals and by combining several features at any time instant, one can define a context. Thus, it is justified to see the problem of context recognition as a problem of the fusion of multiple information sources. In the following we introduce the fusion of multiple information context sources by presenting a principles of fusion at a time instant and over time.

To illustrate the context recognition problem let us consider the following example. At the given time instant using information from sources it would be possible to say, for example, that the user's situation is "walking in a corridor in the middle of the workday". However, this represents one level of the user's context. Additionally, a user can be "browsing a wireless service with a mobile device". Thus the user's situation consists of several simultaneous contexts that can be defined as "walking in a corridor in the middle of the workday while browsing a wireless service with a device".

We state that by combining information from different sources a better, and truer, description of the user's context is obtained. Generalizing this example we say that higher-level contexts can be obtained from the simultaneous fusion of lower-level information. It should be noted that in the example the lower-level of information, ”walking”, “in a corridor”, “in the middle of the workday” and ”browsing a wireless service with a device” are in turn generated from sources.

Sources in the mobile device used in this work produce sensor signals that can be processed to extract features. A set of features can be composed to represent a higher-level entity. The interpretation of context can thus be considered consistent from the lowest to the highest level of states. This approach thus provides a definition of context in terms of lower-level information, where the very lowest level of information is derived from data from sources. Here we denote context information C on a level i derived from source j as $C_{i,j}$, where i = 0 describes the lowest level data, sensor signals, i = 1 denotes the context atom level and i > 1 denotes higher context levels. On levels i > 1, the source information is not relevant since contexts are composed by information fusion.

Figure 7 shows an example of the states being combined to generate higher-levels of data at a time instant. Let the blocks $C_{1,1}(t)$, $C_{1,2}(t)$, $C_{1,3}(t)$, $C_{1,4}(t)$ describe the information on level 1. The blocks correspond to low-level information described in the example: ”walking”, “in a corridor”, “in the middle of the workday” and ”browsing a wireless service with a device”. $C_{1,1}(t)$, $C_{1,2}(t)$ and $C_{1,3}(t)$ are combined to form state of higher-level information $C_{2,1}(t)$. $C_{1,4}(t)$ and $C_{2,1}(t)$ are in turn combined to form the highest-level state $C_{3,1}(t)$, “walking in a corridor in the middle of the workday while browsing a wireless service with a device”. The idea in this example can be applied to context extraction from sensor signals.

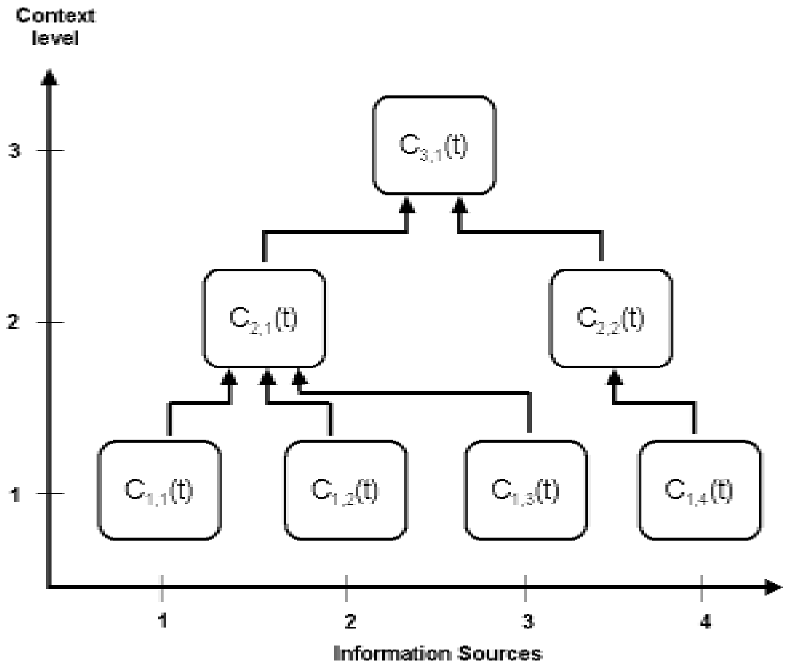


Figure 7. Composing higher-level states from various sources.

The time index *t* is used to indicate time dependency, i.e., the context is dynamic it changes with time, leading us to consider another mechanism for generating higher-level contexts. Another aspect in combining higher-level information is the fusion of the different states over time in a sequential manner. In keeping with the example of the user walking in a corridor, assume that previously the user's situation was identified as ”in his office”. Combining the three states, ”in his office”, ”walking in a corridor in a middle of the workday” and ”browsing a service with a device”, particularly if the service the user is browsing is the lunch menu the user's context could be considered as “going to have a lunch in the restaurant downstairs”.

Generalizing this idea, each state in the sequence represents a lower context level and a fusion of a sequence of lower states generates a higher-level information. This means that higher-level contexts can be generated over time. Figure 8 illustrates the generation of higher-level information $C_{2,1}(t3)$, $C_{2,2}(t5)$ and $C_{3,1}(t5)$ by fusing a sequence of the states over a time interval. Time dependent higher-level information can be generated by fusing several states at a time instant and simultaneously fusing different context states over time.

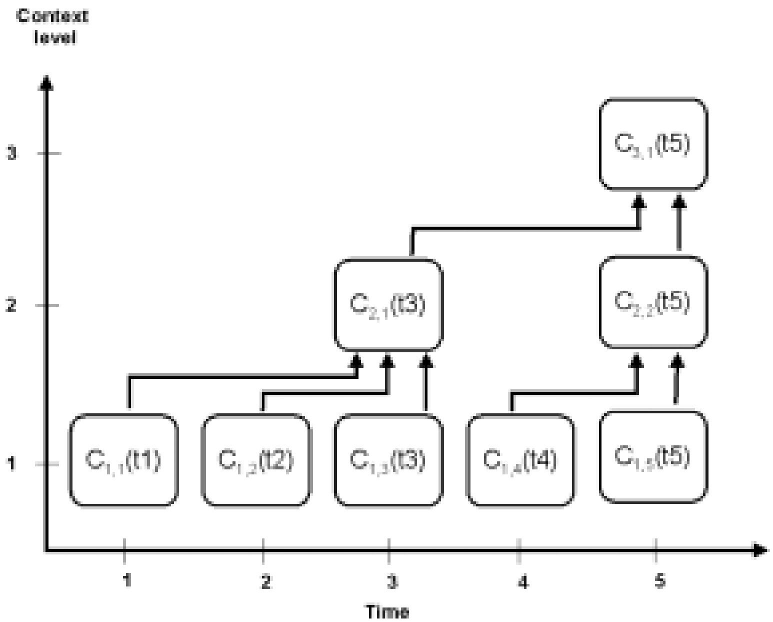


Figure 8. Composing higher-level states by fusing states over time.

Figures 7 and 8 suggest models of contexts to be hierarchical tree structures. With mobile devices situations change quite rapidly and there are several overlapping contexts on different levels at a certain time instant. Several low-level states affect various higher-level states and context models are hierarchically complex structures that evolve with time. This real world fact makes it difficult to model contexts for mobile applications beforehand, hence understanding of the structure and dynamic characteristics of the data as well as some background knowledge are required. This makes the examination of real data with explorative data analysis methods desirable. Models of information fusion in Figures 7 and 8 present basic ideas on how to compose information. However, they do not represent the real world situation of context information fusion and they must be extended.

In Figure 9 the idea of fusion of various states over time in a real world case is illustrated. The highest level state $C_{3,1}(t3)$ is composed by fusing level 2 states $C_{2,1}(t1)$ and $C_{2,2}(t2)$ over time. Another highest level state $C_{3,2}(t4)$ is composed fusing level 2 states $C_{2,1}(t1)$, $C_{2,2}(t2)$ and $C_{2,3}(t3)$ over time. $C_{2,1}(t1)$ is fused from $C_{1,1}(t1)$ and $C_{1,3}(t1)$ at a time instant, $C_{2,2}(t2)$ is composed fusing $C_{1,2}(t3)$ and $C_{1,3}(t1)$ over time and $C_{2,3}(t1)$ is fused from $C_{1,4}(t1)$ and $C_{1,5}(t1)$ at a time instant. Context structures evolve over time and various states may affect other states in the future. This is indicated with dashed arrows. Moreover, context structures can grow, that is there can be a large number of levels.

As a summary, after the lower-level states have been ”recognised” their combination constitutes a new context. Thus an approach for recognising context can be considered as a transformation from a collection of states on a previous level to a state on a next level.

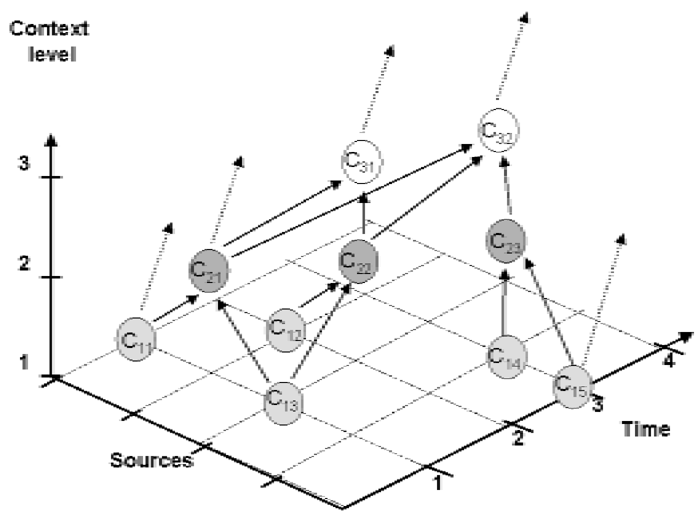


Figure 9. Generalised illustration of formulation of context states.

3.3 Methods

Methods introduced in this section are applied in this thesis for composing representation, for collaborative context recognition and for information fusion. Moreover, they are used for demonstrating utilisation of context information in applications. Methods are presented briefly focusing on basic principles. Justifications for selecting these methods for the context recognition procedure are also provided.

3.3.1 Time domain

Signal characteristics

There exist several methods for calculating signal characteristics in the time domain. In this research we have utilised common parameters such as, mean, root mean square (RMS) value, weighted mean, and standard deviation (STD). Mean, RMS value and STD have been used in calculating features in context atom extraction since, in some cases, which will be explained later, they can be processed to describe semantic context information.

Weighted mean is applied in collaborative context recognition since it provides a basic method for calculating the average for a set of samples by addressing weights for certain samples. This is described in more detail in Publication VI.

Filtering

In the context atom extraction there is a need to examine signals within certain frequency bands setting requirements for filtering, for instance, sharp cut off frequencies. We have used infinite impulse response (IIR) filters. The general IIR filter equation is presented as (Ifeachor & Jervis 1997):

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=1}^{M} b_i y(n-i) \,, \qquad (3.1)$$

where a_i and b_i are the coefficients of the filter, $x(n)$ is an input sample, the current output sample $y(n)$ is a function of past outputs as well as present and past input samples. M and N describe the order of the filter (i.e., number of delay elements) If M = 0 then the system is a finite impulse response filter. The specifications of IIR filters are equivalent to similar analogue filters and high order IIR filters also have sharp transition bands and high throughput. We have used a Butterworth filter that provides a good approximation of the ideal low pass filter response. The stability of filters can be determined in the z-plane; the filter is stable if poles lie within the unit circle or when poles are coincident with zeros on the unit circle so that their effects are nullified (Ifeachor & Jervis 1997). We have used filtering in processing of low-level context information from illumination signals. Data processed with filtering has been used in Publications I, III–VII.

3.3.2 Frequency domain

The frequency domain provides methods for extracting expressive features for processing context data. Feature extraction in the frequency domain has been utilised particularly in the analysis of environmental audio (Peltonen *et al*. 2002). The first step is to estimate a signal spectrum. A common method for this purpose is the estimation of the power spectral density $P(f)$ (Ifeachor & Jervis 1997):

$$P(f) = \sum_{m=0}^{N-1} c_{xx}(m)\exp(-j2\pi fm)\,, \tag{3.2}$$

where m = time delay, f = frequency, N = the length of the sample vector and

$$c_{xx}(m) = E\{[x(n) - \bar{x}(n)][x(n+m) - \bar{x}(n)]\} \tag{3.3}$$

is the autocovariance function of signal $x(n)$ at a lag m. To detect a peak frequency of the estimated spectrum we register an index of a frequency with maximum energy.

The main motivation for using frequency domain methods is that context information to be extracted may appear as periodic signals, e.g., walking. We have utilised frequency domain methods in extracting features from both illumination and acceleration signals. Low-level context information extracted using these methods has been used in Publications III–VII.

3.3.3 Wavelet analysis

Wavelet analysis is a widely used method in the field of signal processing. An advantage compared to Fourier analysis is that it provides a method to examine local time-frequency characteristics of a signal. In wavelet analysis a signal under inspection is divided into different frequency components in which a resolution is matched to their scales (Daubechies 1992).

Here we briefly introduce wavelet analysis presenting continuous and discrete wavelet transforms and a common method for performing wavelet analysis, multiresolution analysis. The introduction is based on (Daubechies 1992).

Continuous wavelet transform (CWT) for a continuous signal $x(t)$ is defined as:

$$Cx(a,b) = \int_{-\infty}^{\infty} x(t)\psi_{a,b}(t)dt\,, \tag{3.4}$$

where the variables a, b determine scaling and shifting. Variations of basic wavelet Ψ for different resolutions and locations is given by the wavelet function $\Psi_{a,b}(t)$:

$$\Psi_{a,b}(t) = |a|^{-\frac{1}{2}}\Psi\left(\frac{t-b}{a}\right), \tag{3.5}$$

where a and b have real values, $a \neq 0$. Wavelets are indexed with the parameters a denoting the width of the wavelet, and b determining the location of the wavelet in time. These issues give wavelet analysis the property of examining local time-frequency characteristics of a signal $x(t)$. A large value of the scaling variable a stretches the basic wavelet Ψ allowing the examination of the low-frequency components of the signal. The small value of a gives a compressed version (in time scale) of the basic wavelet and thus allowes the analysis of the higher frequency components of a signal.

A discrete version of wavelet transform is obtained when a and b are restricted to discrete values $a = a_0^m$, $b= nb_0a_0^m$:

$$Cx(m,n) = a_0^{-\frac{m}{2}} \int_{-\infty}^{\infty} x(t)\psi\left(a_0^{-m}t - nb_0\right)dt\,. \tag{3.6}$$

In the discrete wavelet transform (DWT) variables m, n = 0, ±1, ±2, ±3, …, determine scaling and shifting, and $a_0 > 0$, $b_0 > 0$ are fixed. By choosing values $a_0 = 2$, $b_0 = 1$ one sets the wavelet functions $\Psi_{m,n}(t)=2^{-m/2}\Psi(2^{-m}t - n)$ which are used in multiresolution analysis.

In multiresolution analysis the approach is to represent the original signal with j various resolution levels (Daubechies 1992, Cohen & Kovacevic 1996). The reconstruction of the signal $x(t)$ from the discrete scaling functions $\Phi_{K,k}(t)$, discrete wavelet functions $\Psi_{j,k}(t)$ and wavelet coefficients $c_j(k)$ and $d_K(k)$ with different scales 2^{j} and 2^K containing detailed information about signal $x(t)$ is expressed as (Tikkanen 1999):

$$x(t) = \sum_{j=1}^{K} \sum_{k \in Z} c_j(k) \psi_{j,k}(t) + \sum_{k \in Z} d_K(k) \Phi_{K,k}(t). \tag{3.7}$$

The multiresolution analysis provides wavelet coefficients that describe a signal $x(t)$ with various resolutions at an instant time t. The coefficients describe details of the signals in a compressed format and they can be used as an input for further processing. In this research wavelet coefficients are used as features for user activity classification as described in Publication II. Wavelet analysis is selected as feature extraction method since it provides a way of examining local time-frequency characteristics of a signal that is important when studying acceleration signals describing various types of walking.

3.3.4 Fuzzy sets

The objective in context atom extraction is to extract features describing notions from the real world, for example, level of illumination. The shift from dark to normal illumination conditions and to bright is fuzzy rather than crisp. Quantising the dynamic range of a feature with fuzzy sets results in more expressive representation of context information. To produce expressive context representation we have used fuzzy membership functions.

The concept of fuzzy sets has been introduced by Zadeh (Zadeh 1965). Let X be a space of points, x is an element of X. Fuzzy set A is characterized by a membership function $\mu_A(x)$ associating points in X, with the value of $\mu_A(x) \in [0,1]$ at x representing the degree of membership of x in A (Zadeh 1965). Zero denotes a null degree of membership and one denotes full degree of membership. An example illustrating fuzzy membership functions $\{\mu_A(x), \mu_B(x), \mu_C(x)\} \in [0,1]$ is presented in Figure 10.

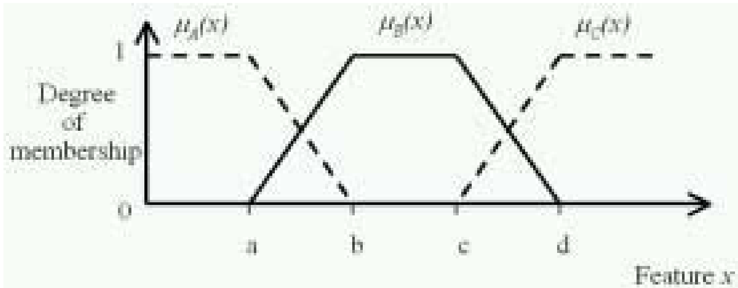


Figure 10. Fuzzy membership functions.

We have applied fuzzy sets for quantising features using one of the basic fuzzy membership functions, the trapezoidal membership function. Mathematical expression for the trapezoidal fuzzy membership function (function μ_B(x) in Figure 10) is presented as:

$$\mu_B(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x < b \\ 1, & b \le x < c \\ 1 - \dfrac{x-c}{d-c}, & c \le x < d \\ 0, & x \ge d \end{cases} . \tag{3.8}$$

Feature quantisation with fuzzy membership functions provides a way to give semantic meaning to extracted features. It is required that features are already scaled to a certain range and the dynamics of features is studied with correspondence to real world situations. When several features are quantised using fuzzy sets the fuzzy logic operations can be utilised to combine several features for control purposes (Driankov *et al*. 1996).

The basic operations of ordinary sets such as, complement, intersection and union are also applicable to fuzzy sets (Zadeh 1965). The membership of the complement of a fuzzy set *A* is

$$\mu_{\neg A}(x) = 1 - \mu_A(x) . \tag{3.9}$$

The membership of the intersection between two fuzzy sets *A* and *B* for aggregating two membership functions can be expressed as

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) . \qquad (3.10)$$

The membership of the union between two fuzzy sets A and B is

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) . \qquad (3.11)$$

The operations (3.10) and (3.11) can be applied to several fuzzy sets at a time. We have applied these fuzzy membership functions to quantise features extracted from sensor signals. Composed features have been used in experiments described in Publications III–VII. The operations of fuzzy sets have been applied in experiments for controlling context-aware applications as described in more detail in Publication VII.

3.3.5 Principal component analysis

Principal component analysis (PCA) is one of the basic methods for the analysis of multidimensional data, particularly in data compression (Jolliffe 1986). The objective in PCA is to find a smaller set of variables with less redundancy, maintaining the representation of the new variables on the same level as the original variables (Jolliffe 1986, Hyvärinen *et al*. 2001). This is done by maximising the variance each variable captures.

In PCA of an n-dimensional zero mean data vector $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^{\mathrm{T}}$ the n PCs $\mathbf{x'} = [x'_1\ x'_2\ \ldots\ x'_n]^{\mathrm{T}}$ are obtained by projection $\mathbf{x'} = \mathbf{U}^{\mathrm{T}}\mathbf{x}$. Projections to principal directions $\mathbf{x'}$ are defined by the eigenvectors $\mathbf{u}_i$, which exist as columns in matrix $\mathbf{U}$. The eigenvalues in matrix $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ corresponding to eigenvectors $\mathbf{u}_i$ are arranged in decreasing order $\lambda_1 > \lambda_2 > \ldots > \lambda_n$. They determine the variance that each PC captures, λ_1 capturing the largest variance. In practice the eigenvectors are calculated from the (sample) covariance matrix $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$. The PCs are uncorrelated and their variances are maximised to the directions orthogonal to direction of previous PC. Another projection of the data $\mathbf{v} = \mathbf{V}\mathbf{x}$ called whitening is obtained when PCs are uncorrelated and scaled to have unit variances using transformation $\mathbf{V} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^{\mathrm{T}}$.

One can also consider these projections as finding interesting directions of data (Hand *et al*. 2001). Maximising the variance of the PCs is a basic approach to carry out PCA. Other methods for PCA exist such as, minimum mean square error compression and the neural approach (Hyvärinen *et al*. 2001).

We have used PCA and in particular whitening in feature extraction from multidimensional sensor signals as presented in Publication II. PCA is also used in examining context information fusion particularly, to compress data for the examination of the dynamic characteristics and structure of the context atom data with several visualisations as presented in Publication IV.

3.3.6 Independent component analysis

The independent component analysis (ICA) is presented as a method for finding statistically independent sources from observations that are mixtures of sources. About ICA, see e.g. (Comon 1994, Jutten & Herault 1999, Hyvärinen *et al*. 2001). In the basic version of ICA the assumption is that the n-dimensional data vector $\mathbf{x} = [x_1, \ldots, x_n]^T$ is observed. The elements of the observation $\mathbf{x}$ are mixtures of the statistically independent, non-gaussian, unknown sources, which are elements of an n-dimensional random vector $\mathbf{s}$. The observed signals $\mathbf{x}$ are related to unknown source signals $\mathbf{s}$ by an unknown $n \times n$ mixing matrix $\mathbf{A}$, denoted as $\mathbf{x} = \mathbf{A}\mathbf{s}$. Here the additive noise component of the ICA model is omitted. The problem is to solve estimates of source signals $\mathbf{s}'$ and demixing matrix $\mathbf{B}$ knowing the observed signal $\mathbf{x}$ as $\mathbf{s}' = \mathbf{B}\mathbf{x}$. An illustration of the procedure for separating independent components is presented in Figure 11.

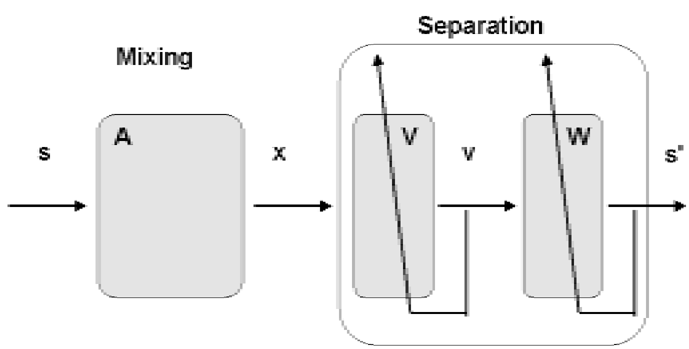


Figure 11. Schematic of the procedure for independent component analysis. Modified from (Vigário 1999).

The approach for solving **s'** consists of two main phases included in the separation, as Figure 11 presents. First, observation signal **x** is preprocessed by whitening using the matrix **V** that gives the whitened data vector **v**. Secondly, the separating matrix **W** is determined from $\mathbf{s'} = \mathbf{W}^{T}\mathbf{v}$ so that the estimates of source components **s'**, independent components, are found.

There are some implications from the ICA for independent components (Hyvärinen & Oja 2000). Firstly, variances of independent components cannot be determined. This is because in the ICA model **s** and **A** are unknown. If there exists any scalar multiplier in any of the sources s_i, it can be eliminated by dividing the corresponding column $\mathbf{a}_i$ of **A** by the same scalar. Thus, an assumption in ICA is to have unit variances for source signals. Another property is that the signs of independent components cannot either be determined since a random multiplier can be –1. The third property of independent components is that their mutual order cannot be determined. Because **s** and **A** are unknown thus, the order of the components can be changed and any of them can be called the first one.

The basic ICA model works under a set of assumptions. First, sources are assumed to be statistically independent. Secondly, source signals carry no noise. Thirdly, the number of linear mixtures n is equal to or larger than the number of independent components m, $n \geq m$. (Hyvärinen 1999)

Solution for the ICA model is initialised considering the Central Limit Theorem. It claims that the distribution of a sum of independent random variables approaches a gaussian distribution, under certain conditions (Papoulis 1991). Finding of the independent components is transformed to an idea of finding independent source signals from their linear mixture by maximising the non-gaussianity of the components (Hyvärinen *et al.* 2001).

Measures for non-gaussianity of a zero mean random variable z presented in this context include kurtosis, which is the fourth order cumulant written as:

$$kurt(z) = E\{z^4\} - 3\left(E\{z^2\}\right)^2, \tag{3.12}$$

that is zero for gaussian variables. Random variables can have negative or positive kurtosis. Supergaussian random variables, i.e., variables with sharp probability density function and long tails, e.g., Laplace distribution have positive kurtosis. Random variables having subgaussian probability density function distribution, e.g., uniform distribution, have negative kurtosis. Kurtosis can be used as a measure for non-gaussianity since it is computationally simple (Hyvärinen & Oja 2000). However, it is not a robust measure being quite sensitive to outliers (Hyvärinen & Oja 2000).

Another measure for non-gaussianity is negentropy, which is defined using differential entropy (Hyvärinen *et al*. 2001). Negentropy is zero for gaussian variables and positive otherwise. In the case of continuous valued random vector **z** the differential entropy defined with the density $f(z)$:

$$H(\mathbf{z}) = -\int_{-\infty}^{\infty} f(\mathbf{z}) \log f(\mathbf{z}) dz \, . \tag{3.13}$$

Negentropy is defined as:

$$J(\mathbf{z}) = H(\mathbf{z}_{\text{gauss}}) - H(\mathbf{z}), \tag{3.14}$$

where $\mathbf{z}_{gauss}$ is a gaussian random vector of the same covariance matrix as **z**. The calculation of negentropy using differential entropy is difficult. Thus the measure is approximated with higher-order cumulant approximation with suitably selected non-quadratic function G and a gaussian variable v of zero mean and unit variance (Hyvärinen & Oja 2000):

$$J(z) \propto [E\{G(z)\} - E\{G(v)\}]^2 . \tag{3.15}$$

These approximations can be used in ICA methods when suitable functions for G are selected (Hyvärinen *et al.* 2001).

Another approach for finding solution to ICA problem is justified with the relation between negentropy and mutual information (Hyvärinen & Oja 2000). Mutual information is a measure for dependence between random variables (Papoulis 1991). Mutual information I between n random variables z_i, $i = 1, \ldots, n$ is defined as:

$$I(z_1,...,z_n) = \sum_{k=1}^{n} H(z_i) - H(\mathbf{z})\,. \qquad (3.16)$$

The relation between negentropy and mutual information is (Hyvärinen & Oja 2000):

$$I(z_1,...,z_m) = D - \sum_{i=1}^{m} J(z_i)\,, \qquad (3.17)$$

where D is a constant. Thus, maximising the non-gaussianity of the mixture is equivalent to minimising the mutual information (Hyvärinen & Oja 2000). The objective is now to solve $\mathbf{W}$ to obtain independent components $\mathbf{s'}$.

There exists a variety of methods for estimating the mixing matrix and the independent components under different assumptions on the data (Hyvärinen *et al*. 2001). In this research we have applied the FastICA algorithm, where the objective is to minimise the approximation of mutual information with contrast functions (Hyvärinen & Oja 1997, Hyvärinen *et al*. 2001).

The main motivation for utilising ICA in this research is that independent components can also be considered as interesting directions of signals. We have used ICA to find interesting directions of accelerometer signals when examining user activity recognition as presented in Publication II. Moreover, ICA is used to study the independence of context atoms as presented in Publication IV.

3.3.7 Vector quantisation through clustering

Clustering can be considered as a method for partitioning the data into groups of similar data items in an unsupervised manner. Two common clustering methods k-means clustering and the SOM are used to find prototype vectors of data.

K-means clustering

One of the fundamental clustering tools is k-means clustering also known as the generalised Lloyd algorithm (Gersho & Gray 1992). The task in k-means is to divide data vectors $\mathbf{x}_n$, $n = 1, ..., N$ into K predefined clusters, sets S_i, $i = 1, ..., K$.

The cluster centers are calculated as the mean points for each set K. The cost function L is defined to be minimised as follows: the sets S_i must be such that the Euclidean distance within each cluster is set to minimum.

$$L = \frac{1}{N}\sum_{i=1}^{k}\sum_{n\in S_i}\left\|\mathbf{x}_n - \bar{\mathbf{x}}_i\right\|^2 \quad . \qquad (3.18)$$

The same cost function is used in the exploration of higher-level contexts with segmentation experiments and visualisations of context atom data time-series as Publication III shows. The detailed examination of a segmentation method is presented in (Himberg *et al*. 2001).

When examining context information the clusterings obtained with k-means and the self-organising map are used to vector quantisation in which the objective can be presented as: A vector quantiser Q of size m is a mapping of a set of data vectors M to a finite set G

$$Q: M \rightarrow G,$$

where vectors of $G = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m\}$, $\mathbf{g}_i \in M$, represent a certain region of the vector space. The collection of prototype vectors $\mathbf{g}_i$ is called a codebook (Theodoridis & Koutroumbas 1999).

Self-organising map

The self-organising map of Kohonen (SOM) (Kohonen 1995, Haykin 1999) is a neural network, which forms spatially organised feature maps from n-dimensional input vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ in an unsupervised manner. Prototype vectors are placed to approximate a high-dimensional input data in an ordered fashion. The method is biologically inspired mimicking the human sensory input mapping in the brain. Each neuron j has an associated n-dimensional weight vector $\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]^T$, $j = 1,2,\ldots,l$, where l corresponds to the total number of neurons in the network. The particular neuron i satisfying the condition, which minimises the Euclidean distance:

$$i(\mathbf{x}) = \arg\min_j \| \mathbf{x} - \mathbf{w}_j \|, \; j = 1,2,\ldots,l, \qquad (3.19)$$

is the winning neuron for the input vector **x**. The topological neighborhood $n_{j,i(\mathbf{x})}$ of the winning neuron $i(\mathbf{x})$ is:

$$n_{j,i(\mathbf{x})} = e^{\left(-\frac{d_{j,i}^2}{2\sigma^2}\right)}, \qquad (3.20)$$

where $d_{j,i}$ is the lateral distance from the winning neuron i to the excited neuron j and σ describes the width of the topological neighborhood. The topological neighborhood decays with the distance from $i(\mathbf{x})$. One of the typical choices of $n_{j,i(\mathbf{x})}$ is the Gaussian function decreasing monotonically as a function of the lateral distance.

The width of the topological neighborhood σ is time dependent shrinking with time. During the training at iteration n weight vectors of the network are shifted towards the input vector **x** by

$$\mathbf{w}_{ji}(n+1) = \mathbf{w}_{ji}(n) + \eta(n) n_{j,i(\mathbf{x})}(n)(\mathbf{x} - \mathbf{w}_{ji}(n)), \qquad (3.21)$$

where $\mathbf{w}_{ij}(n+1)$ is the updated weight vector, η (n) is the learning rate.

The SOM forms a topographically ordered feature map (Haykin 1999, Kohonen 1990). The SOM has four special properties that make it a useful method as an unsupervised clustering tool (Haykin 1999). Firstly, the feature map represents an approximation of the input space. Secondly, the feature map is topographically ordered so that spatial location of the neurons in the lattice corresponds to topological order of the input patterns. Thirdly, the feature map reflects variations in the statistics of the input distributions; regions in the input space from which the sample vectors **x** are selected with a high probability of occurrence are mapped onto larger domains of the output space. Fourthly, the feature map is able to select a set of the most suitable features to describe a non-linear distribution of the data in input space from which sample vectors **x** are drawn. The Kohonen model can be considered as a vector quantiser producing prototype vectors of the data. Additionally, the SOM facilitates data compression since it provides topographical mapping that generates and places a fixed number of prototype vectors into a higher dimensional input space (Haykin 1999). These are the primary motivations for selecting the SOM as a method to examine the unsupervised clustering of static accelerations during gestures, as presented in Publication I.

In this research vector quantisers designed with k-means are used to explore the structure of higher-level contexts data with various visualisations as Publication III presents.

3.3.8 Classification

The objective in classification is to assign an unknown object to the correct category according to its feature vector (Hand *et al*. 2001). A criterion for classifying objects to a certain classes is formed by presenting examples of objects with known classes to a classifier.

Multilayer perceptrons

Multilayer perceptron (MLP) neural networks are widely applied methods for problems in the fields of, e.g., pattern recognition and data analysis. A MLP neural network consists of neurons, computing nodes, which form neuron layers. An input layer feeds the input signal into computational nodes of one or more hidden layers, propagating to computation nodes of an output layer, which produces the output pattern of MLP. The basic functionality of the MLP is based on the supervised training process using the back-propagation algorithm (Haykin 1999). The strength of MLPs in solving difficult pattern recognition tasks relies on the non-linearity of the computation nodes. A neuron j in a local field v_j^l in layer l receives a signal y_i^{l-1} from the output of neuron i in the previous layer l-1. The signal is weighted with the corresponding weight w_{ji}^l between neuron i and j. The local field of a neuron j on layer l assuming that there are m inputs can be expressed as:

$$v_j^l = \sum_{i=0}^{m} w_{ji}^l y_i^{l-1} \ . \qquad (3.22)$$

The weighted input is scaled with a non-linear scaling function φ giving an output of neuron j in a layer l:

$$y_j^l = \varphi_j\left(v_j^l\right). \qquad (3.23)$$

The functionality of the back-propagation algorithm consists of two main computational phases (Haykin 1999); forward pass and backward pass. The summary of the signal-flow graph of back-propagation learning is presented in Figure 12. In a forward pass an output for input signal $x(n)$ is calculated layer-by-layer with Eqs. (3.22) and (3.23). Outputs of different layers are annotated as follows: If a neuron j is in first hidden layer, $l = 1$, $y_i^0 = x_j$, is element of an input pattern x. If neuron j is in the output layer, $l = L$, $y_i^L = o_j$. In this computation phase weights between neurons are not altered.

Secondly, a backward pass is executed. In this phase the computation starts from an output layer propagating to a first hidden layer. An error signal of node j, e_j between js elements of output o_j and the desired output d_j is calculated:

$$e_j = d_j - o_j. \qquad (3.24)$$

Error signals are used to calculate local gradients δ_j^l for neuron j in layer l:

$$\delta_j^l = \begin{cases} e_j^L \varphi'_j\left(v_j^L\right), & \text{for neuron } j \text{ in output layer } L \\ \varphi'_j\left(v_j^l\right)\sum_k \delta_k^{l+1} w_{kj}^{l+1}, & \text{for neuron } j \text{ in hidden layer } l \end{cases}. \qquad (3.25)$$

Local gradients indicate changes in weights. Finally, weights between neurons j and i in layer l are adjusted with the generalisation of a delta rule:

$$w_{ji}^l(n+1) = w_{ji}^l(n) + \alpha\left[w_{ji}^l(n-1)\right] + \eta\delta_j^l(n)y_i^{l-1}(n), \qquad (3.26)$$

where α is the momentum constant for stabilising the update of weights, while η is a learning-rate parameter. In the training process the two computational phases mentioned above are repeated with new sets of training examples in the network until the stopping criteria is met.

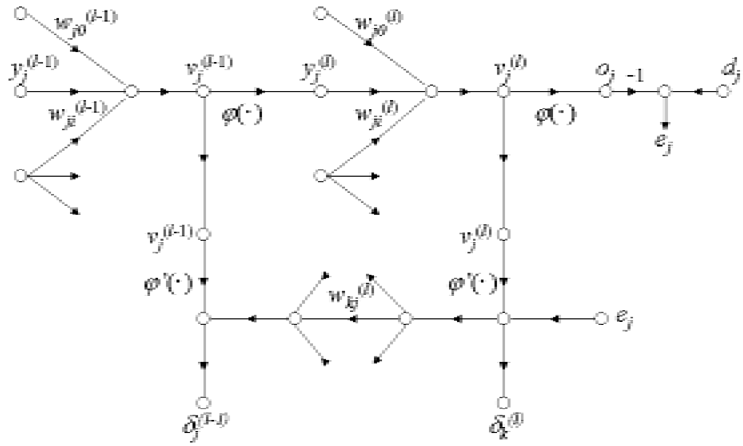


Figure 12. The summary of the signal-flow graph of back-propagation learning. Upper part of the graph presents the forward pass while the lower part of the graph represents the backward pass (Haykin 1999).

After training, in which example patterns with the desired outputs are used to adjust the neuronal units, the network is used to classify input patterns with no class label into known classes.

We have used MLP classifiers to examine the separation of different walking patterns in Publication II since it provides simple and powerful tool for classification. Moreover, MLP has been used to classify acceleration signal patterns to produce context atoms describing the orientation of a mobile device.

3.4 Composing representation

This section presents how we have applied presented methods for extracting relevant low-level context information for composing representation for context data. The main motivation is to find representation with semantic meaning to facilitate the development of context-based mobile applications methods for recognising contexts.

The categorisation of context information is suggested to be achieved by relating context to entities (Schmidt 2002). In that particular conceptual model context consists of an entity, e.g., artefact, type of entity, e.g., artefact's name, and context of entity. This is practical way to present conceptual model when entities are properly defined. Our approach is alike although we do not use name entity but category. Categories with context atom types with examples of context atoms are presented in Table 5. Each context atom type includes a set of context atoms and each context atom has semantic meaning that describes a concept from the real world. This ensures that ontology for representation is clear for the utilisation of context information. This section presents the context atoms and methods for extracting them.

Table 5. Categorisation of context atoms.

Context category:	**Context Atom Type:**	**Context Atom:**
Device	Orientation	Display up/down, sideways to left/right, antenna up/down
	Positional Stability	Stable, not stable
	Touch	In hand, not in hand
Environment	Type of artificial light	Europe/US
	Illumination level	Total darkness, dark, normal, bright
	Type of light	Natural, artificial
	Temperature level	Cold, cool, warm, hot
	Humidity level	Dry, normal, humid
	Loudness level	Modest, normal, loud
User	Activity level	Walking, walking fast, running
	Gesture	Lift up, set down, answer, hang up.

Various context atoms are quantised to features with different methods. Fuzzy sets have not been used in the processing of each of the variable. This is due to fact that some context atoms, e.g., touch and positional stability are crisp and applying of fuzzy sets to them produces no additional value. By fuzzy representation we denote the representation where most of the context atoms are processed with fuzzy sets. By crisp representation we denote the representation where all context atoms are crisp.

3.4.1 Environment

To process context information describing environment we have utilized the following information sources: Illumination sensor, thermometer, humidity sensor and microphone. Signals from these sources are processed into context atoms describing the characteristics of the illumination, temperature, air humidity and audio of the environment of a mobile device. Context atoms describing environment are used in experiments in Publications III–V, which examine methods for generating higher-level contexts, and in Publication VI dealing with collaborative context recognition. Furthermore, illumination and audio are used in experiments to control user interface applications, as presented in Publication VII.

Illumination

Environment illumination loosely couples current contexts into time according to illumination level, and into certain physical spaces such as, outdoors and indoors since indoors there usually are artificial light sources. The illumination signal is processed into two types of context atoms; one describing the level of illumination consisting of four context atoms, and the other describing the type of light source including three context atoms. The illumination level is calculated using mean value of a signal within certain time frame. Context atoms are assigned by quantising the dynamic range of calculated mean value with fuzzy sets: $\{\mu_{BRIGHT}(x),\ \mu_{NORMAL}(x),\ \mu_{DARK}(x),\ \mu_{TOTALDARKNESS}(x)\} \in [0,1]$. In Publication VII we have used three context atoms describing illumination: $\{\mu_{BRIGHT}(x), \mu_{NORMAL}(x), \mu_{DARK}(x)\} \in [0,1]$.

The type of illumination means that the light source is either natural or artificial. The type of light source is determined based on the fact that light sources powered by line current, for example, incasdescent lamp or strip light, carry a rectified signal from power lines, which in Europe appears as a 100Hz peak and in Unites States as a 120 Hz peak respectively. Natural light from the sun does not carry any low frequency component. The type of illumination is determined by detecting the peak frequency of the spectrum. To detect the light source the illumination signal is first bandpass filtered with a fourth order Butterworth type IIR filter. Secondly, the power spectral density $P(f)$ of the signal is estimated. Thirdly, the peak frequency of the $P(f)$ is detected. The use of these methods for preprocessing sensor data for context-aware systems has been introduced by the author of this thesis (Mäntyjärvi 1999).

Context atoms describing the type of illumination are: Natural light, artificial light in Europe and artificial light in United States, and their values are crisp zero or one defined by the presence of the peak frequency.

Temperature and air humidity

Temperature and humidity of a device's environment give hints about the climatice conditions that outdoors may vary according to the time of the day and season. Signals from these sensors vary also between indoors and outdoors environment since outdoor climate changes quite often while indoors climate may be more stable and characteristic over a longer period of time due to air-conditioning. Moreover, these quantities may give hints about certain places, for example, the temperature of a device's environment in a car on a sunny day might be quite high compared to outside temperature at that time.

Temperature and humidity sensor signals are calculated using the mean value of a signal within a certain time frame. The calculated mean value is assigned to context atoms by quantising dynamic ranges of mean values with fuzzy sets. Context atoms describing temperature are $\{\mu_{COLD}(x), \mu_{COOL}(x), \mu_{WARM}(x), \mu_{HOT}(x)\} \in [0,1]$, while three context atoms describing air humidity are $\{\mu_{DRY}(x), \mu_{NORMAL}(x), \mu_{HUMID}(x)\} \in [0,1]$.

Audio

Environmental audio is one of the information types for humans to perceive contexts. Context recognition for mobile devices also concerns environmental audio as one of the potential sources since these devices are used in a number of different environments and some of them may have unique audio environments. As discussed in Section 2.3 there exist studies concerning the recognition of different audio environments according to a diverse set of features, which are calculated in the time and frequency domains. In this research we have used one particular feature for describing audio, the RMS value of signal within certain time frame. A feature is decomposed into three context atoms by quantising the dynamic range of the RMS value with fuzzy sets. Context atoms describing audio are: $\{\mu_{MODEST}(x), \mu_{NORMAL}(x), \mu_{LOUD}(x)\} \in [0,1]$.

3.4.2 User

This category holds context information related to user activity, for example, movements and dynamic gestures. Movements and dynamic gestures of a mobile device user are examined by processing the three accelerometer signals, which provide the acceleration data of a device in three orthogonal directions aligned with the axes of the device. The accelerometers we have used react to dynamic and static accelerations. Movements and dynamic gestures are extracted from dynamic accelerations. Static accelerations are used to extract context atoms describing orientations of a device.

Movements

Activities of a user cause dynamic accelerations to the device a user is carrying, for example, walking causes certain periodic signal characteristics. Figure 13 presents examples of acceleration signals describing three types of walking; walking on the level, walking downstairs and walking upstairs. Shapes of the signals reveal periodicity caused by a gait cycle. The recognition of user movements can be carried out using methods in frequency domain, processing signal waveforms in time domain (Flanagan *et al*. 2002), and using other feature extraction methods such as, wavelets (Sekine *et al*. 1998).

We have used context atoms for describing the movement of a device user; walking, walking fast and running in Publications III–VI. In Publication VII we have used context atoms movements, walking, and running. This is due to experiment design. In Publications III–VI user actions concentrate more on walking and running while in Publication VII there is a need for the descriptive atom name movements when designing application control.

A feature used for extracting the above mentioned context atoms is calculated in the frequency domain. A feature is calculated in a sliding window length of four seconds with a shift of one second. This is due to a compromise between the properties of the gait cycle and the computing factors. First, the magnitude x_m of the signal $\mathbf{x} = [x_1, x_2, x_3]^T$ is calculated by taking squared sum of the signal components;

$$x_s = \sqrt{{x_1}^2 + {x_2}^2 + {x_3}^2}\,, \qquad (3.27)$$

and subtracting the mean $\bar{x}$;

$$x_m = x_s - \bar{x}\,. \qquad (3.28)$$

Secondly, the power spectral density $P(f)$ for total dynamic acceleration signal is estimated. Thirdly, the peak frequency of $P(f)$ is detected. By quantising the dynamic range of peak frequency with fuzzy sets, context atoms $\{\mu_{WALKING}(x), \mu_{WALKING\_FAST}(x), \mu_{RUNNING}(x)\} \in [0,1]$ are composed.

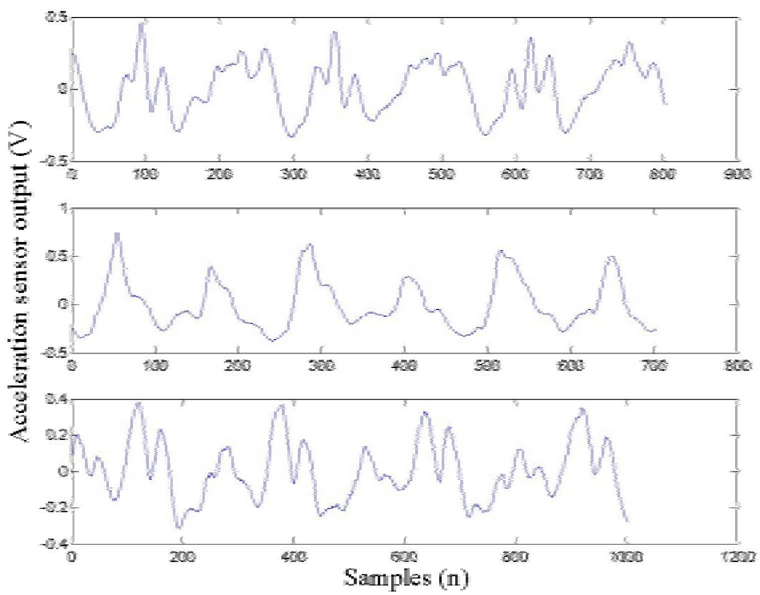


Figure 13. Detailed views of magnitude of the acceleration signals. Top: Walking on the level. Middle: Walking downstairs. Bottom: Walking upstairs (Flanagan *et al.* 2002).

Acceleration signals of movement patterns of a mobile device user are examined more carefully in Publication II, which deals with experiments for classifying various walking patterns; walking on the level, walking down stairs, and walking upstairs. We have used two sensor boxes in Publication II since the research objective in that particular study is that both PCs and ICs may help feature extraction process and thus, more signals are needed. One can see research objective as an attempt to find new directions and scales for the signals computationally such that the signals would be more discriminative. Wavelet analysis in the form of multiresolution analysis is used to extract features from interesting directions. We have further examined the recognition of walking patterns using symbolic representation and coding of acceleration signals (Flanagan *et al.* 2002).

Gestures

The user of a mobile device performs various types of gestures when using a device. A typical one is answering an incoming call. When answering an incoming call we first raise the device so that we see who is calling and then we lift the device to the ear. Figure 14 presents three acceleration signals and their magnitude for answering and set down gesture. Gestures occur as time dependent events in acceleration signals. Publication I presents an approach for gesture recognition where static gestures are recognised with the SOM while dynamic gestures are extracted using HMMs. The approach is validated with experiments. We have further examined the recognition of gestures using symbolic representation and coding of sensor signals (Flanagan *et al.* 2002).

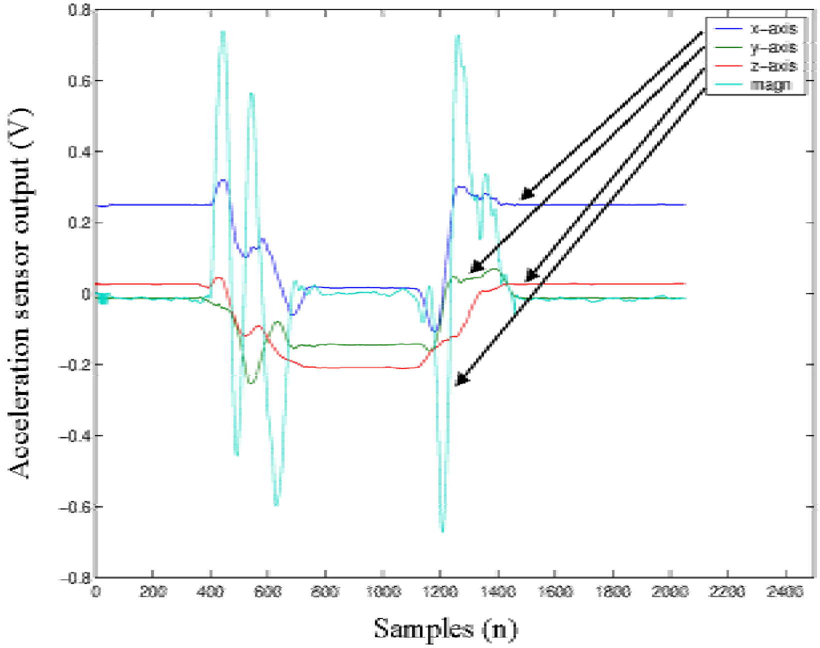


Figure 14. Accelerometer signals for the three axes and magnitude for an answering and set down gesture (Flanagan *et al.* 2002).

3.4.3 Device

Device-related context atoms describe the contextual state of a device more concretely. They are extracted by processing signals from a touch sensing system and from accelerometers. Device related context atoms are used in experiments presented in Publications III–V, which examine methods for generating higher-level contexts, and in collaborative context recognition in Publication VI.

Touch

Context information concerning touch describes the physical interaction between a device and a user. Information can be obtained from a touch sensing system integrated into a device. It provides explicit information when the device is in user's hand and when not. The touch sensing system uses two conducting surfaces to detect touch. Another implementation of the touch sensing system based on capacitive touch detection is presented in (Hinckley *et al*. 1999). In the touch detection system implemented two conducting surfaces are placed close to each other (distance 2–3 mm). Details of the sensor can be found from (Tuulari 2000). The output of the sensing system is 0 when no contact with the skin is detected, and the output is > 0 depending on the conductance of the contact with the skin. The output is coded into one context atom; at hand having crisp values one or zero.

The system is designed for use with bare hands and thus it may be sensitive to errors since it gives output when any conductance between the strips is detected. This may occur with conductive materials such as, metals and moistured clothing. The touch detection system has one obvious drawback; it does not detect touch when the user has gloves. To develop more accurate touch detection system the benefits from both implementations, from conductive and capacitive systems can be combined (Mäntyjärvi *et al.* 2001).

Positional stability

The positional stability of a device provides information about whether the device is on a move or not. As the device is usually with the user the positional stability provides information about user's stability. This information may be useful when other context atoms describing movements give unreliable information. However, there exist situations when a device is not with a user, for example, a device is on a table.

The positional stability is calculated with standard deviation (STD) from the variability of individual acceleration signals x_1, x_2, x_3. The context atom stable is composed by assigning a value of atom to one when the standard deviation of a signal is under a predefined threshold and to zero when the standard deviation exceeds the threshold, respectively.

Orientation

Orientation information of a device can be obtained by processing mean values of signals of the three acceleration signals calculated within a time frame since signals from accelerometers we have used in our experiments, see Table 4, include a static component that is due to gravity. Combinations of mean values are further processed into six context atoms; antenna up, antenna down, sideways to left, sideways to right, display up, display down using an MLP classifier. The MLP is trained with a set of data recorded from example device orientations. An input layer of the MLP consists of three neurons because of a three-dimensional input vector. The classifier consists of one hidden layer with ten non-linear neurons. The output layer provides six outputs $o_i(n) \in [0,1]$, $i = 1, ..., 6$, which are assigned to context atoms described above. Composed MLP model for orientations provide continuous non-linear shifts, values between [0,1], from orientation to another.

3.5 Collaborative context recognition

Mobile devices equipped with sensors are able to recognise aspects of the surrounding context. By combining data with interconnected devices located spatially near each other devices can together perform context-related tasks (Beigl 1999). Reliable context recognition in a single device requires sophisticated algorithms, sometimes involving heavy processing. The more data is available from the sensors, the more welcome would be rough initial assumptions of the context. Moreover, when there are several people with their mobile devices in the same situation, it is more likely that the devices sense aspects of the same context. By combining their current data a more accurate initial assumption about the overall context can be generated. This section presents a method for collaboratively recognising the context of a group of mobile devices located near each other. The main idea in our method is to recognise the need for requesting and locally processing context information from nearby mobile devices.

As explained at the beginning of this chapter, the higher-level context can be defined as a time dependent collection of the contexts from various sources. Assumptions in collaborative context recognition are that mobile devices within the immediate environment of each other sense and share common aspects of that context and mobile devices have similar context sources producing similar context representation. We have experimented with context atoms. The contribution of various devices to collaborative context recognition is dynamic, depending on the properties of the information, content and reliability. The description and simulation results of the method are presented in Publication VI.

3.5.1 Applications for collaborative context recognition

Collaborative context recognition benefits many kinds of context-aware applications. To illustrate the utility of our approach, we discuss two mobile phone application cases where our method may be useful (Mäntyjärvi *et al*. 2002). Not surprisingly, the examples deal with cases where a number of people meet. These applications have not been evaluated with experiments.

The archetypical application with the widely popular mobile phones is automated ringing tone silencing. Many people are familiar with the awkward situations that a suddenly ringing mobile phone causes in sensitive settings such as, museums, libraries, and classrooms. With collaborative context recognition, the problem seems to be solved moderately easily. Where several people are gathered, only the first ones need to set their phones in a suitable mode (which may also be automated). For the rest, their phones can follow the majority or upon user confirmation switch into a silent mode. Note that the technique is less useful where there are not many people present. On the other hand, such situations are also less likely to be sensitive.

The other potential application is presence. The situation often influences our choice of how reachable and visible we want to be: Do we want to receive calls, do we accept unsolicited advertising, do we want to pass our dietary preference profiles to a restaurant. A collaborative context may help in determining a suitable presence profile on the basis of the preferences of the people nearby. For example, if 100 people in the bar tonight have chosen to receive a free soft drink in return for accepting a commercial, maybe I could safely do the same. The freedom to decide must remain with the individual, but peer pressure may be allowed to affect our choices.

3.6 Information fusion

Context recognition can be considered as generating higher-level contexts from a collection of low-level contexts. The objective for context information fusion is to examine and develop methods for extracting higher-level contexts. An overview of the tasks of this approach is presented in Figure 15. The context atom information $C_{1,j}$ can be presented in fuzzy or crisp representation. The first task is to examine which representation yields better results when extracting higher-level information. This is presented in Publication III, which also includes examination of multi-source fusion with clustering and segmentation. The context atom data is analyzed with the basic k-means clustering and segmentation. The applicability of crisp and fuzzy quantisation as a preprocessing method for representation is evaluated. Video recordings are synchronized with the data and segmentations are evaluated qualitatively. The correspondence and consistency of extracted higher-level contexts are examined using video recordings. We chose video recordings as a labelling method for context measurements to eliminate errors in the measurements due labelling carried out by the user. In the examination of extracted higher-level contexts with clustering we are interested in cluster centroids, that is the collection of the prototype vectors since they represent different contexts. In the segmentation experiments we are interested in breakpoints of the clusters. Particularly, we are interested in the order of the appearance of breakpoints of contexts that implies major changes in contexts. Segmentation and clustering of contexts are examined qualitatively and visually. The results are presented in next chapter.

The visual examination of the low-level information is needed to obtain the understanding of relevant context atoms in higher-level contexts. The compression of multidimensional context information while maintaining essential information is necessary to examine the structure and dynamic characteristics of multidimensional context data visually. In Publication IV, PCA and ICA are used to examine these issues. PCA is used to compress multidimensional data into a more compact representation revealing the important phases of information in user scenarios, and to examine relevant context atoms with several visualizations while ICA is applied to examine the independence of context atoms. Next chapter presents the results.

The transformation of the context recognition problem into a problem of generating higher-level contexts presented at the beginning of this chapter is an intuitive definition and it gives no indication to how context should be processed. Publication V provides a framework and method for processing and manipulating multilevel contexts. The framework is verified with experiments. The framework is based on the symbolic representation of context information and unsupervised clustering of symbol strings.

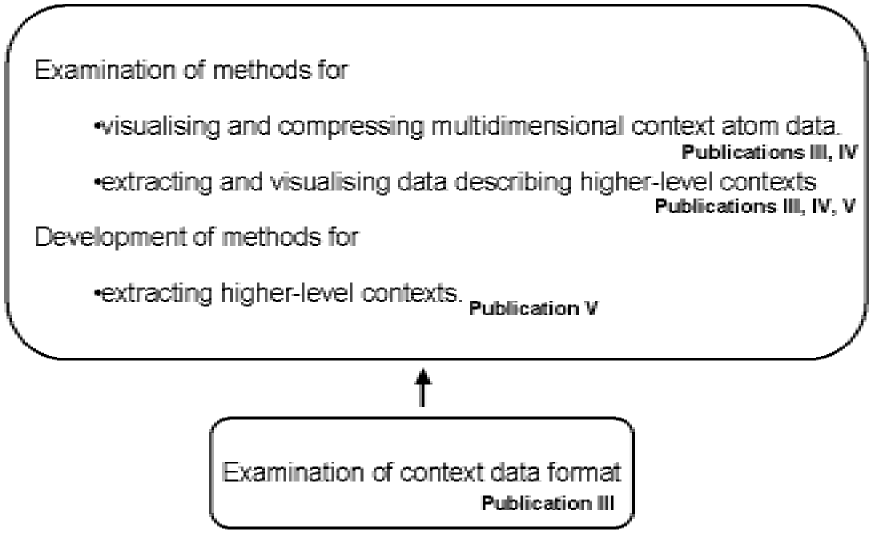


Figure 15. Illustration of an approach for context information fusion.

The context state is represented by a symbol. As presented at the beginning of this chapter, $C_{i,j}(t)$ represents the context state at level i, of context source j at time t. Here, the context state on level k-1 is denoted as source to level k. It is assumed that the state of each of the n sources has been assigned a symbolic representation. The context generated and represented by the fusion of the n context states $C_{i,j}(t)$, j = 1, …, n is denoted by $C_{i+1,k}(t)$, a context state from source k at context level i +1 is given by

$$C_{i+1,k}(t)= (C_{i,1}(t), C_{i,2}(t), \ldots, C_{i,n}(t)). \qquad (3.29)$$

Now the context state $C_{i+1,k}(t)$ has been recognised and assigned with a symbolic representation. In a similar manner the recognition and segmentation problem for a time sequence of context states generated from the same source can be described using this notation. The sequence of context states from source j at context level i is given by ($C_{i,j}(t1)$, $C_{i,j}(t2)$, $C_{i,j}(t3)$,…), where $t_1 < t_2 < t_3$. This describes the intermediate phase where context states are ordered according the time. Once again the recognition of the context is similar to the previous case except this time the higher-level context is given by

$$C_{i+1,k}(t_{l+m})= (C_{i,j}(t_l), C_{i,j}(t_{l+1}), \ldots, C_{i,j}(t_{l+m})), \qquad (3.30)$$

where l, $m > 0$. An important distinction between the two cases must be made. In the first case, notation (3.29) it is likely that the temporal ordering of the symbols is not important. However, in the second case given by notation (3.30) it is very likely that the ordering of the symbols is highly important in determining the context.

When we have a set of n sources of context information that produce the context states as in notation (3.29) at time t, then one could suggest that the recognition of the context could be simply a lookup table where the resulting context state $C_{i+1,k}(t)$ is listed as the resulting context. However, the user's context world is highly personal and the mobile devices are personal to the user. This implies that any sort of lookup table would be personalized to the user. Secondly, in any real world situation there will always be noise. This implies that there will not be a unique one to one relationship between a set of symbols and their higher-level context representation. Well-known methods for clustering data cannot be applied in an efficient manner to the clustering of symbolic data, thus we have developed a suitable approach described in Publication V.

The general idea of the approach developed is presented in Figure 16. During learning the algorithm takes an instant time t, an input string of symbols coded from context atoms and clusters them. The formed clusters are automatically labeled. Thus, the first stage presented in notation (3.29) is performed. Generating a second set of symbol strings on a first clustering layer, at each time instant the symbol string from the first context level is clustered and the state at that time instant is denoted by the cluster label to which the symbol string is clustered. This results in a second set of symbol strings. The second set of symbols strings is used in the learning of the second layer clustering with automatic labelling, where labels refer to higher-level contexts with low-level history involved. This is explained in more detail in Publication V.

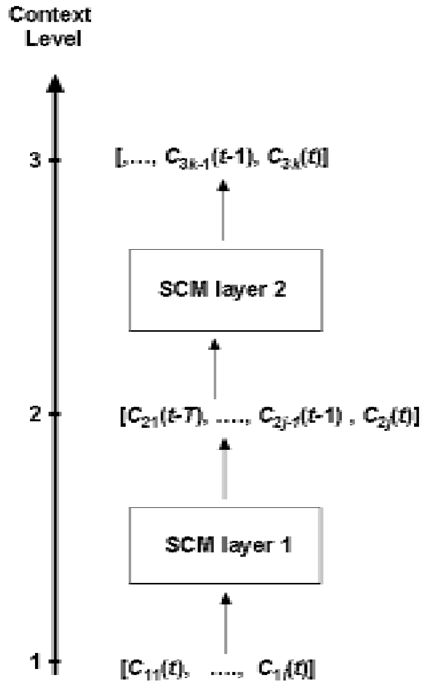


Figure 16. Overview of symbolic context information clustering (Publication V).

3.7 Context information for applications

This section introduces our approach to utilise context information in applications. First, the sharing of context information in a basic mobile device application is introduced. Secondly a mobile application control using the representation developed is explained. Thirdly we discuss a manipulating representation with multilevel contexts for applications.

3.7.1 Sharing of context in remote communication

Mobile devices and particularly, mobile phones play an important role in forming and maintaining the social relations of their users. One objective in utilising context information in these devices is to enhance remote mobile communication by sharing situated information, as explained in Publication VIII. Moreover, the important issue is the presentation of context information to a user. A common method for presenting context information in mobile applications is to use small iconic symbols (Schmidt *et al*. 2001, Keränen *et al*. 2003)

By viewing users behavior with their mobile devices in various situations we can obtain insight as to initiating remote mobile communication in unsuitable situations. We have introduced the wireless application protocol (WAP) based solution for representing contexts between terminals. Our approach provides a solution for sharing context information before initiating a call, Publication VIII. Profile information has been used as the context source. It is already available in terminals and it can be considered to describe a coarse approximation of user's situation since users configure and set profiles to operate in different way in various situations. The system developed consists of a remote context server and two WAP based applications; context selector and context-call. Context selector enables user to update profile information to a remote server while context-call takes care of checking context of a communication party. When context is delivered to a caller he/she can decide how to proceed, for example, call or use messaging.

3.7.2 Application control based on representation

The exploitation of the representation developed for context information in mobile applications is not straightforward. The representation consists of various types of sensor-based information that are characterised with uncertainty, i.e., when the extracted features are quantised with fuzzy sets and labeled with semantic meaning according to the developed ontology, the interpretation of the data is seldom 100% reliable. Moreover, dynamic environments set special requirements related to usability issues and social acceptance of novel functionalities of mobile applications. Since composed representation describes approximations of current situations at best, the use of context information in application control still maintains the risk of launching applications in unsuitable situations and of applications behaving in an unwanted manner. By utilizing fuzzy logic methods in designing control mechanisms for adapting mobile applications and user interfaces we can exploit the uncertainty that is characteristic for fuzzy representation developed.

We have developed an approach for adaptating information representation in mobile device applications. This is presented in Publication VII. The approach uses multiple fuzzy controllers and it provides continuous context-based application control. Information representation applications include the volume of operating tunes, display illumination, font size of a text presented on a screen and content of a service presented on a screen as text. These features use inputs from context atom types; user and environment. The adaptation of applications is performed using representation and fuzzy control. Design of controllers is carried out with fuzzy rule bases that are built by using the fuzzy logic operations presented. The approach is validated with experiments with real context data from user scenario, and users' reactions to application adaptation are gathered to obtain the user's preferences about the application adaptability.

3.7.3 Example of high-level context application

The context recognition framework enables us to construct a layered representation that can be utilised in context-aware systems and in applications. As discussed earlier, Publication VII presents how representation is utilised in adapting applications according to multiple fuzzy contexts. Moreover, the conversion of fuzzy context information into crisp symbolic representation enables the composition of higher-level contexts as presented in Publication V. Here, we briefly introduce how to manipulate symbolic higher-level representation to be utilized in context-aware systems and in applications.

First, let us consider the symbolic coding of context atoms presented in Table 6. The coding is performed to the categorisation presented in Table 5. Secondly, let us consider the following example that aligns with experiments in Publication V. We have two sequential higher-level contexts in Table 7. They are represented on a C_1 level. At a time instant t–1, a symbol string represents a single higher-level context, user's action "walking inside in a lobby". The C_1 level representation is [4 9 12 14 18 23 25 29 34 38]. Corresponding higher-level, C_2 representation is a symbol [281]. At a time instant t, a symbol string represents a single higher-level context, the user's action "walking outside". The C_1 level representation is [4 9 12 16 17 22 25 29 33 38] and corresponding higher-level, C_2 representation is a symbol [379]. Obviously time dependent context of these events is "going outside", which can be represented on three different levels, on

C_3 level as a symbol 254, on a C_2 level as symbols [281 379] or on C_1 level as consecutive sequences of symbols. Higher-level contexts [281] and [379] represent labels of the first layer context clustering. Thus, C_1 level representations present typical combinations of context strings in context clusters, as presented in Publication V.

Table 6. Symbolic coding of context categories, context atom types and context atoms. ND represents not defined and means that the context atom cannot be assigned. DD=display down, DU=display up, AD=antenna down, AU=antenna up, SR=sideways right, SL=sideways left.

Context category	**Context Atom Type**	**Context Atom**
Device	Orientation	(1=DD), (2=DU), (3=AD), (4=AU), (5=SR), (6=SL), (7=ND)
	Positional Stability	(8=Stable), (9=Unstable), (9=ND)
	Touch	(11=In hand), (12=Not in hand), (13=ND)
Environment	Type of artificial light	(14=50Hz), (15=60Hz), (16=ND)
	Illumination level	(17=Bright), (18=Normal), (19=Dim), (20=Dark), (21= ND)
	Type of light	(22=Natural), (23= Artificial)
	Temperature level	(24=Hot), (25=Warm), (26=Cool), (27=Cold), (28= ND)
	Humidity level	(29=Humid), (30=Normal), (31=Dry), (32= ND)
	Loudness level	(33=Silent), (34=Modest), (35=Loud), (36= ND)
User	Activity level	(37=Walking), (38=WalkingFast), (39=Running), (40=ND)

Thirdly, let us consider an application to be executed in context “walking inside in a lobby”. An application needs a defined set of variables. Both representations on levels C_1: [4 9 12 14 18 23 25 29 34 38] or C_2: [281] can be used. Accordingly, an application to be executed in context ‘walking outside’ may utilize both representations on levels C_1: [4 9 12 16 17 22 25 29 33 38] or C_2:

[379]. However, it is commendable to use the representation on level C_2 since it represent context cluster and it is more general describing that particular context better, as Publication V shows. Finally, let us consider an application to be executed in time dependent context "going outside". Again, C_2 level representation [281 379] can be used. However, by using level C_3 representation 254, more a general representation for higher-level context can be expected. The utilisation of higher-level context information in UI-application is demonstrated jointly with authors of Publication V in (Himberg *et al.* 2003).

Table 7. Categorisation and hierarchical structure of context information.

Level C_3 representation		254	
Level C_2 representation		281	379
Context category	**Context Atom Type**	**Context Atoms (*t*–1)**	**Context Atoms (*t*)**
Device	Orientation	(4=AU)	(4=AU)
	Positional Stability	(9=Unstable)	(9=Unstable)
	Touch	(12=Not in hand)	(12=Not in hand)
Environment	Type of artificial light	(14=50Hz)	(16=ND)
	Illumination level	(18=Normal)	(17=Bright)
	Type of light	(23=Artificial)	(22=Natural)
	Temperature level	(25=Warm)	(25=Warm)
	Humidity level	(29=Humid)	(29=Humid)
	Loudness level	(34=Modest)	(33=Silent)
User	Activity level	(38=Walking Fast)	(38=Walking Fast)

3.8 Summary

In this chapter an approach for the design and development of a context recognition procedure is presented. The objectives of this approach are to extract low-level contexts from several sensors to provide information representation, to examine dynamic characteristics and structure of data and to provide information with several levels enabling diverse use of extracted context information.

Context recognition for extracting higher-level context information includes the following tasks:

- Context information measurements from various sources that are low-cost sensors integrated into a mobile device.

- Composing the representation. The measured sensor signals are processed into context representation by categorising and labelling them with semantic meaning. We have applied signal processing and pattern recognition methods for extracting features describing low-level concepts from the real world.

- Collaborative context recognition describes a method for negotiating jointly the current context between nearby context-aware devices. The method provides a way to obtain initial assumption of the current context that is determined together with nearby devices. The method is demonstrated using the representation developed. This task is not necessarily carried out before context information fusion.

- Context information fusion involves methods for examining dynamic characteristics and structure of multidimensional data in the form of context representation, and a framework for extracting and representing higher-level contexts. We have applied data analysis methods for examining the extraction of higher-level context information. The methods for time series segmentation and symbolic clustering are demonstrated to be suitable for context recognition.

We have also presented how the utilisation of context information in mobile devices enables the development of novel applications and development of adaptability to existing ones. These issues are validated with three approaches: Firstly, distribution of context information already available in mobile devices enables us to enhance remote mobile communication with context-aware features. Secondly, the representation developed enables the adaptation of mobile applications for information representation according to multiple uncertain context variables. Thirdly, using the representation higher-level contexts can be provided for applications. The next chapter presents summary of the main results of this work.

4. Results

This chapter summarises the main results of this thesis answering the main research problem:

RP: How should sensor-based context recognition be defined and performed to facilitate the utilisation of context representation in mobile applications?

The results are discussed in two parts. The first part deals with context information processing, answering the subproblems SP_1 and SP_2. The second part deals with context information and applications, answering the subproblem SP_3.

4.1 Context information processing

We have presented a context recognition procedure including processing of sensor data into a representation, collaborative context recognition, examination of context data and the framework for generating higher context levels. The subproblems SP_1 and SP_2 deal with context information processing.

SP_1: How should low-level context information be extracted from sensor signals to obtain rich and usable context representation?

The low-level context information is produced in the first phases of the context recognition procedure; measurements-phase and representation-phase. The measurements-phase provides sampled sensor signals. In the representation-phase features are first extracted from signals and they are further processed to an expressive set of context atoms using signal processing methods, which are selected so that the extracted features describe concepts from the real world. Context information is transformed into a context representation consisting of three main categories: Environment, Device, and User.

The environment category includes context atom types; loudness level, humidity level, illumination level and temperature level. It is shown that the quantisation of the dynamic ranges of humidity, illumination and temperature sensor output values with fuzzy sets enables the construction of a representation for the environment category for these context atom types. In the case of the audio signal the RMS-value of a signal is first calculated. To obtain an element for the representation describing the loudness level of an environment, fuzzy sets are applied. Moreover, it is shown that the processing of illumination signals in the frequency domain enables us to assign context atoms describing the type of light and type of artificial light in the representation.

Device category includes context atom types; positional stability and orientation of a device, as well as touch between a device and a user. It is shown that a context atom describing positional stability can be processed from accelerometer signals using STD. It is also shown that by processing acceleration signals with the MLP classifier, orientation information of a device can be extracted. Moreover, it is shown that context information describing touch between the user and a device can be processed from a touch detection system.

The user category contains context atom types activity level and gestures. It is shown that context information describing user gestures and the user activity can be extracted from acceleration sensor signals. The results of Publication I show that it is possible to extract gesture information from the usage of a mobile device. Gesture information can be utilised in the development of implicit input methods for the interaction with mobile devices.

It is shown that context information describing user activity can be extracted from acceleration signals by processing data in the frequency domain and by applying fuzzy sets to quantise the peak frequency. The results of Publication II showthat by applying ICA and PCA (whitening) to multidimensional sensor signals interesting directions of the data can be revealed. The results of Publication II show that user activity particularly, users’ walking patterns can be recognised from multidimensional sensor data using PCA (whitening), ICA and wavelet transformations for feature extraction. Results are obtained from the distributed sensing scenario where a mobile device has an accessory containing accelerometers. Results are presented in Figure 17 as bar chart. The results show that using PCA and ICA in feature extraction improves the walking pattern recognition considerably. Utilisation of PCA (whitening) and ICA produces nearly equal results.

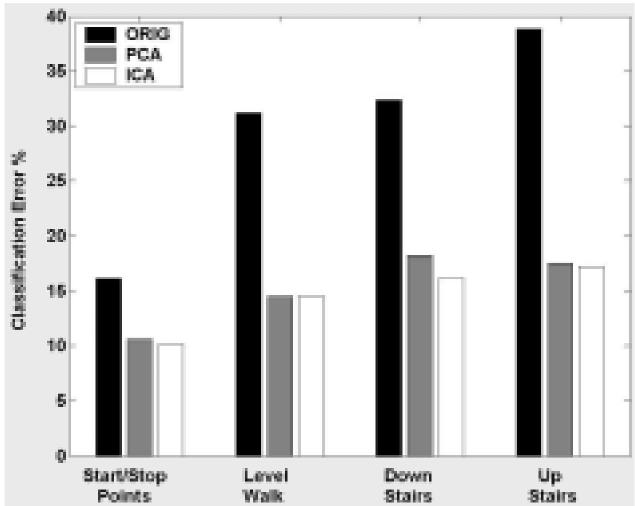


Figure 17. Walking pattern classification errors obtained with various feature generation methods; Original data, PCA (whitening), ICA (Publication II).

SP_2: How should low-level context information be processed and examined to obtain higher-level contexts?

The two later phases in the context recognition procedure are for examining actual context atom data; the collaborative context recognition-phase for determining the current context together with several devices, and the information fusion-phase for examining and extracting higher-level contexts. The composed representation as a multidimensional variable array enables the examination of the structure and dynamic characteristics of data with explorative data analysis methods.

In Publication VI a novel method for collaboratively recognising the context of a group of devices is demonstrated. The idea of the method is that several devices within communication range of each other are able to negotiate the current context together and control their applications accordingly. The idea of the method is derived from the viewing of peoples' behaviour in contexts. The results of Publication VI show that the method is capable of collaboratively recognise contexts. Thus it provides a reasonable initial guess for the current higher-level context. The similarity measures from the experiments with 20 nodes are presented in Figure 18.

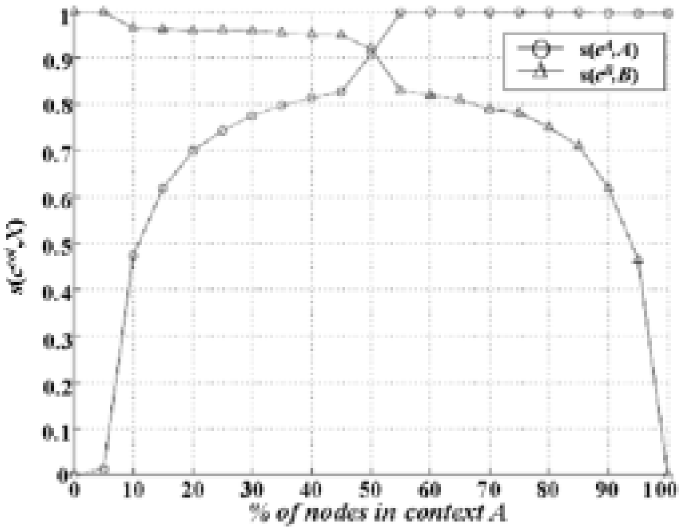


Figure 18. Similarity measure between collaboratively recognised context and contexts (X=A,B) as a function of the amount of nodes in context A. Number of nodes = 20 (Publication VI).

The similarity measure $s(c^{col},\mathrm{A})$ increases rapidly above level 0.5 indicating that the recognized collaborative context is closer to the actual context when ~10% of the nodes belong to context A. Correspondingly, the similarity measure $s(c^{col},B)$ decreases rapidly after 94% showing that a method is able to recognise the collaborative context B among the nodes in context B, even if the majority of all devices is in context A.

Jointly determined context information can be utilised in certain types of applications. We have discussed potential applications that are related to situations where a number of people meet, for example, ring tone silencing and presence profile negotiation.

We have presented experiments in Publications III, IV and V to examine multidimensional context data with data analysis methods. In the clustering and segmentation experiments, Publication III, the motivation is to find a low-level context representation that gives consistent results for higher-level context extraction. The clustering is performed with k-means clustering while minimum-variance segmentation is used for time series segmentation. The results of Publication III show that data processed with fuzzy quantisation gives more consistent clustering. The clustering with crisp data divides diverse sets of context atoms into clusters. The reason for this is that binary data forms a data cloud where data points are located in the corners of a hypercube.

The segmentation is examined qualitatively with video recordings from measurements. The segmentation of the context atom time series gives an adaptive length time window where the higher-level context within the window is constant, i.e., the breakpoints separating the adjacent segments indicate the change in the total variance of data and thus the change in higher-level context. Examples of segmentations for crisp and fuzzy data are shown in Figure 19.

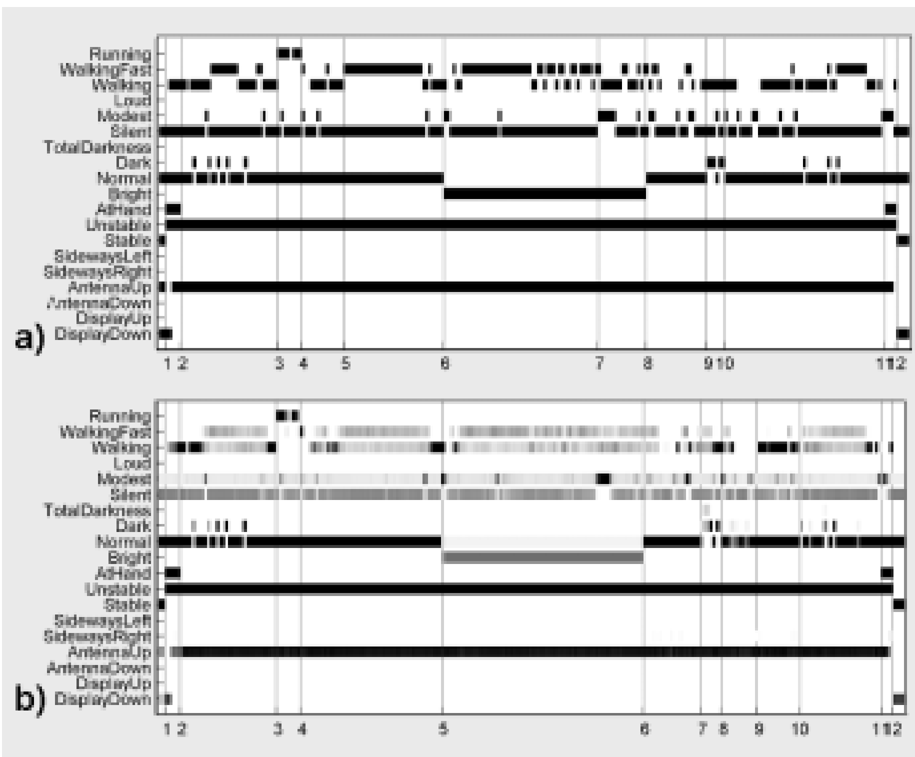


Figure 19. Segmentations using a) crisp and b) fuzzy data sets. The context atoms are presented as a function of time using grey level bars. Black denotes one and white zero (Publication III).

The segmentation of data processed with fuzzy quantisation gives a more stable location of breakpoints over several iterations than the segmentation obtained with crisp data. Moreover, the segmentation of fuzzy data corresponds better to real context changes when examining segmentation with video recordings, Figure 20.

Figure 20. a) Video snapshots corresponding to the breakpoints of the segmentation using the crisp data set presented in Figure 19a. b) Video snapshots corresponding to the breakpoints of the segmentation using the fuzzy data set presented in Figure 19b (Publication III).

The low-level context data is also examined with PCA and ICA. PCA is used to compress multidimensional data into a more compact representation to be used in the visual examination of context data. ICA is applied to extract patterns containing independent low-level information. The results of Publication VI show that context atoms turned out to be already independent. When examining the data compression in the particular scenario twelve atoms out of 27 explained 96% of the data variance, and in the case of PCs seven dominant PCs explained the same amount of data variance.

Figure 21 shows the visualisation of the data using two PCs in one particular test. The arrows show how some phenomena are related to PC1. Changes in illumination related context atoms lead PC1 value towards 1 while changes in stability related context atoms lead PC1 value to zero. PC1 and PC2 are used in several visualisations in Publication IV and we have found the use of PCA useful in examining context data.

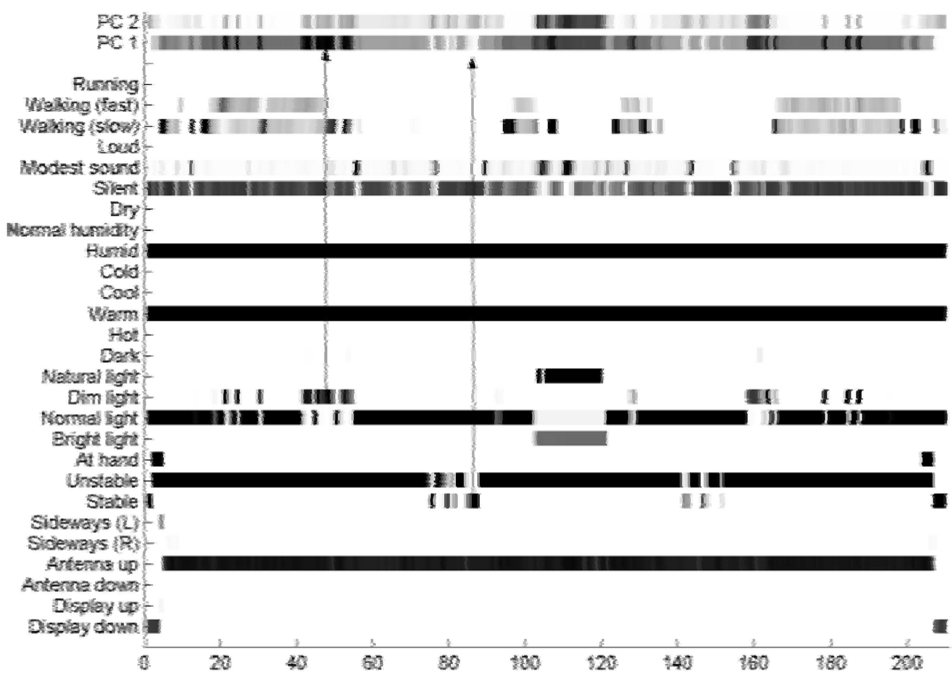


Figure 21. Data and the two most dominant PCs of one test (Publication IV).

In Publication V a general framework for generating higher-levels of context information is presented. In the framework, data is processed using symbolic representation. We have also described a novel method for extracting higher-level contexts from lower-level context information by using an unsupervised clustering approach designed particularly for symbolic data (Flanagan 2003a, Flanagan 2003b). The context recognition framework is verified with experiments with a context data set. A requirement is that the fuzzy data is converted to symbolic data. The results in Publication V show that unsupervised clustering for symbolic data applied to segmented context information can be efficiently use to extract and label context information.

In the extraction of the highest level context in our data, that is, trying to identify the scenarios using the method, each of the symbol strings associated with each of the repetitions are classed to a particular cluster. Table 8 shows the results. It is shown that scenarios, 1, 2 are quite similar with little similarity between scenarios 3, 4, and 5. This is because scenarios 1, 2 are actually quite similar and other scenarios differ from each other (Table 2, Publication V).

Table 8. The number of repetitions of each of the scenarios assigned to each cluster in the second clustering stage (Publication V).

	Scen. 1	**Scen. 2**	**Scen. 3**	**Scen. 4**	**Scen. 5**
Cluster 1	43	47	4	0	0
Cluster 2	1	1	45	0	0
Cluster 3	0	0	0	43	4
Cluster 4	0	0	0	7	46

The principle of generating higher-level context and recognizing context has been demonstrated. We have shown that the context recognition framework developed produces a feasible way to represent context information with different abstraction levels. The framework carries out source fusion and temporal fusion.

The summary of the context recognition procedure developed is presented in Table 9. Arrows on the left show that any phase (3, 4 or 5) can be performed using the composed representation (context atoms). It must be noted that context atoms must be converted to crisp, symbolic representation to be used in stage 5.2.

Table 9. Summary of context recognition procedure.

1. Measurements

   1.1 Sampling and calibration of source signals

2. Composing representation

   2.1 Feature extraction from source signals with particular methods described.

   2.2 Labeling and quantisation of features with fuzzy sets (if needed)

   → Context atoms

3. Collaborative context recognition

4. Examination of context atom data

   4.1 Visualisations

5. Extraction of higher level contexts

   5.1 Segmentation

   5.2 Symbolic clustering of data

4.2 Context information and applications

This section presents answers to the last subproblem.

SP_3: How to utilize context representation in applications?

By composing context representation on the context atom level we facilitate the utilisation of context information in mobile context-aware systems and applications. This is demonstrated in Publication VII, which presents the adaptation of mobile device applications according to fuzzy context information from the context categories, environment and user. Selected applications exemplify basic types of applications for representing information to a user with different means. The approach developed utilises fuzzy logic controllers, which are designed to adapt the selected applications according to fuzzy rules. The approach is experimented and evaluated with a user scenario. The real context data from a scenario is presented in Figure 22.

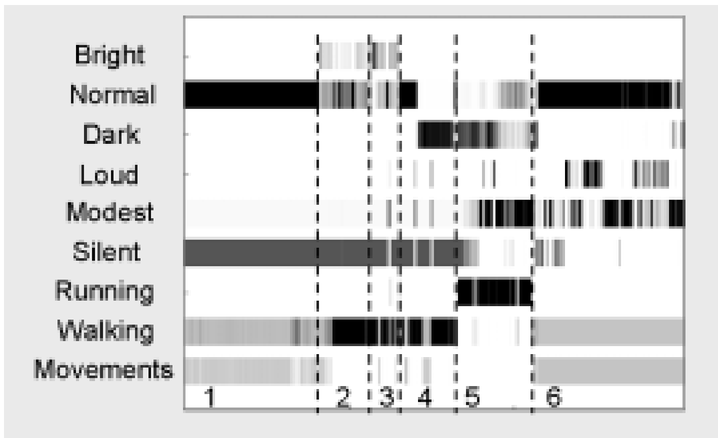


Figure 22. Context data from a user scenario. The context atom values are represented as gray level bars. Numbers of segments correspond to certain activities of the user (Publication VII).

The examination of the outputs of the controllers shows that applications adapt to the current context of the user's mobile device, Figure 23. The control signals can be examined by comparing them to the users' actions expressed as context atoms in Figure 22. User activities in the segments are presented in Table 2 in Publication VII. In the adaptation of the information content of the service, the most compressed information is presented when the user is running. The basic information representation with small font size is provided when user activity is low. This demonstrates the adaptation of the service content representation.

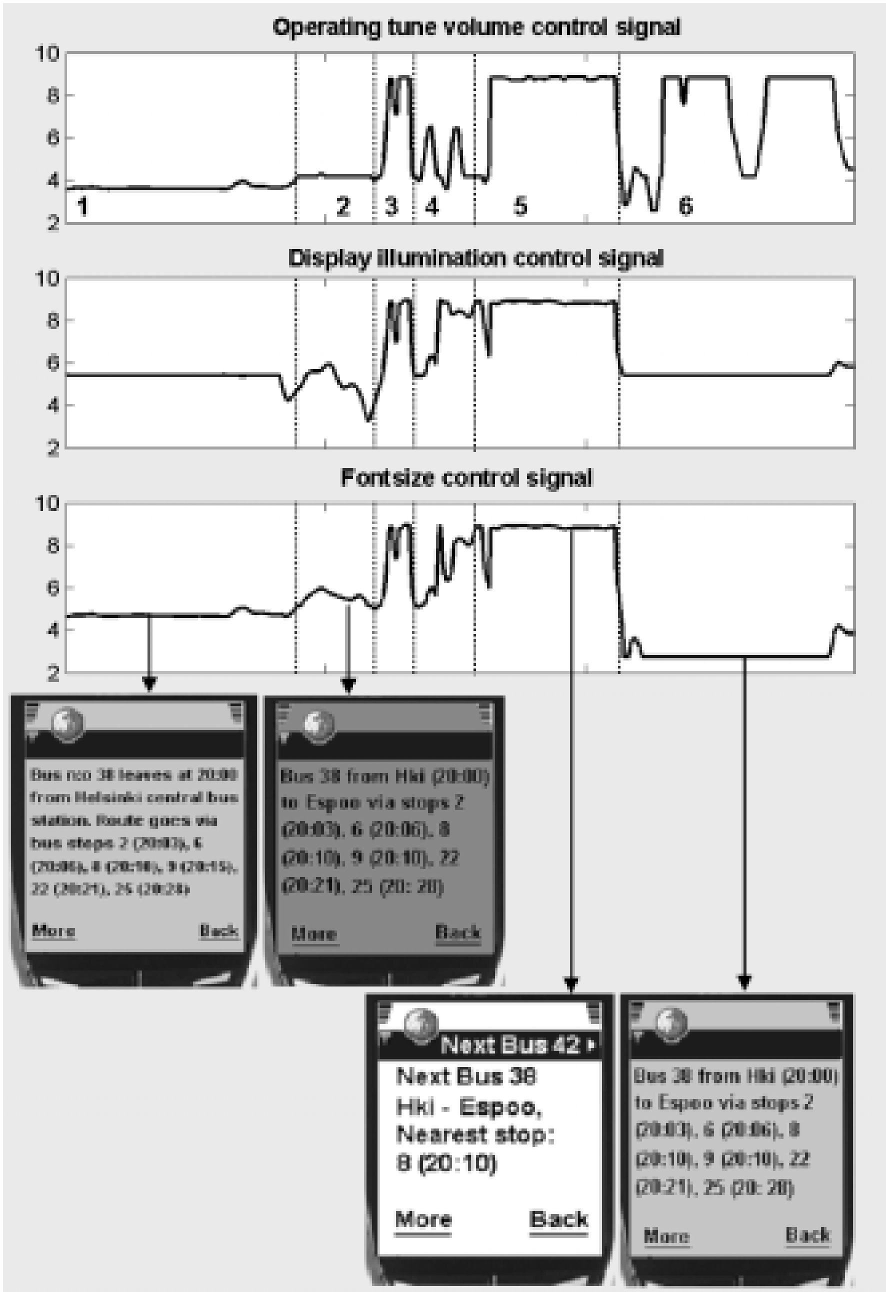


Figure 23. Operating tune volume, display illumination and font size control signals. Numbers correspond to the segments of the scenario. The adapted information contents of the bus timetable service with illumination control in different phases of the scenario are presented (Publication VII).

The results of Publication VII show that the context representation with fuzzy logic controllers is suitable for adapting applications in mobile devices.

Context information can appear in many formats in a mobile device, since a device has various types of context sources, as introduced in Chapter 2. One common context information source is profile information that contains information about how a device operates in different profiles. The profiles are set by the user and they can be considered as context settings. We have viewed the behaviour of people when making a call with a mobile phone, revealing that the lack of knowledge about the context at the other end leads to initiation of calls in inappropriate situations. As the result of Publication VIII we present a novel solution to this problem, a method to exchange mutual context information that is, profile information, when establishing remote communication. Our solution may make it more natural to initiate the remote communication since an overview of the context of the communication partner is already known. This allows a caller to choose how to proceed with a remote communication, e.g., call or send a message.

Table 10 shows that context information extracted in various phases of the context recognition procedure can be provided to applications (arrows on the right). Some applications can utilise source information directly as presented in Publication VIII. The context information at the level of the context atom can be effectively used in applications as showed in Publication VII. Collaborative context recognition utilises context representation and thus it can be assumed to provide context data for applications in the same format. Since it has not been validated with experiments it is illustrated using dashed arrows. It has been shown that higher-level contexts can be also provided to applications as cluster labels.

Table 10. Utilisation of information from context recognition procedure in applications.

1. Measurements

   1.1 Sampling and calibration of source signals

2. Composing representation

   2.1 Feature extraction from source signals with particular methods described.

   2.2 Labeling and quantisation of features with fuzzy sets (if needed)

   → Context atoms

3. Collaborative context recognition

4. Examination of context atom data

   4.1 Visualisations

5. Extraction of higher level contexts

   5.1 Segmentation

   5.2 Symbolic clustering of data

→ Applications

4.3 Discussion

The representation of sensor based context information can be obtained by applying carefully designed quantisation to features extracted from sensor signals. We have used fixed limit values in quantisation that may be a problem for developing online context-aware systems, since limiting values in quantisation for determining representation might be personal, for example, one person may consider 25 degrees of Celsius warm whereas the other feels it cold.

Data analysis methods used; PCA, ICA, clustering and segmentation are suitable methods for examining the dynamic characteristics and the structure context information for the design of context recognition. PCA is an efficient tool for processing, compressing visualising and examining context variables. The use of PCA in context-aware systems may become useful when there is a need for compressing data, for example, in communicating context when the number of variables in a representation increases. PCA, whitening and ICA are good methods for finding interesting directions of the data when examining sensor signals and context information. Methods for unsupervised clustering applied here, the SOM and k-means are useful tools for making vector quantisers for processing sensor and context data. Segmentation is a suitable method for examining the dynamic characteristics of context data from a time series. Symbolic clustering and segmentation are demonstrated to be together a suitable tool for extracting higher-level contexts in an unsupervised manner. To apply them for online context recognition additional studies, e.g., examination of memory and processing requirements and examination of online implementation are required.

The viewing of peoples' behaviour in contexts has encouraged us to innovate novel solutions for context data processing and for applications. Collaborative context recognition and the suggested applications that may benefit from the collaborative behaviour of mobile devices are introduced. The method for collaborative context recognition resembles the way people adapt their behaviour in groups. Due to the preliminary stage of the study the performance of the method is presented using only simulations with small artificial and real context data sets. The method is neutral to the chosen representation. However, all participating devices must share a common context representation. The method supports mobile systems, and it does not require any central context server. Various aspects of the method require additional investigation. For example, it may have problems with stability in dynamic situations where the contexts of devices fluctuate rapidly, suggesting that the stability issues of the method should be studied in various cases. In static situations when the contexts of neighboring devices remain different, the joint context will converge, but the result may be an artificial context that does not correspond to any physical situation. In crowded places with highly mobile nodes the method may not converge and it might lead to oscillating behavior. This suggests that criteria for stopping unnecessary signaling should be developed and further studied. The reliability measures are conducted from the data. Because the variability of contexts from various nodes changes, the reliability measures should be adaptive. This issue should also be further studied. To obtain more realistic results simulations including aspects like energy consumption and delays caused by communication protocols and node movements should be carried out. Moreover, the identified research aspects must be solved before a prototype of the method with applications can be implemented.

The representation facilitates the development of application control in mobile devices. The control of applications with fuzzy logic controllers enhances the capabilities of a mobile device in representing information according to context. The control signals tend to fluctuate when the values of individual contexts change rapidly. This can be considered as the main problem in the system's performance. The fluctuation of control signals may lead to the oscillation of various information representation levels on a display. The variance of the control signals can be decreased by smoothing the input data, or outputs of the controllers.

In mobile device usage environments the continuous adaptation that changes too fast may confuse the user. User reactions indicate that in general application adaptation is an acceptable feature, however, mistakes in adaptation are not tolerated. Users accept the adaptability of minor application features while they want to have control over major application adaptation. Moreover, the behavior of mobile device applications should follow the rules of human social behavior, i.e., a device should not lead their users into embarrassing situations. The experiments with the control of mobile applications with fuzzy logic raised some new research aspects, such as how the optimisation of controllers affects online adaptation of applications.

The sharing of context data in remote communication allows the development of novel types of context-based applications. The context-based call operation is developed to meet the need to perceive context when making a mobile phone call. The solution developed requires that user actively updates his/her profile information into a remote context server using a WAP application. However, this may become a nuisance. This suggests that mobile device applications requiring or providing context information should automatically communicate context into a remote server or directly to terminals to make it visible for applications located in other devices. However, the shared profile information of a mobile communication device does not necessarily mean that the user is in that particular context, since the operation for setting profiles is manual and the user may forget to use it. The context recognition system providing context information continuously would make it more accurate. When context information is transferred to other terminals the privacy issues become relevant; perhaps users do not want to share their context information with other people even if they are close friends or family members.

The results for sensor-based context recognition for mobile applications presented in this thesis are promising. However, it must be noted that they are obtained with a quite limited set of data with a relatively small number of users and user scenarios. To test the methods developed more generally, data collected from several user scenarios from various operating environments, and from several context sources must be gathered. Moreover, methods should tested with hardware designed for mobile devices.

5. Summary and conclusions

Context-aware computing is a relatively young research discipline. It is proposed as an enabling technology for adaptating different functions in computer aided devices. The adaptation is proposed to be carried out by recognising and combining implicit information from the usage situations and the environment of a device and its user, and to provide this information, e.g., for various types of applications and services, which can adapt their appearance and functions accordingly.

There is an acknowledged, evident need for context-aware computing. Research into various subtopics such as, HCI, system architectures, context recognition, various types of information sources, and the utilisation of context information in applications and services are ongoing. It might take several years to see mobile fully context-aware systems. As long as there is a lack of understanding for a common representation to describe dynamic context information and a lack of methods for extracting context information in an online manner the integration of the main components of context-aware systems is difficult. This thesis aims to develop a solution for one narrow area on this field. The thesis presents a review of context-aware computing, focusing on sensor-based context recognition and the utilisation of context information in mobile devices. The extraction of relevant information and its representation are examined and they are shown to facilitate the development of novel mobile applications and new features for them.

Sensors integrated into mobile devices are potential sources providing useful information about the environment and user actions. The context recognition of a mobile device and its user from several onboard sensors is a challenging task. When extracted relevant low-level context data describe some notions of the current situation they are just approximations at best. However, even if the extracted context information is not 100% reliable, relevant higher-level information can be composed. This is validated with experiments using the context recognition framework developed. In the framework a novel method for symbolic clustering is applied for segmented data. On the other hand, the sensors selected affect the methods for context recognition and feasible applications. It is important that a rich and semantic low-level representation of context can be composed. From the context recognition viewpoint; the more relevant context atoms that can be composed from the sensor signals, the more useful the sensor is. From the application point of view the sensor is useful if it provides data that can be used in various applications, i.e., the added value it provides is maximised. For example, accelerometers are useful sensors since they provide information on device orientations, positional stability, user movements, and gestures that can be used in various applications. To enable accurate movement and gesture recognition the methods should be adaptive since users are individuals and their movement and gesture patterns are personal.

When the number of context sources and atoms increase, the utilisation of explorative data analysis methods emphasises. The examination of context variables meaningful for certain applications is needed. These issues set a need for developing context data analysis toolkits and context simulators. To implement the methods developed into mobile devices it is required that they operate in real time and the system is able to process the data flow continuously. It is obvious that when implementing a context recognition system for online processing the methods should be designed according to the processing and memory limitations set by the platform.

To enable the context communication and the facilitation of context information in applications the representation and ontology must be standard. The developer of context-aware applications for mobile devices does not necessarily want to be aware what is happening in the context recognition framework, he just wants to have a standard and static list of available contexts that applications may use. Considering the context communication, the devices should have a common understanding of the context transferred so that they can exploit the data obtained, i.e., in context communication a common vocabulary must exist. For example, higher-level contexts can be restored to standard lower-level representation. This enables the mutual understanding of higher-level contexts between devices.

Although context-aware computing proposes novel mechanisms for human-computer interaction the main challenge is to keep the interaction model of applications consistent and to align the recognised model of the context with the user’s idea of the context. This is a big challenge since mass market mobile devices have tens or hundred millions of users each of them may have own idea of his/her context.

References

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R. & Pinkerton, M. 1997. Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, Vol. 3, 1997: 421–433.

Aoki, H., Schiele, B. & Pentland, A. 1999. Realtime personal positioning system for a wearable computer. In the digest of papers of the 3rd Intl. Symposium on Wearable Computers: 37–43.

Bahl, P. & Padmanabhan, V.N. 2000. RADAR: An in-building RF-based user location and tracking system. In Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 2: 775–784.

Bartlett, J.F. 2000. Rock 'n' scroll is here to stay [user interface]. IEEE Computer Graphics and Applications, Vol. 20(3): 40–45.

Beigl, M., 2000. Memoclip: A location based remembrance applicance, Journal of Personal Technologies, Springer Press, Vol. 4(4): 230–234.

Beigl, M, 1999. Using spatial co-location for coordination in ubiquitous computing environments. In proc. of the First Intl. Symposium on Handheld and Ubiquitous Computing: 259–273.

Brown, P.J. 1998. Triggering information by context, Personal Technologies, Vol. 2(1): 18–27.

Brown, P. J. 1996. The Stick-e Document: A framework for creating context-aware applications. In the Proc. of the Electronic Publishing: 259–272.

Brown, P.J. & Jones, G.J.F. 2001. Context-aware retrieval: exploring a new environment for information retrieval and information filtering. Personal and Ubiquitous Computing, Vol. 5(4): 253–263.

Brown, P.J., Bovey, J.D. & Chen, X. 1997. Context-aware applications: from the laboratory to the marketplace. IEEE Personal Communications Vol. 4: 58–64.

Chen, G. & Kotz, D. 2000. A Survey of context-aware mobile Computing Research. Darthmouth Computer Science Technical Report TR2000-381. http://www.cs.dartmouth.edu/reports/abstracts/TR2000-381/

Cheverst, K., Davies, N., Mitchell, K., Friday, A. & Efstratiou, C. 2000. Developing a context-aware electronic tourist guide: Some issues and experiences. CHI Letters, Vol. 2(1): 17–24.

Cheverst, K., Davies, N., Mitchell, K. & Friday, A. 1998. Design of an object model for a context-sensitive tourist guide. In Proc. of the Workshop on Interactive Applications of Mobile Computing: 24–25.

Clarkson, B., Mase, K. & Pentland, A. 2000. Recognizing user context via wearable sensors. In the digest of papers of the 4th Intl. Symposium on Wearable Computers: 69–75.

Clarkson, B. & Pentland, A. 1999. Unsupervised clustering of ambulatory audio and video. In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 6: 3037–3040.

Cohen, A. & Kovacevic´, J. 1996. Wavelets: A mathematical Background. In proceedings of the IEEE Vol. 84(4): 514–522.

Comon, P. 1994. Independent component analysis, a new concept? Signal Processing Vol. 36: 287–314.

Daubechies, I. 1992. Ten Lectures on Wavelets. SIAM.

Dey, A.K. 2000. Providing architectural support for building context-aware applications. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, USA.

Dey, A.K. & Abowd, G.D. 2000. CybreMinder: A context-aware system for supporting reminders. In the Proc. of the 2nd Intl. Symposium on Handheld and Ubiquitous Computing: 172–186.

Dey, A.K. & Abowd, G.D. 2000. Towards a better understanding of context and context-awareness. In the CHI 2000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness.

Dey, A.K., Abowd, G.D. & Wood, A. 1998. CyberDesk: A framework for providing self-integrating context-aware services. Knowledge-Based Systems, Vol. 11(1): 3–13.

Dey, A.K., Salber, D., Abowd, G.D. & Futakawa, M. 1999. The conference assistant: combining context-awareness with wearable computing. In the digest of papers of the 3rd Intl. Symposium on Wearable Computers: 21–28.

Drane, C., Macnaughtan, M. & Scott, C. 1998. Positioning GSM telephones. IEEE Communications, Vol. 36(4): 46–54.

Driankov, D., Hellendoor H. & Reinfrank, M. 1996. An Introduction to Fuzzy Control, 2nd edition, Springer.

Engelmore, R.S. & Morgan, A.J. 1988. Blackboard Systems. Addison-Wesley.

Erickson, T. 2002. Some problems with the notion of context-aware computing. Communications of the ACM, Vol. 45(2): 102–104.

Farrington, J., Moore, A.J., Tilbury, N., Church, J. & Biemond, P.D. 1999. Wearable sensor badge and sensor jacket for context-awareness. In the digest of papers of the 3rd Intl. Symposium on Wearable Computers: 107–113.

FCC Docket No. 94-102, 2000. Fourth Memorandum Opinion and Order. http://www.fcc.gov/Bureaus/Wireless/Orders/2000/da002336.html.

Flanagan J.A. 2003. A Non-Parametric Approach to Unsupervised Learning and Clustering of Symbol Strings and Sequences, In Proc. of the Workshop on Self-Organising Maps (WSOM): 128–133.

Flanagan J.A., 2003. Unsupervised Clustering of Symbol Strings. In Proc. of the Intl. Joint Conference on Neural Networks (IJCNN): 3250–3255.

Flanagan, J.A., Mäntyjärvi, J., Korpiaho, K. & Tikanmäki, J. 2002. Recognizing movements of a handheld device using symbolic representation and coding of sensor signals. In the Proc. of the 1st Intl. Conference on Mobile and Ubiquitous Multimedia: 104–112.

Franklin, D., Budzik, J. & Hammond, K. 2002. Plan-based interfaces: Keeping track of user tasks and acting to cooperate. In Proc. of the Intl. Symposium on Intelligent User Interfaces: 79–86.

Gersho, A. & Gray, R.M. 1992. Vector Quantization and Signal Compression. Kluwer.

Hand, D., Mannila, H. & Smyth, P. 2001. Principles of Data mining. MIT Press.

Harter, A., Hopper, A., Steggles, P., Ward, A. & Webster. P. 1999. The anatomy of a context-aware application. In Proc. of the 5th ACM/IEEE Intl. Conference on Mobile Computing and Networking: 59–68.

Haykin, S. 1999. Neural Networks: A Comprehensive Foundation. 2nd edition, Prentice Hall.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J. & Toivonen, H. 2001. Time series segmentation for context recognition in mobile devices. In Proc. of the IEEE Conference on Data Mining: 203–210.

Himberg, J., Flanagan, J.A. & Mäntyjärvi, J. 2003. Towards context awareness using symbol clustering map. In Proc. of the Workshop on Self-Organizing Maps (WSOM): 249–254.

Hinckley, K. & Sinclair, M. 1999. Touch-sensing input devices. In proc. of the Intl. ACM Conference on Human Factors in Computing Systems: 223–230.

Hinckley, K., Pierce, J., Sinclair, M. & Horwitz, E. 2000. Sensing techniques for mobile interaction. CHI Letters 2(2): 91–100.

Hull, R., Neaves, P. & Bedford-Roberts, J. 1997. Towards situated computing. In the digest of papers of the 1st Intl. Symposium on Wearable Computers: 146–153.

Hyvärinen, A. & Oja, E. 1997. A fast fixed-point algorithm for independent component analysis. Neural Computation, Vol. 9(7): 1483–1492.

Hyvärinen, A. 1999. Survey on independent component analysis. Neural Computing Surveys: 94–128.

Hyvärinen, A. & Oja, E. 2000. Independent component analysis: Algorithms and applications. Neural Networks, Vol. 13(4): 411–430.

Hyvärinen, A., Karhunen, J. & Oja, E. 2001. Independent Component Analysis. Wiley-Interscience.

Ifeachor, E.C. & Jervis, B.W. 1997. Digital signal processing: A practical Approach. Addison-Wesley.

Iyengar, S. & Brooks, R.R. 1997. Multi-Sensor Fusion: Fundamentals and Applications with Software. Prentice Hall.

Jolliffe, I.T. 1986. Principal Component Analysis. Springer-Verlag.

Jutten, C. & Herault, J. 1999. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Processing, (24): 1–10.

Keränen, H., Rantakokko T. & Mäntyjärvi, J. 2003. Sharing and presenting multimedia and context information within online communities using mobile terminals. In Proc. of the IEEE Intl. Conference on Multimedia and Expo, Vol. 2: 641–644.

Kohonen, T. 1990. The self-organizing map. In Proc. of the IEEE, Vol. 78(9): 1464–1480.

Kohonen, T. 1995. Self-Organizing Maps. Berlin, Springer Verlag.

Korpipää, P., Koskinen, M., Peltola, J., Mäkelä, S-M. & Seppänen T. 2003. Bayesian approach to sensor-based context awareness. Personal and Ubiquitous Computing, Springer-Verlag, Vol. 7(2): 113–124.

Kortuem, G., Segall, Z. & Bauer, M. 1998. Context-aware, adaptive wearable computers as remote interfaces to 'intelligent' environments. In the digest of papers of the 2nd Intl. Symposium on Wearable Computers: 58–65.

Lamming, M., Brown, P., Carter, K., Eldridge, M., Flynn, M., Louie, G., Robinson, P. & Sellen, A. 1994. The design of a human memory prosthesis. The Computer Journal, Vol. 37(3): 153–163.

Lee, S. & Mase, K. 2001. Recognition of walking behaviour for pedestrian navigation. In Proc. of the IEEE Intl. Conference on Control Applications: 1152–1155.

Lenat D.B. 1998. The dimensions of context space. CYCorp Document. http://www.casbah.org/resources/cycContextSpace.shtml

Lenat, D.B., Guha R. V., Pittman, K., Pratt, D. & Shepherd, M., 1990. Cyc: toward programs with common sense. Communications of the ACM, Vol. 33(8): 30–49.

Marmasse, N. & Schmandt, C. 2000. Location aware information delivery with ComMotion. In the Proc. of the Conference on Handheld and Ubiquitous Computing: 157–171.

Merriam-Webster's Collegiate Dictionary, http://www.m-w.com/home.htm.

Minsky, M. 2000. Commonsense-based interfaces. Communications of the ACM, Vol. 43(8): 66–73.

Mäntyjärvi J. 1999. Context Recognition with sensor fusion and SOM, Master of Science Thesis, University of Oulu, Finland.

Mäntyjärvi J., Tuomela U., Känsälä I. & Häkkilä J. 2003. Context studio – tool for personalizing context-aware applications in mobile terminals. To Appear in the Proc. of the conference for the Computer-Human Interaction Special Interest Group of the Ergonomics Society of Australia.

Mäntyjärvi J., Huuskonen P. & Himberg J. 2002. Collaborative context determination to support mobile terminal applications. IEEE Wireless Communications Vol. 9(5): 39–45.

Mäntyjärvi J., Takaluoma A. & Tuomela U. 2001. Controlling a terminal of a communication system. European patent application, EP 1109382A2.

Nakanishi, Y., Takayuki, T., Ohyama., M. & Hakozaki, K. 2000. Context-aware messaging service: A dynamical messaging delivery using location information and schedule information. Journal of Personal Technologies, Vol. 4: 221–224.

Oppermann, R. & Specht, M. 2000. A Context-sensitive nomadic exhibition guide. In Proc. of the Conference on Handheld and Ubiquitous Computing: 127–142.

Papoulis, A. 1991. Probability, random variables, and stochastic processes, 3rd edition, McGraw-Hill.

Pascoe, J. 1998. Adding generic contextual capabilities to wearable computers. In the digest of papers of the 2nd Intl. Symposium on Wearable Computers: 92–99.

Pascoe, J. 1997. The stick-e note architecture: Extending the interface beyond the user. In Proc. of the 2nd Intl. Conference on Intelligent user interfaces: 261–264.

Peltonen, V., Tuomi, J., Klapuri, A., Huopaniemi, J. & Sorsa, T. 2002. Computational auditory scene recognition. In Proc. of the Intl. Conference on Acoustics, Speech, and Signal Processing, Vol. 2: 1941–1944.

Pyle, D., 1999. Data Preparation for Data Mining. Morgan Kaufmann.

Rhodes, B.J. 1997. The wearable remembrance agent: A system for augmented memory. In the digest of papers of the 1st Intl. Symposium on Wearable Computers: 123–128.

Salber D. & Abowd G. D. 1998. The design and use of a generic context server. In the Proc. of the Perceptual User Interfaces Workshop: 63–66.

Salber, D., Dey, A.K. & Abowd, G.D. 1999. The context toolkit: Aiding the development of context-enabled applications. In the Proc. of the Conference on Human Factors in Computing Systems: 434–441.

Schilit, B.N. 1995. System architecture for context-aware mobile computing. Ph.D. Thesis, Columbia University, New York.

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

Schilit, B., Adams, N. & Want. R. 1994. Context-aware computing applications. In Proc. of the Workshop on Mobile Computing Systems and Applications: 85–90.

Schmidt, A. 2002. Ubiquitous computing – computing in context. Ph.D. Thesis, Lancaster University, UK.

Schmidt, A. 2000. Implicit human computer interaction through context. Journal of Personal Technologies, Vol. 4: 191–199.

Schmidt, A. & Van Laerhoven, K. 2001. How to build smart appliances? IEEE Personal Communications Vol. 8(4): 66–71.

Schmidt, A., Stuhr, T. & Gellersen, H-W. 2001. Context-phonebook – extending mobile phone applications with context. In 3rd Mobile Human-Computer Interaction Workshop, Lille.

Schmidt, A., Aidoo, K.A., Takaluoma, A, Tuomela, U., Van Laerhoven, K. & Van de Velde. W. 1999. Advanced interaction in context. In Proc. of the 1st Intl. Symposium on Handheld and Ubiquitous Computing: 89–101.

Sekine, M., Tamura, T., Ogawa, M., Togawa, T. & Fukui, Y. 1998. Classification of acceleration waveform in a continuous walking record. In Proc. of the 20th Annual International Conference of the Engineering in Medicine and Biology Society, Vol. 3: 1523–1526.

Shneiderman, B. 1998. Designing the User Interface, Strategies for Effective Human-Computer Interaction, 3rd edition, Addison Wesley.

Starner, T., Schiele, B. & Pentland, A. 1998. Visual contextual awareness in wearable computing. In the digest of of papers of the 2nd Intl. Symposium on Wearable Computers: 50–57.

Sumi, K. & Nishida, T. 2001. Telme: a personalized, context-aware communication support system. IEEE Intelligent Systems, Vol. 16(3): 21–27.

Suomela, R. & Lehikoinen, J. 2000. Context compass. In digest of papers of the 4th Intl. Symposium on Wearable Computers: 147–154.

Theodoridis, S. & Koutroumbas, K. 1999. Pattern Recognition. Academic Press.

Tikkanen, P. 1999. Characterization and application of analysis methods for ECG and time interval variability data, A 323, Acta Universitatis Ouluensis.

Tuulari E. 2000. Context aware hand-held devices. Licenciate Thesis, Technical Research Centre of Finland.

Van Laerhoven, K. & Cakmakci, O. 2000. What shall we teach our pants? In digest of papers of the 4th Intl. Symposium on Wearable Computers: 77–83.

Vigário R., N. 1999. Independent component approach to the analysis of EEG and MEG signals. Ph.D. Thesis, Acta Polytechnica Scandinavia, Mathematics and Computing Sciences, The Finnish Academy of Technology.

Want, R., Hopper, A. & Falcao, V., Gibbons J. 1992. The active badge location system. ACM Transactions on Information Systems, Vol. 10(1): 91–102.

Want, R., Schilit, B., Adams, N., Gold, R., Petersen, K., Goldberg, D., Ellis, J. & Weiser M. 1995. An overview of the PARCTAB ubiquitous computing experiment. IEEE Personal Communications Vol. 2(6): 28–43.

Weiser, M. 1991. The computing for the 21th century. Scientific American, Vol. 265(3): 94–104.

Zadeh, L. A. 1965. Fuzzy sets. Information and Control, Vol. 8: 338–353.

Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.vtt.fi/inf/pdf/)

Published by

Series title, number and report code of publication

VTT Publications 511
VTT–PUBS–511

Author(s)
Mäntyjärvi, Jani

Title
Sensor-based context recognition for mobile applications

Abstract
Context-aware computing is proposed as an enabling technology for adaptating different functions in computer-aided devices. The development of context-awareness for mobile devices requires recognition and extraction of implicit context information from the usage situations and environment of a device. Context information is provided for applications and services, which adapt their appearance and functions accordingly. Mobile devices contain several potential context data sources such as, location, time and applications. In this thesis, sensors integrated into a mobile device are utilised as sources for context information.

The main challenge in sensor-based context-aware computing for mobile devices is how to define and carry out context recognition from sensor signals to facilitate use of context information in mobile applications. In this thesis we have divided this into specific research problems: How should low-level context information be extracted from sensor signals to obtain a rich and usable representation? How should low-level context information be processed and examined to obtain higher-level contexts? How to utilise context representation in applications? An empiric and data centric approach including signal processing, feature extraction and explorative data analysis methods is used in examining and defining a procedure for sensor-based context recognition.

The main result of this work is a procedure for sensor-based context recognition that is demonstrated with experiments and with the applications developed. The main technical solutions developed include methods for extracting context information and converting it into a suitable context representation, solution for collaborative recognition of the context of a group of mobile devices, an approach for controlling mobile applications and a solution for enhancing remote communication with context information. The thesis includes a review of context data processing and the utilisation of context information in mobile devices.

Keywords
Ubiquitous computing, pervasive computing, mobile computing, pattern recognition, data exploration, human-computer interaction

Activity unit
VTT Electronics, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland

ISBN	Project number
951–38–6253–4 (soft back ed.) 951–38–6254–2 (URL:http://www.vtt.fi/inf/pdf/)	

Date	Language	Pages	Price
November 2003	English	118 p. + app. 60 p.	D

Name of project	Commissioned by

Series title and ISSN	Sold by
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)	VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374

The utilisation of context information in mobile devices requires recognition and extraction of implicit information from the usage situations and environment of a device. In this thesis, sensors integrated into a mobile device are used as context sources. Main challenges in sensor-based context awareness are how to perform context recognition from sensor signals and how to represent extracted information to facilitate the development of context-aware mobile applications. This thesis presents a context recognition procedure that is demonstrated with experiments and with the mobile applications developed. Technical solutions developed also enable collaborative recognition of the context of a group of mobile devices, an approach for controlling mobile applications and enhancing remote communication with context information.

Tätä julkaisua myy

VTT TIETOPALVELU
PL 2000
02044 VTT
Puh. (09) 456 4404
Faksi (09) 456 4374

Denna publikation säljs av

VTT INFORMATIONSTJÄNST
PB 2000
02044 VTT
Tel. (09) 456 4404
Fax (09) 456 4374

This publication is available from

VTT INFORMATION SERVICE
P.O.Box 2000
FIN–02044 VTT, Finland
Phone internat. +358 9 456 4404
Fax +358 9 456 4374

ISBN 951–38–6253–4 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6254–2 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)

VTT PUBLICATIONS 511 Sensor-based context recognition for mobile applications Jani Mäntyjärvi