Space, Time and Ambient Intelligence

IJCAI 2011 Workshop Proceedings

Mehul Bhatt, Hans W. Guesgen, Juan Carlos Augusto (Eds.)

SFB/TR 8
SPATIAL COGNITION

SFB/TR 8 Report No. 027-06/2011
Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition
Universität Bremen / Universität Freiburg

Contact Address:

Dr. Thomas Barkowsky
SFB/TR 8
Universität Bremen
P.O.Box 330 440
28334 Bremen, Germany

Tel +49-421-218-64233
Fax +49-421-218-64239
barkowsky@sfbtr8.uni-bremen.de
www.sfbtr8.uni-bremen.de

IJCAI 2011

Workshop proceedings

Space, Time and Ambient Intelligence

STAMI 2011
International Joint Conference on Artificial Intelligence
Barcelona, Spain

16 July 2011

Organizing and Editorial Committee

Mehul Bhatt
SFB/TR 8 Spatial Cognition
University of Bremen
P.O. Box 330 440, 28334 Bremen, Germany
T +49 (421) 218 64 237
F +49 (421) 218 98 64 237
bhatt@informatik.uni-bremen.de

Hans W. Guesgen
School of Engineering and Advanced Technology
Massey University
Private Bag 11222, Palmerston North 4442, New Zealand
T +64 (6) 356 9099 extn 7364
F +64 (6) 350 2259
h.w.guesgen@massey.ac.nz

Juan Carlos Augusto
School of Computing and Mathematics
University of Ulster at Jordanstown
Shore Road, BT37 0QB Newtownabbey, Co. Antrim
United Kingdom
JC.Augusto@ulster.ac.uk

Advisory and Program Committee

André Trudel (Acadia University, Canada)
Asier Aztiria (University of Mondragon, Spain)
Boi Faltings (EPFL, Switzerland)
Christian Freksa (University of Bremen, Germany)
Christophe Claramunt (Naval Academy Research Institute, France)
Frank Dylla (University of Bremen, Germany)
Fulvio Mastrogiovanni (University of Genoa, Italy)
Hans Guesgen (Massey University, New Zealand)
Jason Jingshi Li (EPFL, Switzerland)
Juan Carlos Augusto (University of Ulster, United Kingdom)
Kai-Florian Richter (University of Melbourne, Australia)
Kathleen-Stewart Hornsby (University of Iowa, USA))
Mehul Bhatt (University of Bremen, Germany)
Michel Klein (Vrije University, Netherlands)
M. Sasikumar (CDAC Mumbai, India)
Norbert Streitz (Smart Future Initiative, Germany)
Nico Van de Weghe (Ghent University, Belgium)
Paulo E. Santos (Technical University FEI, Brazil)
Reinhard Moratz (University of Maine, USA)
Seng Loke (La Trobe University, Australia)
Shyamanta Hazarika (Tezpur University, India)

Organizing Institutions

SFB/TR 8 Spatial Cognition. University of Bremen, and University of Freiburg (Germany)

Massey University (New Zealand)

University of Ulster at Jordanstown (United Kingdom)

Contents

Preface

Welcome to the Workshop on Space, Time and Ambient Intelligence (STAMI) at the International Joint Conference on Artificial Intelligence 2011 in Barcelona, Spain. This workshop is a follow-up to the first edition of STAMI held at the COSIT 2009 conference in France, and is one of the initiatives being conducted within the overall STAMI framework. STAMI is focussed on the theoretical and application-centred questions pertaining to reasoning about space, time, actions, events, and change in the domain of intelligent and smart environments, and ambient intelligence in general.

A wide-range of application domains within the fields of ambient intelligence and ubiquitous computing environments require the ability to represent and reason about dynamic spatial and temporal phenomena. Real world ambient intelligence systems that monitor and interact with an environment populated by humans and other artefacts require a formal means for representing and reasoning with spatio-temporal, event and action based phenomena that are grounded to real aspects of the environment being modelled. A fundamental requirement within such application domains is the representation of dynamic knowledge pertaining to the spatial aspects of the environment within which an agent, system or robot is functional. At a very basic level, this translates to the need to explicitly represent and reason about dynamic spatial configurations or scenes and desirably, integrated reasoning about space, actions and change. With these modelling primitives, primarily the ability to perform predictive and explanatory analyzes on the basis of available sensory data is crucial toward serving a useful intelligent function within such environments.

The emerging fields of ambient intelligence and ubiquitous computing will benefit immensely from the vast body of representation and reasoning tools that have been developed in Artificial Intelligence in general, and the sub-field of Spatial and Temporal Reasoning in specific. There have already been proposals to explicitly utilise qualitative spatial calculi pertaining to different spatial domains for modelling the spatial aspect of an ambient environment (e.g., smart homes and offices) and also to utilize a formal basis for representing and reasoning about space, change and occurrences within such environments. Through this workshop, and the STAMI initiative, we aim to bring together academic and industrial perspectives on the application of artificial intelligence in general, and reasoning about space, time and actions in particular, for the domain of smart and intelligent environments.

Mehul Bhatt, Hans Guesgen, Juan Carlos Augusto
(STAMI 2011 Co-Chairs)

Predicting User Movements in Heterogeneous Indoor Environments by Reservoir Computing*

Davide Bacciu and **Claudio Gallicchio**
Alessio Micheli and **Stefano Chessa**
Universita di Pisa
Pisa, Italy
{bacciu,gallicch,micheli,ste}@di.unipi.it

Paolo Barsocchi
ISTI-CNR
Pisa, Italy
paolo.barsocchi@isti.cnr.it

Abstract

Anticipating user localization by making accurate predictions on its indoor movement patterns is a fundamental challenge for achieving higher degrees of personalization and reactivity in smart-home environments. We propose an approach to real-time movement forecasting founding on the efficient Reservoir Computing paradigm, predicting user movements based on streams of Received Signal Strengths collected by wireless motes distributed in the home environment. The ability of the system to generalize its predictive performance to unseen ambient configurations is experimentally assessed in challenging conditions, comprising external test scenarios collected in home environments that are not included in the training set. Experimental results suggest that the system can effectively generalize acquired knowledge to novel smart-home setups, thereby delivering an higher level of personalization while decreasing costs for installation and setup.

1 Introduction

Localization and tracking of mobile users in indoor environments are important services in the construction of smart spaces, and they are even considered enabling, baseline services for Ambient Assisted Living (AAL) [AAL, 2009] applications. In fact, AAL aims at improving the quality of life of elderly or disabled people, by assisting them in their daily life, in order to preserve their autonomy and by making them feeling included, protected and secure in the places where they live or work (typically their home, their office, the hospital and any other places where they may spend significant part of their time). These objectives can be granted only if the appropriate services are delivered to the users in the right time and in the right pace.

In AAL applications, localization aims at the real time estimation of the user position, while tracking refers to the activity of reconstructing the path of the user, with the purpose of anticipating its future position and thus to prepare the system to the timely delivery of the appropriate services. Localization and tracking of objects can be achieved by means of a large number of different technologies, however only few of them are suitable for AAL applications, as they should be non-invasive on the users, they must be suited to the deployment in the user houses at a reasonable cost, and they should be accepted by the users themselves. On the other hand, accuracy in the position estimation is subject to less requirements than it may occur in other applications (accuracies in the order of the centimeter or below are typically not required). Considering all these constraints, a promising technology for this services is based on Wireless sensor networks (WSN) [Baronti *et al.*, 2007], due to their properties of cost and time effective deployment. Within such WSN, it is possible to estimate the location of a user by exploiting Received Signal Strength (RSS) information, that is a measure of the power of a received radio signal that can be obtained from almost any wireless device.

The measurement of RSS values over time provides information on the user trajectory under the form of a time series of sampled signal strength. The relationship between the RSS and the location of the tracked object cannot be easily formulated into an analytical model, as it strongly depends on the characteristics of the environment as well as on the wireless devices involved. In this sense, computational learning models have received much interest as they allow to learn such relationship directly from the data. These approaches typically exploit probabilistic learning techniques to learn a probabilistic estimate of user location given RSS measurements at known location [Zàruba *et al.*, 2007]. However, such models have considerable computational costs connected both with the learning and the inference phase, which might grow exponentially with the number of sensors in the area. Further they do little to exploit the sequential nature of the RSS streams, whereas they provide static pictures of the actual state of the environment. There exist several machine learning approaches capable of explicitly dealing with signals characterized by such time-dependent dynamics including, for instance, probabilistic Hidden Markov Models (HMM), Recurrent Neural Networks (RNN) and kernel methods for sequences. In this paper, we focus on a computationally efficient neural paradigm for modeling of RNNs, that is known as Reservoir Computing (RC). In particular, we con-

[*] This work is partially supported by the EU FP7 RUBICON project (contract n. 269914).

sider Echo State Networks (ESNs) [Jaeger and Haas, 2004; Jaeger, 2001], that are dynamical neural networks used for sequence processing. The contractive reservoir dynamics provides a fading memory of past inputs, allowing the network to intrinsically discriminate among different input histories [Jaeger, 2001] in a suffix-based fashion [Tiño *et al.*, 2007; Gallicchio and Micheli, 2011], even in absence of training.

The most striking feature of ESNs is its efficiency: training is limited to the linear outputs whereas the reservoir is fixed; additionally, the cost of input encoding scales linearly with the length of the sequence for both training and test. In this regard, the ESN approach compares favorably with competitive state-of-the-art learning models for sequence domains, including general RNNs, in which the dynamic recurrent part is trained, e.g. [Kolen and Kremer, 2001], probabilistic Hidden Markov Models, that pay consistent additional inference costs also at test time, and Kernel Methods for sequences, whose cost scales at least quadratically with the input length, e.g. [Gärtner, 2003]). ESNs have been successfully applied to several tasks in the area of sequence processing, often outperforming other state-of-the-art learning models (see [Jaeger and Haas, 2004; Jaeger, 2001]). Recently, ESNs have shown good potential in a range of tasks related to autonomous systems modeling, e.g. as regards event detection and localization in robot navigation [Antonelo *et al.*, 2008; 2007] and multiple robot behavior modeling [Waegeman *et al.*, 2009]. However, such applications are mostly focused on modeling robot behaviors and often use artificial data obtained by simulators.

In this paper, we apply the ESN approach to a real-world scenario for user indoor movements forecasting, using real and noisy RSS input data, paving the way for potential applications in the field of AAL. The experimental assessment is intended to show that the proposed technology has a strong potential to be deployed in real-life situations, in particular as regards the ability of generalizing the prediction performance to unknown environments. In this sense, we expect that the proposed solution will increase the level of service personalization by making accurate prediction of the user spatial context, while yielding to a reduction of the setup and installation costs thanks to its generalization capability.

2 User Movement Prediction in Indoor Environments

2.1 Localization by Received Signal Strength

The exploitation of wireless communication technologies for user localization in indoor environments has recently received much attention by the scientific community, due to the potential of service personalization involved in an accurate identification of the user spatial context. Cost efficiency is a critical aspect in order to determine the success of such localization technologies. In this sense, the most promising localization approaches are certainly those based on Received Signal Strength (RSS) information, that is a measure of the power of a received radio signal. RSS measurements can be readily obtained from (potentially) any wireless communication device, being a standard feature in most radio equipments. In a, not so far-ahead, scenario, we foresee an ubiquitous diffusion of wireless sensors in the environment (e.g. monitoring temperature, humidity, pollution, etc.), together with a wide availability of radio devices on the user's body (e.g. personal electronics, sensors monitoring health status, etc.). Therefore, irrespectively of the intended use of such sensors and devices, we expect to be able to exploit their radio apparatus to obtain noisy, yet potentially informative, RSS traces for realtime user localization.

Indoor positioning systems based on RSS information are getting increasing attention due to the widespread deployment of WLAN infrastructures, given that RSS measures are available in every 802.11 interface. Mainly, we distinguish between two alternative approaches to localize users leveraging the RSS measurements, i.e. *model-based* and *fingerprinting* positioning. Model-based positioning is popular approach in literature that founds on expressing radio frequency signal attenuation using specific path loss models [Barsocchi *et al.*, 2011]. Given an observed RSS measurement, these methods triangulate the person based on distance calculations from multiple access points. However, the relationship between the user position and the RSS information is highly complex and can hardly be modeled due to multipath, metal reflection, and interference noise. Thus, RSS propagation may not be adequately captured by a fixed invariant model. Differently from model-based approaches, fingerprinting techniques, such as [Kushki *et al.*, 2007], create a radio map of the environment based on RSS measurements at known positions throughout an offline map-generation phase. Clearly, the localization performance of fingerprinting-based model relies heavily on the choice of the distance function that is used to compute the similarity between the RSS measured in the online phase, with the known RSS fingerprints. Further, the offline-generated ground truth needs to be revised in case of changes to the room/environment configuration which result in relevant discrepancies in the known fingerprints.

The user localization approaches discussed above focus on finding accurate estimates of the current user position, but lack the ability of anticipating his/her future location. Being capable of predicting the future user context is of fundamental value to enhance the reactivity and personalization of smart services in indoor environments. In the following, we describe a real-life office scenario targeted at adaptive user movement prediction using RSS traces: a brief discussion of the wireless technology involved is provided together with a detailed description of the experimental indoor environment.

2.2 Movement Prediction Scenario

A measurement campaign has been performed on the first floor of the the ISTI institute of CNR in the Pisa Research Area, in Italy. The scenario is a typical office environments comprising 6 rooms with different geometry, arranged into pairs such that coupled rooms (referred as Room 1 and Room 2 in the following) have fronting doors divided by an hallway, as depicted in Fig. 1. Rooms contain typical office furniture: desks, chairs, cabinets, monitors that are asymmetrically arranged. From the point of view of wireless communications, this is a harsh environment due the to multi-path reflections caused by walls and the interference produced by electronic

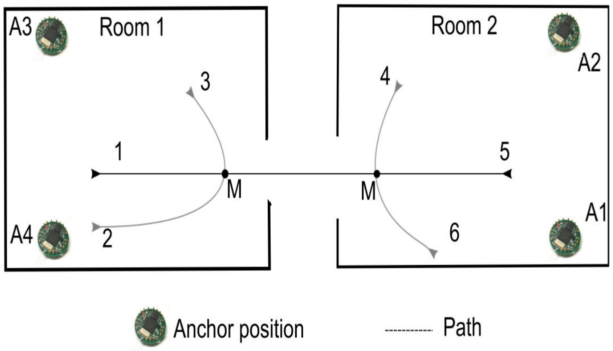


Figure 1: Schematic view of the experimental setting: anchors' position and prototypical user movements are shown. Straight paths, labeled as 1 and 5, yield to a room change, while curved movement (paths 2, 3, 4 and 6) preserve the spatial context. The M markers denote the points where we predict if the user is about to change its location. The actual setting differs from the schematics by the presence of office furniture (covering roughly 50% of the space) that is asymmetrically arranged and influences the actual user trajectories in the different rooms.

Dataset Number	length [m]	width [m]
1	4.5	12.6
2	4.5	13.2
3	4	12.6

Table 1: Physical layout of the 3 room couples.

devices. Experimental measurements have been performed by a sensor network of 5 IRIS nodes[1] embedding a Chipcon AT86RF230 radio subsystem that implements the IEEE 802.15.4 standard. Four sensors, in the following *anchors*, are located in fixed positions in the environment and one sensor is placed on the user, hereafter called *mobile*.

The measurement campaign comprises experiments on three different couple of rooms with a total surface spanning from 50 m^2 to about 60 m^2. Table 1 details the environment dimensions for the three couple of rooms, hereby referred as dataset 1, dataset 2 and dataset 3. Experiments consisted in measuring the RSS between anchors and mobile for a set of repeated user movements. Figure 1 shows the anchors deployed in the environment as well as a prototypical trajectory for each type of user movement. The height of the anchors has been set to $1.5m$ from the ground and the mobile was worn on the chest. The measurements were carried out on free paths to facilitate a constant speed of the user of about 1 m/s. Measures denote RSS samples (integer values ranging from 0 to 100) collected by sending a beacon packet from the anchors to the mobile at regular intervals, 8 times per second, using the full transmission power of the IRIS.

Path Type	Dataset 1	Dataset 2	Dataset 3
1	26	26	27
2	26	13	12
3	-	13	12
4	13	14	13
5	26	26	27
6	13	14	13
Tot. Changed	52	52	54
Tot. Unchanged	52	54	50
Lengths *min-max*	19-32	34-119	29-129

Table 2: Statistics of the collected user movements.

Experimentation gathered information on 6 prototypical paths that are shown in Fig. 1 with arrows numbered from 1 to 6: two movement types are considered for the prediction task, that are straight and curved trajectories. The former run from Room 1 to Room 2 or viceversa (paths 1 and 5 in Fig. 1) and yield to a change in the spatial context of the user, while curved movements (paths 2, 3, 4 and 6 in Fig. 1) preserve the spatial context. Table 2 summarizes the statistics of the collected movement types for each dataset: due to physical constraints, dataset 1 does not have a curved movement in Room 1 (path 3). The number of trajectories leading to a room change, with respect to those that preserve the spatial context, is indicated in Table 2 as ”Tot. Change” and ”Tot. Unchanged”, respectively. Each path produces a trace of RSS measurements that begins from the corresponding arrow and that is marked when the user reaches a point (denoted with M in Fig. 1) located at $0.6m$ from the door. Overall, the experiment produced about 5000 RSS samples from each of the 4 anchors and for each dataset. The marker M is the same for all the movements, therefore different paths cannot be distinguished based only on the RSS values collected at M.

The experimental scenario and the gathered RSS measures can naturally be exploited to formalize a binary classification task on time series for movements forecasting. The RSS values from the four anchors are organized into sequences of varying length (see Table 2) corresponding to trajectory measurements from the starting point until marker M. A target classification label is associated to each input sequence to indicate wether the user is about to change its location (room) or not. In particular, target class $+1$ is associated to location changing movements (i.e. paths 1 and 5 in Fig. 1), while label -1 is used to denote location preserving trajectories (i.e. paths $2, 3, 4$ and 6 in Fig. 1). The resulting dataset is made publicly available for download[2].

3 Reservoir Computing for Movement Prediction

Reservoir Computing (RC) is a computational paradigm covering several models in the Recurrent Neural Network (RNN) family, that are characterized by the presence of a large and sparsely connected hidden *reservoir* layer of recurrent non-linear units, that are read by means of some read-out mechanism, i.e. typically a linear combination of the reservoir

[1] Crossbow Technology Inc., `http://www.xbow.com`

[2] `http://wnlab.isti.cnr.it/paolo/index.php/dataset/6rooms`

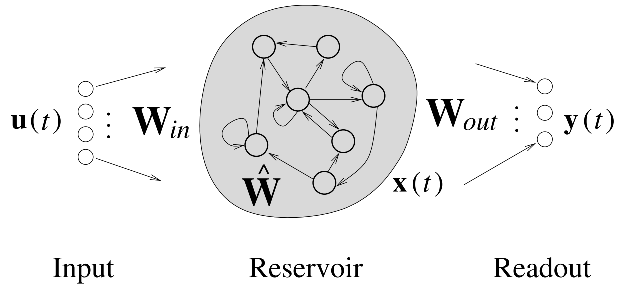


Figure 2: The architecture of an ESN: $\mathbf{W}_{in}$, $\hat{\mathbf{W}}$ and $\mathbf{W}_{out}$ denote the input, the reservoir and the output weights, respectively. Terms $\mathbf{u}(t)$ and $\mathbf{y}(t)$ identify the input at time t and the corresponding predicted read-out; $\mathbf{x}(t)$ is the associated reservoir state. Further details are given in the text.

outputs. With respect to traditional RNN training, where all weights are adapted, RC performs learning mainly on the output weights, leaving those in the reservoir untrained. As other RNNs, RC models are well suited to modeling of dynamical systems and, in particular, to temporal data processing. As the movement prediction problem discussed in this paper is, from a machine learning perspective, a time-series prediction task, we are naturally interested in analyzing and discussing the effectiveness of the RC paradigm on such a scenario. In particular, we focus on the computationally efficient ESNs [Jaeger, 2001; Jaeger and Haas, 2004; Lukosevicius and Jaeger, 2009], that are one of the best known RC models, that are characterized by an input layer of N_U units, an hidden reservoir layer of N_R *untrained recurrent non-linear units* and a *readout* layer of N_Y feed-forward linear units (see Fig. 2). Within a time-series prediction task, the untrained reservoir acts as a *fixed* non-linear temporal expansion function, implementing an encoding process of the input sequence into a state space where the trained linear readout is applied.

Standard ESN reservoirs are built from simple additive units with a sigmoid activation function which, however, has been shown to weakly model the temporal evolution of slow dynamical systems [Jaeger *et al.*, 2007]. In particular, [Gallicchio *et al.*, 2011] have shown that indoor user movements can be best modeled by a *leaky integrator* type of RC network (LI-ESNs) [Jaeger *et al.*, 2007]. Given an input sequence $\mathbf{s} = [\mathbf{u}(1), \ldots, \mathbf{u}(n)]$ over the input space $\mathbb{R}^{N_U}$, at each time step $t = 1, \ldots, n$, the LI-ESN reservoir computes the following state transition

$$\mathbf{x}(t) = (1-a)\mathbf{x}(t-1) + af(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t-1)), \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ denotes the reservoir state (i.e. the output of the reservoir units) at time step t, $\mathbf{W}_{in} \in \mathbb{R}^{N_R \times N_U}$ is the input-to-reservoir weight matrix (possibly including a bias term), $\hat{\mathbf{W}} \in \mathbb{R}^{N_R \times N_R}$ is the (sparse) recurrent reservoir weight matrix and f is the component-wise applied activation function of the reservoir units (we use $f \equiv tanh$). The temporal recursion in (1) is based on a null initial state, i.e. $\mathbf{x}(0) = \mathbf{0} \in \mathbb{R}^{N_R}$. The term $a \in [0, 1]$ is a *leaking rate* parameter, which is used to control the speed of the reservoir dynamics, with small values of a resulting in reservoirs that react slowly to the input [Jaeger *et al.*, 2007; Lukosevicius and Jaeger, 2009]. Compared to the standard ESN model, LI-ESN applies an exponential moving average to the state values produced by the reservoir units (i.e. $\mathbf{x}(t)$), resulting in a low-pass filter of the reservoir activations that allows the network to better handle input signals that change slowly with respect to the sampling frequency. LI-ESN state dynamics are therefore more suitable for representing the history of input signals.

For a binary classification task over sequential data, the linear readout is applied only after the encoding process computed by the reservoir is terminated, by using

$$\mathbf{y}(\mathbf{s}) = sgn(\mathbf{W}_{out}\mathbf{x}(n)), \quad (2)$$

where sgn is a sign threshold function returning $+1$ for non-negative arguments and -1 otherwise, $\mathbf{y}(\mathbf{s}) \in \{-1, +1\}^{N_Y}$ is the output classification computed for the input sequence $\mathbf{s}$ and $\mathbf{W}_{out} \in \mathbb{R}^{N_Y \times N_R}$ is the reservoir-to-output weight matrix (possibly including a bias term).

The reservoir is initialized to satisfy the so called *Echo State Property* (ESP) [Jaeger, 2001]. The ESP asserts that the reservoir state of an ESN driven by a long input sequence only depends on the input sequence itself. Dependencies on the initial states are progressively forgotten after an initial *transient* (the reservoir provides an echo of the input signal). A sufficient and a necessary condition for the reservoir initialization are given in [Jaeger, 2001]. Usually, only the necessary condition is used for reservoir initialization, whereas the sufficient condition is often too restrictive [Jaeger, 2001]. The necessary condition for the ESP is that the system governing the reservoir dynamics of (1) is locally asymptotically stable around the zero state $\mathbf{0} \in \mathbb{R}^{N_R}$. By setting $\tilde{\mathbf{W}} = (1-a)\mathbf{I} + a\hat{\mathbf{W}}$, where a is the leaking rate parameter, the necessary condition is satisfied whenever the following constraint holds:

$$\rho(\tilde{\mathbf{W}}) < 1 \quad (3)$$

where $\rho(\tilde{\mathbf{W}})$ is the *spectral radius* of $\tilde{\mathbf{W}}$. Matrices $\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$ are therefore randomly initialized from a uniform distribution, and $\hat{\mathbf{W}}$ is successively scaled such that (3) holds. In practice, values of ρ close to 1 are commonly used, leading to reservoir dynamics close to the edge of chaos, often resulting in the best performance in applications (e.g. [Jaeger, 2001]).

In sequence classification tasks, each training sequence is presented to the reservoir for a number of $N_{transient}$ consecutive times, to account for the initial transient. The final reservoir states corresponding to the training sequences are collected in the columns of matrix $\mathbf{X}$, while the vector $\mathbf{y}_{target}$ contains the corresponding target classifications (at the end of each sequence). The linear readout is therefore trained to solve the least squares linear regression problem

$$\min \|\mathbf{W}_{out}\mathbf{X} - \mathbf{y}_{target}\|_2^2 \quad (4)$$

Usually, Moore-Penrose pseudo-inversion of matrix $\mathbf{X}$ or ridge regression are used to train the readout [Lukosevicius and Jaeger, 2009].

4 Experimental Evaluation

We evaluate the effectiveness of the RC approach to user movement prediction on the real-life scenario described in Section 2.2. In particular, we assess the ability of the proposed approach to generalize its prediction to unseen indoor environments, which is a fundamental property for the deployment as a movement prediction system in real-life applications. To this end, we define an experimental evaluation setup where RC training is performed on RSS measurements corresponding to only 4 out of 6 rooms of the scenario, while the remaining 2 offices are used to test the generalization capability of the RC model.

In [Gallicchio *et al.*, 2011], it has been analyzed the baseline performance of different ESN models on user movement prediction with a small 2-rooms dataset. Such an analysis suggests that the LI-ESN model, described in Section 3, is best suited to deal with slowly changing RSS time series. Therefore, in the remainder of the section, we limit our analysis to the assessment of a leaky-integrated model, with meta-parameters chosen as in [Gallicchio *et al.*, 2011]. In particular, we consider LI-ESNs comprising reservoirs of $N_R = 500$ units and a 10% of randomly generated connectivity, spectral radius $\rho = 0.99$, input weights in $[-1, 1]$ and leaking rate $a = 0.1$. Results refer to the average of 10 independent and randomly guessed reservoirs. The readout ($N_Y = 1$) is trained using pseudo-inversion and ridge regression with regularization parameter $\lambda \in \{10^{-i} | i = 1, 3, 5, 7\}$.

Input data comprises time series of 4 dimensional RSS measurements (N_U = 4) corresponding to the 4 anchors in Fig. 1, normalized in the range $[-1, 1]$ independently for each dataset in Table 1. Normalized RSS sequences are feed to the LI-ESN network only until the marker signal M. To account for the the initial reservoir transient, each input sequence is presented consequently for 3 times to the networks.

We have defined 2 experimental settings (ES) that are intended to assess the predictive performance of the LI-ESNs when training/test data comes from both uniform (ES1) and previously unseen ambient configurations (ES2), i.e. providing an external test set. To this aim, in ES1, we have merged datasets 1 and 2 to form a single dataset of 210 sequences. A training set of size 168 and a test set of size 42 have been obtained for the ES1, with stratification on the path types. The readout regularization parameter $\lambda = 10^{-1}$ has been selected in the ES1, on a (33%) validation set extracted from the training samples. In ES2, we have used the LI-ESN with the readout regularization selected in the ES1, and we have trained it on the union of datasets 1 and 2 (i.e. 4 rooms), using dataset 3 as an external test set (with measurements from 2 unknown environments). Table 3 reports the mean test accuracy for both the ESs. An excellent predictive performance is achieved for ES1, which is coherent with the results reported in [Gallicchio *et al.*, 2011]. Such an outcome is noteworthy, as the performance measurements in [Gallicchio *et al.*, 2011] have been obtained in a much simpler experimental setup, comprising RSS measurements from a single pair of rooms (that differ from those considered in this study). This seems to indicate that the LI-ESN approach, on the one hand, scales well as the number of training environments increases while, on the other hand, it is robust to changes to the training room configurations. Note that RSS trajectories for different rooms are, typically, consistently different and, as such, the addition of novel rooms strongly exercises the short-term memory of the reservoirs and their ability to encode complex dynamical signals (see RSS examples in Fig. 3).

ES 1	**ES** 2
$95.95\%(\pm 3.54)$	$89.52\%(\pm 4.48)$

Table 3: Mean test accuracy (and standard deviation) of LI-ESNs for the two ESs.

		LI-ESN Prediction	
		+1	**-1**
Actual	**+1**	$44.04\%(\pm 5.17)$	$7.88\%(\pm 5.17)$
	-1	$2.60\%(\pm 2.06)$	$45.48\%(\pm 2.06)$

Table 4: Mean confusion matrix (expressed in % over the number of samples) on the ES2 external test-set.

The result on the ES2 setting is more significative, as it shows a notable generalization performance for the LI-ESN model, that reaches a predictive accuracy close to 90% on the external test comprising unseen ambient configurations. Table 4 describes the confusion matrix of the external test-set in ES2, averaged over the reservoir guesses and expressed as percentages over the number of test samples. This allows appreciating the equilibrium of the predictive performance, that has comparable values for both classes. Note that total accuracy is obtained as the sum over the diagonal, while error is computed from the sum of the off-diagonal elements.

5 Conclusion

We have presented a RC approach to user movement prediction in indoor environments, based on RSS traces collected by low-cost WSN devices. We exploit the ability of LI-ESNs in capturing the temporal dynamics of slowly changing noisy RSS measurements to yield to very accurate predictions of the user spatial context. The performance of the proposed model has been tested on challenging real-world data comprising RSS information collected in real office environments.

We have shown that, with respect to the work in [Gallicchio *et al.*, 2011], the LI-ESN approach is capable of generalizing its predictive performance to training information related to multiple setups. More importantly, it can effectively generalize movement forecasting to previously unseen environments, as shown by the external test-set assessment. Such flexibility is of paramount importance for the development of practical smart-home solutions, as it allows to consistently reduce the installation and setup costs. For instance, we envisage a scenario in which an ESN-based localization system is trained off-line (e.g. in laboratory/factory) on RSS measurements captured on a (small) set of sample rooms. Then, the system is deployed and put into operation into its target environment, reducing the need of an expensive fine tuning phase.

In addition to accuracy and generalization, a successful context-forecasting technology has also to possess sufficient

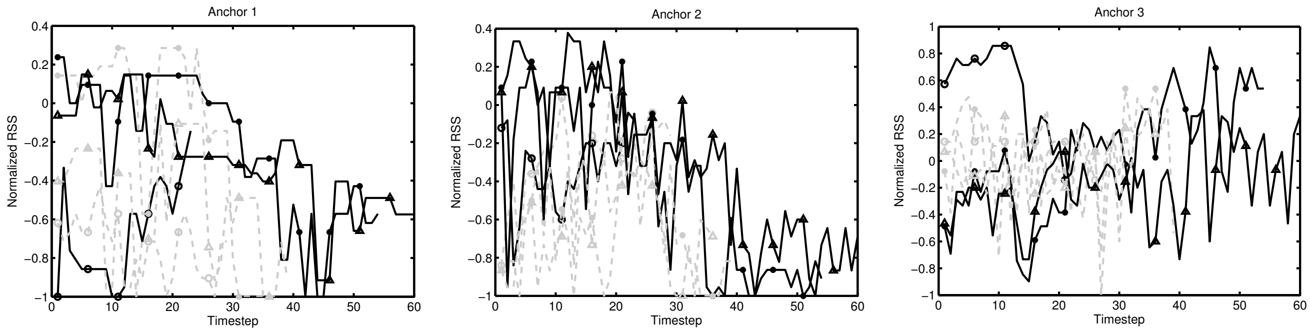


Figure 3: Examples of RSS sequences in the 3 datasets: trajectories leading to a room change are denoted as continuous lines, while dashed curves are examples from the negative class. Circles, stars and triangles denote sequences from dataset 1, 2 and 3, respectively. Due to space constraints, RSS streams are shown only for 3 out of 4 available anchors.

reactivity, so that predictions are delivered timely to the high-level control components. In this sense, ESN is a good candidate to optimize the trade-off between accuracy, generalization and computational requirements among machine learning models for sequential data. Such potential can be further exploited by developing a distributed system that embeds the ESN learning modules directly into the nodes of the wireless networks. By virtue of ESN's limited computational requirements, we envisage that such solution could be cost-effectively realized on WSNs comprising simple computationally constrained devices (e.g. see the objectives of the EU FP7 RUBICON project[3]).

References

[AAL, 2009] AAL. Ambient assisted living roadmap, 2009.

[Antonelo *et al.*, 2007] E. A. Antonelo, B. Schrauwen, and J. M. Van Campenhout. Generative modeling of autonomous robots and their environments using reservoir computing. *Neural Proc. Lett.*, 26(3):233–249, 2007.

[Antonelo *et al.*, 2008] E. A. Antonelo, B. Schrauwen, and D. Stroobandt. Event detection and localization for small mobile robots using reservoir computing. *Neural Netw.*, 21(6):862–871, 2008.

[Baronti *et al.*, 2007] P. Baronti, P. Pillai, V. W.C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655 – 1695, 2007.

[Barsocchi *et al.*, 2011] P. Barsocchi, S. Lenzi, S. Chessa, and F. Furfari. Automatic virtual calibration of range-based indoor localization systems. *Wireless Comm. and Mobile Comp.*, 2011.

[Gallicchio and Micheli, 2011] C. Gallicchio and A. Micheli. Architectural and markovian factors of echo state networks. *Neural Netw.*, 24(5):440 – 456, 2011.

[Gallicchio *et al.*, 2011] C. Gallicchio, A. Micheli, S. Chessa, and P. Barsocchi. User movements forecasting by reservoir computing using signal streams produced by mote-class sensors. In *To Appear in the Proc. of the MOBILIGHT 2011 Conf., LNICST*. Springer, 2011.

[Gärtner, 2003] T. Gärtner. A survey of kernels for structured data. *SIGKDD Expl. Newsl.*, 5:49–58, 2003.

[Jaeger and Haas, 2004] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[Jaeger *et al.*, 2007] H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

[Jaeger, 2001] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical report, GMD, 2001.

[Kolen and Kremer, 2001] J.F. Kolen and S.C. Kremer, editors. *A Field Guide to Dynamical Recurrent Networks*. IEEE Press, 2001.

[Kushki *et al.*, 2007] A. Kushki, K.N. Plataniotis, and Anastasios N. A.N. Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Trans. Mobile Comp.*, 6(6):689 –705, jun. 2007.

[Lukosevicius and Jaeger, 2009] M. Lukosevicius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127 – 149, 2009.

[Tiño *et al.*, 2007] P. Tiño, B. Hammer, and M. Bodén. Markovian bias of neural-based architectures with feedback connections. In *Perspectives of Neural-Symbolic Integration*, pages 95–133. Springer-Verlag, 2007.

[Waegeman *et al.*, 2009] T. Waegeman, E. Antonelo, F. Wyffels, and B. Schrauwen. Modular reservoir computing networks for imitation learning of multiple robot behaviors. In *8th IEEE Int. Symp. on Comput. Intell. in Robotics and Autom.*, pages 27–32. IEEE, 2009.

[Zàruba *et al.*, 2007] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac. Indoor location tracking using RSSI readings from a single wi-fi access point. *Wireless Netw.*, (13):221235, 2007.

[3] `http://www.fp7rubicon.eu/`

A Variable Order Markov Model Approach for Mobility Prediction

Halgurt Bapierre, Georg Groh, Stefan Theiner
Fakultät für Informatik
Technische Universität München, Germany
halgurt@gmx.de, grohg@in.tum.de, stefan.theiner@mytum.de

Abstract

The contribution suggests a new approach for mobility prediction that avoids some of the drawbacks of existing approaches, whose characteristics and drawbacks are briefly reviewed. The new approach allows to integrate temporal and social context. The influence on improving prediction accuracy by integrating these contexts into the model is investigated. We use a variant of a Variable Order Markov Model incorporating spatial, temporal and social contexts which allows for improvements in accuracies compared to current state of the art approaches and alleviates critical drawbacks such as cold start and zero frequency problems. Furthermore it is easier to train and retrain. A heuristic density based clustering method is used to identify stay-points and hot-spots in a user's pattern of activity. We evaluate the overall approach using two large Open-Source data-sets of location traces.

1 Introduction

For various context-aware applications or for general insights into human behavior, it is interesting to acquire and model individual mobility patterns (see e.g. [Kim *et al.*, 2006]), allowing for various types of analysis and prediction tasks [González *et al.*, 2008]. Context-aware applications that can profit from mobility analysis and mobility prediction range from flexible, intelligent delivery of goods and provision of services to privacy preserving assistance for children, the elderly or disabled persons. Analysis of driver's mobility patterns can furthermore be used for environment protecting fuel management (see e.g. [Ericsson *et al.*, 2006] for a related approach). The advent of high-precision, personal location technologies such as GPS, broadly available in mobile devices such as smart-phones and navigation systems, allows a large share of the population to easily log their locations. Thus interest in human mobility models was further boosted because large precise data-sets of real mobility patterns become available [Phithakkitnukoon *et al.*, 2010][Pentland and Eagle, 2009][Pentland *et al.*, 2009]. Mobility models may be perceived as models of context, usable in context-aware applications such as the applications mentioned above. Spatial context (e.g. location) is certainly a key element of individual context [Tamminen *et al.*, 2004] but context is not limited to location [Schmidt *et al.*, 1999]. Besides representations of sequences of spatial context (e.g. location-measurements), mobility models may also need to incorporate representations of temporal context (e.g. local time or data on whether a day is a workday or a holiday) and resulting connections of temporal and spatial context. Context-elements to be considered may also encompass indications of social context ("who else is around?", "what are they doing?"), application contexts (e.g. contents of a personal or group calendar), or representations of activity semantics or short term interests etc. All of these context elements may influence each other.

The investigation of mobility models for the prediction of a user's future locations based on sequences of location measurements may thus be generalized to the investigation of models which predict vectors of future contexts on the basis of context histories. As our research question we specifically investigate models based on sequences of representations of a user's spatial, temporal and social context and their interrelations. It is our goal to show that the incorporation of temporal and social context can substantially improve the accuracy of prediction of the user's next location.

An efficient prediction model will have to mine for patterns of various combinations of spatial, temporal and social context sequences: Purely spatial patterns, that is patterns in sequences of past spatial contexts (locations), clearly influence the location prediction. In these cases the future location of the user only depends on sequences of previous locations (e.g. rules like "if in location A, the user will visit location B afterwards" may be derived). Purely temporal patterns such as "every evening the user goes home regardless of his previous and current locations" neglect spatial context altogether. Spatio-temporal patterns express more specific connections between temporal and spatial context, connecting temporal periodicities with locations ("on a workday morning, a visit of location C is always followed by a visit to location D and on a weekend C is followed by E"). The model will have to precisely specify and formalize concepts such as "visit", "location" etc.

Social contexts may also influence a location prediction e.g. as in "if with my friends, a stay in the park is usually followed by a visit in the pub, if with my children, it is usually followed by a visit to the candy shop". Social contexts may be derived from an analysis of co-locations. Long-term social network information, such as friendship relations in social networking platforms can also be regarded.

This contribution is structured as follows: First we will review the key concepts from related work. After that, we introduce our own methodology of incorporating and combing several contextual elements with the help of an adaptedion of a Variable Order Markov Model (VOMM) [Begleiter *et al.*, 2004], which is a variant of a context specific Bayesian network. We then discuss an extensive quantitative evaluation on two structurally different large datasets of location traces. Finally we discuss elements of future work.

2 Related Work

Spatial and temporal context and their influence on human motion and activities were subject of several independent studies using a variety of techniques (nth-order HMMs, Kalman filters, Conditional Random Fields etc. (see e.g. [Bui *et al.*, 2001; Brockmann *et al.*, 2006; Cheng *et al.*, 2003; Clarkson, 2003; González *et al.*, 2008; Hightower *et al.*, 2005; Kang *et al.*, 2005; Kim *et al.*, 2006; Krumm and Horvitz, 2006; Liao *et al.*, 2007a; Liu *et al.*, 2002])). We will briefly discuss two important studies as examples.

Ashbrook and Starner [Ashbrook and Starner, 2003] devised a model for individual user's mobility in the context of context-aware wearable computing. After investigating suitable thresholds, they classified a measured location as a "place" if the stay time of the user at that place was ≥ 10 min. They used adapted K-Means clustering with a cut-off radius to group places to "locations". Varying the cut-off radius, a hierarchy of locations and "sub-locations" can be generated. On sequences of such location-types, they trained a second order Markov Model. Unfortunately no exact figures of performance were reported. They also did not take temporal pattern types and other contexts into account which can be helpful in cases were the user enters previously unseen locations. Clearly, a fixed order Markov-model will also induce a certain level of inflexibility and for larger orders will require a large amount of training data to populate the large transitions matrix. Our own implementation of their approach revealed an average accuracy of 0.797 on one of our datasets (see below).

Eagle and Pentland [Pentland and Eagle, 2009] collected a large dataset of context-sequences of MIT subjects (mostly students) over a 9 month period. The location- information was recorded using cell tower ids. Long term social context was gathered in form of a sociomatrix of relations and background information via questionnaires. Via grouping the location data into days (patterns) and hours of day (contributing to pattern dimension) and a PCA-based analysis, the authors showed that the daily behavior of students represented by the sequence of locations they visited can be characterized by a few eigenvectors ("Eigenbehaviors"). Representing the first 12 hours of a previously unseen day-behavior in the eigenbasis, they could predict the remaining 12 hours with an averaged overall accuracy of roughly 0.79. Furthermore, they were able to analyze the behavior of social groupings of the users with the same method, using average daily behaviors of the group members as patterns. By comparing the characteristic behaviors of a group to a given average behavior of a person, they were able to predict aspects of the long-term social context (group affiliation). However, the study only distinguished six very coarse location states (work, home, else, no signal, off) and three groups of students. It remains to be seen, whether the same levels of accuracy can be accomplished if more fine grained contexts have to be modeled.

To our knowledge, none of the existing approaches allow for the explicit flexible inclusion of other context elements such as social context or deeper dependencies between context elements, seamless integration of online-learning and prediction while being conceptually simple and accurate on fine grained locations.

In the next sub-sections we will further review notions and related work in some of the key areas relevant for our approach and the subsequent discussion.

2.1 DBN Approaches for Prediction

In Dynamic Bayesian Networks (DBN) [Russell and Norvig, 2003], usually a Markov assumption restricts the order of conditional dependencies in the the transition model (defining conditional probabilities on internal state variables X_t) and the sensor model (defining conditional probabilities involving internal state variables X_t and evidence variables E_t). In mobility models, state variables may correspond to locations and evidence variables to location measurements. If (notationally assuming first order Markov condition) the sensor and transition models are assumed to be Gaussian ($P(X_t|X_{t-1}) \sim \mathcal{N}(AX_t, B)$, $P(E_t|X_t) \sim \mathcal{N}(CX_t, D)$ with suitable matrices A, B, C, D these models are called Gauss Markov Models and play an important role as predictive mobility models [Liang and Haas,]. One may set $X_t = (S_t, V_t)$, interpreting S_t as actual locations and V_t as speed and choose the model parameters A, B so, that S_t and V_t are linearly dependent plus an added Gaussian noise $P(X_t|X_{t-1}) \sim \mathcal{N}(S_{t-1} + \Delta V_{t-1}, B)$ where Δ is the time difference between the discrete time steps t and $t-1$. These (linear) Kalman-Filter-models may also be used for short term (seconds to a few minutes) mobility predictions if the assumptions are justified [Liao *et al.*, 2007b]. However, the assumption concerning Gaussian distributions and fixed Markov order are usually not applicable to human motion on intermediate timescales of several minutes to hours, since human trajectories show a high degree of spatial and temporal regularity [González *et al.*, 2008] which is not well captured by the aforementioned random models. Other studies also confirmed the highly non-random nature of human mobility [Song *et al.*, 2010]. Hidden Markov Models drop the Gaussian assumption and are restricted to a single, discrete state variable x_t and usually a fixed Markov order and allow for more flexible distributions to be introduced for mobility modeling [Krumm, 2003]. However, they usually require a large amount of training data and thus the inclusion of further context elements can be very problematic. If we, for example, aim at including temporal periodicities and social elements, each new context aspect and its values multiplicatively contribute to the size of the state space. Furthermore, the fixed order of a standard HMM also may not fit well with the complex dependencies in human mobility patterns. If nevertheless applied to mobility modeling, since exact inference in an unrolled DBN is difficult (see e.g. [Russell and Norvig, 2003]), methods of approximate inference have to be taken into account such as Particle Filtering possibly augmented with Rao-Blackwellization to control the number of samples [Krumm, 2003] [Gustafsson *et al.*, 2002].

2.2 Discovering Significant Places

Most of significant place detection approaches assume that a geographic location is only significant if the user spends at least some time above a certain threshold there [Ye *et al.*, 2009]. Unfortunately in practice there's no evidence for an ideal temporal threshold th_t that leads to the detection of all the significant places [Liao *et al.*, 2007a]. Furthermore, if precise GPS data are available, those fine grained significant locations need to be grouped together to meaningful hot-spots [Kang *et al.*, 2005] which requires a spatial threshold th_s. Kang et al. [Kang *et al.*, 2005] find $th_s \approx$ 40 m and $th_t \approx 300s$ useful and use a K-Means clustering variant for their experiments. However, since K-Means has a tendency to find spherically shaped clusters of same size, this approach might not ideally reflect the true hot-spot structure.

2.3 Connections between Social and Spatial Context

Pentland has shown [Pentland and Eagle, 2009][Pentland *et al.*, 2009] that the social and spatial contexts are closely related (e.g. via similar Eigenbehaviors w.r.t. location visits). Furthermore, several studies discussing the topology of social networks emphasize that social relatedness has an influence on spatial relatedness and vice versa. Routing in social networks often can be accomplished via distances [Liben-Nowell *et al.*, 2005], and motion analysis can reproduce social networks [González *et al.*, 2006]. The more location history two users share, the more socially correlated these users are [Zheng *et al.*, 2001]. We conclude that the incorporation of social context is likely to improve the accuracy of location prediction.

2.4 Periodic Pattern Extraction

A key element of temporal context relevant for mobility prediction are periodic patterns. As we will see, we use a heuristic a priori approach to the problem of detecting periodic patterns for simplicity reasons, which exploits the "naturally occurring" periodicities in western society and humankind in general which are induced by natural (year, month, day) and cultural periods (week) assuming their universal validity. Statistically orientated mining for periodicities e.g. find patterns like Eigenbehaviors [Pentland and Eagle, 2009] via Principal Component Analysis, other approaches use a Conditional Random Fields[Liao *et al.*, 2007a]. [Agrawal and Srikant, 2002] present an algorithm for mining patterns from spatio-temporal sequences. They use a sub-string tree structure to store all possible sub-patterns and provide a counter to each node of a tree, that indicates the frequency of this pattern. After evaluating the tree structure they perform a level-wise mining method to detect all frequent patterns. The presented method works very efficiently and scales well in time $O(m \log m)$ with m being the number of track-points. [Roddick and Spiliopoulou, 1999] provides a comprehensive overview of different algorithms for mining spatio-temporal patterns.

3 Our Approach

As discussed in 1, our approach for medium-term location prediction aims to allow for the incorporation of temporal and social contexts and their connections with spatial context especially exploiting regular patterns over various time-scales and should be able to easily deal with previously unseen locations. A key requirement is a better ability to deal with an amount of training data that is small compared to the number of locations involved e.g. in view of cold start and zero frequency problems that previous approaches exhibit.

For the detection of *stay-points* we first use a group-filtering of the raw GPS traces with a temporal threshold of 10 minutes yielding locations where the user stayed for at least 10 minutes. We consider clusters of these stay-points as a *hot-spot* for a user, if the spatial distance between stay-points is less than 10 m. Hot-spots furthermore contain at least 5 stay-points. We use density based clustering (DB-Scan [Zaiane and Lee, 2002]) instead of k-means clustering because it alleviates some of its disadvantages (see section 2.2). The heuristic choice of the mentioned parameters is motivated from previous work and behavioral analysis of own location traces over one year and is intended to inject independent common sense knowledge into the approach. Statistically deriving them from data-sets is difficult and may lead to an inappropriate level of fuzziness in the definition of hot-spots.

As has been mentioned before, using fixed order Markov models for location prediction causes some problems. An Nth-order Markov model using $|\Sigma|$ labeled states (with labels in Σ) will have a $|\Sigma|^N \times \Sigma$ transition matrix. Neglecting evidence, the model has to learn probabilities $p(q|s)$ where $q \in \Sigma$ and $s \in \Sigma^N$ and later use them for predicting the next state, given s. For location prediction Σ may correspond to the set of locations and s may be perceived as the spatial context. If we do not have a sufficiently large training-set, we will often encounter zero-frequency problems when training the model (sparsity problem). Furthermore, naively introducing new contexts such as temporal context will multiplicatively enlarge the state-set $\Sigma_{new} = \Sigma_{loc} \times \Sigma_{temp}$ and worsen the sparsity problem considerably. Another aspect is that a fixed order will not flexibly allow for regarding temporal regularities and periodicities on varying scales (e.g. monthly routine, weekly routine, daily routine).

Using a Variable Order Markov Model (VOMM) can contribute to solving these problems. A key application for VOMMs is lossless compression but they can also be used for prediction [Begleiter *et al.*, 2004]. Besides being structurally simpler than HMMs, VOMMs allow for learning and using variable length contexts an efficient way. We use an adaptation of Prediction by Partial matching (PPM), which is an instance of general VOMM that outperformed other instances in the prediction task in a comparative study [Begleiter *et al.*, 2004]. VOMM approaches generally use a tree structure to address the sparseness problem of the transition matrix. If the maximal order of the VOMM is N, the tree has a maximal depth $N+1$ and each path in it defines a subsequence of symbols appearing in the training sequence. Each node of the tree is labeled with a symbol q from the alphabet Σ and has a counter c for bookkeeping the number of occurrences of the context constructed through concatenating all the symbols from the root to that node 1. Each VOMM variant differs slightly in the way, for example, the zero-frequency problem is treated. If the current context is represented by s, the prediction algorithm works on that tree by simply predicting $\mathrm{argmax}_q p(q|s)$ and stepwise escaping to suffixes of s if no entries are represented in the tree for s. One of the main advantages is that training and prediction do not have to be separated. PPM defines an escape mechanism as defined in [Begleiter *et al.*, 2004]. For all symbols that did not appear after s yet, the escape mechanism assigns a probability mass $P(escape|s)$. The remaining mass $1 - P(escape|s)$ is distributed between the symbols appearing after context s. Equation 1 determines the probability of any symbol q occurring after context s recursively. If we define Σ_s to be the set of symbols that have already appeared after context s (and thus have been accounted for in the tree) we have

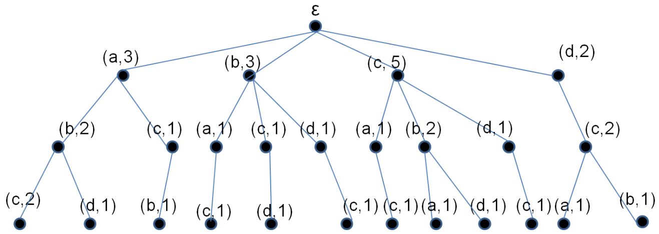


Figure 1: The prefix-tree (trie) constructed by PPM for a training sequence "abcdcacbdcbac". Node marks: $(q \in \Sigma, c)$

$$P(q|s) = \begin{cases} \tilde{P}(q|s) & \text{if } q \in \Sigma_s \\ \tilde{P}(escape|s)\ P(q|\mathrm{suf}(s)) & \text{else} \end{cases} \qquad (1)$$

where $\mathrm{suf}(s)$ denotes the longest suffix of s.

If s is empty, the probability of any symbol after an empty context is $P(q|\varepsilon) = \frac{1}{|\Sigma|}$. For symbol q and context s, let $C(sq)$ be the counter that counts the occurrences of sq. We then define

$$\tilde{P}(q|s) = \frac{C(sq)}{|\Sigma_s| + \Sigma_{q' \in \Sigma_s} C(sq')} \qquad (2)$$

$$\tilde{P}(escape|s) = 1 - \sum_{q \in \Sigma_s} \tilde{P}(q|s) = \frac{|\Sigma_s|}{|\Sigma_s| + \Sigma_{q' \in \Sigma_s} C(sq')} \qquad (3)$$

The escape mechanism is a technique to deal with the zero-frequency problem, which implies the Laplace estimator-like summand $|\Sigma_s|$ in the denominator of eq. (2) and (3) (Compare the Rule of Succession in general statistics (see e.g. [Zabell, 1989])). Σ_s is the set of all symbols appearing after s.

Thus training of a PPM VOMM tree effectively corresponds to instantiating or updating node counters in the tree. If the order is bounded by $n_0 = 6$ and a sequence $sq = ABC \ldots G$ appears in the training data all the counters along the path from G up to A and finally the root ϵ have to be incremented.

3.1 Inclusion of Temporal Context

For location prediction, the symbols correspond to locations. In order to include temporal periodicities as temporal contexts, we modify the tree and expand each spatial node with a sub-tree built from the temporal annotations determined from the timestamps of the occurrences of the context s corresponding to that node. Let $\lambda = (D_1, D_2, ..D_i)$ be the set of temporal features we want to use. In our evaluation we use three temporal features ($\lambda = (ts, wo, dw)$) where and ts is a partition of the day (e.g. Morning(mor), Afternoon(afn), Evening(evn)), wo indicates whether it is weekend (we) or working day (wd) and dw indicates the day of week ($mon, tue, \ldots$). For each symbol q appearing in the training sequence, we determine its temporal annotation $(\tau_1, \tau_2, ..., \tau_i)_q$ with $\forall j : \tau_j \in D_j$. A temporal annotation example would be (mor, we, sun). According to the temporal annotation we assign each node in the tree (corresponding to a symbol q and a context path s) a temporal-sub-tree. Each node in the temporal sub-tree has a label from λ and a counter indicating the occurrence of the sequence sq fitting the temporal feature indicated by the label. Figure 2 depicts an exemplary temporal sub-tree for a spatial node. In the temporal sub-tree the temporal features $(\tau_1, \tau_2, ..., \tau_i)$ are ordered such that the temporal feature τ_x on tree level x is conceptually more specific than the temporal feature τ_{x-1} on level $x-1$. Conceptual specifity can be implemented by an inclusion semantics which implies that temporal partitions on level x are sub-divided into temporal partitions at level $x+1$ (e.g. a day is sub-divided into 24 hours). It is possible to argue towards ts as the least specific temporal feature being determined by given general biological and physical rhythms of our existence and artificial features such as week of the year being more specific. It is also possible to argue in favor of the inverse hierarchy of specifity. The construction principle of VOMM and the escape mechanism used will ensure in both cases that the counters are correctly maintained and used during training and prediction. In our case we opted for the sequence of $wo \sqsubseteq dw \sqsubseteq ds$ given by the simple temporal inclusion semantics. As in the general case described above, each node is associated with a counter that stores the visits matching the particular temporal context. During the prediction phase we can descend the temporal sub-tree matching the current sub-tree and escape to more general temporal features whenever there is not enough evidence available for the more specific context. Besides those natural temporal patterns it is also possible to include more data-specific temporal patterns which sophisticated algorithms, mentioned in section 2.4, can discover or adapt to the user's cultural background (e.g. concerning which days are considered "weekend") When using individually learned temporal features the inclusion of social contexts described in the next section becomes more complicated. Hence we decided to use these static temporal features for all users. If, for example, a spatial sequence sq has temporal context (mor, wd, mon), we have to increment all

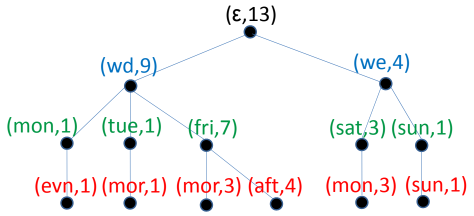


Figure 2: An example of a temporal context sub-tree

counters along this path in exactly the same way as in the spatial case described above. Via this step-wise temporal abstraction multi-bookkeeping we are able to use the standard PPM algorithm with its ability to consider various sequence lengths without modification. If, for a context s, we have not seen that particular (ts, wo, dw) temporal context we escape to the more general (wo, dw), if we have not seen this we escape to dw. If we have not seen this dw we escape to purely spatial relations.

In effect we use standard PPM except for inserting the temporal sub-tree after each node in the tree. In effect we use $\Sigma = \Sigma_{loc} \cup \Sigma_{temp}$ where Σ_{loc} are the location symbols and $\Sigma_{temp} = \cup_{j=1}^{i} D_j$ which in our case is $\Sigma_{temp} = \{mor, afn, evn, we, wd, mon, tue, \dots, sun\}$.

3.2 Inclusion of Social Context

We define socializable hotspots (SHs) as hotspots where at least 2 users were co-located at a given time. For a specific user we distinguish between those SHs where she participated herself (class I SHs) and those SHs, were "friends" of herself participated (class II SHs). "Friends" can be won from explicit declaration (as in Facebook) or by implicit mining (e.g. class I SHs can be an implicit measure for social relatedness). Individual hotspots can be treated as a special form of a SH, where the number of people visiting the place is equal to one. This notion allows us to incorporate SHs elegantly in the tree. In the tree containing spatial and temporal knowledge we annotate each node with counters for each set of persons that have been seen in the corresponding context. The presence of other users can be identified by Social Situation detection [Groh *et al.*, 2010] using device to device communication. This type of communication is also used to exchange spatio-temporal VOMM sub-trees between friends during encounters to determine the class II SHs. The exchange of the class II SHs allows to inject a vast amount of additional knowledge gathered by other users into the individual tree, the use of which for prediction is suitably mediated by comparing it to the current social context. Class II SHs furthermore demand that the hotspots of different users can be compared to detect matching hotspots. This can be achieved either by comparing the labels assigned by the user to his hotspots or by clustering hotspot within a small radius among all users. The elegance of this method lies in the simple incorporation of the additional social knowledge. Class I SHs are detected by the user herself. Whenever a friend is detected at the same place, the counters for the individual visits in the current spatio-temporal context and the counter for the set of present friends (including the user herself) are both incremented. Recently communicated class II SHs have to be matched against the paths in the exisiting spatio-temporal tree of the user. If the the path already exists, we can simply add another counter corresponding to the new class II social situation (set of friends) that have been learned from the friend's tree. If the path doesn't exist it is inserted together with the acquired counter values corresponding to SHs in the friend's tree. Hence inclusion of social context information only requires some adjustments in the prediction tree instead of a multiplicatively growing tree as observed in naïve Markov model implementations. We assume that for every point in time the social situation [Groh *et al.*, 2010] of a user is known.

If for a given spatio-temporal context of a user u_{i1} other users $u_{i2}, \dots, u_{im}$ are present and the tree for the given context contains counters for some previous visits (either individual hotspots, class I or class II SHs) the algorithm has to weigh which previously observed behavior is the most relevant for this situation. A weighted sum of all the counters $C(sq)^{\text{soc}}_{U'}$ of a node q in the temporal sub-tree is computed with a measure of overlap between U' and U as weight (we use Jaccard coefficient for simplicity reasons):

$$\hat{C}^{\text{soc}}_{U}(sq) = \sum_{U'} C(sq)^{\text{soc}}_{U'} * \text{Jacc}(U', U) \qquad (4)$$

Because the set of persons involved in social situations/socializable hotspots does not allow an interpretation of 'more specific' as in the temporal sub-tree or 'happened before' as in the spatial tree we cannot employ a corresponding sub-tree escape mechanism, instead the social context is respected via introducing the Jaccard-coefficients. The usage of the Jaccard coefficient provides various benefits. Whenever the user is alone the paths learned from his individual behavior has weight 1, while every other node has maximum weight of 0.5 (class I SHs with the user and another person).

4 Evaluation

We considered two datasets for evaluation: Dataset 1 is a dataset publicly available from Microsoft research (165 users over 2 years, GPS coordinates) [Zheng *et al.*, 2008][Zheng *et al.*, 2009]. Dataset 2 is the Reality Mining dataset [Pentland and Eagle, 2009] (97 persons over 9 months, cell-tower-based locations) described in section 2. Dataset 2 contains a social network between the actors and Bluetooth encounters between them which allow to at least determine class I SHs. Class II SHs are hard to determine because of user specific location labels, which cannot be easily resolved. We only regarded the named cell-towers as hot-spots, because we did not have the actual physical location of the cell-towers to perform our hot-spot identification preprocessing steps.

Using a 10-fold cross validation and maximum Markov order of 2, we compute the accuracy of our predictions on both datasets for each user with more than 5 locations. Weighted average accuracy is determined by weighting each user's prediction accuracy with the relative number of predictions. This is a pessimistic estimation, since for users with few locations, prediction accuracies are, on average, higher. On dataset 1 we achieve an average weighted accuracy of 0.6584 (± 0.194)

Order o	Fixed Order ($= o$)	Variable Order ($\leq o$)	Variable Order + temporal
2	0.797 ($\pm$ 0.106)	0.798 ($\pm$ 0.119)	0.819 ($\pm$ 0.107)
3	0.795 ($\pm$ 0.103)	0.797 ($\pm$ 0.126)	0.821 ($\pm$ 0.106)
4	0.782 ($\pm$ 0.114)	0.784 ($\pm$ 0.135)	0.819 ($\pm$ 0.110)
5	0.769 ($\pm$ 0.129)	0.775 ($\pm$ 0.148)	0.814 ($\pm$ 0.113)

Table 1: Weighted average accuracies: comparison between fixed order Markov models (Order=o) and VOMM(Order$\leq$ o): varying Markov order. (10 fold cross validation → 90 % of data used for training.)

Share for training	Fixed Order	Variable Order	Variable Order + temporal
50%	0.763 ($\pm$ 0.152)	0.797 ($\pm$ 0.131)	0.815 ($\pm$ 0.107)
33.3%	0.759 ($\pm$ 0.149)	0.793 ($\pm$ 0.133)	0.816 ($\pm$ 0.109)
25%	0.745 ($\pm$ 0.155)	0.792 ($\pm$ 0.133)	0.815 ($\pm$ 0.108)
10%	0.714 ($\pm$ 0.178)	0.786 ($\pm$ 0.134)	0.814 ($\pm$ 0.108)

Table 2: Weighted average accuracies: comparison between fixed order Markov models and VOMM: varying training set size. 'Share for training' specifies the share of the dataset that was used for training. The remaining data was used for testing. Markov order was 3 (for fixed order MM) and $\leq$ 3 (for VOMM) respectively.

without considering social and temporal context. Including temporal context, the accuracy is improved to 0.781 % ($\pm$0.151). Total number of predictions made is 7088. These numbers are computed pessimistically, because we leave out users with less than 5 hotspots, because with few hot-spots the accuracy is naturally very high. On dataset 2, purely spatial weighted average accuracy for maximum Markov order 2 was 0.798 ($\pm$0.119) and weighted average accuracy including temporal context was 0.819 ($\pm$0.107). Total number of predictions made is 264529.

In order to more thoroughly compare our approach to a fixed order Markov model as in [Ashbrook and Starner, 2003], we evaluated varying Markov order and the size of the training set on dataset 2. The results are shown in tables 1 and 2. Table 1 first of all shows that including temporal context substantially increases the prediction accuracy (see column 3) compared to the case where only spatial information is used (see column 2). Furthermore when comparing the purely spatial VOMM approach (column 2) to a fixed order Markov model (column 1) we note the following: table 1 shows that the VOMM approach is always slightly more accurate because of the corrected zero frequency problem. While our approach remains roughly stable when increasing the Markov order, the accuracy for the fixed case decreases with growing Markov order which can be attributed to the limited size of the training data. The effects of small sizes of the training data can be seen more clearly in table 2. Here, we vary the size of the training data from 50 percent to 10 percent. We see that the performance of the fixed order Markov model substantially decreases (and σ increases) while the results of the VOMM remain stable. Fixed order Markov models require a huge amount of training data in order to deliver good performance and thus suffer from cold start problems. This is one of the key advantages of our approach.

Figure 3 shows the relation between accuracy and the the number of different locations for each user with more than 5 locations. Shown are (1) Fixed order Markov model (spatial only, order 3), (2) VOMM (spatial only, max. Markov order: 3) and (3) VOMM (spatial + temporal, max. Markov order: 3). All figures are computed on dataset 2, training data size was 10 % (90 % were used for testing (10 fold cross validation)). The linear regression lines for each show a general tendency that more locations correspond to a decreased prediction accuracy as expected. Fixed order Markov Model performs worst while VOMM spatial and VOMM spatial + temporal are less sensitive of the number of locations, which is on of the key advantages of our approach.

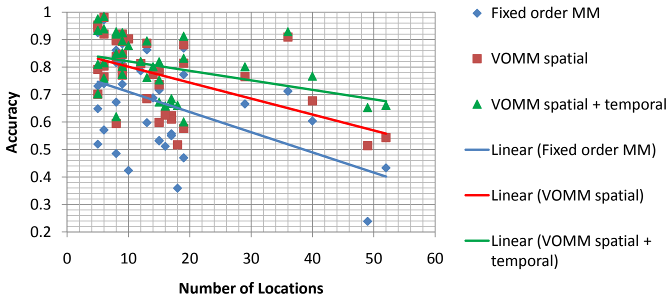


Figure 3: The relation between accuracy and number of locations. Shown: (1) Fixed order Markov model (spatial only, order 3), (2) VOMM (spatial only, max. Markov order: 3), (3) VOMM (spatial + temporal, max. Markov order: 3). All on dataset 2, training data size: 10 %. Also shown: linear regression lines for each. Pearson-correlations: $PC_{(1)} = -0.49, PC_{(2)} = -0.51, PC_{(3)} = -0.38$ and Spearman's $R_{(1)} = -0.43, R_{(2)} = -0.53, R_{(3)} = -0.46$

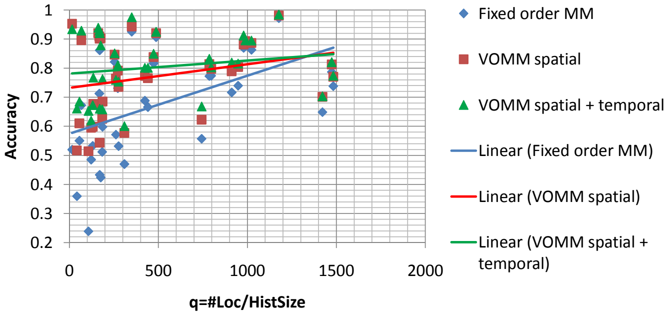


Figure 4: The relation between accuracy and q =history size/number of locations. Shown: same elements as in figure 3. Pearson-correlations: $PC_{(1)} = 0.48, PC_{(2)} = 0.27, PC_{(3)} = 0.18$ and Spearman's $R_{(1)} = 0.58, R_{(2)} = 0.25, R_{(3)} = 0.20$

Figure 4 shows the relation between accuracy and the quotient of history size and the number of different locations. Here we can also see a tendency that a larger history and smaller number of locations increases accuracy. Again, our approach outperforms fixed order Markov.

Figure 5 shows the relation between accuracy and the history size. This comparison shows the smallest correlation, which can be attributed to the fact that the model requires only comparatively few training examples to provide a reasonable accuracy. We see that fixed order Markov suffers from cold-

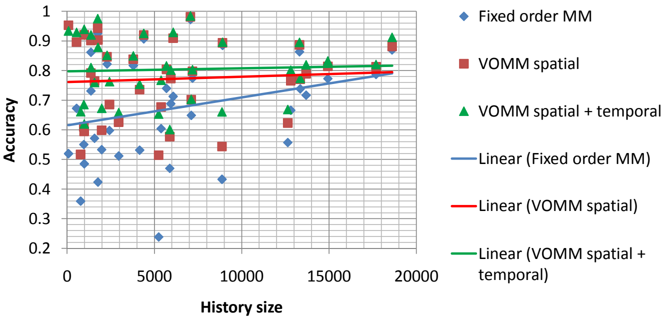


Figure 5: The relation between accuracy and history size. Shown: same elements as in figure 3. Pearson-correlations: $PC_{(1)} = 0.28, PC_{(2)} = 0.07, PC_{(3)} = 0.05$ and Spearman's $R_{(1)} = 0.29, R_{(2)} = -0.03, R_{(3)} = -0.02$.

start problems. Including more training data (larger history size) does not improve VOMM spatial + temporal very much on our datasets because users with a larger history size will have more different locations and also exhibit a higher behavioral mobility entropy.

Thus we investigated reasons for inadequate predictions in our approach by computing the behavioral mobility entropy $H = \sum_{q \in \Sigma} p(q) \log_2 p(q)$ of each user and correlating it with the per-user prediction accuracy via Pearson-correlation PC and Spearman's R. The results are shown in figure 6. On Dataset 1 we have $PC = -0.62$ and $R = -0.50$ and on dataset 2 we have $PC = -0.69$ and $R = -0.70$. The correlation coefficients indicate a negative linear correlation between entropy and accuracy: the prediction accuracy decreases with increasing entropy. Thus, as expected, users living a more 'regular' life are easier to predict.

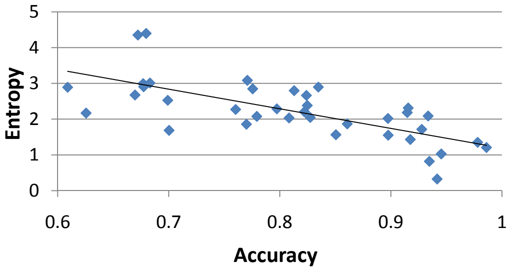


Figure 6: The relation between entropy and accuracy (dataset 2). Also shown: linear regression line

4.1 Including social context

As for social context inclusion, dataset 1 (although fairly large) unfortunately did not provide enough co-locations (contributing to class I SHs) for a statistically meaningful estimation of its benefit. An analogous problem exists with dataset 2 where not enough Bluetooth encounters involving named cell-towers to make a statistically significant estimation. However, for dataset 2 we were able to identify 2384 social situations where more at least two users were co-located. For these cases we were able to compare the performance of our approach without regarding the social context with the performance of our approach including the social context (we used Markov order 3 plus including temporal context). The study yielded an increase of 0.0069 when using the social context. Although this figure is not statistically significant because of the lack of social situations in our data we see that the inclusion of social contexts is tendentially beneficial.

5 Conclusion

We presented a novel approach to location prediction based on PPM VOMM. We were able to demonstrate that the inclusion of temporal context improves prediction accuracy considerably. The overall performance of the approach can be regarded as satisfactory. The inclusion of social context was able to improve prediction in several cases, but due to few usable training sequences containing social context the improvement may not be regarded as significant yet. Future work will focus on developing and evaluating finer grained versions of the social context inclusion and further investigating the suitability of the inclusion of other types of context information such as calendar contents etc. Furthermore we will investigate the effect of using more sophisticated per-user temporal periodicity detection methods (e.g. yielding other partitions of the time of the day as temporal patterns). In view of possible applications mentioned in section 1, the limits of our approach with respect to predicting further into the future (before reaching the inevitable limits of an equivalent of mixing time in DBNs ([Russell and Norvig, 2003]) will have to be investigated more thoroughly. As a last point it might also be interesting to investigate applicability of an adapted version of our approach to a smaller spatio-temporal scale e.g. for predicting indoor location changes.

References

[Agrawal and Srikant, 2002] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 11th IEEE Int'l Conf on Data Engineering*, pages 3–14, 2002.

[Ashbrook and Starner, 2003] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):286, 2003.

[Begleiter *et al.*, 2004] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22(1):385–421, 2004.

[Brockmann *et al.*, 2006] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[Bui *et al.*, 2001] H.H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. In *Hidden Markov models*, pages 177–196. World Scientific Publishing Co., Inc., 2001.

[Cheng *et al.*, 2003] C. Cheng, R. Jain, and E. van den Berg. Location prediction algorithms for mobile wireless systems. In *Wireless internet handbook*, pages 245–263. CRC Press, Inc., 2003.

[Clarkson, 2003] B.P. Clarkson. Life Patterns: structure from wearable sensors. *PhD thesis, MIT Media Lab (Pentland)*, 2003.

[Ericsson *et al.*, 2006] E. Ericsson, H. Larsson, and K. Brundell-Freij. Optimizing route choice for lowest fuel consumption-Potential effects of a new driver support tool. *Transportation Research Part C: Emerging Technologies*, 14(6):369–383, 2006.

[González *et al.*, 2006] M.C. González, P.G. Lind, and H.J. Herrmann. System of mobile agents to model social networks. *Physical review letters*, 96(8):88702, 2006.

[González *et al.*, 2008] M.C. González, C.A. Hidalgo, and A.L. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[Groh *et al.*, 2010] Georg Groh, Alexander Lehmann, Jonas Reimers, Rene Friess, and Loren Schwarz. Detecting social situations from interaction geometry. *Proc. IEEE SocialCom 2010, Minneapolis USA*, 2010.

[Gustafsson *et al.*, 2002] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, 2002.

[Hightower *et al.*, 2005] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. *UbiComp 2005: Ubiquitous Computing*, pages 159–176, 2005.

[Kang *et al.*, 2005] J.H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):58–68, 2005.

[Kim *et al.*, 2006] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proc. IEEE Infocom*, pages 1–13. Citeseer, 2006.

[Krumm and Horvitz, 2006] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. *UbiComp 2006: Ubiquitous Computing*, pages 243–260, 2006.

[Krumm, 2003] J. Krumm. Probabilistic inferencing for location. In *2003 Workshop on Location-Aware Computing*, pages 25–27. Citeseer, 2003.

[Liang and Haas,] B. Liang and Z.J. Haas. Predictive distance-based mobility management for PCS networks. In *Proc. INFOCOM'99*, volume 3, pages 1377–1384.

[Liao *et al.*, 2007a] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int'l J. of Robotics Research*, 26(1):119, 2007.

[Liao *et al.*, 2007b] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

[Liben-Nowell *et al.*, 2005] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *PNAS*, 102(33):11623, 2005.

[Liu *et al.*, 2002] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *IEEE J. on Selected Areas in Communications*, 16(6):922–936, 2002.

[Pentland and Eagle, 2009] A. Pentland and N. Eagle. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology 63:7*, 2009.

[Pentland *et al.*, 2009] A. Pentland, N. Eagle, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS Vol 106(36), pp.15274-15278*, 2009.

[Phithakkitnukoon *et al.*, 2010] S. Phithakkitnukoon, T. Horanont, G. Di Lorenzo, R. Shibasaki, and C. Ratti. Activity-aware map: Identifying human daily activity pattern using mobile phone data. *Human Behavior Understanding*, pages 14–25, 2010.

[Roddick and Spiliopoulou, 1999] J.F. Roddick and M. Spiliopoulou. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM SIGKDD Explorations Newsletter*, 1(1):34–38, 1999.

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

[Schmidt *et al.*, 1999] A. Schmidt, M. Beigl, and H.W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[Song *et al.*, 2010] C. Song, T. Koren, P. Wang, and A.L. Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 2010.

[Tamminen *et al.*, 2004] S. Tamminen, A. Oulasvirta, K. Toiskallio, and A. Kankainen. Understanding mobile contexts. *Personal and Ubiquitous Computing*, 8(2):135–143, 2004.

[Ye *et al.*, 2009] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Proc. IEEE MDM'09*, pages 1–10, 2009.

[Zabell, 1989] S.L. Zabell. The rule of succession. *Erkenntnis*, 31(2):283–321, 1989.

[Zaiane and Lee, 2002] O.R. Zaiane and C.H. Lee. Clustering spatial data when facing physical constraints. In *Proc. IEEE ICDM02*, pages 737–740, 2002.

[Zheng *et al.*, 2001] Y. Zheng, L. Zhang, and X. Xie. Recommending friends and locations based on individual location history. In *Proc. Conf. Advances in GIS*, volume 247, page 256, 2001.

[Zheng *et al.*, 2008] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.Y. Ma. Understanding mobility based on GPS data. In *Proc. 10th Int'l. ACM Conf. on Ubiquitous Computing*, pages 312–321, 2008.

[Zheng *et al.*, 2009] Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. ACM WWW09*, pages 791–800, 2009.

Fusion of Audio and Temporal Multimodal Data by Spreading Activation for Dweller Localisation in a Smart Home*

Pedro Chahuara, François Portet, Michel Vacher
LIG UMR 5217, UJF-Grenoble 1 / Grenoble INP / UPMF-Grenoble 2 / CNRS,
Laboratoire d'Informatique de Grenoble
Grenoble, F-38041, FRANCE
{pedro.chahuara,francois.portet,michel.vacher}@imag.fr

Abstract

In this paper, an approach to locate a person using non visual sensors in a smart home is presented. The information extracted from these sensors gives uncertain evidence about the location of a person. To improve robustness of location, audio information (used for voice command) is fused with classical domotic sensor data using a two-level dynamic network and using an adapted spreading activation method that considers the temporal dimension to deal with evidence that expire. The automatic location was tested within two different smart homes using data from experiments involving 25 participants. The preliminary results show that an accuracy of 90% can be reached using several uncertain sources.

1 Introduction

The objective of this work is the continuous location of an inhabitant in their home using sources that are non-visual (i.e., without camera) and indirect (the person does not wear a sensor). It is part of the Sweet-Home (`http://sweet-home.imag.fr/`) project which aims to design an intelligent controller for home automation through a voice interface for improved comfort and security. In this vision, users can utter vocal orders from anywhere in their house, thanks to microphones set into the ceiling. It is thus particularly suited to assist people with disabilities and the growing number of elderly people in living autonomously as long as possible in their own home. Within the smart home domain, this concept is known as *Aging-In-Place* [Marek and Rantz, 2000] and consists in allowing seniors to keep control of their environment and activities, to increase their autonomy, well-being and their feeling of dignity.

Among the main data processing tasks a smart home must implement, detecting the correct location of the person plays a crucial role to make appropriate decisions in many applications (e.g., home automation orders, heating and light control, dialogue systems, robot assistants) and particularly for health and security oriented ones (e.g., distress call, fall, activity monitoring). For instance, in the Sweet-Home context, if the person says "turn on the light", the location of the person and the lamp which is referred to must be deduced. However, in the context of a vocal command application, noise, reverberation and distant speech can alter the recognition quality and lead to incorrect inferences [Vacher *et al.*, 2011]. To improve the robustness of the automatic location, we propose to combine audio source with other sources of information.

Automatic location becomes particularly challenging when privacy issues prevent the systematic use of video cameras and worn sensors. In the Sweet-Home project, only classical home automation and audio sensors are taken into account. These sensors —Presence Infra-red Detector (PID), door contacts, and microphones— only inform indirectly and transiently about the location of a person. Automatic location is thus a challenging task that must deal with indirect, uncertain and transient multiple sources of information.

In this paper, we present a new method developed for automatic dweller location from non-visual sensors. After a brief state of the art of location techniques in Section 2, the approach we adopted to locate a person is presented in Section 3. It is based on a fusion of information obtained from various sensors (events) through a dynamic network that takes into account the previous activations and the uncertainty of the events. The adaptation of the method to two smart homes is described in Section 4 and the results of the experiments are summarised in Section 5. The paper ends with a brief discussion of the results and gives a future work outlook.

2 Location of an Inhabitant: Common Techniques in Smart Homes

Techniques for locating people in a pervasive environment can be divided into two categories: those that use sensors explicitly dedicated to this task and worn by people such as a GPS bracelet, and those that use sensors that only inform implicitly about the presence of a person in a confined space such as infrared presence sensors or video cameras.

The wearable sensors are often used in situations where the person has a social activity (museum visit) or for professional, health or safety reasons (e.g., patients with Alzheimer's disease running away). Despite their very good location performance, they are not adapted to an informal and comfortable

*This work is part of the Sweet-Home project founded by the French National Research Agency (Agence Nationale de la Recherche / ANR-09—VERS-011)

in-home use. Indeed, these sensors can be awkward and annoying and, except for passive sensors (e.g., RFID), they require the systematic checking of the batteries. Moreover, if the goal is to improve the daily living comfort, the constraint of a wearable sensor may be a strong intrusion into the intimate life. That is why this paper focuses on techniques using environmental sensors (video, sound, motion sensor, door contacts, etc..).

Video analysis is a very interesting modality for home automation which is used in many projects [Marek and Rantz, 2000; Moncrieff *et al.*, 2007]. However, video processing requires high computational resources and can be unreliable and lacking in robustness. Moreover, installing video cameras in a home may be perceived as too much intrusion into intimate life, depending on the kind of video processing that is installed (e.g., plain vs. silhouette based video processing or hiding [Moncrieff *et al.*, 2007]).

Another usual source of localization can be derived from household appliances and surveillance equipment. For instance, infrared sensors designed for automatic lighting were used to evaluate the position and the activity of the person [Le Bellego *et al.*, 2006; Wren and Tapia, 2006]. The use of some devices can also be detected using new techniques that identify the signatures of an electrical appliance on the household electric power supply [Berenguer *et al.*, 2008].

Another interesting modality in home automation is the analysis of the audio channel, which, in addition to providing a voice command, can bring various audio information such as broken glass, slamming doors, etc. [Vacher *et al.*, 2010]. By its omnidirectional or directional nature, the microphone is a promising sensor for locating events with a high sensitivity or high specificity. There is an emerging trend to use such modality in pervasive environment [Bian *et al.*, 2005; Moncrieff *et al.*, 2007; Vacher *et al.*, 2010]. Audio sources require far less bandwidth than video information and can easily be used to detect some activities (e.g., conversations, telephone ringing). However, if the video is sensitive to changes in brightness, the audio channel is sensitive to environmental noise. The audio channel, while a relevant and affordable modality is therefore a noisy source and sometimes highly ambiguous.

Throughout this state of the art, it appears that no source taken alone makes a robust and resource-efficient location possible. It is therefore important to establish a location method that would benefit from the redundancies and complementarities of the selected sources. There is a large literature in the domain of activity recognition on such methods mainly using probabilistic graph-based methods such as Bayesian networks [Dalal *et al.*, 2005; Wren and Tapia, 2006] or Markov models [Wren and Tapia, 2006; Chua *et al.*, 2009]. However, given the large number of sensors, building HMM models taking all the transition states into account for really time processing would be extremely costly. More flexible graph-based approaches based on sensor network that include a hierarchy of processing levels (Bayesian and HMM classifiers) were proposed [Wren and Tapia, 2006]. However, if temporal order is often taken into account, the temporal information about duration or absolute date is rarely considered in these models .

Recently, Niessen et al. [Niessen *et al.*, 2008] proposed to apply dynamic networks to the recognition of sound events. In their two-level network, the input level is composed of sound events, the first level represents the assumptions related to an event (e.g., ball bounce or hand clap), and the second level is the context of the event (e.g., basketball game, concert, play). Each event activates assumptions according to the input event and the contexts to which these assumptions are linked. These assumptions then activate the contexts, reenforcing them or not. Thus, there is a bidirectional relationship between contexts and assumptions. For instance, if several previously recognized sounds are linked to a concert context, the next sound will be more likely related to a concert context. The method imposes no pattern but the notion of time is explicitly taken into account by a time constant that reduces the importance of an event according with its age. Given the flexibility provided by this approach, we chose to adapt it to the location of a person in a flat using multisource information.

3 Location of an Inhabitant by Dynamic Networks and Spreading Activation

The method developed for locating a person from multiple sources is based on the modelling of the links between observations and location assumptions by a two-level dynamic network. After a brief introduction to dynamic networks and spreading activation, the method adapted to work with multiple temporal sources is described.

Dynamic Networks and Spreading Activation

The spreading activation model, is employed in AI as a processing framework for semantic or associative networks and is particularly popular in the information retrieval community [Crestani, 1997]. Briefly, the considered network $\mathcal{N}$ is a graph where nodes represent concepts and where arcs, usually weighted and directed, represent relationships between concepts. The activation process starts by putting some 'activation weight' at an input node that spreads to the neighbouring nodes according to the strength of their relationships and then spreads to the other neighbours and so on until a termination criteria is satisfied. The activation weight of a node is a function of the weighted sum of the inputs from the directly connected nodes. For detailed introduction to spreading activation, the reader is referred to [Crestani, 1997]. Within this model, dynamic network have also bee proposed to represent knowledge that evolves with time [Niessen *et al.*, 2008]. A network is dynamic in the sense that it changes according to inputs that can modify the structure and/or the strength of the relationships between nodes. The spreading activation in a dynamic network provides a flexible and intuitive framework to represent associations between concepts which is particularly interesting to fuse evidence that decay. However, to the best of our knowledge, few approaches have focused on the case of temporal sources whose activation decreases with time [Niessen *et al.*, 2008]. In the following, we present the temporal multisource approach to locate a person in an flat.

Temporal Dynamic Networks for Multisource Fusion

The dynamic network that we designed is organized in two levels: the first level corresponds to location hypotheses generated from an event; and the second level represents the occupation context for each room whose weight of activation indicates the most likely location given the previous events. Location hypotheses correspond to area where the person can be at a specific time while occupation contexts correspond to rooms in which the person is over time. Our approach uses the following definitions:

Definition 1 (Observation) *An observation o_n is a data structure generated when a sensor reacts to the event e_n at time $t_n \in \mathbf{R}^+$ with $n \in \mathbf{N}$. Each observation is related to a sensor $o.sensor$ and has a sensor type $o.type$.*

Definition 2 (Simultaneous observations) *Two observations o_n^i and o_k^j are simultaneous if $t_k \in [t_n - d, t_n + d]$, with $d \in \mathbf{R}^+$ a predefined delay.*

Definition 3 (Observation activation) *The activation $A_n^o \in [0, 1]$ of an observation o_n represents the intensity of the evidence being integrated into the network. It can be derived from the weight or probability of the classifier/detector generating the observation. In our case, A_n^o is based on its ambiguity such that for a set of simultaneous observations of same type O_n, $\sum_{o \in O_n} A_n^o = 1$.*

Definition 4 (Location hypothesis) *$h_n^i \in L$, where $L = \{Loc_1, \dots, Loc_R\}$ is the hypothesis that the inhabitant is at location i at time t_n. These hypotheses are created only from the observations at time t_n.*

Definition 5 (Occupation context) *$c^i \in R$ where $R = \{Room_1, \dots, Room_S\}$ is the occupation context of the i^{th} room.*

Definition 6 (Relationship weight) *$w \in [0, 1]$ is the importance of the relationship between two nodes in the network. w_{o,h^i} is the weight between an observation and the i^{th} hypothesis whereas w_{h^i,c^j} is the weight between the i^{th} hypothesis and the j^{th} context.*

Definition 7 (Decay function) *The decay function $f(t_n, t_{n-1}) = e^{-\frac{\Delta t}{\tau}}$, with $\Delta_t = t_n - t_{n-1}$ represents the decrease of the context through time. It makes it possible to keep a short-term memory about contexts.*

The dynamic network evolution is summarised by the following algorithm:

1. for every new observation o_n^k, a new node is created;
2. thereupon hypothesis nodes h_n^i are created and connected to o_n^k with weights w_{o^k,h^i};
3. hypothesis nodes h_n^i are connected to occupation context nodes c^j with weights w_{h^i,c^j};
4. activation spreads from o_n^k to h_n^i and the activation of each h_n^i is calculated;
5. activation spreads from h_n^i to c^j and the activation of each c^j is recalculated;
6. the node c^j with the highest activation becomes the present location;

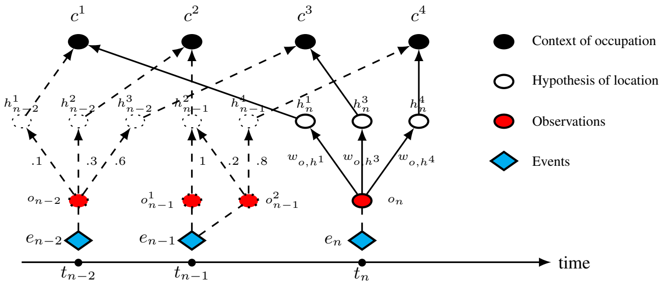


Figure 1: Example of Dynamic Network

7. all the nodes h_n^i and o_n^k are deleted from the network.

An example of dynamic network is shown in Figure 1. At time t_{n-2}, the event e_{n-2} is detected by a sensor which generates the observation o_{n-2} from which 3 hypotheses are derived : h_{n-2}^1 with a relationship weight of 0.1 towards the context c^1, h_{n-2}^2, with weight 0.3 towards c^2 and h_{n-2}^3 with 0.6 towards c^3. If no previous events occurred, then c^3 would be the most probable location. At time t_{n-1}, two simultaneous observations caused by event e_{n-1} are integrated within the network. Every node created previously (i.e., at t_{n-2}) is discarded, except the contexts which are always kept. Active contexts are weighted by $f(t_{n-1}, t_{n-2})$ and the activation of the hypothesis h_{n-1}^2 is added to c^2 and h_{n-1}^4 is added to c^4. The method is applied subsequently at time t_n.

Spreading Activation

The activation of a node is typically defined [Crestani, 1997] by the formula $n_i(t) = \sum_{i \neq j} w_{i,j} \times A^j(t)$ where $w_{i,j}$ is the weight, j corresponds to a neighbour of i and $A^j(t)$ is the activation of its neighbours at time t. A node that has been activated by a neighbour node cannot spread its activation back to it. In our case, activations are always triggered by an observation o with a bottom-up spreading. Once the accumulated activation from neighbours $n(t)$ is obtained, the output activation of the node is calculated. It differs according to the node level. For location hypotheses, the activation $A_n^{h^i} \in [0, 1]$ is computed using Formula 1:

$$A_n^{h^i} = n_i(t_n) = \sum_{o \in O_{t_n}} w_{o,h^i} \, A_n^o \tag{1}$$

Regarding the occupation contexts, the output activation results from the previous activation weighted by the decay function and the accumulated activation of hypotheses. Equation 2 describes the activation of an occupation context A^{c^i} as a consequence of an external activation at time t_n.

$$A^{c^i}(t_n) = M \times A_n^{h^i} + e^{-\frac{\Delta t}{\tau}} A^{c^i}(t_n - \Delta t) \times [1 - A_n^{h^i}] \tag{2}$$

where $A^{c^i}(t_n - \Delta t)$ is the previous activation, $M = 1$ is the maximal activation and $e^{-\frac{\Delta t}{\tau}}$ is the decay function. Therefore, if no event appears during $5 \cdot \tau$ seconds, the contexts activation can be considered zero. The introduction of M constrains the activation value between 0 and 1.

Computation of the node level Relationship
Given that the network is composed of two layers, two types of relationship exist: *Observation-Hypothesis* and *Hypothesis-Context*. The links between the different layers depend strongly on the application and the environment considered.

The **Hypothesis-Context** relationship is in our case of type one-to-one because a hypothesis of location is only related to a unique room. It is an experimental choice since some hypotheses about rooms loosely separated could activate several occupation contexts. Thus $w_{h^i,c^j} = 1\,\forall i = j,\ 0$ otherwise.

The **Observation-Hypothesis** relationship is unidirectional and of type one-to-many. Weights and hypotheses vary depending on the observations and prior knowledge about this relationship. In order to include this prior knowledge in the network, the relationship weight is defined by formula 3 in the form of probabilities where the relationship weight between the current (possibly set of simultaneous) observation(s) O_n and hypothesis h^i is defined by the probability of observing the inhabitant at location i given the current observation(s) and the context $\mathcal{C}$.

$$w_{o,h^i}(t_n) = P(loc = i \mid O_n, \mathcal{C}) \qquad (3)$$

4 Adaptation of the Method to Pervasive Environments

Two pervasive environments were considered in our study: the DOMUS smart home and the Health Smart Home(HIS) of the Faculty of Medicine of Grenoble. Every experiment in these smart homes considered only one inhabitant at a time. In section 4.1 details of both corpora are given, then sections 4.2 and 4.3 explain how relationships between layers are computed for each smart home and which *a priori* information is taken into account to derive the inhabitant's location.

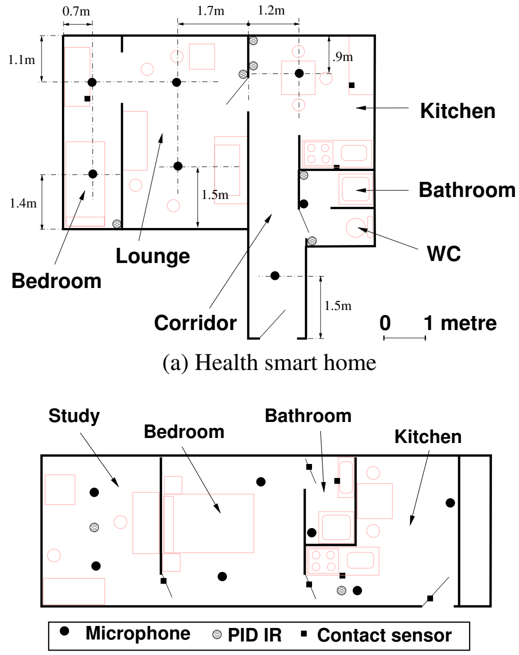
4.1 Pervasive Environments and Data Used

The HIS corpus was acquired during experiments [Fleury *et al.*, 2010] aiming at assessing the automatic recognition of Activities of Daily Living (ADL) of a person at home in order to automatically detect loss of autonomy. Figure 2a describes the 6-room Health Smart Home of the Faculty of Medicine of Grenoble at the TIMC-IMAG laboratory [Le Bellego *et al.*, 2006]. The data considered in this study consisted of about 14 hours of 15 people recordings using the following sensors:

- 7 microphones (Mic) set in the ceiling;
- 3 contact sensors on the furniture doors (DC) (cupboards in the kitchen, fridge and dresser in the bedroom);
- 6 Presence Infrared Detectors (PID) set on the walls at about 2 metres in height.

The Sweet-Home corpus was acquired in realistic conditions, using the DOMUS smart home. This smart home was designed and set up by the Multicom team of the Laboratory of Informatics of Grenoble. Figure 2b shows the details of the flat. The data considered in this study consisted of about 12 hours of 10 people recordings performing daily activities using the following sensors:

- 7 microphones (Mic) set in the ceiling;



(a) Health smart home

(b) DOMUS Smart Home

Figure 2: Layout of the smart homes used and position of the sensors.

- 3 contact sensors on the furniture doors (DC) (cupboards in the kitchen, fridge and bathroom cabinet);
- 4 contact sensors on the 4 indoor doors (IDC);
- 4 contact sensors on the 6 windows (open/close);
- 2 Presence Infrared Detectors (PID) set on the ceiling.

4.2 Weight computation for the HIS

$w_{o,h^i}(t_n)$, was computed differently for each kind of sensors. For observation o with $o.type \in \{DC, PID\}$ a single hypothesis node is created with a weight $w_{o,h} = 1$. Indeed, spatial informations about PID an DC are unambiguous and certain. For example, opening the fridge can only occurs if the inhabitant is in the fridge area. For both PID an DC, the activation of observations is $A^o_n = 1$.

The microphones cannot be treated in the same manner. Microphones can detect theoretically all the acoustic waves produced in the home making this information highly ambiguous. However, it is possible to estimate from the position of the Mic the areas they can most likely sense and thus the location they are most related to. To take this into account, formula 3 was approximated with $w_{o,h^i}(t_n) = P(loc = i|Mic = j)$ that is the probability that the inhabitant is in the i^{th} room given an observation o_n generated by the j^{th} microphone. To acquire this *a priori* knowledge, two approaches were tested: a naïve approach and a statistical one. Succinctly, for the naive approach, the circle outside which the loss of energy is greater than 75% is considered for each Mic. The weight is calculated as the surface of the intersection between the circle and the rooms with a penalty of 2 when the circle goes beyond a wall. The statistical approach acquired

$P(Loc|Mic)$ estimated by the naïve approach

Mic	1	2	3	4	5	6	7
bedroom	.14	.07	.70	.85			
lounge	.86	.93	.27	.14	.01	.13	.03
kitchen			.03	.02	.10	.87	.50
bathroom					.06		.18
wc					.06		.18
corridor					.77		.10

$P(Loc|Mic)$ estimated from corpus

Mic	1	2	3	4	5	6	7
bedroom	.28	.29	.42	.43	.25	.18	.20
lounge	.59	.56	.47	.41	.07	.07	.09
kitchen	.05	.08	.06	.09	.45	.63	.37
bathroom	.06	.05	.04	.04	.09	.04	.10
wc	.01	.01	.01	.01	.12	.05	.21
corridor		.02	.01	.02	.03	.03	.02

Table 1: Estimation of $P(Loc|Mic)$ for HIS smart home

the probabilities from the annotated corpus. Table 1 shows the weights obtained for both approaches. Apart from this static information, dynamic information, such as the signal energy, was taken into account in case of simultaneous observations on the Mic. The Mic activation, it was computed using the signal-to-noise ratio (SNR) estimated in real-time. Given that A_n^o summarises the observation ambiguity, it was computed as $A_n^o = o.snr / \sum_{obs \in O_n} obs.snr$ where O_n is the set of simultaneous observation at time n.

The fusion of prior and dynamic information provides a better disambiguation. For example, for the HIS flat if two simultaneous observations are detected by the microphones in the kitchen and bathroom with a similar SNR of $12dB$, the formula 1, and the prior information from the naive estimation give as activation $A^{h^{kitchen}} = .87 \times A^{o^6} + .5 \times A^{o^7} = .69$ which is higher than the bathroom activation $A^{h^{bath}} = .09$ even when the SNR is similar.

4.3 Weight computation for Sweet-Home

$w_{o,h^i}(t_n)$ was computed in the same way as for the HIS for DC and PID. However, conversely to contact sensors on furniture and windows which are always linked to a unique hypothesis, contact sensors on the indoor doors (IDC) can be ambiguous regarding the location. The problem is to decide which of the two rooms around the door should have the highest weight. In that case, formula 3 was approximated with the conditional probability $w_{o,h^i} = P(loc = i \mid o.sensor, o.state, \mathcal{C})$, where $o.state \in \{Open/Close\}$ and $\mathcal{C}$ is the inhabitant's location at time t_{n-1}. This *a priori* knowledge was statistically acquired from an annotated corpus different from the one used in the test.

Results of the conditional probabilities estimation indicate that in 97% of cases when a door is open from a room then a transition to the contiguous room is produced, whereas when the door is closed the transition is less certain (66% of cases).

5 Experimentation

For each participant's record, the events from DC, PID and Mic were used to activate a dynamic network to estimate the location of the inhabitant. Location performance was evaluated every second by comparing the context of the highest weight to the ground truth. If they matched, then it was a true positive (TP), otherwise it was a confusion. The accuracy was given by $Acc = nb(TP)/nb(test)$ where $nb(test)$ corresponds to the duration of the record in seconds and $nb(TP)$ the number of seconds in which a TP was obtained.

Sensor and prior information	PID	DC	Mic+ DC	PID+ DC	PID+ Mic	PID+ Mic+DC
SH no prior info.	62.9	59.9	63.7	71.6	64.5	**73.2**
SH prior info.	62.8	73.3	77.4	81.7	64.6	**84.0**
HIS no prior info.	88.9	26.5	32.8	**89.4**	87.7	88.2
HIS prior naive info.	88.9	26.5	34.1	89.4	89.0	**89.5**
HIS prior stat. info.	88.9	26.5	34.8	89.4	89.7	**90.1**

Table 2: Accuracy with several combinations of sources

For Sweet-Home a first experiment was done without using the prior probabilistic knowledge about room doors contact sensors, afterwards the method was executed using these probabilities to evaluate how significant their contribution is. Likewise, three independent experiments were carried out with the HIS corpus: without a priori ($P(loc|Mic) = 1$ when the microphone was in the room, 0 otherwise), with the naïve approach and with the statistical approach. Table 2 shows the results of both corpora using several combinations of sensor.

In the case of Sweet-Home, it is clear that the fusion of information improved the accuracy since it rises as more sensors information is combined. Even when the precision of infrared sensors was good, the overall results of the method using only these sensors was low (63%) as only two of them were set in the 4-room flat. This led to a poor sensitivity. In the second row, the accuracy using the information of door contact on room doors is reported. It can be noticed that the learned probabilities had a significant impact on the performance. Here, the column DC includes also the results with IDC. In every case, *a priori* information about IDC had a positive impact on the performances, confirming that prior knowledge can be used to improve the performance.

From HIS experiement, it can be noticed that, in some cases, fusion of information did not improve the accuracy. The door contact information slightly improved the accuracy compared to that obtained only with the infrared sensors. On the other hand, adding the sound information decreased the performance (88.2 % versus 89.4 %). One reason for this may the high level of confusion between sound and speech of the AUDITHIS system which reached 25% of classification errors. Nevertheless, once again an increase of performance was achieved by means of the prior knowledge introduction: results are better or similar in every combination of sensors when using the probabilities. There is a slight advantage of statistical approach over the naïve one but the naïve approach does not require any dataset to be acquired and thus simplifies the set up in new pervasive environements.

6 Discussion and Perspectives

The results showed that the information fusion by spreading activation is of interest even when the sources have very good accuracy. It is the case for infrared sensors (but with imperfect sensitivity) and for door contact sensors. The use of less certain localisation sources, such as speech recognition, can then improve performance in many cases. Another important finding is that *a priori* knowledge about sensors is a possible leverage to gain a higher accuracy as it was done with the contact sensors of the room doors for the Sweet-Home corpus and the microphones for the HIS corpus. In those cases, the introduced knowledge was expressed in terms of conditional probabilities and its exploitation was demonstrated to be useful. Furthermore, the approach is general enough to include different kinds of ambiguous sensor as input to the dynamic network. Given the source, the probability of the inhabitant being in a room given some feature of the sensor data can be estimated and this prior knowledge can be applied to enhance localisation. This is the case, for instance, of the water meter. Even if this information cannot be directly used for localisation, it is feasible to estimate the probability about the inhabitant's location given the change of flow rate in order to use this probability when generating hypotheses in the dynamic network.

Several ways to improve this method can be followed. So far, besides direct information given by sensors, we have applied some knowledge based on specific sensors features. However, it would be advantageous to use other characteristics of the environment. One way could be to use the topology of the flat as in [Wren and Tapia, 2006]. For instance, an occupant can not move from the bedroom to the front door without going through the lounge, etc. Further extensions of our method include Markovian techniques to estimating the probability of the present inhabitant's location given their precedent location. We believe that it could be a relevant contribution when fusioned with the sources of information already described in this work. The next step is to apply this method to classify the sounds of everyday life using the location context to disambiguate the sound classification, and to test the general suitability of the approach by confronting the system to actual users (elderly and frail people).

References

[Berenguer *et al.*, 2008] M. Berenguer, M. Giordani, F. Giraud-By, and N. Noury. Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line. In *HealthCom'08, 10th IEEE Int. Conf. on e-Health Networking, Applications & Service*, 2008.

[Bian *et al.*, 2005] Xuehai Bian, Gregory D. Abowd, and James M. Rehg. Using sound source localization in a home environment. In *Third International Conference of Pervasive Computing*, pages 19–36, 2005.

[Chua *et al.*, 2009] Sook-Ling Chua, Stephen Marsland, and Hans W. Guesgen. Spatio-temporal and context reasoning in smart homes. In *Proceedings of the COSIT Workshop on Spatial and Temporal Reasoning for Ambient Intelligence Systems*, 2009.

[Crestani, 1997] F Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

[Dalal *et al.*, 2005] S Dalal, M Alwan, R Seifrafi, S Kell, and D Brown. A rule-based approach to the analysis of elders activity data: Detection of health and possible emergency conditions. In *AAAI Fall 2005 Symposium*, 2005.

[Fleury *et al.*, 2010] Anthony Fleury, Michel Vacher, François Portet, Pedro Chahuara, and Norbert Noury. A multimodal corpus recorded in a health smart home. In *LREC Workshop Multimodal Corpora and Evaluation*, pages 99–105, Matla, 2010.

[Le Bellego *et al.*, 2006] G. Le Bellego, N. Noury, G. Virone, M. Mousseau, and J. Demongeot. A model for the measurement of patient activity in a hospital suite. *IEEE Transactions on Information Technologies in Biomedicine*, 10(1):92 – 99, 2006.

[Marek and Rantz, 2000] KD Marek and MJ Rantz. Aging in place: a new model for long-term care. *Nursing Administration Quarterly*, 24(3):1–11, 2000.

[Moncrieff *et al.*, 2007] Simon Moncrieff, Svetha Venkatesh, and Geoff A. W. West. Dynamic privacy in a smart house environment. In *IEEE Multimedia and Expo*, pages 2034–2037, 2007.

[Niessen *et al.*, 2008] Maria E. Niessen, Leendert van Maanen, and Tjeerd C. Andringa. Disambiguating sounds through context. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 88–95. IEEE Computer Society, 2008.

[Vacher *et al.*, 2010] Michel Vacher, Anthony Fleury, François Portet, Jean-François Serignat, and Norbert Noury. *Complete Sound and Speech Recognition System for Health Smart Homes: Application to the Recognition of Activities of Daily Living*, pages 645 – 673. Intech Book, 2010.

[Vacher *et al.*, 2011] Michel Vacher, François Portet, Anthony Fleury, and Norbert Noury. Development of audio sensing technology for ambient assisted living: Applications and challenges. *International Journal of E-Health and Medical Communications*, 2(1):35–54, 2011.

[Wren and Tapia, 2006] Christopher R. Wren and Emmanuel Munguia Tapia. Toward scalable activity recognition for sensor networks. In *Location- and context-awareness*, 2006.

Semi-supervised Learning for Adaptation of Human Activity Recognition Classifier to the User

Božidara Cvetković, Mitja Luštrek, Boštjan Kaluža, Matjaž Gams
Jožef Stefan Institute, Department of Intelligent Systems
Jamova cesta 39, Ljubljana, Slovenia
{boza.cvetkovic, mitja.lustrek, bostjan.kaluza, matjaz.gams}@ijs.si

Abstract

The success of many ambient intelligence applications depends on accurate prediction of human activities. Since posture and movement characteristics are unique for each individual person, the adaptation of activity recognition is essential. This paper presents a method for on-line adaptation of activity recognition using semi-supervised learning. The method uses a generic classifier trained on five people to recognize general characteristics of all activities and a user-specific classifier briefly trained on the user using a reduced number of activities. The final decision on which classification to use for a given instance is done by a meta-classifier trained to decide which of the classifiers is more suitable for the classification. An instance classified with a sufficient confidence is added into the training set of the generic classifier. Experimental results show that the activity recognition accuracy increases by up to 11 percentage points with the proposed method. In comparison with Self-training proposed method performs better for up to five percentage points.

1 Introduction

Ambient intelligence (AmI) applications aim to provide relevant response to the human presence and have been widely researched and used in a variety of fields such as healthcare, eldercare, ambient assisted living, security, etc. Applications focused on user monitoring can benefit from efficient recognition of the activity in many ways. When the recognition is reliable the system can accurately detect deviations in the user's behavior, provide proper assistance and support in everyday life as well as adjust the environment and application to the user's habits, etc.

The most commonly used approach in activity recognition is supervised machine learning [Lester *et al.*, 2006]. Applications based on this approach are usually deployed with a generic classifier trained on the data collected in the laboratory environment and not on the behavior of the new end-user. In most cases once the system is trained and deployed it does not change anymore. The accuracy of activity recognition is thus affected by the difference in physical characteristics between the end-user and the people used in training. Consequently, the accuracy on real-life end-users with different characteristics may be substantially lower than in laboratory tests. Some approaches improve the activity recognition by using spatio-temporal information [Wu *et al.*, 2010].

The method we propose is trying to overcome the gap between end-users and the people used in training. This is achieved by employing two additional classifiers along with the generic classifier trained on general characteristics of the activities. The user-specific classifier is briefly trained during the initialization procedure on user specifics and the meta-classifier is trained to designate which of the activity recognition classifiers will label an instance. If the classification confidence value surpasses a specified threshold, the instance is added into the training set of the generic classifier. This method was deployed and validated in the project Confidence [2011], which uses a real-time localization system based on Ultra-wideband (UWB) technology with four wearable tags. The experimental results show that the activity recognition accuracy increases for up to 11 percentage points with the proposed method and in comparison with Self-training it performs better for up to 5 percentage points.

The paper is structured as follows. The related work on semi-supervised learning and adaptation of activity recognition is reviewed in Section 2. Section 3 introduces our experimental domain; Section 4 presents the proposed semi-supervised method and specifics of the learning procedures. In Section 5 we present the experimental results including method validation and comparison. Finally, Section 6 concludes the paper.

2 Related Work

Semi-supervised learning is a technique in machine learning that can use both labeled and unlabeled data. It is gaining popularity because the technology makes it increasingly easy to generate large datasets, whereas labeling still requires human effort, which is very expensive. The approach where the human annotator is required when the classifier is less confident in labeling is called Active learning [Settles, 2009]. Since in our case the human interaction is undesirable the Active learning approach is inappropriate, therefore we will focus on other semi-supervised learning techniques.

There are two categories of semi-supervised learning [Zhu, 2005]: single-classifier that use only one classifier and multi-classifier that use multiple classifiers, which can be split into multi-view and single-view approach. Key characteristic of a multi-view method is to utilize more feature independent classifiers on one classification problem. Single-view methods use classifiers with the same feature vector but differentiate considering the algorithm used for learning. We will review the techniques that relate to our proposed method.

The most common method that uses a single classifier is called Self-training. After an unlabeled instance is classified, the classifier returns a confidence in its own prediction, namely the class probability. If the class probability threshold is reached the instance is added to its training set and the classifier is retrained. The Self-training method has been successfully used on several domains such as handwriting word recognition [Frinken and Bunke, 2009], natural language processing [Guzmán-Cabrera *et al.*, 2008], protein-coding gene recognition [Guo and Zhang, 2006], etc.

Self-training was also applied to activity recognition by Bicocchi et al. [2008]. The initial activity recognition classifier was trained on the acceleration data and afterwards used to label the data from a video camera. The classified instances from the camera were added into the feature vector of the initial classifier and used for further activity recognition. This method can be used only if the initial classifier achieves high accuracy, since errors in confident predictions can decrease the classifier's accuracy.

Co-training [Blum and Mitchell, 1998] is a multi-view method with two independent classifiers. To achieve independence, the attributes are split into two feature subspaces, one for each classifier. The classifier that surpasses a confidence threshold for a given instance can classify the instance. The instance is afterwards added to the training set of the classifier that did not surpass the confidence threshold.

Democratic Co-learning [Zhou and Goldman, 2004] is a single-view technique with multiple classifiers. All the classifiers have the same set of attributes and are trained on the same labeled data with different algorithms. When an unlabeled instance enters the system, all the classifiers return their class prediction. The final prediction is based on the weighed majority vote among *n* learners. If the voting results returned 95% confidence or more the instance is added into the training set of all classifiers.

The modified multi-view Co-training algorithm called En-Co-training [Guan *et al.*, 2007] was used in the domain of activity recognition. The method uses information from 40 sensors, 20 sensors on each leg to identify the posture. The multi-view approach was changed into single-view by using all data for training three classifiers with the same feature vector and different learning algorithm which is similar to previously mentioned democratic Co-learning. The final decision on the classification is done by majority voting among three classifiers and the classified instance is added into the training set for all classifiers. This method improves the activity recognition; however the number of sensors is to high for unobtrusive system.

The method we propose is a single-view approach with two classifiers. Both are trained with the same algorithm but on different data. We use a third classifier to make the final prediction.

3 Confidence: A Brief Overview

The Confidence is an intelligent system for remote eldercare. The main objective is to detect deviations in short-term and long-term behavior of the end-user. There are currently three prototypes of the system in the verification phase in multiple European countries.

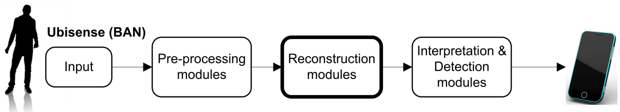


Figure 1: Simplified structure of the Confidence system. The method described in this paper is implemented as one of the reconstruction modules.

The simplified structure of the system is shown in Figure 1. The inputs to the system are the coordinates of four tags worn by the user. The coordinates are provided by the UWB real-time localization system Ubisense [Ubisense, 2010]. The user has a tag attached to the chest, waist and both ankles. The stated accuracy is approximately 15 cm but in practice larger deviations were observed.

The received data is sent to the pre-processing, where all four position coordinates are assembled into the current state in time denoted as snapshot. Each snapshot is processed by three filters. First, a median filter is applied, which eliminates large short-term changes in tag locations due to noise. Second, a filter that enforces anatomic constraints is used. This filter corrects errors such as an apparent lengthening of a limb. Third, the Kalman filter is applied, which smoothes sharp changes in both locations and speed.

The attributes for the recognition classifier are calculated from the filtered values. The attributes are the distances between the tags, velocity of the sensors and raw coordinates. For detailed explanation of the attributes the reader is referred to [Luštrek and Kaluža, 2009] where the authors used up to twelve tags to find appropriate attributes. The majority of the attributes computed by this module are for activity recognition by machine learning. The goal of activity recognition is to accurately identify the following eight human postures: lying, standing, sitting, falling, sitting on the ground, on all fours, going down and standing up.

The recognized activities serve as one of the inputs for the interpretation and detection modules focused on determining possible short-term or long-term behavior deviations [Luštrek *et al.*, 2009], that may indicate a health problem. Additional inputs are the characteristics of the user's movement, such as the speed of movement and various gait properties, and the user's location in the room with respect to the furniture (bed, chair). When a short-term deviation is detected, an alarm is raised and the output module issues a call for immediate help. When a long-term deviation is detected, a warning is sent describing the deviation, which may help a medical professional to determine whether it is a sign of an emerging disease.

Misclassification of the activity can result in a false positive alarm and in the worst case even in a false negative alarm, which can directly jeopardize the end-user's wellbeing. This shows that it is essential to accurately classify the activities in order to avoid such hazardous situations.

The main reason for misclassification of the activity, if we discard the noise, is the difference in the physical characteristics among users. The generic classifier employed in the system is trained on data of isolated set of people and does not contain the specific characteristics of the end-user. To overcome this problem we apply the method presented in the next section, which enables the system to learn the specifics of the user with semi-supervised learning.

4 The Adaptation Method

The Confidence system as well as other AmI systems that continuously monitor a user produce a large amount of unlabeled data for a particular end-user. These data are usually discarded, but they can be used to adapt the activity recognition classifier to the particular user.

We propose a method that adapts a system equipped with a generic classifier for activity recognition to a particular end-user.

The method consists of two steps:

- Initialization step
- On-line learning step

The initialization step is executed only once when the system is introduced to the end-user for the first time. During this process short labeled recordings of a subset of activities are made and used for training the user-specific classifier. The on-line learning operates in a non-supervised fashion, where both the user-specific and the generic classifiers are utilized for activity recognition. Activities classified with a sufficient confidence are used as additional training data for the generic classifier, which over time becomes adapted to the end-user. User-specific classifier is never retrained.

4.1 Initialization Step

The initialization step is performed only once at the beginning to introduce a new user to the system.

During this step the user is briefly recorded while performing basic activities that are defined in the recognition repertoire, namely standing, lying and sitting, since they are easy to perform. The transition activities such as falling, going down, standing up, sitting on the ground and on all fours are non-basic activities, since they are either uncomfortable to perform or very hard to label. The user is asked to perform each basic activity for a certain amount of time, in our case 60 seconds. During the recording procedure the captured data is labeled and used for the initial training of the new user-specific classifier.

The initialization step also involves modification of the generic classifier. The attributes related to the user's height are scaled by multiplying the value with the quotient of the user's height and the average height of the people used for generic classifier training. After the normalization the generic classifier is retrained.

The initialization step results in a new user-specific classifier and a modified generic classifier, both involved in the next step.

4.2 On-line Learning

The on-line learning step starts after the initialization and is performed until the stopping criterion is met, that is when the generic classifier is chosen to label most of the new istances.

The flow chart of the algorithm is presented in Figure 2. An unclassified instance is separately classified by two classifiers, the generic classifier and the user-specific classifier. Each of them returns the class distribution for the current instance. The meta-classifier decides which of the activity recognition classifiers is more likely to predict the class correctly. If the probability for the class returned by the chosen classifier surpasses a threshold, the instance is added to the training set of the generic classifier. In our case the threshold is 100%. After a period of time the generic classifier containing additional instances is retrained and thus adapted to the characteristics of the user. In our case we retrained the classifier every five minutes.

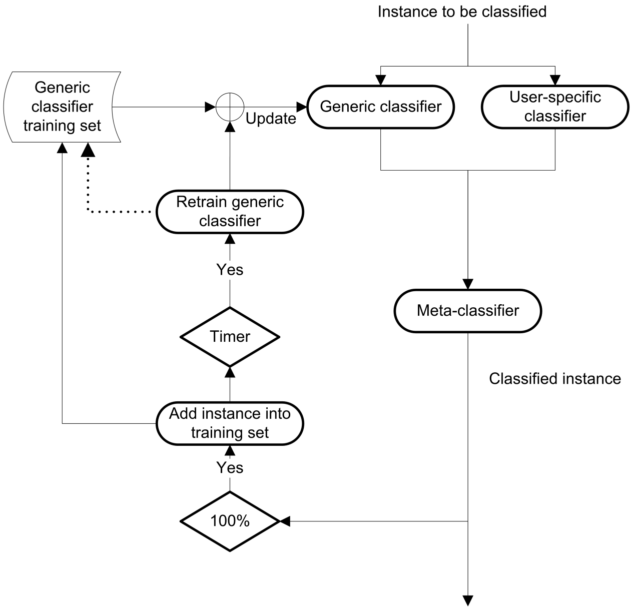

Instance to be classified
Generic classifier training set
Update
Generic classifier
User-specific classifier
Retrain generic classifier
Yes
Meta-classifier
Timer
Classified instance
Add instance into training set
Yes
100%

Figure 2: A work-flow of the on-line adaptation method.

To achieve a degree of balance between the classes and to add weight to the non-basic class instances added to the training set of the adapted generic classifier, the basic class instances are added only once, whereas the non-basic instances are added in triplicate. Adding only one or two instances resulted in slower learning. The reason for adding instances into the generic classifier and not the user-specific classifier is that the latter one is not equipped to handle all known activities, only those on which it was trained during the initialization step.

4.3 The Classifiers

The **Generic classifier** was build from the data we contributed to UCI Machine Learning Repository, under the title Localization Data for Person Activity, which was also used by Kaluža et al. [2010]. This dataset contains recordings of five

ML Algorithm	Attribute combination and accuracy %				
	Snapshot + Set 1	Set 1	Set 3	Set 1 + Set 3	**Set 1 + Set 2**
SVM	86.6	92.9	88.9	87.8	88.3
C4.5	96.8	95.4	96.1	96.6	95.9
Random Forest	90.9	95.9	96.6	96.9	**97.4**
Naive Bayes	61.0	75.7	70.1	68.8	82.3
AdaBoost	88.6	84.8	84.6	84.6	79.0
Bagging	96.9	94.7	95.8	96.2	95.8

Table 1: Attribute and machine learning algorithm combinations tested with the Meta-classifier.

people performing a scenario composed of eight activities: lying, standing, sitting, going down, standing up, sitting on the ground, on all fours and falling. The output of the generic classifier is the probability distribution over the classes corresponding to the eight activities given by Equation 1.

$$Pr_G = [Pr_G(C_1), \ldots, Pr_G(C_8)] \tag{1}$$

For validation of this classifier we used leave-one-person-out approach, where a classifier is built using the data of four persons and tested on the data of the fifth person. The classifier was trained using the Random Forest algorithm [Breiman, 2001] with attributes as described in Section 3. For the improvement of this classifier, we used the height of the end-user to scale the values of the height-related attributes. The scaled attributes are only the distances between the tags regarding the z-coordinate, since other attributes do not reflect the height. The measured accuracy was 86%.

The **User-specific classifier** is trained on the data recorded during the initialization procedure. Each posture is recorded for 60 seconds and given the sampling rate of 10 Hz we get approximately 1200 instances for the classifier training. This classifier was trained with the Random Forest algorithm. The feature vector is the same as in the generic classifier. The user-specific classifier is not able to recognize all activities. In our case it is trained to recognize basic activities: lying, standing and sitting; it has no knowledge about other activities. The output is the probability distribution over the eight classes given by Equation 2, where the unknown classes have zero probability, i.e. sitting on the ground, falling, on all fours, going down and standing up.

$$Pr_U = [Pr_U(C_1), \ldots, Pr_U(C_8)] \tag{2}$$

The **Meta-classifier** is used to determine the final activity of the current instance. It is trained before the system is deployed and is not adapted to the end-user. We compared the accuracy using several possible attribute sets for the meta-classifier. The results of the sets with best results are shown in the Table 1, where snapshot presents a current state of four tags.

The attributes in set 1 are represented by Equations from 3 to 8.

$$C_G = argmax_{i=1\ldots8}(Pr_G(C_i)) \tag{3}$$

$$C_U = argmax_{i=1\ldots8}(Pr_U(C_i)) \tag{4}$$

$$P_{GC_G} = Pr_G(C_G) \tag{5}$$

$$P_{UC_U} = Pr_U(C_U) \tag{6}$$

$$B_{CLASS} = \begin{cases} 1, & \text{if } C_i \in \{\text{standing, sitting, lying}\} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$EqualC = \begin{cases} 1, & \text{if } C_G = C_U \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The C_G and C_U represent the classification of the Generic and User-specific classifier, which are the classes with the highest probabilities in the class distribution. These probabilities are represented by P_{GC_G} and P_{UC_U}. The binary attribute B_{CLASS} tells whether the classification returned by the classifier selected by meta-classifier is a basic activity. The attribute represented by $EqualC$ tells whether the generic and user-specific classifier returned the same class.

Set 2 contains only the two attributes represented by Equations 9 and 10: the probability for the class selected by the user-specific classifier as computed by the generic classifier P_{GC_U} and the probability for the class selected by the generic classifier as computed by the user-specific classifier P_{UC_G}.

$$P_{GC_U} = Pr_G(C_U) \tag{9}$$

$$P_{UC_G} = Pr_U(C_G) \tag{10}$$

The attributes in set 3 are the z-coordinates of all tags, the distance between the chest and ankles and the distance between the chest and waist. Experiments showed that the distances in set 3 are not person-independent. Since the meta-classifier is not adapted to the end-user, these attributes had to be omitted.

The training of the meta-classifier was done on 60 minutes of labeled data of a person not used for further experiments. The data was collected from the recordings of a person performing a sequence of activities defined by the scenario. Each instance from the recording was passed over to the generic and user-specific classifier for classification. The class of the meta-classifier was defined according to the true class of the input instance and the relation to the prediction of the activity classifiers. We have tested all sensible combinations of the sets and the results five with the best results are shown in Table 1. The results show that the highest accuracy was achieved using attributes from the sets 1 and 2 and the Random Forest algorithm.

Activity Class	Person 1		Person 2		Person 3		Person 4	
	Start	End	Start	End	Start	End	Start	End
Lying	81.6	87.8	96.8	98.4	75.2	75.7	94.3	98.0
Standing	95.5	98.5	92.8	98.6	96.2	98.8	89.3	99.4
Sitting	35.9	80.1	88.7	99.1	52.2	76.5	75.0	97.7
Going down	52.0	52.9	42.7	54.6	51.8	55.4	16.7	12.8
Standing up	56.7	57.5	57.8	58.4	42.6	43.0	44.3	50.5
Sitting on the ground	28.8	63.4	22.0	40.2	83.3	86.6	46.5	36.1
On all fours	100	77.8	20.0	24.0	82.6	84.8	38.5	42.3
Falling	3.6	18.7	42.0	46.0	14.3	24.3	1.0	2.1
Overall	73.0	84.1	76.8	83.4	76.4	82.0	77.1	83.1

Table 2: The results of the on-line semi-supervised learning on four people. The results show the accuracies for each class and the overall accuracy (%) before the normalization and after the adaptation.

	Person			
	1	2	3	4
Difference in height (cm)	-18	-16	-5	+12
Starting accuracy (%)	73.0	76.8	76.4	77.1
Accuracy after normalization (%)	79.9	77.1	79.0	77.2
Accuracy after on-line adaptation (%)	84.1	83.4	82.0	83.1

Table 3: The difference in height per person according to the average height of the generic classifier, accuracy of the generic classifier before the adaptation process, increase in accuracy after normalization and the accuracy of the generic classifier after on-line adaptaion.

5 Experimental results

The method was integrated as one of the reconstruction modules in the Confidence system and was run on four different people with different physical characteristics. For the test set, every person performed the same sequence of activities defined in a scenario. The scenario captured typical daily activities during entire day, as well as some falls. A part of the scenario that represents the morning is for example lying in the bed, waking up, walking to the bathroom, sitting in the bathroom and falling in the bathroom. Each continuous sequence of the scenario lasted approximately 20 minutes and was repeated by the same person five times. Four of the recordings of each person were used for on-line learning and the final one to test the accuracy of the adapted classifier.

The experimental procedure was as follows: the system was initialized for the specific user (1 minute each basic activity), the user-specific classifier was trained and the generic classifier was normalized to the user's height. We learned in preliminary experiments that the scaling of all attributes for all instances can lead to higher noise for the activities taking place close to the ground. The misclassification happens because the lying activity is often classified as other activities where the z-coordinates of the chest and the waist are relatively close, for example on all fours. To avoid these types of misclassification we omitted the normalization of the lying instances. Attributes that are representing the distances between tags were selected for the normalization.

After the initialization process the on-line learning was started. The algorithm was run on four 20-minute recordings for each tested person and the accuracy of the adapted generic classifier was calculated every five minutes. The accuracy evaluation was done on the fifth recording of the person that was not used in the on-line learning procedure. The analysis of the progress of the adaptation process has shown that in the beginning all the instances added to the training set belonged to a basic class. During the fourth recording the majority of instances belonged to a non-basic class. In the beginning of the on-line learning the superior knowledge of the user-specific classifier was exploited to teach the generic classifier about the basic classes' specifics for the current user. As a consequence, later in the process generic classifier was more confident in the classification of the non-basic activities.

The results of the adapted generic classifier after the last processed recording are shown in Table 2. The table presents the accuracies of each class and the overall accuracy of the generic classifier before normalization and after the stopping criteria of the on-line learning was reached. The stopping criterion was reached in case the generic classifier classified all instances in the last 10 minutes.

The improvement of the generic classifier accuracy after normalization can be seen in Table 3. The table presents the difference in height regarding the average height of the people used in generic classifier, accuracy of the generic classifier before the process of adaptation started, accuracy of the generic classifier after normalization and accuracy of the adapted generic classifier. In the case of Persons 2 and 4 we see that normalization does not improve the generic classifier much and with the proposed method we can gain more than 5 percentage points of accuracy as seen in Table 2.

The proposed method was compared with the well known

Method	Gain in accuracy per person (pp)			
	1	2	3	4
Self-training	+8.63	+1.14	+2.08	+3.29
Proposed method	+11.10	+6.60	+5.60	+6.00
Difference	+2.47	+5.46	+3.52	+2.70

Table 4: The comparison between Self-training and our proposed method.

method for semi-supervised learning called Self-training. The results are presented in Table 4. We can observer that Self-training did increase the accuracy of the generic classifier, however our proposed method outperformed the Self-training by at least 2.47 percentage points and in best case by up to 5.46 percentage points.

6 Conclusion

This paper describes a method for on-line semi-supervised learning. The method uses generic, specific and meta-classifier. It was validated on the adaptation of the activity recognition. We showed that because of the difference in physical characteristics among the people, this method can be used to select informative instances in real-time and re-train the generic classifier to adapt it to a specific user. If we omit the gain in accuracy by simple height normalization, we can still show an increase in accuracy of 5 percentage points. The method was compared with Self-training method and the results showed that our proposed method outperformed it by 3.5% on average.

In thefuture the method should be compared with other known methods for semi-supervised learning and additionally verified on more people. To improve the method we will introduce a measure to balance the classes, since some of them have considerably more instances than others. For long-term use of the method it would be necessary to introduce aging of data. Finally, since this method has proven successful on our activity recognition domain, it should be tested on other domains as well.

Acknowledgments

The research leading to these results has received funding from the European Community's Framework Programme FP7/2007-2013 under grant agreement no 214986.

References

[Bicocchi *et al.*, 2008] Nicola Bicocchi, Marco Mamei, Andrea Prati, Rita Cucchiara, and Franco Zambonelli. Pervasive self-learning with multi-modal distributed sensors. In *Proceedings of the 2nd IEEE IWSOS*, pages 61–66, Washington, DC, USA, 2008. IEEE Computer Society.

[Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th COLT*, pages 92–100, New York, NY, USA, 1998. ACM.

[Breiman, 2001] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, October 2001.

[Confidence, 2011] Project Confidence. http://www.confidence-eu.org/, 2011.

[Frinken and Bunke, 2009] Volkmar Frinken and Horst Bunke. Self-training strategies for handwriting word recognition. In *Proceedings of the 9th ICDM*, pages 291–300, Berlin, Heidelberg, 2009. Springer-Verlag.

[Guan *et al.*, 2007] Donghai Guan, Weiwei Yuan, Young-Koo Lee, Andrey Gavrilov, and Sungyoung Lee. Activity recognition based on semi-supervised learning. In *Proceedings of the 13th IEEE RTCSA*, pages 469–475, Washington, DC, USA, 2007. IEEE Computer Society.

[Guo and Zhang, 2006] Feng-Biao Guo and Chun-Ting Zhang. Zcurvev: a new self-training system for recognizing protein-coding genes in viral and phage genomes. *BMC Bioinformatics*, 7(1):9, 2006.

[Guzmán-Cabrera *et al.*, 2008] Rafael Guzmán-Cabrera, Manuel Montes-Y-Gómez, Paolo Rosso, and Luis Villaseñor Pineda. A web-based self-training approach for authorship attribution. In *Proceedings of the 6th GoTAL*, pages 160–168, Berlin, Heidelberg, 2008. Springer-Verlag.

[Kaluža *et al.*, 2010] Boštjan Kaluža, Violeta Mirchevska, Erik Dovgan, Mitja Luštrek, and Matjaž Gams. An agent-based approach to care in independent living. In *AmI*, volume 6439 of *LNCS*, pages 177–186, Berlin, Heidelberg, 2010. Springer-Verlag.

[Lester *et al.*, 2006] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Pervasive Computing*, volume 3968, pages 1–16, Berlin, Heidelberg, 2006. Springer-Verlag.

[Luštrek and Kaluža, 2009] Mitja Luštrek and Boštjan Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):197–204, 2009.

[Luštrek *et al.*, 2009] Mitja Luštrek, Boštjan Kaluža, Erik Dovgan, Bogdan Pogorelc, and Matjaž Gams. Behavior analysis based on coordinates of body tags. In *AmI '09 Proceedings of the ECAmI*, pages 14–23, Berlin, Heidelberg, 2009. Springer-Verlag.

[Settles, 2009] Burr Settles. Active learning literature survey. CS Technical Report 1648, University of Wisconsin–Madison, 2009.

[Ubisense, 2010] Ubisense. http://www.ubisense.net, 2010.

[Wu *et al.*, 2010] Chen Wu, Amir Hossein Khalili, and Hamid Aghajan. Multiview activity recognition in smart homes with spatio-temporal features. In *Proceedings of the 4th ACM/IEEE ICDSC*, pages 142–149, New York, NY, USA, 2010. ACM.

[Zhou and Goldman, 2004] Yan Zhou and Sally Goldman. Democratic co-learning. In *Proceedings of the 16th IEEE ICTAI*, pages 594–202, Washington, DC, USA, 2004. IEEE Computer Society.

[Zhu, 2005] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, CS, University of Wisconsin-Madison, 2005.

Beat-based gesture recognition for non-secure, far-range, or obscured perception scenarios[*]

Graylin Trevor Jay
Department of Computer Science
Brown University
tjay@cs.brown.edu

Patrick Beeson
TRACLabs Inc.
Houston, TX
pbeeson@traclabs.com

Odest Chadwicke Jenkins
Department of Computer Science
Brown University
cjenkins@cs.brown.edu

Abstract

Gesture recognition is an important communication modality for a variety of human-robot applications, including mobile robotics and ambient intelligence domains. Most gesture recognition systems focus on estimating the position of the arm with respect to the torso of a tracked human. As an alternative, we present a novel approach to gesture recognition that focuses on reliable detection of time-dependent, cyclic "beats" given by a human user. While the expressiveness of "beat-based" gestures is limited, beat-based gesture recognition has several benefits, including reliable 2D gesture detection at far ranges, gesture detection anywhere in the image frame, detection when the human is mostly hidden or obscured, and secure detection via randomly rotated beat patterns that are known only by the user and the perception system. In addition to discussing this complimentary approach to gesture recognition, we also overview a preliminary implementation of beat-based gestures, and demonstrate some initial successes.

1 Introduction

Gestures form the basis of most non-verbal human communication. Thus, reliable gesture recognition is an important communication modality for a variety of human-robot applications [Kojo *et al.*, 2006; Jenkins *et al.*, 2007; Waldherr *et al.*, 2000], including mobile robotics and ambient intelligence domains. Gesture recognition can be used alone, or in conjunction with speech [Nicolescu and Mataric, 2003; Rybski *et al.*, 2007], to communicate spatial information, deliver commands, or update an intelligent observer on the status of the human. Pose-based gesture recognition techniques that estimate the position and orientation of the arms with respect to the torso have recently received a great deal of attention, especially depth-based efforts like Microsoft's Kinect and other infrared systems [Knoop *et al.*, 2006a].

Much of the research in both 2D and 3D gesture recognition [Dalal *et al.*, 2006; Knoop *et al.*, 2006b; Sminchisescu and Telea, 2002] (including our own previous work [Loper *et al.*, 2009]) has utilized a familiar perception precessing sequence: 1) segment the human from the background, 2) estimate the pose of the limbs and torso, 3) classify the configuration as a particular gesture (or no gesture). For 2D perception systems, the full object segmentation, including the recognition of individual body parts for the purposes of pose estimation, is the dominant computational cost. One of the reasons for the emerging popularity of depth-based systems is easier object segmentation, easier 3D pose estimation, and well-defined scale. However, even for 3D perception systems, where these computational costs are lessened, there may still be high cost in reliably recognizing an evolving series of poses as a gesture.

Regardless of the sensors used, there are both practical and engineering disadvantages to any pose-based gesture recognition system. First, creating a sensor suite that can reliably track, estimate, and classify human limbs and torso in a variety of environments is a large challenge. 3D sensors that work well indoors do not often work well outdoors, and 3D sensors that work well in all illumination conditions are often prohibitively expensive. Additionally, any 2D or 3D sensor will have difficulty maintaining reliable pose estimation of the human's limbs over large distances. In fact, there are many situations where gestures may be needed but where the human's torso or arms may not even be fully observable.

In these situations, where practical or resource considerations make pose-estimation techniques less appropriate, it is ideal to have an alternative system that is less reliant on the need for accurate object recognition and modeling. We present just such an alternative approach that is based purely on motion observed by a 2D camera—specifically on the detection of cyclic motions. Our "beat-based" system can reliably detect well-timed waving of the arm in a horizontal or vertical direction without the need for object recognition of any kind. As an example, our initial implementation recognizes when an operator waves their hand back-and-forth roughly once a second—in other words, a cyclic motion to a 1Hz "beat". While the expressiveness of "beat-based" gestures is limited compared to pose-estimation system, beat-based gesture recognition has several benefits. Beat-based gestures can be detected in a 2D image stream at near or far distances. Detection can occur anywhere in the image frame, which allows the human to be hidden, or allows attention to

[*] This work was funded by U.S. Army Phase I STTR Contract W56HZV-10-C-0443.

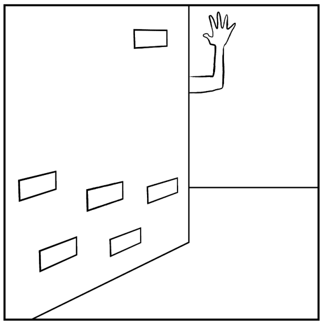


Figure 1: A motion-based gesture system allows for signals to be provided even if the user is hiding.

be focused on a particular location of the image. Also, by allowing the perception system to randomly change the required beat frequency, only a user with knowledge of the current beat will be able to signal the robot or sensor-equipped environment. These characteristics make beat-based recognition a natural complement to more traditional recognition systems that try to determine the exact posture and motion of a well-perceived human.

2 Motivation and Approach

While close-proximity, unobscured gesture recognition is possible in a large number of scenarios, many situations require a user to send (at least simple) signals to the robot in more obfuscated environments and at greater distances. For example, an operator may lead a robot to a particular location and ask it to begin patrolling. Later, the operator may wish to interrupt the patrol and ask the robot to "heel". Ideally, the operator should be able to do this at the limits of the visual range of the robot. Another relevant example is that of a medic or bomb disposal soldier needing to signal an autonomous robot without exposing their body to an unsafe line of sight (see Figure 1). Similarly, an emergency worker may want to signal a robot that is partially occluded by debris, smoke, or fire.

Our attempts at beginning to address this problem have been guided by the observation that, in terms of difficulty for monocular perception: pose-estimation $\gg$ recognition $\gg$ feature detection. "Difficulty" here refers both to the computational and data/sensor resources required. For example, pose-estimation usually requires not only a fast system to perform real-time spatial modeling but also relatively high-resolution data. For a practical example, consider that even analog consumer cameras have long been built with the ability to be triggered by motion, as motion detection is simple. While most modern digital consumer cameras can perform rudimentary facial recognition to locate possible humans inside a photo, most consumer camera capabilities (as of this writing) do not yet have the computational ability to reliably track the orientation of bodies, arms, etc. As such, it would be ideal for camera systems to focus on regions and trigger upon detecting specific human-made motion signals. Such a system would always have greater range than pose detection (or even face detection) and would require fewer resources.

Inspired by similar work commonly done with time-based signals [Carlotto, 2007], this paper represents an exploration into the viability of the approach to gesture recognition. In adapting the idea of purely motion based recognition to the 2D spatial domain, we have come to focus on the detection of repetitive motion at specific frequencies. Thus, we call the resulting gestures "beat-based" gestures. By focusing on motion alone, there is no need for higher level recognition (e.g., of arms and hands), and motions involving relatively few pixels can be detected, which support ranges further than most 3D sensors can handle and at the effective range of most 2D sensors. Our initial experiences indicate that these advantages may well be realizable in a low-cost, easy-to-build sensor system, and that the topic warrants further study.

3 Implementation

Our beat-based gesture approach has been implemented in two distinct ways. In the first iteration, the software was completely correlation based, where changes in pixels were correlated to know cycle times. In order to provide for more flexibility and to require less precise gestures, the more recent implementation is based on motion analysis similar to optical flow.

3.1 Initial Implementation

The first iteration of our system was correlation based. A predefined oscillating signal, a beat, was given to a sensing program and to the operator (in the form of a flashing light). Motion, perceived as texture (i.e. pixel intensity) changes within the video stream was correlated with this signal. If the correlation reached a certain threshold, the pixels involved were considered to represent an executed gesture. To perform a gesture, the operator simply timed his or her movements to the flashing light.

In practice, this system was capable of a greater range of gestures than the simple binary set implied by the correlation threshold because depending on the type of movement, for example pendulation with the hand facing up versus down, different shaped patterns of pixels would reach the correlation threshold before others. These patterns could be distinguished from each other and this allowed for a simple two gesture take-off and land control of an AR Drone (see Figure 2).

In practical testing (e.g., in outdoor environments) the system required the operator to perform only five or six cycles of a gesture before detection. However, the high amount of user feedback needed was impractical and the need for strong synchronization in the system made implementation and parameter changes difficult. To address these problems the system was extended to allow for the detection of oscillations simply approaching (rather than being very accurately synced) with a target frequency at *any* phase. This required a more complex motion perception system.

3.2 Motion Perception

One of the traditional tools for motion analysis is optical flow [Lucas and Kanade, 1981]. Optical flow is often calculated by searching for a collection of per-pixel motion vectors that best explains the evolution of a series of images with

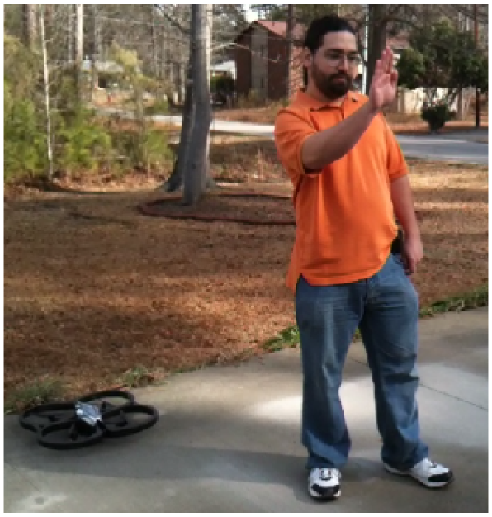


Figure 2: The initial implementation, here used to launch and land a drone, was based on a correlation threshold between a supplied signal (a "beat") and perceived motion.

respect to a number of constraints. Such constraints usually include: 1) that the changes in the pixel intensity values of an image sequence are due purely to motion, 2) that movements that change the relationship of a pixel to its neighbors (in terms of intensity values) are less likely than those that do not, 3) that a pixel is more likely to be involved in a motion similar to one its neighbors are involved in. In the nomenclature of the literature, these constraints are known as the optical flow constraint, the gradient constancy assumption, and the smoothness constraint. Typically, assumptions and constraints of optical flow are treated as costs and the set of best motion vectors is discovered by solving a minimization/optimization problem. Solving such an optimization problem can be quite computationally intense and is often not possible in real-time without specialized hardware or programming techniques.

Our motivation for having a motion-focused system is that motion is a salient and local feature. Many of the constraints of optical flow, when taken together, are equivalent to a non-local analysis of image structure. For example, many fast approximations of "true" optical flow are implemented in terms of tracking higher-level features such as corners. To make our motion analysis as local as possible and to avoid the computational complexity of optical flow, we implemented our own simpler alternative based on a single assumption/constraint: All detected texture changes are due to the motion of persistent objects. In the implementation, this assumption is applied by keeping a record of the time since each pixel experienced a texture (i.e. pixel intensity) change greater than a set threshold. When the assumption above holds, gradients within the resulting value field capture the direction of the motion of objects (see Figure 3). The current system classifies such motion as mainly left-right or up-down, but future systems could use more nuanced information.

On top of this motion perception facility, we implemented a visual oscillation detector that could be tuned to a particular frequency. We adapted a state machine approach similar to the one often used in visual synchronization. Initial results have been encouraging, with low overhead for accurate detection. In this state machine approach every pixel is assigned a counter and a timer. For horizontal detection, when a leftward motion is detected the timer is reset. When a rightward motion is detected the current value of the timer is examined. If the timer's value is close (by a pre-determined tolerance) to the amount that would be expected given the oscillation frequency being sought out, the counter is incremented. Pixels whose counter exceeds a threshold are considered to be involved in oscillations at the target frequency. The system detects a gesture when a target number of pixels are actively involved in such oscillations.[1] Because we classify motion as horizontal or vertical based on the motion gradients, oscillations based on horizontal motion can be detected as distinct from oscillations made by vertical motion.

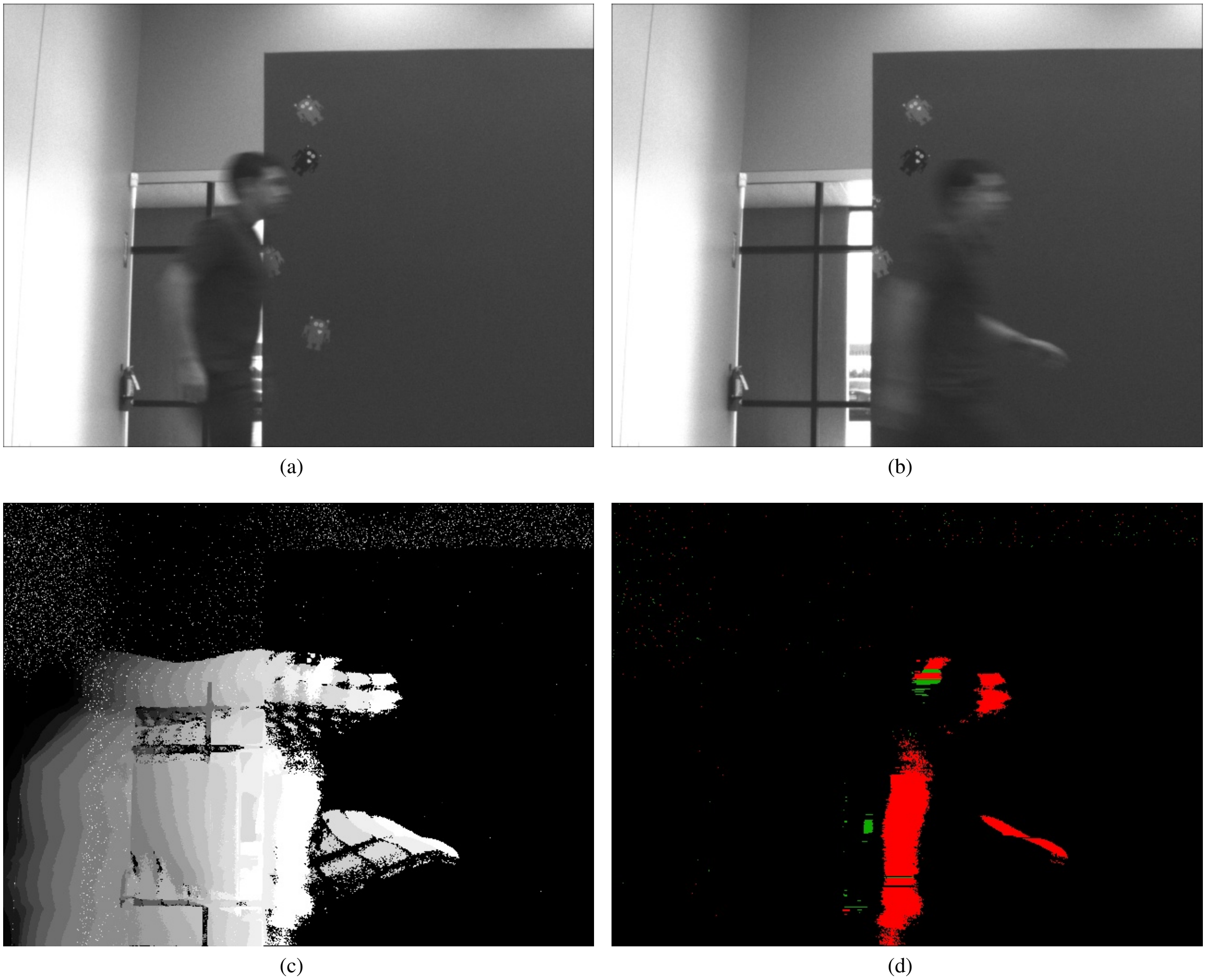
It is important to note that the beat-based gesture detection is based the frequency of *alternating* left-right or up-down motions and *not* simply on the frequency of motion. The system is thus only sensitive to properly oscillating motions and not other forms of cyclic motion. The system would not, for example, respond to a blinking light, even if it was blinking at the target frequency, nor would it respond to a rotating object whose period of rotation was the target frequency.

To test this algorithm's reliability in far-range, outdoor situations, we tuned the system to trigger a detection after seeing three consecutive back-and-forth oscillations at 1 Hz. 1Hz was chosen to eliminate the need for operator training or an externally provided beat: 1 Hz corresponds roughly to counting out loud. The oscillations could occur at any location in the 640×480 images of the motionless camera. Even with this low-resolution camera, we were able to achieve a working distance of ~25.6 meters (~84 ft) with highly reliable detection of 1 Hz arm gestures and with no false positives (even with palm trees blowing and vehicles driving in the distance). We did notice a reduction in recognition quality whenever the background has a similar intensity to the human's arm. Wearing dark sleeves against a light background (or light sleeves against a dark background) can overcome this issue in the initial implementation. Figure 4 shows images from this testing, where the camera used for beat-based gestures is mounted on a mobile robot base.

3.3 Increasing Expressiveness

Although the system could demonstrate highly reliable oscillation detection at far ranges using a still camera, our beat-based recognition suffered from occasional false positives when mounted on mobile platforms operating in highly dynamic environments. These false positives were overcome in two ways. First, whenever the camera itself is moving (known by monitoring any pan-tilt devices or the robot base), the gesture software ignores the incoming camera frames. This

[1] One advantage to this entire approach is that all operations are performed on a per-pixel basis except for the final counting of oscillating pixels and the calculation of the local motion gradients; however, even in these cases a very limited number of neighboring pixels are involved.



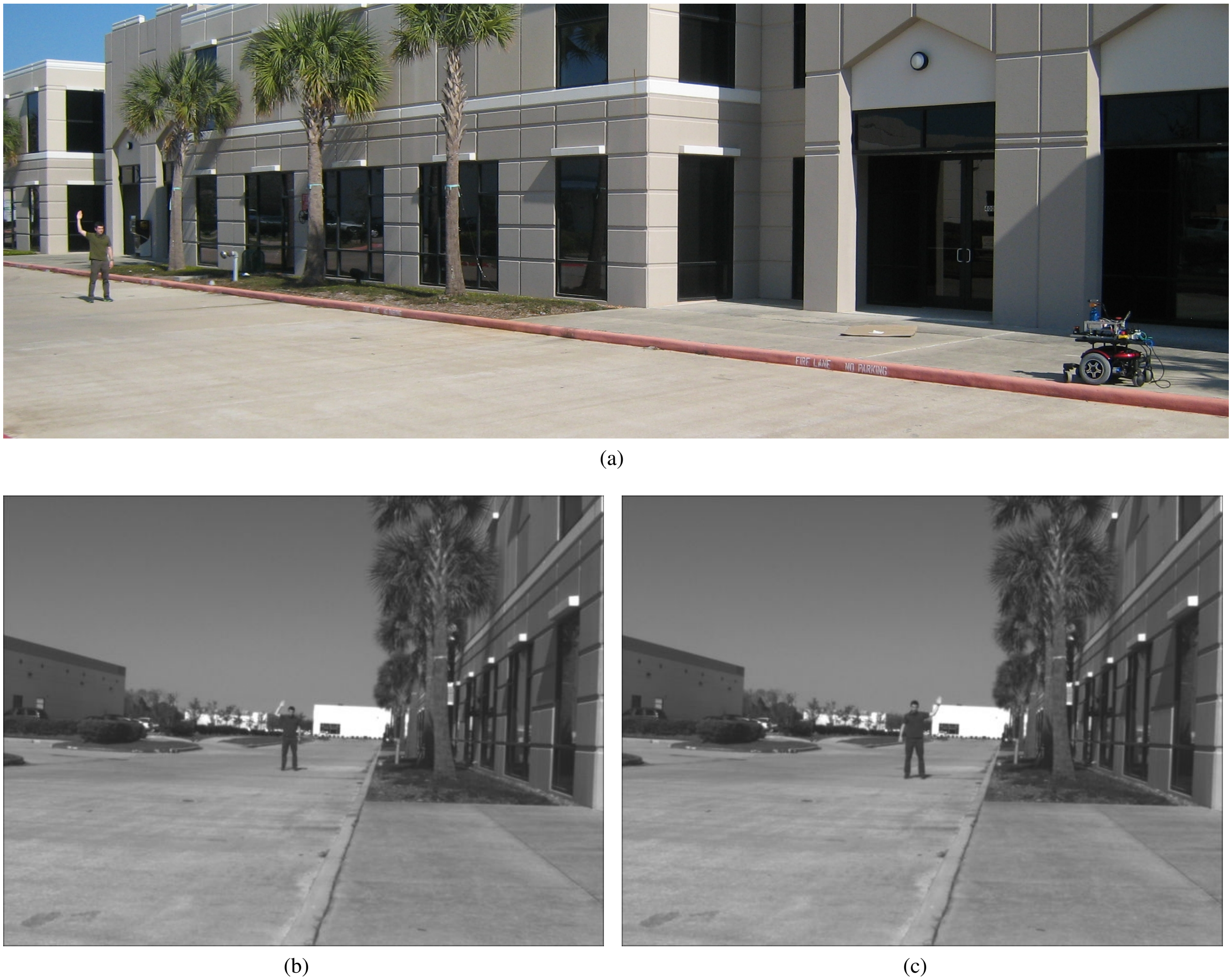
(a)

(b)

(c)

(d)

Figure 3: (a, b) Motion causes temporal changes in pixel values. (c) A "trail" image is created recording which pixels were active changing over time (brighter indicates more recently changed pixels). (d) Determining the gradient of a pixel in the trail image can be used to determine its direction of motion. Here, the high degree of red corresponds to a rightward gradient and indicates the person must be moving rightward. (There is a low number of green ("leftward") pixels in the image.)



(a)

(b)

(c)

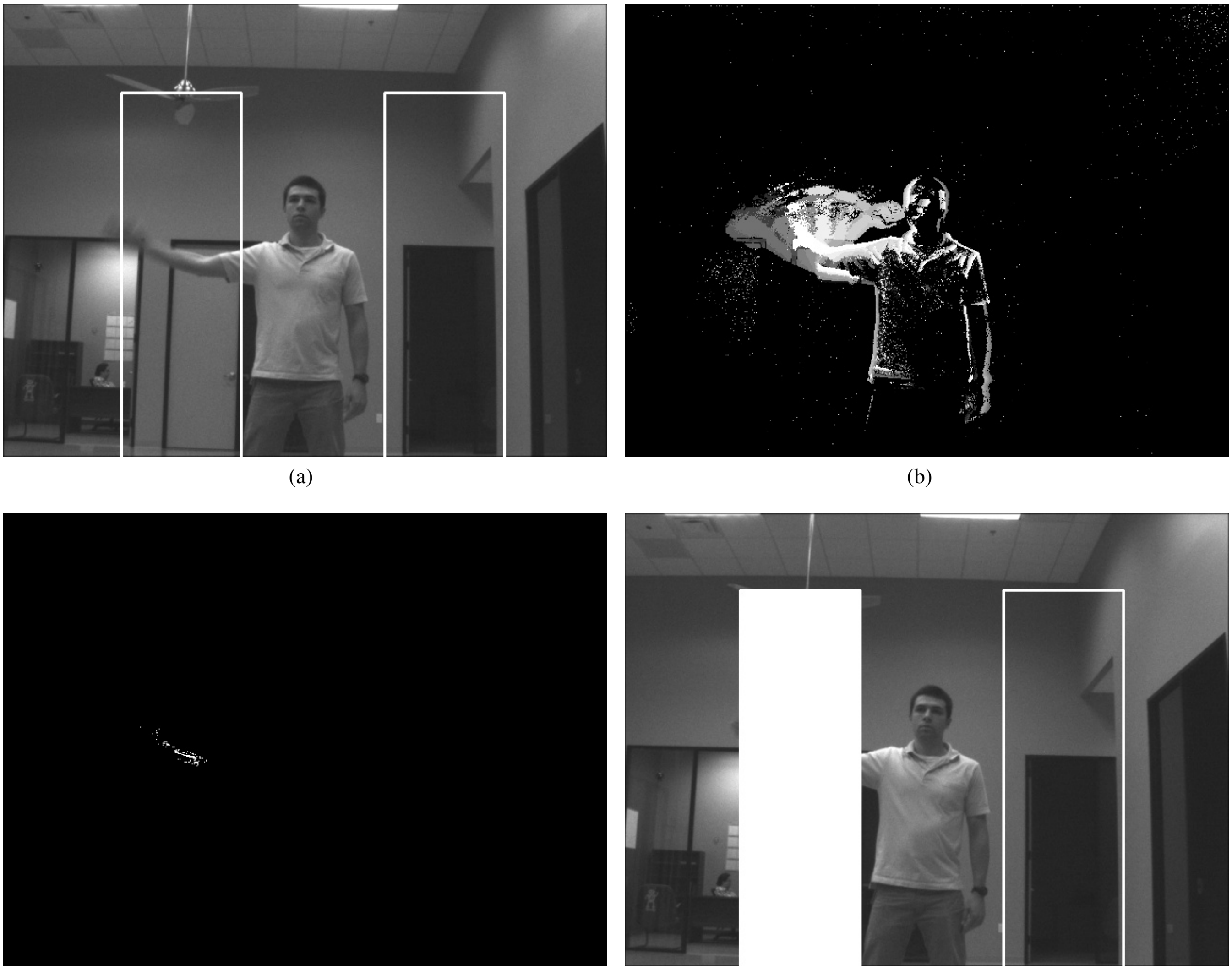
Figure 4: (a) Even with 640x480 resolution images, the beat-based gestures work well at far distances. (b) An example image from the camera at the maximum distance where gestures work reliably. Notice the small number of pixels that fall onto the user's right arm. (c) When the background intensity (grayscale) is similar to the user's arm intensity (left arm in image), the arm motion cannot be separated from the background, resulting in a failure. Future efforts will focus on overcoming these threshold problems.

reduces false positives caused by $\sim$1 Hz oscillations due to panning or tilting of the camera back and forth. This reduces many false positives, but also means that the robot must be perfectly still when beat-based gestures are given. This is accomplished by having an adequate "dead-zone" for any actuation (pan-tilt, zoom, or mobile base control) during tracking behaviors.

To further reduce any false positives and to extend the expressiveness of the system, we decided to augment the system with human tracking information. Given the rough location of the tracked human, the 2D camera image is divided into regions of interest that correspond to areas to the left and right of the operator's torso (see Figure 5(a) and `http://www.youtube.com/watch?v=83Six7g8lMM`). Regions of interest reduce false positive detections caused by a number of factors—most often the human's own body rocking back and forth during conversation or other natural activities. By estimating the distance/scale of the person, the regions of interest grow and shrink as the leader moves closer and nearer to the camera. This has the added benefit of allowing the beat-based software to be more sensitive when the leader is further away, resulting in improved performance at far distances.

In addition to reducing false positives and increasing operating range, knowing the leader location also allows us to distinguish beat gestures made with the left or right arm (Figure 5), **giving us 4 signals** (left/vertical, left/horizontal, right/vertical, right/horizontal). Videos showing left/right arm beat-based gesture recognition working both indoors and outdoors and turning on and off different tracking and following behaviors can be seen at `http://www.youtube.com/watch?v=55F928QVXOI`.

As it was available on the mobile platform, the hu-



(a)

(b)

(c)

(d)

Figure 5: Utilizing human tracking information, right and left arm oscillations can be distinguished; thus, in addition to left-right and up-down beats, the system can determine whether the beats came from the left or right arm of the torso. This results in 4 different gestures that the current system can recognize. (a) Given 3D torso tracking, depth-scaled activation regions to the left and the right of the torso can be used to further refine “beat-based” gestures. (b) An oscillation of the right arm is performed. (c) The final frame of the trace is used to classify the location of the oscillation with respect to the tracked individual. (d) Here, we show the system correctly identifying the activation region that contained the gesture.

man tracking used for augmentation of the system was depth-based. Specifically, a custom 3D template matching approach was used. As such, the low level of tracking accuracy required to distinguish left from right arm beat gestures makes it very likely that an alternative, less resource-intensive, approach such as facial recognition, traditional people-following [Schulz, 2006; Gockley *et al.*, 2007], texture-based tracking, or even sound or radio-based localization could have been used to achieve similar results.

4 Future Work

As mentioned, we have noticed that the intensity thresholds used for real-time motion detection might not always distinguish the human's arm from backgrounds of similar intensity. In the future, will focus on refining the algorithms to increase the reliability of far-range gesture recognition and to recognize more complex motions. First, we will use more principled approaches to segmenting the moving portions of the image stream from the static background. Rather than using static thresholds, we can use thresholds that quickly adapt to particular environment and lighting conditions. KaewTraKulPong and Bowden [2001] present a method that we have used in the past to quickly and reliably pick out motions from image sequences while ignoring sensor noise (see Figure 6). Similarly, we plan to create alternative methods for motion perception, such as adding an additional smoothness constraint, which should eliminate sensitivity to spurious or unintended movement. This would allow beat-based gesture detection to compensate for camera movement, from pan/tilt actuation or from the base rotation/translation.

Next, we will investigate a larger variety of beat-based motions. Our initial implementation cannot distinguish between oscillations at the target frequency (typically 1 Hz in testing) and higher frequency harmonics. This means that providing a beat gesture at a higher multiple of the desired frequency (e.g., at 2 or 3 Hz) is equivalent to providing the gesture at the desired frequency; however, it may be beneficial to allow beat gestures at 2 Hz or 3 Hz to mean different things. Categorizing the exact frequency of beat-based gestures would allow for the same basic motions performed at different speeds to be assigned different semantics.

Additionally, we would like to continue to utilize the simplifying assumption that all detected texture changes are due to the motion of persistent objects. In the past this was only true when the camera was motionless. In the future, we should be able to leverage our tight perception/control loop in order to remove motion between frames caused by pan-tilt or mobile base actuation. This way, beat-based gestures can be detected even on continuously moving platforms, like unmanned aerial vehicles.

Finally, we would like to further explore some of the implications of our initial correlation-based approach. Potentially, when based on a shared signal, beat-based gestures could provide highly secure visual communication between a leader and the robot. Suppose instead of always looking for a 1 Hz cyclic gesture, the robot only acknowledges motions that synchronize with a randomly determined beat, and suppose this beat changes periodically, much like the login information of a secure key-fob. If the operator has access to this information (e.g., via a headset tuned to an encrypted signal or simply through a synchronized clock), then only the operator can command the robot, as only the leader will know the current beat required to control the robot. Such a scenario could be used to acquire a robot's attention, to change between operators, and to keep a threat from taking control of the robot. It would also allow for multiple operator/robot pairs to operate at once in a shared space.

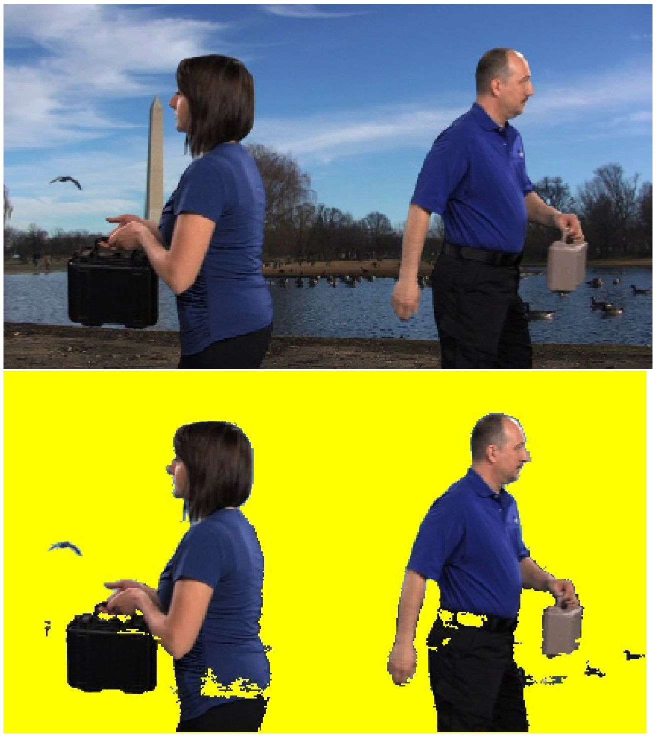


Figure 6: Using the method proposed by KadewTraKuPong and Bowden [2001], small repetitive motions (like leaves or water) are ignored by constantly adapting a mixture of Gaussians to recent frames. Large motions across many pixels are quickly determined via adaptive background subtraction.

In the context of our larger work, which focuses on gesture-controlled human-following systems, we hope to utilize beat-based gestures to develop a hybrid-approach to far-range 2D visual tracking. The approach will utilize well accepted approaches to motion-based tracking [Dang *et al.*, 2002] and cutting-edge work in attention-based recognition [Mishra *et al.*, 2009]. Attention-based techniques attempt to address the “chicken and egg” problem of background separation—that it is easy to separate a recognized object from a background or to recognize an object properly separated from the background, but that either is difficult to perform alone with a natural image. Attention-based techniques solve this problem by performing segmentation based on an assumed object center. In previous work, these object centers were supplied by an operator or some other system (such as stereo disparity); however, due to its ability to perceive motions without object recognition, the current motion perception system presents a unique opportunity to obtain these centers directly.

5 Conclusion

We have presented our initial explorations of low-cost, 2D, motion-based gesture recognition through the implementation of a beat-based gesture system. While this paper described a preliminary investigation of the practicality of the technique, we believe our experience illustrates that the approach is practical and warrants further, more controlled, study. We see great promise for the use of such approaches in hybrid systems integrating a large number of interaction modalities [Stiefelhagen *et al.*, 2004; Rogalla *et al.*, 2002; Haasch *et al.*, 2004; Kennedy *et al.*, 2007] or in applications where pose-based systems are either infeasible, impractical, or overly expensive.

We hope to focus our future work on empirical investigations of our current system's performance in a variety of scenarios—examining the effects of different lighting, background terrains, clutter, environment sizes, distances, dynamics, and human sizes and speeds. Additionally, we want to address some of the main practical flaws in the approach that our initial investigation has revealed. For example, we would like to implement various low-cost motion compensation techniques that would allow for the system to be used by a robot in motion, and we would like to examine the use of alternative sensing techniques (such as thermal imaging) to eliminate the need for the operator to be distinguished from the background by color alone.

Acknowledgments

The authors would like to thank Ben Conrad for his help in creating the images and videos that demonstrate this work.

References

[Carlotto, 2007] M. J. Carlotto. Detecting patterns of a technological intelligence in remotely sensed imagery. *Journal of the British Interplanetary Society*, 60:28–30, 2007.

[Dalal *et al.*, 2006] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the European Conference on Computer Vision*, 2006.

[Dang *et al.*, 2002] T. Dang, C. Hoffmann, and C. Stiller. Fusing optical flow and stereo disparity for object tracking. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 112–117, September 2002.

[Gockley *et al.*, 2007] R. Gockley, J. Forlizzi, and R. G. Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE International Conference of Human-Robot Interaction*, 2007.

[Haasch *et al.*, 2004] A. Haasch, S. Hohenner, S. Huwel, M. Kleinehagenbrock, S. Lang, I. Toptsis, G. Fink, J. Fritsch, B. Wrede, and G. Sagerer. BIRON – the Bielefeld robot companion. In *Proceedings of the International Workshop on Advances in Service Robotics*, 2004.

[Jenkins *et al.*, 2007] O.C. Jenkins, G. Gonzalez, and M.M. Loper. Interactive human pose and action recognition using dynamical motion primitives. *International Journal of Humanoid Robotics*, 4(2):365–385, June 2007.

[KaewTraKulPong and Bowden, 2001] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of the European Workshop on Advanced Video-Based Surveillance Systems*, 2001.

[Kennedy *et al.*, 2007] W. G. Kennedy, M. Bugajska, M. Marge, W. Adams, B. R. Fransen, D. Perzanowski, A. C. Schultz, and J. G. Trafton. Spatial representation and reasoning for human-robot collaboration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2007.

[Knoop *et al.*, 2006a] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3D human body tracking with an articulated 3D body model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[Knoop *et al.*, 2006b] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3D human body tracking with an articulated 3D body model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[Kojo *et al.*, 2006] N. Kojo, T. Inamura, K. Okada, and M. Inaba. Gesture recognition for humanoids using proto-symbol space. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2006.

[Loper *et al.*, 2009] M. Loper, N. Koenig, S. Chernova, O. Jenkins, and C. Jones. Mobile human-robot teaming with environmental tolerance. In *Proceedings of the ACM/IEEE International Conference of Human-Robot Interaction*, 2009.

[Lucas and Kanade, 1981] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Imaging Understanding Workshop*, 1981.

[Mishra *et al.*, 2009] A. Mishra, Y. Aloimonos, and C. Fermuller. Active segmentation for robotics. In *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, 2009.

[Nicolescu and Mataric, 2003] M. N. Nicolescu and M. J. Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[Rogalla *et al.*, 2002] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2002.

[Rybski *et al.*, 2007] P. E. Rybski, K. Yoon, J. Stolarz, and M. Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the ACM/IEEE international Conference on Human-Robot Interaction*, 2007.

[Schulz, 2006] D. Schulz. A probabilistic exemplar approach to combine laser and vision for person tracking. In *Proceedings of the Robotics: Science and Systems Conference*, 2006.

[Sminchisescu and Telea, 2002] C. Sminchisescu and A. Telea. Human pose estimation from silhouettes: A consistent approach using distance level sets. In *Proceedings of the WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.

[Stiefelhagen *et al.*, 2004] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[Waldherr *et al.*, 2000] S. Waldherr, S. Thrun, and R. Romero. A gesture-based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

Models of Indoor Environments – a Generic Interactive Model for Design and Simulation of Building Automation (on Example of BNT-project)

Kalev Rannat[a,], **Merik Meriste**[a,b], **Jüri Helekivi**[b] and **Tõnis Kelder**[b]
[a] Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia
[b] University of Tartu, Ülikooli 18, 50090, Tartu, Estonia
Kalev.Rannat@dcc.ttu.ee

Abstract

Energy saving and comfort in modern buildings is supported by complex automation system. The key issue for achieving adequate energy efficiency is to determine how the environmental impacts can affect the automation system. The system ought to implement the control of devices over dynamic feedback in accordance with dynamical changes in the environment. Without capturing spatial information it is hard to apply any physical impact to a spatially selected area.

This paper presents the concept of an interactive situation-aware model for building automation systems - the Virtual Model of the Building (VMB). VBM is a generic agent-based model developed for intelligent building automation system design, monitoring and simulation. Spatial information captured by VMB reflects selectively impact arising from the environment. VMB consists of active objects (of construction elements e.g. walls, floors, windows, etc.) and automation agents embedded in the in-door environment (e.g. controllers, sensors, actuators, software etc.).

Pilot application of VMB is considered in the context of a pilot software/hardware tool for the design and monitoring of building automation systems based on BACnet protocol.

Keywords. Environment, generic models, simulation, software agents, virtual model of the building, automation, ambient intelligence

1 Introduction

Modern buildings can be considered as systems of systems including heating, ventilation, automation and conditioning, security and many other systems to offer user comfort and applying a wide spectrum of control mechanisms [Dounis and Caraiscos, 2009], [Perumal et al., 2010]. However, often all the subsystems have been considered as independent subsystems of the same building. The spatial context (e.g BIM approach) i.e some model of the in-door environment where the subsystems for automation are embedded in, is typically not considered. The aim of present work is to point at the need to consider the building automation for a certain building as a whole, embedded to physically constrained spatial environment as well as influenced by common physical effects and user factors. It is easy to notice that from the user point of view there exists a lot of partly overlapping functionalities that can be shared between the (sub)systems to offer more complex and flexible control scenarios. The solution is in finding tools and methods to represent similar types of devices, situations and activities in the same way.

The key issue for achieving adequate energy efficiency is to determine how the environmental impacts can affect the automation system. The system ought to implement the control of devices over dynamic feedback in accordance with dynamical changes in the environment. Without capturing spatial information it is hard to apply any physical impact to a spatially selected area.

The key for that can be found in using generic models providing spatial context awareness. In the following we try to point out that an intelligent building with its automated systems can be considered as a "complete organism", where the behavior of components involved is modeled by generic agents The approach considered integrates by means of agent-based models both the building with its construction elements and automation systems including all subsystems and their components as one sophisticated system into a dynamic model of their environment.

Looking for models and methods for modeling the buildings one can find a lot of solutions where the modeling/simulation is based on a minimized physical model of a real building or a certain part of it. In contrary, the generic models and the model components representing the equivalents of certain real devices can be set up (initialized) by the modeler case by case, depending on the situation and needs.

Generic models and generic software agents offer a possibility to model and integrate completely different devices into one virtual environment and simulate the interactions between the system components. In general, it is indifferent if the software agent in the large system of agents represents an equivalent of an electronic device or a human operator with well-determined and restricted degrees of freedom for decision-making. The same concept used by the authors for modeling and simulation of the building automation can be applied for the territory surveillance and security systems modeling. The tests based on the prototype solution are realized and shortly illustrated in the following subtopics).

2 The role of the spatial information

Without spatial information it is hard (if not impossible) to apply any physical impact to a geometrically selected area either for indoor or field conditions. For energy efficiency modeling in the building it is a key issue to determine how the environmental impacts can affect the automation system, what is the temperature (humidity, CO_2, etc) distribution and how to organize dynamic feedback for optimal control.

There can be almost 3 different aspects in the role of spatial information: (1) – spatial information for design, construction and maintenance (2) - spatial information for automation design purposes (3) - spatial information for monitoring the systems and buildings.

2.1 Building Information Model (BIM)

The contemporary building design has met a need for a common method or environment where the building can be examined for any situation and the needed modifications added with minimum human effort. The designing software (e.g. ArchiCAD, AutoCAD, Cadsoft, etc.) has several options to vary with materials, types of constructions and to change certain parts from already finished project. Every change in the building construction will lead to changes in subsystems like ventilation for example. The more sophisticated is the project the more difficult it is to verify that the change in one part does not lead to technical conflicts in another part. This verification is already built into the design software. Management of the building (including the design) has initiated a need to have a continuous record/history on all possible changes in the building. The information on all building construction elements, materials, etc (incl. the changes history) is kept in databases accessible to every authorized counterpart related to a certain building at present and in the future. From this the BIM-concept (Building Information Modeling) has born. BIM is an intelligent 3D model-based process that helps design teams more efficiently incorporate geospatial data into planning, design, construction, operations and asset management. A Building Information Model is a digital representation of the physical and the functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its life cycle from inception onward. Creating a BIM is different from making a drawing in 2-D or 3-D CAD. To create a BIM, a modeler uses intelligent objects to build the model [Conover et al., 2009]. Those who integrate BIM and geospatial data show increased productivity and efficiency [Speden, 2011]. However, in any definitions of BIM [Cerovsek, 2011] we don't notice building automation as a part of it. There is no evidence on using spatial information needed for estimating impact of the physical environment, for providing spatial context for devices behavior.

2.2 Spatial information for automation

The authors have enhanced similar to BIM concept - Virtual Building Model (VBM) to the building automation design, monitoring and maintenance. The work presented is initially realized in 2D approach (each floor separately).

There exist several tools on the market for design of electric installations, monitoring of automation for industrial plants, buildings, but there is no known tool for design and concept proofing of building automation as an integrated system of systems in a specified building/environment. The reasons can be different, mostly it is expected that the BIM-concept has been too novel and complicated for automation design and the automation into the building is usually designed and installed the last. It is not always straightforward to notice that the optimal solution for automation comes from sharing/utilizing the common resources and the regulation needs are tightly related to the same environmental constraints.

Spatial information gives a possibility to apply selective impact to a certain part of the building/automation and to register the effects on control process, either localized or on the whole. The automation system interacts with the environment to where it is installed. The information from geospatial databases can be effectively linked with spatial information of system elements in VBM. With this concept the building with everything included is not an isolated part of environment but just one finite subarea. If the environmental conditions change, the automation realises the control over dynamic feedback, to keep the processes and local parameters in desired limits. The designer can verify his/her technical concepts and the service engineer can find optimal parameters for control.

3 The ways to offer user comfort, environmental/situation awareness in process control, dynamic feedback

A lot of examples can be given where the same building may have rooms for several different needs. The user, operator or owner may also have a need to make some temporary changes in room's microclimate because of varying needs. The fast-growing and modern building market in Asia has initiated a lot of investigations related to the "intelligence of the buildings" [Wong et al., 2008a] and the need for validation of the related analytical models [Wong et al., 2008b]. Many of similar needs have been pointed out also in MASBO [Booy et al., 2008], but mostly considered from the viewpoint of human rationality, not from the system's functionality.

Most common changes for the building come from external temperature (seasonal variability). It can also be foreseen, that some unwanted external effects (smoke for example) will not penetrate to the building. For the automation it means, that the common control algorithm must be temporarily changed – this is the effect of dynamic feedback, where the system behavior is controlled by varying environmental conditions or user needs.

It is known that not all people feel themselves comfortable in mostly unified 21 degrees temperature [Karjalainen, 2007a]. Research has been made on possibilities to offer user-defined conditions in buildings with numerous offices [Karjalainen 2007b].

What is common in small buildings (private houses, where the user can fine-tune mostly everything) is somehow not realised for large offices. Because of individual differences in experiencing thermal environments, no thermal environment can satisfy everybody. But in addition to thermal comfort also productivity and certain health reasons may support the needs for individual thermal control.

Similar needs for automated control can be noticed for applications not related to the buildings. Let's have a look at environmental monitoring for example – seasonal and diurnal changes have an effect on sensor systems and it must be compensated/recalibrated to guarantee the system's reliability. The monitoring system must stay reliable even in case of malfunction of certain sensing nodes in it (meteorological network for example).

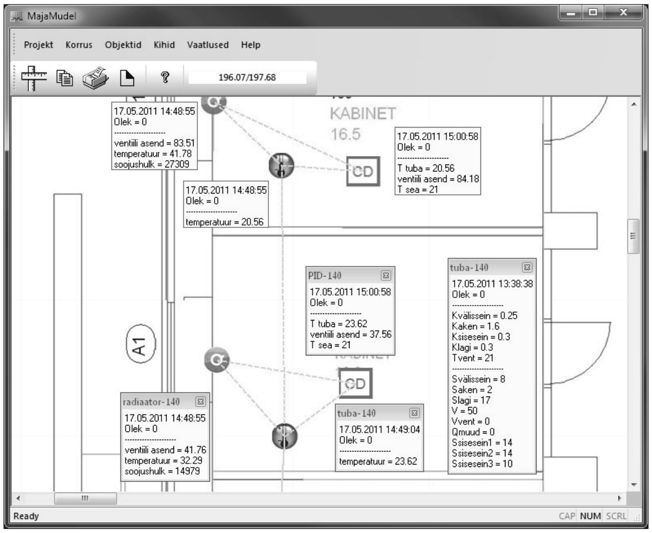
In field conditions (environmental monitoring, territorial surveillance) we need to know how the terrain, meteorological conditions, different objects on the scene will affect the sensors, which group of the sensors can be most affected and by what. To make correct interpretation on results of any measurements we need to know where a certain sensor is situated, the relative distances from other components/objects, sources of radiation or intruders. Space information offers a possibility to determine to which part of automation system the environmental changes have the most effects. The modeling environment must offer a possibility to simulate different scenarios for different subareas, using sliding borders for certain effects in space, etc. Of course, the effectiveness of the modeling and analysis relies on the system analyser who must make a choice where to measure and what (e.g. it means that the operator must be able to situate a suitable agent at a right place).

4 VMB (Model of the Building)

Model of the spatial information of a building is implemented by *BuildingModel* software working in tight cooperation with *BuildingAgents* providing data exchange with physical devices or their simulation. *BuildingModel* implement a visual 2D interface for creating and monitoring objects involved in a spatially smart model of a building equipped with automatization devices. The realization (outlook) of the VMB is illustrated (Figure 1) as a screen snapshot from the prototype software. The initial information comes from the documentation (floor plan) of a real building.

Some definitions

Model of a particular building forms a *Project*, all information of a particular *Project* is kept in a specific database. *Project* i.e the spatial model of a building has one or more floors. Each floor has its base map(s) – coordinated raster maps generated typically on the basis of the building project documentation (CAD drawings). Each floor consists of information about *BuildingElements* (e.g. its construction elements - doors, windows, walls etc) as well as information about devices of automation systems placed on that floor. These devices as well as building elements a placed on the base map also coordinated – information about their locations and spatial relations are available for modeling and simulation on the basis of the model of the building.



```
17.05.2011 14:48:55
Olek = 0
-------------------
ventiili asend = 83.51
temperatuur = 41.78
soojushulk = 27309

17.05.2011 14:48:55
Olek = 0
-------------------
temperatuur = 20.56

17.05.2011 15:00:58
Olek = 0
-------------------
T tuba = 20.56
ventiili asend = 84.18
T sea = 21

PID-140
17.05.2011 15:00:58
Olek = 0
-------------------
T tuba = 23.62
ventiili asend = 37.56
T sea = 21

tuba-140
17.05.2011 13:38:38
Olek = 0
-------------------
Kvälissein = 0.25
Kaken = 1.6
Ksisesein = 0.3
Klagi = 0.3
Tvent = 21
-------------------
Svälissein = 8
Saken = 2
Slagi = 17
V = 50
Vvent = 0
Qmuud = 0
Ssisesein1 = 14
Ssisesein2 = 14
Ssisesein3 = 10

radiaator-140
17.05.2011 14:48:55
Olek = 0
-------------------
ventiili asend = 41.76
temperatuur = 32.29
soojushulk = 14979

tuba-140
17.05.2011 14:49:04
Olek = 0
-------------------
temperatuur = 23.62
```

Figure 1. Screen snapshot of VMB.

Types of devices and their properties

Devices of the same type are specified in the model by *DeviceType* agents. The type of a device specifies (common) attributes of devices belonging to that type. Attributes of a device are: (a) attributes that describe common (similar and fixed by value) to all devices of a type properties and, (b) generic attributes (e.g. parameters) of a particular device with values in some domain. Each type of devices is related to a category of devices where all these devices are belonging to. Devices of the same category have visually the same icon.

Any device of a type is represented in the model by its agent – an instance of the agent of appropriate type. The behavior of agents (i.e. access to physical devices or their behavior at a simulation are described similarly. To an instance of a device in the model is related to a separate service provided by the appropriate agent, which actually can also be treated as a separate instance of an agent.

For example, temperature sensors of various kinds could form each a separate device type, but all of them belong to the category of temperature sensors. These sensors are represented in the model by the same icon, but the connection to particular physical devices could be implemented by different agents exhibiting different behavior. Typical property of temperature sensors is the temperature measured, this property does not depend on the behavior of other sensors and could change in time and space, e.g. it is a parameter of that device. Temperature sensors will be placed on the base raster map of a floor in the building model according to their *de facto* location in that floor. To each of these sensors corresponds a service of the according agent, the description of that service will specify parameters to access a particular sensor.

Elements of the building

In similar manner as device types etc are specified, could also be specified types of elements of the building and the elements themselves (e.g. doors, windows, walls etc). In our current implementation these objects of the model are treated as passive object of the building model – such an object is not related to some (simulation) agent, exhibiting some behavior. Naturally, one can change values of parameters of such an object. Also, these values could be set also by agents of devices embedded in the model of the building.

Relations of objects

There must also be specified *Relations* between devices and/or building elements. By relations one can specify, which devices are connected to each other and/or which objects have influence to each other. For example, a window could influence a movement sensor without being explicitly connected to it.

Relations created between objects will serve for specifying connections of agent services (as representatives of instances of devices).

To objects involved into a building model could be attached user-defined software fragments – scripts. These scripts will be started automatically at model composition time, their serve for imitation and/or checking specific properties or behaviors of devices. Scripts appear to be useful also in monitoring/simulating devices.

Transfers

In connection with any relation of devices there could be also specified one or more *transfers*. By a transfer is defined the sender device, the receiver device and the parameter (value) to be transferred. Agents related to devices will send to each other information according to transfers specified in the building model. Transfers cannot be specified for building elements because they don't have agents.

Agents and the model of the building

Agents are software components implementing specific behaviors of devices at the monitoring/simulation time of the model constructed. Typically, an agent includes a code for interaction with a particular physical device (for example for reading data from a sensor) or, algorithms for emulating the behavior of a device. To the devices of the same type correspond agents with similar behavior. An agent could be instantiated (and is thereafter applicable) in different models for devices belonging to the appropriate device type.

Agents have similar structure, what simplifies creation of new agents as well as guarantees similar structural and semantic properties of them.

5 BACnet Tester (BNT)

A huge amount of different agent based modeling platforms [Nikolai et al.] and agent-based simulation tools [Agent-Based Models] can be found. BNT is developed using C# and .NET, not using any of the "ready-to-use" solutions mentioned above. The development platform for BNT is chosen due to long term activities in agent-based modeling related to interactive maps [Meriste et al., 2004, 2005].

In frames of a project "BACnet Tester" (BNT) is realized a set of hard- and software tools for building automation modeling, including HVAC and security systems. Additionally an option to use smart mobile sensor network based on mote's technology is offered.

The basis of BNT is a Virtual Model of the Building (VMB). The VMB consists of all construction elements (walls, floors, windows, etc.) of the building and the automation components (controllers, sensors, actuators, etc.) in it. VMB elements are supported (tagged) with space coordinates in a local reference frame. Simply - all elements of a virtual model of the building and automation systems are supported with space information.

BNT offers two main scenarios for its users:

For Design:

1. Graphical plan of the building → 2. Virtual model of the building → 3. Add automation, define connections → 4. Start simulations → 5. Optimize the configuration → 6. Finalize the project

For Debugging and Monitoring:

1. Graphical plan (figures) of the building → Virtual model of the building → Add automation, define connections →
2. Define „points of interest" for the measurements or software aided monitoring, connect the hardware (if needed) → Compare the „model" contra „reality" → make changes if needed →
3. Optimize the configuration → Finalize

In practice, the number of test scenarios is not limited - each user can easily develop its own, using the BNT software facilities. BNT is targeted but not limited to HVAC, it can be applied to security and lighting systems as well.

BNT consists of certain application software and laboratory tools (Figure 2). An essential part of the software is made for composing the VMB (including facilities for converting the graphical building plan to the virtual building model together with automation systems) and running simulations on it. BNT *is not* an apparatus (or device), but can be ported as a complex of special software-supported instruments to perform analysis at different sites (buildings). For a different building a new VMB must be generated as a first step.

The software provides monitoring of the system parameters in both VMB and reality, handling the databases, making analysis and outputting the reports.

One part of the software is communicating with the BNT hardware (laboratory equipment like network analysers, etc., + all the automation systems installed to the building). During test/simulation processes, the real object (building + automation) works together with the VMB as a whole. The

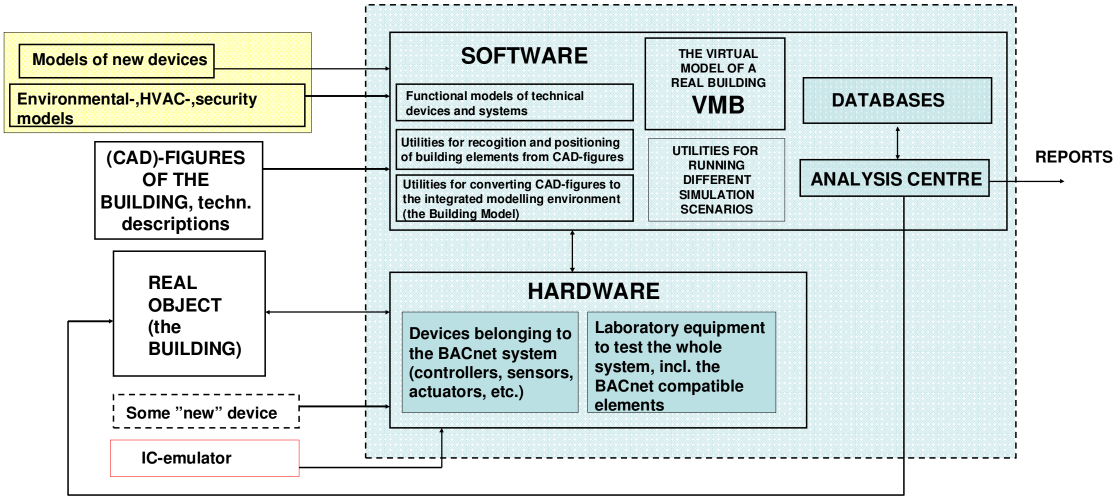

Models of new devices
Environmental-,HVAC-,security models
(CAD)-FIGURES OF THE BUILDING, techn. descriptions
REAL OBJECT (the BUILDING)
Some "new" device
IC-emulator
SOFTWARE
Functional models of technical devices and systems
Utilities for recogition and positioning of building elements from CAD-figures
Utilities for converting CAD-figures to the integrated modelling environment (the Building Model)
THE VIRTUAL MODEL OF A REAL BUILDING
VMB
UTILITIES FOR RUNNING DIFFERENT SIMULATION SCENARIOS
DATABASES
ANALYSIS CENTRE
REPORTS
HARDWARE
Devices belonging to the BACnet system (controllers, sensors, actuators, etc.)
Laboratory equipment to test the whole system, incl. the BACnet compatible elements

Figure 2. General description of functional blocks

result comes from comparing „real measurements" ←→ „what was planned". The optimal solution can be reached as stepwise iteration by changing the control parameters and/or modifying the installation.

BNT software is realised on agent-based programming paradigm [Ferber, 1998] and generic models of automation system elements. BNT is easily expandable and can be used for different communication standards (BACnet is not obligatory, for BNT the BACnet was chosen as a well documented and expandable standard [Bushby, 1997] for HVAC and in our case for the security system as well). The BNT can be easily expanded to adapt new hardware components on the market and the user can fast develop a wide spectrum of different conceptual tests (limited to one's own fantasy).

The instrumental part of BNT is realised in form of a small test laboratory, equipped with the instruments for network traffic analysis, outdoor conditions monitoring, BACnet compatible controllers for both HVAC and security systems, Crossbow motes based wireless sensor's network and naturally the server for running the BNT software. As a future vision, the instrumental part should fit into a suitcase and gives the operator a possibility to work at the site of interest.

5.1 Application of the BNT

BNT is designed for both long-term trends analysis and monitoring of certain parameters at a point of interest. The first test-bed for BNT has been the building of Institute of Technology, Tartu.

For buildings:

a) large buildings

b) private houses (small buildings)

BNT offers a possibility for remote monitoring (getting raw information from the remote object) and running simulation on BNT, consisting of virtual model of certain part of the remote building and automation in it.

The plan of the remote building can be used as an interactive map of the building. It is possible to localize an area of interest, to choose the sensors and actuators on it and to activate data monitoring at the points of interests. The controller(s) at the localized area will be replaced with a software agent or agents, acting exactly as the real controller(s) would do. Based on data from "hot points" the trend curves can be compiled.

By analysing long-term trends of certain parameters it is possible to evaluate the „energy efficiency" of the remote system. It is also possible to detect some „unusual behaviour" in the remote system (system's generation for example) and to offer a solution how to avoid it. For example, cooling down and warming up a certain room in the remote building will give specific information on thermodynamic behaviour of the area of interest. This valuable information is used for making suggestions, how the real controller should be programmed for a certain room. During installation the controller is programmed with default settings based on expert estimation what cannot be optimal – optimization can be done only based on analysis of real situations. Shortly – the expert can give analysis based suggestion for the programmer how a certain controller must be programmed to obtain the optimal energy efficiency and comfort.

However, the correct analysis of the remote system and energy efficiency of the building is possible only if we have good enough knowledge about the environmental conditions and their impact on the building. At today's level of development the BNT needs to get the environmental information and thermodynamic effects for the building from aside (from different model or experts). Knowledge about thermodynamic processes in a certain building gives BNT a possibility to simulate „close to real" situation and to give suggestions (based on simulations), how the situation can be changed/optimized.

For private houses the task can be easier (no need to split the building into different zones/rooms to estimate its total energy efficiency).

Some additional applications:

a) Simulation of automation system itself (optimizing the system).

b) Simulation of automation system together with a „human factor" (by adding an agent to organize some disturbances or making local adjustments in the office to improve his/her personal comfort).

c) Simulation of environmental monitoring or territorial surveillance.

Example: (room cooling/heating, room's parameters analysis and simulations):

Let's assume a situation described (Figure 3), where a single room gets unheated while the neighbouring rooms have normal temperature at 21°C. The application of BNT helps to analyse, what happens with the room temperature (on which temperature it will stabilize) and what is the influence on heating system in the neighbourhood. The outdoor temperature is assumed stable at -15°C (typical situation for Tartu in winter 2011).

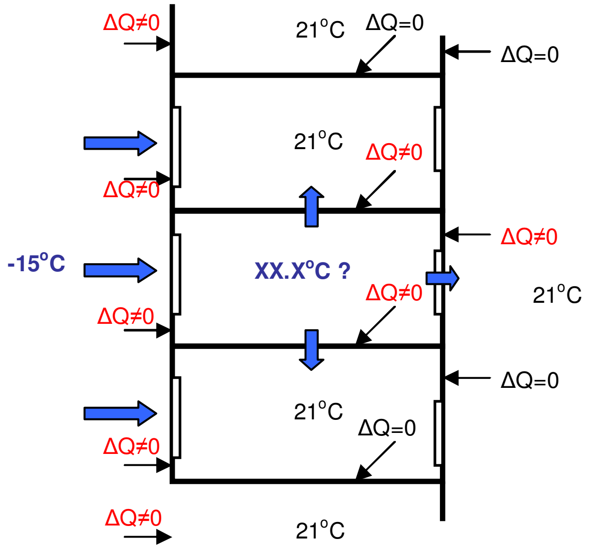


Figure 3. Initial question, what happens if...

To get any idea about the room's physical parameters (thermal inertia, the total heat capacity, heating budget, energy efficency, etc.) some real tests must be made. On (Figure 4) one can see a time series of room temperature (upper curve) and outdoor temperature (lower diurnally oscillating curve). The almost 5-days experiment is made in two parts a) cooling the room (the temperature decreases exponentially) and b) heating it up (the temperature rises by a power law). For the cooling session the ventilation was switched off to avoid heating by the constant 21°C air inflow.

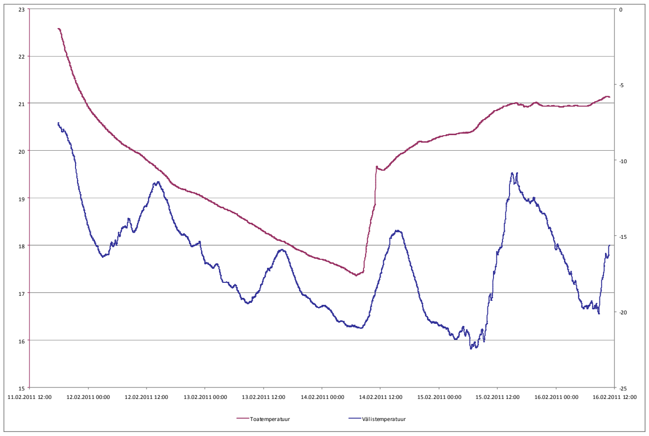


Figure 4. Real measurements to find the physical parameters.

The real measurements look quite unsmooth and noisy, they must be post-processsed for further data analysis (Figures 5 and 6).

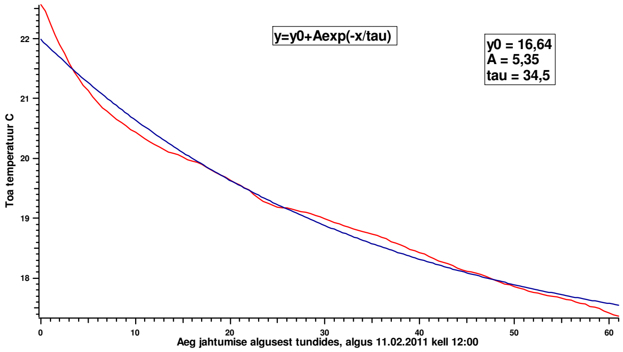


Figure 5. Analysis I (tau is directly related to the room integrated heat capacity).

From the analysis it can be found the most needed parameters for all similar rooms in the building (assuming the size, geometry and configurations unchanged). To get optimal results, the preliminary cooling/heating tests must be made exactly for the room we investigate (not by using test results form a different room). If using measurement results just from similar rooms, we must consider that each room is unique in details – the mixing of air and the radiating heat exchange can be suboptimal because of the furniture, the wall materials can be different, etc.

For a test situation (Figure 3) we need to find the total thermal capacity for the room (incl. the air, furniture, etc). After that we can run the simulations and check the influence of one unheated room to the adjacent neighbors.

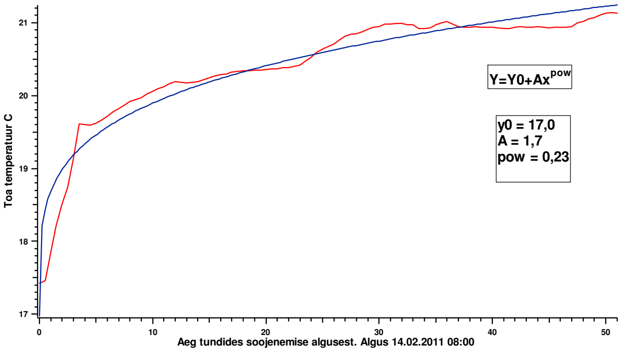


Figure 6. Analysis II

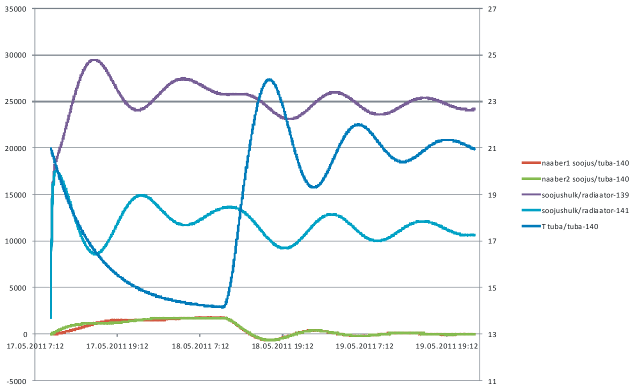


Figure 7. Simulations based on Analysis I, II

From (Figure 7) we can see that the room temperature gets stabilized at around 14^{o}C. The oscillation of heating curves indicates the characteristic process of PID-regulators and the oscillation can be damped by changing the control parameters for PID controllers (for the model's case the PID-controller agents). The cooling stops because of the heat inflow through the internal walls (additionally through the roof and the floor) and constant air inflow from the ventilation. It can be noticed that the heating power in adjacent rooms will increase.

The illustrative simulation is trivial but it helps to better explain the functionality and possibilities the BNT realisation can offer today. However, demonstration of all the developed and tested features of the BNT is outside the scope of the article.

6 Related works

Hereby we want to emphasize that VMB is not the same as Multi-Agent System for Building cOntrol (MASBO) as presented in [Booy et al., 2008]. BNT software agents represent real equipment in the automation system and by monitoring the changing environmental conditions (either real or simulated) at positions they are virtually fixed, the energy efficiency for a building is assured by dynamic feedback.

More related work can be found additionally from [Perumal et al., 2010] and [Dounis and Caraiscos, 2009]. VMB is developed to manage monitoring/controlling of changes in environmental conditions at a certain finite area of the building (or environment). VMB acts as a generic model to evaluate the impact of changes to the building automation and to organize dynamic feedback to support user's comfort, safety and system's integrity. The modeling/simulation with VMB incorporates spatial information as one of the principal cornerstone – every single element of the model is tagged with its space coordinate and therefore the changes of environmental conditions may have their impact selectively on a certain part of the automation system. This concept helps easily to mask the environmental changes/conditions to a geometrically preselected part of the system only (for example, a group of rooms) and to investigate the impact to the whole system.

An extensive research has been made on trying to identify what makes the building "intelligent" or "intelligent enough" to fulfill the contemporary requirements on energy consumption and user comfort [Wong et al., 2008a, 2008b] and how to apply the optimal control methods [Dounis and Caraiscos, 2009].

It can be noticed that there exists tens of "intelligence indicators" without clear consensus between different user groups what is the most important. The authors of present work have tried to search information about the need to use spatial information as one of the intelligence indicators. None of the tens of indicators is clearly related to space coordinates or space coordinate usage. It is somehow astonishing, that the intelligent indicators from the research [Wong et al., 2008a, 2008b] do not reflect much need for utilizing information about the space (spatial information, spatial coordinates).

The ranking results for "the most important indicators" for appraisal of buildings intelligence may differ between different user groups and the statistical analysis based on analytic models [Wong et al., 2008b] is not trivial.

A good example about modeling and monitoring zonal information in a building can be found from: [Dibley et al., 2010], where a software system has been developed that exploits the combination of a number of technologies. As well as generating useful knowledge for decision support, the software is reported to be self configuring, continually adapting to the environment, and employing learning to evolve its performance. The agents utilize a distributed network of readily available wired and wireless sensors and associated data storage providing access to near real time and historical data, as well as an Industry Foundation Classes (IFC) [BuildingSMART, 2011] model describing building geometry and construction. The agents used in [Dibley et al., 2010] work individually and cooperatively for identifying the usage and dynamics of arbitrarily sized spaces in buildings. The building model itself with construction and positioning of the sensors is described by IFC-file.

(In fact, it can be the way how to link VMB–data with BIM) The multi-agent system used in [Dibley et al., 2010] is designed to support decision-making for facilities management, trying to track the room's occupancy, to identify user behaviour and individual preferences to provide as efficient management of the building as possible.

Research and publications can be found about functional modeling of spatio-temporal aspects of buildings [Bhatt et al., 2010], [Hois et al., 2009] and indoor spatial models and reasoning with spatial structure [Schultz and Bhatt, 2010], [Bhatt et al., 2009] for Spatial Assistance Systems. The works are more related to spatial representation of automation (motion sensors) and their range space to offer optimal coverage of the sensors and to avoid "static" conflicts with building construction elements. Here the BNT solution offers a complementary concept where the sensors as a part of automation system can register the environmental changes at certain location in space (determined by local coordinates) and the software-based agents use this information to guarantee the energy efficiency.

7 Conclusions

The history for any building starts from a vision of the architect who has got a specification of user needs/wishes from the owner of the building. The realization of the project is made with a wide variety of CAD-software and the result can be offered in both human- and machine-readable format. At design phase the architect can use a full set of visualization possibilities known from 3D design and the result can even be demonstrated to the end-user before final acceptance. Ventilation, drainage, water supply etc. belong to every building as an inseparable part that cannot be designed without knowing the technical details and physical positioning of all the building construction elements.

For the BNT the work starts from converting the input data from graphical representation of building (usually the Auto-CAD format technical figures) to the internal data format of VMB. As a first approach, the process is semi-automatic.

The most challenging for the VMB is the implementation of thermodynamic modeling of the building. Most contemporary simulation programs are based either on response function methods or on numerical methods in finite difference or, equivalently, finite volumes form [Clarke, 2001]. For BNT the first option is chosen. It looks unreasonable to integrate everything into one solution. One possibility is to use commercially available software packages for thermodynamic modeling with finite elements methods. This approach needs an additional software module for data communication between BNT and COMSOL for example. Due to the internal structure of BNT it can be easily expanded or linked to third party software.

In the future more options can be added the software package: automated conversion from CAD-figures to VMB, automated design/pre-configuration of HVAC and security systems components, preliminary cost calculations, etc.

Acknowledgments

The work has been supported by Enterprise Estonia as an applied research project "BACnet Tester" EU 29103.

References

[Agent-Based Models] Agent-Based Models, Methodolgy and Philosophy, http://www.agent-based-models.com/blog/resources/simulators/

[Bhatt et al., 2009] M. Bhatt, F. Dylla, J. Hois. Spatio-Terminological Inference for the Design of Ambient Environments. Proceedings of the 9th International Conference on Spatial Information Theory (COSIT), France, September 2009, Lecture Notes in Computer Science, Vol. 5756, ISBN 978-3-642-03831-0, Springer, 2009, pp. 371-391

[Bhatt et al., 2010] M. Bhatt, J. Hois, O. Kutz, F. Dylla. Modelling Functional Requirements in Spatial Design. In Proc. of the 29th International Conference on Conceptual Modeling (ER-2010), Vancouver, BC, Canada, 2010.

[Booy et al., 2008] Darren Booy et al. A Semiotic Multi-Agent System for Intelligent Building Control. In *Conference Proceedings, Ambi-sys 2008,* February 11-14, Quebec, Canada, 2008

[BuildingSMART, 2011] BuildingSMART, Model – Industry Foundation Classes (IFC), 2011. http://buildingsmart.com/standards/ifc/model-industry-foundation-classes-ifc/ , 2011

[Bushby, 1997] Steven T. Bushby. BACnet^{TM}: a standard communication infrastructure for intelligent buildings. Automation in Construction, Volume 6, Issues 5-6, September 1997, pp.529-540

[Cerovsek, 2011] Tomo Cerovsek. A review and outlook for a 'Building Information Model' (BIM): A multi-standpoint framework for technological development. Advanced Engineering Informatics, Volume 25, Issue 2, April 2011, pp. 224-244

[Clarke, 2001] J.A. Clarke, *Energy Simulation in Building Design*. Butterworth-Heinemann, 2nd Edition, 2001.

[Conover et al., 2009] Dave Conover et al., An introduction to building information modelling. A guide to ASHARE members, 2009 American Society of Heating Refrigerating and Air-Conditioning Engineers, Inc. Nov. 2009

[Dibley et al., 2010] M.J. Dibley, H. Li, J.C. Miles, Y. Rezgui, Towards intelligent agent based software for building related decision support, Advanced Engineering Informatics 25, Elsevier, doi:10.1016/j.aei.2010.11.002, 2011, pp. 311-329

[Dounis and Caraiscos, 2009] A.I. Dounis, C. Caraiscos. Advanced control systems engineering for energy and comfort management in a building environment—A review. Renewable and Sustainable Energy Reviews, Volume 13, Issues 6-7, August-September 2009, pp.1246-1261

[Ferber, 1998] J. Ferber. Multi-agent systems: an introduction to distributed artificial intelligence. 509 p., Addison Wesley, 1998.

[Hois et al., 2009] J. Hois, M. Bhatt, O. Kutz. Modular Ontologies for Architectural Design. Formal Ontologies Meet Industry (FOMI 2009), Proceedings of the 4th Workshop FOMI 2009, September 2, 2009, Vicenza, Italy, in association with the 10th European Conference on Knowledge Management. Frontiers in Artificial Intelligence and Applications, Vol. 198, ISBN: 978-1-60750-047-6, IOS Press, 2009, pp 66-77

[Karjalainen, 2007a] Sami Karjalainen. Gender differences in thermal comfort and use of thermostats in everyday thermal environments. *Building and Environment*, Vol. 42, No.4, pp. 1594-1603

[Karjalainen, 2007b] Sami Karjalainen. *The characteristics of usable room temperature control*. VTT Publications 662, Espoo, 2007

[Meriste et al., 2004] M. Meriste, L. Motus, T. Kelder, J. Helekivi. Agent-based Templates for Implementing Proactive Real-time Systems. The 3rd International Conference on Computing, Communications, and Control Technologies; Texas, USA; 24–27 July, 2004, pp. 199–204

[Meriste et al., 2005] M. Meriste, J. Helekivi, T. Kelder, A. Marandi, L. Motus, J. Preden. Location Awareness of Information Agents. *In: Advances in Databases and Information Systems, Proceedings: 9th East European Conference on Advances in Databases and Information Systems, ADBIS2005; Tallinn; 12–15 September, 2005. (Eds.) Eder, J; Haav, H.M.; Kalja, A.; Penjam, J.,* Springer, (Lecture notes in computer science), 2005, pp. 199–208.

[Nikolai et al.] C. Nikolai, G. Madey. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms *Journal of Artificial Societies and Social Simulation* vol. 12, no. 2, http://jasss.soc.surrey.ac.uk/12/2/2.html

[Perumal et al., 2010] Thinagaran Perumal, Abd Rahman Ramli, Chui Yew Leong, Khairulmizam Samsudin, Shattri Mansor. Middleware for heterogeneous subsystems interoperability in intelligent buildings. Automation in Construction, Volume 19, Issue 2, March 2010, pp. 160-168.

[Speden, 2011] Liam Speden. Are you ready for BIM? *Geo World*, March, 2011, pp. 18-22.

[Wong et al., 2008a] Johnny Wong, Heng Li, Jenkin Lai. Evaluating the system intelligence of the intelligent building systems: Part 1: Development of key intelligent indicators and conceptual analytical framework. Automation in Construction, Volume 17, Issue 3, March 2008, pp. 284-302.

[Wong et al., 2008b] Johnny Wong, Heng Li, Jenkin Lai. Evaluating the system intelligence of the intelligent building systems: Part 2: Construction and validation of analytical models. Automation in Construction, Volume 17, Issue 3, March 2008, pp. 303-321.

[Schultz and Bhatt, 2010] C. Schultz, M. Bhatt. A Multi-Modal Data Access Framework for Spatial Assistance Systems. In Proc. of Second ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness (ISA 2010), In conjunction with ACM SIGSPATIAL GIS 2010, San Jose, CA, USA, 2010.

Calculating Meeting Points for Multi User Pedestrian Navigation Systems Using Steiner Trees

Bjoern Zenker, Alexander Muench
University of Erlangen-Nuernberg
Erlangen, Germany
bjoern.zenker@cs.fau.de

Abstract

Most pedestrian navigation systems are intended for single users only. Nevertheless, there are often cases when pedestrians are not alone e.g. when meeting or going somewhere with friends. Motivated by such situations, we built a navigation system that allows routes to be calculated for multiple people who want to meet, but who depart from different locations. In this paper we present how satisfying meeting points can be found. We discuss two approaches, one based on the Steiner Tree Problem in Networks and one based on the Euclidian Steiner Problem, which neglects the street network. Both approaches are evaluated and a user study demonstrates the applicability of our solutions.

1 Introduction

People often go out with other people, meet friends and prefer covering distances together. According to [Moussad *et al.*, 2010] 70% of all pedestrians travel in a group. [James, 1951] found that 71.07% of pedestrian groups groups consisted of two individuals and 28.93% groups consisted of two to seven individuals. This results in a average group size of 2.41 individuals. Nevertheless, most pedestrian navigation systems are intended for single users only.

To close this gap, we built a mobile pedestrian navigation system for groups (PNS4G) of users who want to meet. As a motivating example, imagine two first-year students living in two different student dormitories in different parts of a city. They arrange to go to the cinema together. On their way they want to talk to each other about their current lectures, but they also have to finish their homework first so they do not want to have too much detour. Thus a compromise between detour and the time walking together is needed. Where should they meet? Figure 1 shows four different possibilities for the individuals p and q with the common destination g. Current navigation systems cannot help users to solve this problem.

The focus of this paper is on finding satisfying meeting points for individuals who depart at different locations and who have a common goal. We will present two approaches to find satisfying meeting points and corresponding routes for the users. Together these routes compose a meeting tree. We will cover this problem for two or more individuals. Figure 2 shows a possible meeting tree for three individuals.

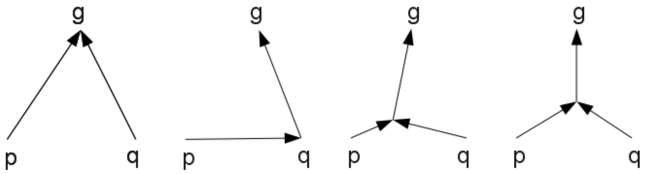


Figure 1: *F*rom left to right: Users p and q meet at the destination g, at q's position, at some intermediate place, at the Torricelli point.

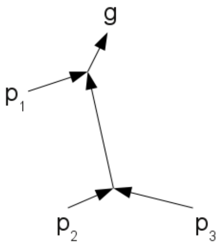


Figure 2: Meeting tree of three individuals p_1, p_2, p_3 heading to a common goal g meeting at intermediate points.

First, we will present the state of the art and introduce our multi user pedestrian navigation system in Section 2. In Section 3 we first present a practical and theoretical formulation of the problem of finding good meeting points. We then propose two methods for finding such meeting points in Section 4. The first method is based on the Steiner Tree Problem in Networks. As the running time of the used algorithm is too high, we investigate a second method based on a relaxation of our problem, namely neglecting the underlying street network. By changing the problem definition in this way we can employ Euclidian Steiner Trees to model our problem and therefore utilize faster algorithms. Finally, Section 5, we compare both methods and show the applicability in a user study in . We conclude by giving an outlook for further research and development in Section 6.

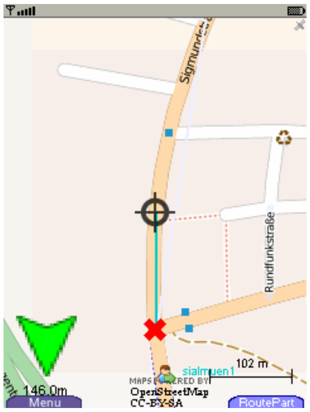


Figure 3: Screenshot: User (cross hairs) will meet his friend (user icon) soon at the meeting point (red cross).

2 Towards Multi User Pedestrian Navigation

2.1 Pedestrian Navigation

Current pedestrian navigation systems help pedestrians to find their way to their goals by giving turn-by-turn instructions. Today all pedestrian navigation systems known to the authors are for single users only. Examples of such systems are Google Maps Navigation, ovi maps, MobileNavigator, PECITAS [Tumas and Ricci, 2009], P-Tour [Maruyama *et al.*, 2004], RouteCheckr [Voelkel and Weber, 2008], COMPASS [van Setten *et al.*, 2004] and many more. Additionally, most commonly used routing algorithms for pedestrian navigation in street networks like Dijkstra and A* as well as algorithms for public transport networks, e.g. [Huang, 2007] and [Ding *et al.*, 2008] are single source only. Thus, such algorithms cannot be used for calculating routes for multiple users or rather meeting trees. However, known from the literature, the majority of pedestrians go out in groups in their leisure time.

To address this, we created the multi user pedestrian navigation system GroupROSE , an extension of [Zenker and Ludwig, 2009] . Each user runs a client software (currently in J2ME) on his mobile phone. One user can invite other users to a location, e.g. a certain cinema. After the users have accepted the invitation, their GPS positions are sent to the server. There, routes for all users are calculated with appropriate meeting points. These routes are displayed on the mobile phones and allow turn-by-turn navigation for each user. Figure 3 shows the client of a user who is going to meet shortly another user. Then, they will continue their journey to their destination conjoint. An overview map of the routes of two users in the city of Berlin can be seen in Figure 4.

2.2 Steiner Tree Problem

From a theoretical point of view the Steiner Tree Problem resembles the problem of finding meeting points and corresponding routes. Hence we will give a short introduction to this problem, namely to "Find the shortest network spanning a set of given points…" [Winter, 1987]. One can find two similar versions of the Steiner Tree Problem in literature:

Steiner (Tree) Problem in Networks (SPN) Given an undirected network $N = (V, E, c)$ with vertices V, edges E and cost function $c : E \to \mathbb{R}^+$ and a terminal set $T \subseteq V$, find the subnetwork S of N such that all nodes $t \in T$ are connected and that the total costs $\sum_{x \in E_S} c(x)$ are a minimum. S is called Steiner Minimum Tree (SMT). Vertices $SinV_S \setminus T$ are called steiner points. SPN is NP-complete [Karp, 1972]. An overview of several exact and approximative algorithms as well as a introductions to SPN are given by [Winter, 1987] and [Proemel and Steger, 2002].

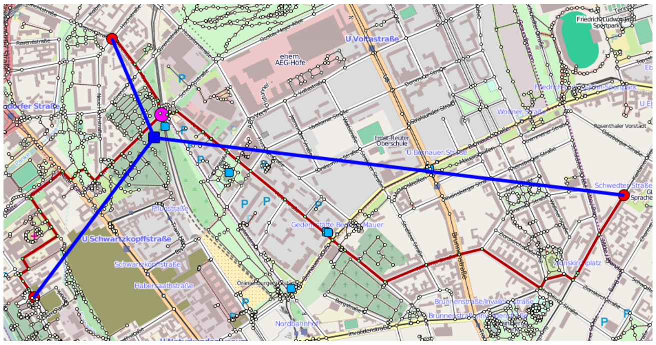


Figure 4: Comparison between Toricelli point (and lines of sights) and Steiner point (and routes in the street network).

Euclidian Steiner (Tree) Problem (ESP) Given a set T of n points in the Euclidian plane, find a tree S connecting all $t \in T$ minimizing the length of S. Note that this might introduce new points at which edges of the tree meet. A minimal tree for 3 terminals shows the rightmost graph in figure 1. The three edges meet at the Torricelli point. A prominent exact algorithm was given by [Melzak, 1961], heuristics e.g. by [Smith *et al.*, 1981] and [Chang, 1972]. *Geosteiner* [Warme *et al.*, 2000] [Warme *et al.*, 2000] is an implementation of an exact algorithm with special pruning techniques to rule out implausible SMTs. Detailed information about this problem can be found in [Winter, 1985].

3 Meeting Problem

Currently, there is no literature known to the author regarding meeting behaviour and in particular, the finding of suitable meeting points. Thus we will give our own definition of the problem.

3.1 Practical Problem Formulation

Given is a map of a city and a set of individuals starting positions and a goal position in that city. For each individual, a route has to be found from his starting position to the goal position. All these routes together compose a meeting tree (MT). The problem ist to find a MT such that the following requirements are optimized:

- the distances travelled together should be maximized
- the detour for doing so is minimized.

These requirements correspond to common sense every day meeting behavior of people. From a social psychological point of view, we can formulate a different requirement: Find a MT such that the costs for all individuals are minimized.

By assigning costs for detour and negative costs to distances travelled conjoint we transform the common sense requirements to the social psychological requirements. These requirements also conform to rational choice theory (see e.g. [Becker, 1976]), which thinks of individuals as if they would choose actions to maximize personal advantage.

In this paper we assume that positions are given as pairs of latitude and longitude and that all positions are in a city. The latter assumption means, that individuals can walk from one position to another by following roads.

Next, we will give a more formal definition of this problem.

3.2 Theoretical Definition: Meeting Tree Problem (MTP)

We start with some definitions. A path $r = \{v_1, v_2, \ldots\}$ is an ordered set of vertices. r_p is the path of individual p. An edge $e_w = (v_w, v_{w+1})$ is a pair of consecutive vertices in a path. The union of all individuals' paths $\{r_1, r_2, \ldots, r_n\} \in MT$ is the meeting tree. Each vertex has a position. Positions are given as pairs of latitude and longitude. The length of an edge $e = (e_1, e_2)$ will be measured using the great-circle distance $d(e) = ||e_1, e_2||$.

The cost of a path which is covered by one individual is $c_{single}(r) := \sum_{e \in r} d(e)$. To consider the fact that people prefer to walk together we set costs for paths who are covered conjoint by m individuals to $c_{conjoint}(r) = \frac{c_{single}(r)}{s}$. This means that the costs of a path covered by more individuals is weighted by a factor s. In this paper we will always set $s = m$, which means, that the *conjoint* costs of a path only depend on the distance and not on the number of individuals travelling on the path. The costs of the MT are calculated by $c(MT) = \sum_{r \in MT} c_{conjoint}(r)$.

Now we can formulate the problem: Given is a set of n users with starting positions $\{p_1, p_2, \ldots, p_n\} \in P$ and a goal position g. Find the routes $r_1, r_2, \ldots, r_n\} \in R$ such that $c(MT)$ is minimized.

In the case of $s = m$ the cost function of the MTP equals the cost function of the Steiner Tree Problem. Under the assumption that it does not matter whether we exchange starting positions and goal position, hence writing $T = P \cup g$, finding meeting points in the MTP can be reduced to a Steiner Tree Problem. We are currently preparing a study to investigate whether this assumption produces a simplification that holds.

In the next section we will present two methods to solve this problem. Section 5.2 shows that our theory achieves good results when compared to meeting points from users.

4 Two approaches

4.1 Solving MTP in the street network

At the beginning of our research we interpreted MTP as an instance of a SPN. Thus, we used an extended Dreyfus-Wagner algorithm as described in [Proemel and Steger, 2002] to calculate the SMT in the network of the streets. The algorithm first computes the transitive hull of shortest paths for all pairs of nodes. This equals all SMTs for all pairs of nodes. In further steps these results are used to calculate SMTs for subsets of $3, 4, 5, \ldots$ nodes. As the first step is the same for all possible sets of terminals, this step can be precompiled.

city	nodes	shortest paths	t_h	t_r
Hamburg	3 787	7 168 791	$225.26s$	$86.49s$
Berlin	3 216	5 169 720	$114.69s$	$56.77s$
Madrid	2 496	3 113 760	$65.87s$	$33.28s$

Table 1: Time used by Dreyfus-Wagner algorithm

On clippings of maps (from OpenStreetMap) of six different cities we calculated SMTs for three terminal nodes respectively. Note that three terminals equal a MTP with two individuals who want to go to one destination. All terminal nodes were randomly picked and within walking distance to each other. We measured the time t_h of the first step from the Dreyfus-Wagner algorithm to calculate the shortest path transitive hull and the time t_r for the following steps. The results for some cities are shown in Table 1. You can clearly see the exponential increase in t_h and t_r. We also measured the times needed for problem instances with more terminals: for each additional terminal the time t_r doubled. In clippings with a practical amount of streets calculating a MT took over 30 seconds. As we wanted to construct a system with a responce time smaller two seconds we explored a second method.

4.2 Solving MTP geometrically

This method works on a relaxed version of the problem. [Wuersch and Caduff, 2005] observed that "Pedestrian navigation [...] is not confined to a network of streets, but includes all passable areas, such as walkways, squares, and open areas, within or out- side buildings.". In other words, pedestrian movement can be seen as largely independent of the street network. Inspired by their observation we neglect the actual structure of the street network in a first step. Now our problem resembles the ESP. Such, we can use for example the Melzak algorithm to estimate meeting points on a geometric basis only. Afterwards routes to these meeting points are calculated in the street network. The course of action is detailed in the following paragraphs. Our evaluation in Section 5.2 affirms that this assumption is a reasonable one to make.

First Step: Estimate Theoretical Meeting Points

To calculate meeting points in the ESP we relied on the program GeoSteiner (see Section 2.2). The runtime of GeoSteiner for our limited set of terminals is negligible. We call meeting points obtained by solving an ESP theoretical meeting points (TMPs).

Second Step: Find MPOIs

TMPs calculated in the previous step can be situated in unaccessible places such as buildings or lakes or unintuitve places such as "42 meters west of house number 14". Thus, we move these points to better locations nearby, which we call practical meeting points (PMPs). A PMP can be in front of what we call a meeting point of interest (MPOI). MPOIs can be for example restaurants, bars, bus stops, subway stations, some points-of-interest (POIs), places open to the public or large crossroads.

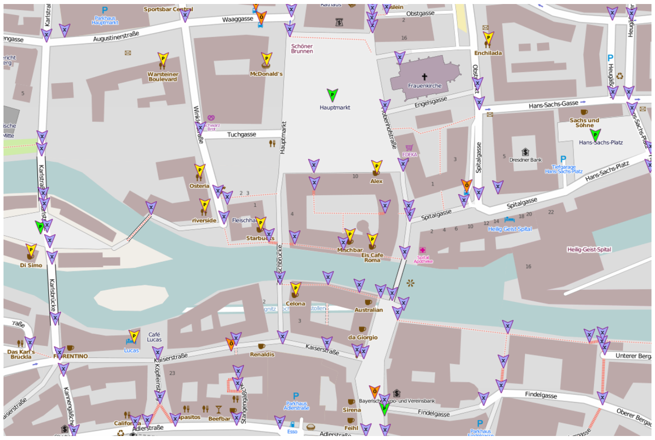


Figure 5: MPOIs in downtown Nuremberg in front of restaurants (yellow) and bus stops (orange), crossroads (blue) and public open spaces (green).

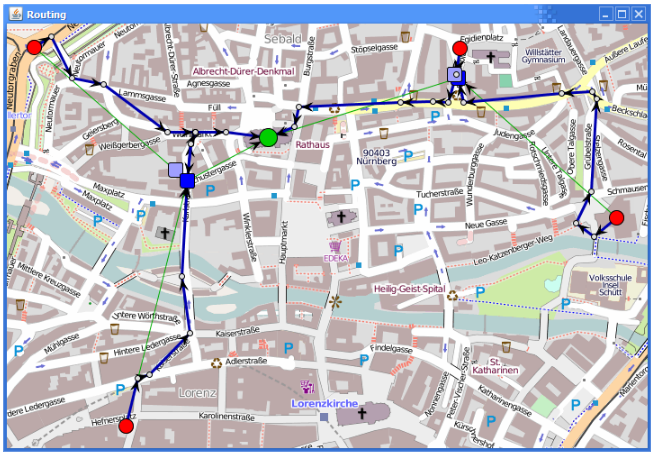


Figure 6: TMPs (light blue) PMPs (dark blue) and Routes (arrows) of four users (red) to the destination (green).

We used OpenStreetMap as source for finding MPOIs. Figure 5 shows various MPOIs in Nuremberg, city centre.

Third Step: Times and Routes

Now, we calculate walking routes in the street network for the users. Each route spans from the currents users' position through zero or more PMP to the goal.

Next, times at which users have to be at specific locations are calculated. Therefore a tree-traversal starting at the destination location labels all vertices in the meeting tree with the time, at which users have to leave when they want to arrive in time at the next vertice. The destination node is initialized with the time at which users want to arrive there, an additional time buffer can be inserted at meeting points.

Using the above tree-traversal, also the individual routes of all users can be extracted from the MT. An example for the result of these three steps is shown in Figure 6.

Note that in this figure the user from the top left corner has to walk a small stretch of way twice. This is due to our heuristic approach. But one could easily move in an additional step meeting points causing these indirections such, that walking distances back and forth are minimized.

n	$\bar{l}_N$	$\bar{l}_G$	$\lvert\bar{l}_G - \bar{l}_N\rvert = \Delta$	$\sigma(\Delta)$	$\Delta : l_N$
4	$3\,003m$	$3\,310m$	$308m$	$237m$	9.9%
5	$3\,267m$	$3\,585m$	$317m$	$192m$	10.3%
6	$3\,657m$	$4\,212m$	$554m$	$480m$	15.6%
7	$3\,224m$	$4\,863m$	$640m$	$571m$	14.6%

Table 2: Influence from ESP and SPN based approaches on meeting tree length

5 Evaluation

We conducted three studies to show the suitability of our theory on meeting points and meeting trees.

5.1 Comparing ESP and SPN

To check whether the relaxation, neglecting the street network, yields suitable results, we compared the lengths l_G of meeting trees calculated geometrically with lengths l_N of meeting trees calculated in the street network.

For various numbers of terminals we calculated meeting tree lenghts for MTPs in five cities. Average lengths for both methods are shown in table 2.

In the case of four terminals the length of geometric meeting trees are by 9.9% longer than network meeting trees. In our study this means an overall detour of 308 meters for all three individuals in the average. This results in even smaller detours for each individual. As [Helbing *et al.*, 2001] finds "[d]etours of up to about 25% seem to be acceptable to pedestrians.", detours between 9.9% and 15.6% ($12.5\%'$ in average) in contrast to the optimal route look very acceptable.

5.2 Meeting points for two persons

In a study we asked 6 participants to mark their preferred meeting points in three different situations in five citys. All situations diplayed positions of two individuals and their common destination on a map. The participants had to mark the location which they thought is the best meeting point.

The distance between the Torricelli point and the peoples choices was in the average 287 meters with a root mean square deviation of 183 meters. As the average meeting tree length in the graph in this study is 2558 meters, the avarage maximum detour for both individuals is 11.2%. This result also looks acceptable.

5.3 Subjective evaluation

Additionally we conducted a subjective evaluation of our approach. We presented six scenarios to seven participants. Each scenario consisted of a map with positions of several individuals and a position of their destination. The participants had to draw a meeting tree for each scenario. Afterwards, the probands were presented the automatically created meeting trees from our system using the geometrical approach. Now, they had to answer a questionnaire on a discrete five point Likert scale (0: full reject, 5: full accept). For "Do you think the automatically created meeting tree is a good compromise?" the rated in average 4.31, whether they find the routes practically 4.0, whether the quality of the automaticlly created meeting tree is equal to their meeting tree was answered with 2.83. We summarize that they rated the automatically created routes positively relating to compromise and practicabillity. It is plain that they still prefere their own routes, as we considered only a very limited set of aspects when calculating MTs.

6 Conclusion

Modeling the meeting behaviour of individuals is a new territory. We proposed a practical and theoretical formulation of the MTP. To solve MTP we presented and evaluated two methods which lead back to the Steiner tree problem.

The runtime of the method based on SPN is (at least when using the exact Dreyfus-Wagner algorithm) too high for our needs. For the future it would be interesting to evaluate the behaviour of approximative algorithms like [Smith *et al.*, 1981] on the MTP.

For achieving better runtime we invented another method working on ESP which neglects the street network. Heuristics of GeoSteiner guarantee short runtime. Compared to the optimal solution this method results in an average of 12.5% overall detour, which is according to [Helbing *et al.*, 2001] acceptable for pedestrians. Further, our study showed that theoretical meeting points obtained by solving ESP are in average only 287 meters away of meeting points people would choose.

As PNS4G is a new area of research there are still many open questions to answer and many aspects of meeting behaviour to be researched. One problem raised in this paper is, how to set the parameter s, which weights the costs of route parts travelled conjoint. We think that s reflects the public spirit of the individuals who are meeting. Currently we are conducting studies to estimate this parameter. Besides that we focus on integrating support for considering means of public transportation in multi user routing.

References

[Becker, 1976] G.S. Becker. *The economic approach to human behavior*. University of Chicago Press, 1976.

[Chang, 1972] S.K. Chang. The generation of minimal trees with a Steiner topology. *Journal of the ACM (JACM)*, 19(4):699–711, 1972.

[Ding *et al.*, 2008] D. Ding, J. Xu Yu, and L. Qin. Finding time-dependent shortest paths over large graphs. *EDBT Proceedings*, pages 697–706, 2008.

[Helbing *et al.*, 2001] D. Helbing, P. Molnar, I.J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B*, 28(3):361–384, 2001.

[Huang, 2007] R. Huang. A schedule-based pathfinding algorithm for transit networks using pattern first search. *Geoinformatica, Greece*, 11:269–285, 2007.

[James, 1951] J. James. A preliminary study of the size determinant in small group interaction. *American Sociological Review*, 16(4):474–477, 1951.

[Karp, 1972] R.M. Karp. Reducibility Among Combinatorial Problems. *Complexity of computer computations: proceedings*, page 85, 1972.

[Maruyama *et al.*, 2004] A. Maruyama, N. Shibata, Y. Murata, and K. Yasumoto. P-tour: A personal navigation system for tourism. *Proc. of 11th World Congress on ITS*, pages 18–21, 2004.

[Melzak, 1961] Z.A. Melzak. On the problem of Steiner. *Canad. Math. Bull*, 4(2):143–148, 1961.

[Moussad *et al.*, 2010] Mehdi Moussad, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, 04 2010.

[Proemel and Steger, 2002] H.J. Proemel and A. Steger. *The Steiner tree problem: a tour through graphs, algorithms, and complexity*. Friedrick Vieweg & Son, 2002.

[Smith *et al.*, 1981] J.M.G. Smith, DT Lee, and J.S. Liebman. An O (n log n) heuristic for Steiner minimal tree problems on the Euclidean metric. *Networks*, 11(1):23–39, 1981.

[Tumas and Ricci, 2009] G. Tumas and Francesco Ricci. Personalized mobile city transport advisory system. *ENTER Conference 2009*, 2009.

[van Setten *et al.*, 2004] M. van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244, 2004.

[Voelkel and Weber, 2008] T. Voelkel and G. Weber. Routecheckr: personalized multicriteria routing for mobility impaired pedestrians. *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 185–192, 2008.

[Warme *et al.*, 2000] D.M. Warme, P. Winter, and M. Zachariasen. Exact algorithms for plane Steiner tree problems: A computational study. *Advances in Steiner Trees*, pages 81–116, 2000.

[Winter, 1985] P. Winter. An algorithm for the Steiner problem in the Euclidean plane. *Networks*, 15(3):323–345, 1985.

[Winter, 1987] P. Winter. Steiner problem in networks: a survey. *Networks*, 17(2):129–167, 1987.

[Wuersch and Caduff, 2005] M. Wuersch and D. Caduff. Refined route instructions using topological stages of closeness. *Web and Wireless Geographical Information Systems*, pages 31–41, 2005.

[Zenker and Ludwig, 2009] B. Zenker and B. Ludwig. ROSE: assisting pedestrians to find preferred events and comfortable public transport connections. In *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*, page 16. ACM, 2009.

Towards an automatic diary: an activity recognition from data collected by a mobile phone

Rudolf Kadlec and Cyril Brom
Charles University in Prague
Faculty of Mathematics and Physics
Czech Republic
rudolf.kadlec@gmail.com, brom@ksvi.mff.cuni.cz

Abstract

We present our initial work on an "automatic diary" recognizing and storing episodes of human daily activities. The goal is to create an application that will perform activity recognition based on data collected from a mobile phone. This includes a GPS location, WiFi and Bluetooth signals. Our aim is to combine these sensory data with information from publicly available databases of points of interest thus identifying restaurants, schools etc. Until now we have collected several months of hierarchically annotated data, created a desktop viewer of the logged data and experimented with inference of activities on two different levels of abstraction. Several machine learning algorithms were tested in these experiments.

1 Introduction

Human activity recognition is a tool that enables wide range of possible applications for a healthy lifestyle [Consolvo *et al.*, 2008] , helping elderly people [Kröse *et al.*, 2008] etc. Activity recognition also fits well into the context of lifelogging [Bell and Gemmell, 2009] – continuous logging of all possible information related to a person's life. Decreasing prices of storage capacity enables us to continuously store many details of our daily lifes. We can store our location, photographs and even audio at an acceptable price. The key question is how to make the log accessible to humans. So far we can search it by time, or use full text search on recorded audio [Vemuri *et al.*, 2006]. We think that an automatic activity recognition can add a valuable key that can be used to search the lifelog in various applications.

The idea of lifelogging is fueled by current advances in mobile technologies. Smart mobile phones provide a wide range of sensors that can be used for human activity recognition. Thanks to extended battery life time it is now possible to continuously store information from GPS, WiFi, Bluetooth and accelerometer almost for a whole day. Imagine that just by wearing your Android mobile phone a summary of your daily activity could be automatically computed for you. During the day values from mobile phone sensors will be logged and at the end of the day the computer will present you several possible explanations of your todays activity. Among these explanations you will pick the one that best matches what you really did. Then you can share this information with your friends or family through Facebook or any other social networking service. Without any effort you will get statistics showing how and where you spend your time, these statistics can help you improve managing your time in the future. With the use of your diary you will be also able to better recall old episodes, e.g. a medieval castle visit last year. Recall of this episode will immediately show similar episodes of your life just like YouTube shows similar videos to the one you are just watching.

In this paper we approach the goal of enriching the lifelog by experimenting with the activity recognition on two levels of abstraction. The first is the level of atomic activities like sleep, work, watch TV etc. The second is the level of higher activities like visiting school, shopping, training etc.

The rest of the paper continues by describing the hierarchical activity representation used in our application. Then we detail the architecture of our system and procedure for collecting data. Further we review related works in the field of activity recognition. After that we will show two machine learning experiments, the first experiment focuses on low level activity recognition while the second deals with classification of longer time periods.

2 Activity representation

Findings from psychology suggest that people often perceive activities in a hierarchical way [Zacks and Swallow, 2007], e.g. an episode *Work day* can consist of a *Commute*, *Work* and again *Commute* episodes, where the *Commute* episodes can be further decomposed into *Walk*, *Travel by bus*, *Waiting at a bus stop* etc. Our system uses this hierarchical activity representation where activities can be decomposed down to atomic activities that are not further decomposable. The activity log is then a forest of trees representing high level activities, children of every activity are also activities ordered by time of their start. Each point in time has associated *activity trace* which is a trace from the high level activity down to the atomic activity.

We believe that this hierarchical activity representation will make the lifelog more accessible to a human user. Users will be able to "zoom" their activity to the level of detail that suits their needs, thereby focusing their search.

3 System architecture

The overall architecture of the system is shown in Figure 1.

1. The Android mobile phone logs GPS location and presence of WiFi and Bluetooth signals. During learning phase, user inputs activity annotation through a simple GUI. The logged data are stored in an internal SQLite database.

2. There is a set of desktop command line utilities used to extract features that are later used in the machine learning phase. For instance at this point the online geospatial database Gowalla [Gowalla, 2011] is used to add points of interest (POIs) to places detected in the movement log.

3. Machine learning is done inside the RapidMiner[1], an open source machine learning framework.

4. A GUI application is used for viewing and editing the data both from the database with the lifelog and for results of the machine learning. Figure 2 shows a screenshot of the GUI.

3.1 Implementation

Both the Android client and desktop applications were developed using Java. The desktop GUI log viewer and the editor was developed on the top of the Netbeans RCP[2]. Machine learning was performed mainly using RapidMiner. Since RapidMiner does not support Hidden Markov Model (HMM) used in some of our experiments, we used the JAHMM library[3] which implements HMM. We created a plugin[4] for RapidMiner that provides JAHMM's functionality. The plugin was released under a GNU GPL license.

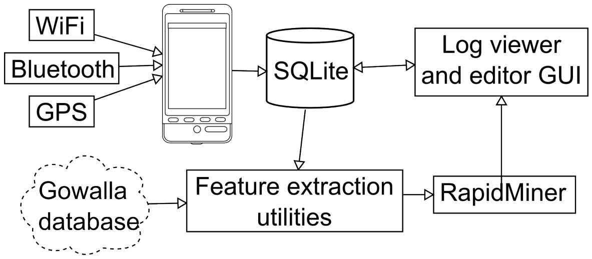
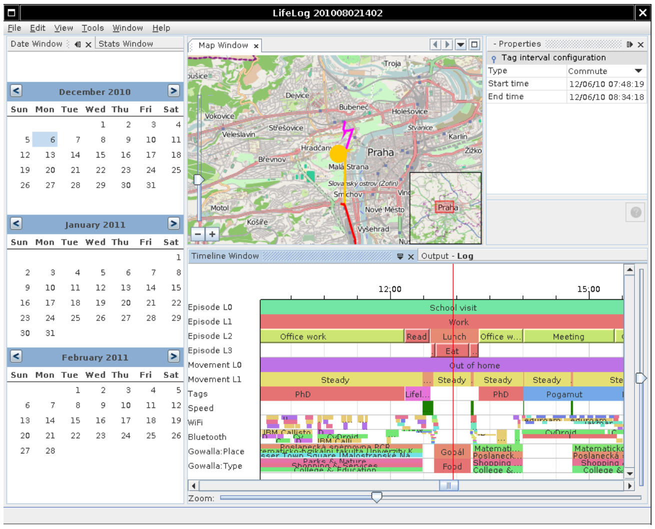


Figure 1: Architecture of the system.

4 Data collection

In this paper we present data collected by one participant between mid November 2010 and April 2011. We have other three participants who collected data over shorter time periods but each used different sets of activities. The application allowed users to mark beginnings and ends of their activities and use custom hierarchy for their description.

The GPS, WiFi and Bluetooth data was collected with period from 10 seconds to 2 minutes. Longer period was used to decrease power consumption when the phone was going out of power.



- Properties

Tag interval configuration

Type	Commute
Start time	12/06/10 07:48:19
End time	12/06/10 08:34:18

Figure 2: Screenshot of the desktop log editor. The editor contains a map view showed in the center, a timeline with activity log in the bottom and a calendar for time interval selection in the left.

5 Related work

In recent years several human activity recognition algorithms were published. They differ in sensors used as an input to the system (GPS/WiFi/accelerometer/video). Time scales considered in these recognizers vary from hours to weeks. As a formal model for activity recognition Dynamic Bayes Networks, including variants of Hidden Markov Models, are the most popular option. Table 1 shows related work published in the recent years.

Considering indoor activity recognition most of the systems use setup of accelerometers bound to specific parts of the body [Tapia, 2008; Huynh *et al.*, 2008; Stikic and Schiele, 2009]. This restriction is not applicable when considering normal mobile phone usage. Works using mobile phones without significant restriction on their usage [Lu *et al.*, 2010] predicted only classes like *walk*, *run*, *cycling* etc. [Lu *et al.*, 2010] presents system implemented in a mobile phone that is optimized for lower power consumption. [Liao *et al.*, 2007] is the closest to our work's aim. Iit features hierarchical activity representation but operates only in outdoor environments. An example of work from different domain is [Blaylock and Allen, 2006]. It performs activity recognition using hierarchy of Hidden Markov Models on data generated by a planning algorithm. Inputs of this system are purely symbolic, thus it allows for a higher level activity inference.

[1] RapidMiner 5.1 Homepage, http://sourceforge.net/projects/rapidminer, March, 2011.

[2] Netbeans RCP, Oracle Corporation, http://platform.netbeans.org, March, 2011.

[3] JAHMM Homepage, http://http://code.google.com/p/jahmm, March, 2011.

[4] The plugin is available for download at http://http://code.google.com/p/rapidminerhmm/, April, 2011.

Citation	Hier. act.	Input data	Time scale	Environment	Algorithm
[Lu *et al.*, 2010]	×	GPS, Audio, Accel.	Days	City	SVM, GMM, NB, DT
[Liao *et al.*, 2007]	√	GPS	1month	City	DBN
[Huynh *et al.*, 2008]	√	Accelerometer	1 week	Indoor	SVM, HMM, NB, LDA
[Stikic and Schiele, 2009]	×	Accel.	Days	Indoor	Multi instance SVM
[Oliver and Horvitz, 2005]	√	A/V, Keyboard, Mouse	Hours	Indoor	DBN, HMM
[Yin *et al.*, 2004]	×	WiFi	Hours	Indoor	DBN, N-gram
[Tapia, 2008]	×	Accel., heart rate	Hours	Indoor	DT, NB
[Kautz *et al.*, 2003]	√	Noisy location	3 weeks	Indoor	Hierarchical HSMM
[Blaylock and Allen, 2006]	√	Artificial symbols	5000 plans	Monroe corpus	Hierarchical HMM

Table 1: Several existing activity recognition algorithms. Shortcuts used for algorithms: SVM = Support Vector Machine, GMM = Gaussian Mixture Model, NB = Naive Bayes, DT = Decision Tree, CRF = Conditional Random Field, LDA = Latent Dirichlet Analysis, DBN = Dynamic Bayes Network, HMM = Hidden Markov Model, HSMM = Hidden Semi-Markov Model.

6 Experiments

We have made two experiments with the logged data just to test the applicability of well known machine learning algorithms oo our data. In the first experiment we tested the ability to infer low level atomic activities like sleeping, working, hygiene etc. The second experiment focuses on inference of high level activities like visiting school, friends or parents, shopping, training etc.

6.1 Low level activities inference

Method

Logged data were transformed into feature vectors $f_1, f_2 ... f_T$. Each feature vector $f_t = \langle lat_t, lon_t, speed_t, hour\_of\_day_t, w_t^1 ... w_t^m, b_t^1 ... b_t^n, g_t^1 ... g_t^o \rangle$, where $w_t^i \in \langle 0, 100 \rangle$ is a WiFi network's signal strength $w^i \in W = \{$*a WiFi network whose first and last occurrence were at least 1 week apart and it was present for at least 4 hours it total*$\}$ in time t, $b_t^j \in \{0, 1\}$ indicates presence of a Bluetooth device $b^j \in B = \{$*a Bluetooth device whose first and last occurrence were at least 1 week apart and it was present for at least 30 minutes it total*$\}$, finally $g_t^k \in \{0, 1\}$ indicates presence of a place obtained from the Gowalla database with type $k \in \{$*Travel, Food, Parks & Nature, Shopping, Entertainment, Architecture & Buildings, College & Education, Nightlife, Art*$\}$ in time t. The feature vectors were sampled at a constant rate of 1 minute. There were 30 different types of atomic actions, the actions were: *Alpine skiing, Car repair, Clean car, Concert, Cook, Cross-country skiing, Cycling, Eat, Hair cut, Hand work, Home Office, Household, Hygiene, Idle, Meeting, Other, Packing, Play games, Program, Shop, Sleep, Spinning, Strengthening, Teaching, Travel, Wait, Walk, Watch TV, Working, Writing.*

In preliminary experiments we tested a CART decision tree [Breiman *et al.*, 1984], Hidden Markov Model [Rabiner, 1989], 1-NN classifier [Hart, 1967] and a zero classifier that predicts the most probable class no matter what the sensory input is. The decision tree, the zero classifier and the k-NN were used directly on a sequence of feature vectors. In case of the HMM feature vectors were clustered using k-means clustering into 1000 and 4000 clusters used as discrete observations. Hidden states were atomic actions, matrices of observation probabilities for states and state transitions were computed directly from the data. Laplace correction was used, hence none of the probabilities was zero. A Viterbi algorithm [Rabiner, 1989] was used for inference of the most probable sequence of hidden states.

Method	Accuracy in %
Zero classifier	32.3
1NN	48.5
HMM (1000)	50.0
Decision tree	50.9
HMM (4000)	52.1

Table 2: Comparison of performance of Zero classifier, Decision tree, 1-NN and two variants of HMM

Results

Table 2 shows performance of tested algorithms. The best performing was Hidden Markov Model (HMM) with observation space clustered into 4000 observations, but its accuracy was only 52.1%

The zero classifier predicted *Sleep* that was the most frequent class with almost 8 hours of sleep a day, this lead to accuracy of 32.3%. The other three classifiers performed comparably well with accuracy around 50%. The Hidden Markov Model succeeded in capturing several temporal dependencies in the data. For example most days begin with sequence: *sleep, hygiene, eat*, which was correctly revealed by the HMM.

The class best predicted by the HMM was *Sleep* with precision of 95% and recall of 89%, class *Work* had precision 73% and recall of 70%. Other classes were predicted with significantly lower accuracy.

Discussion

The performance of 52% is not satisfactory. This could be caused by presence of too many classes and by lack of some important information in the context provided to the machine learning. Based on this results and related works we have extended the logging application with a pedometer that will be used in future experiments. Sleep was predicted relatively well because this activity was bound with a specific place that was infered from WiFi network's signals. We originally thought that inclusion of Bluetooth data will increase recognition rate of activities like *Meeting* that can be bound to presence of specific people. However due to the fact that most people have switched off the Bluetooth discovery mode of their mobile phone this does not proved to be right.

6.2 High level activities inference

Method

In this experiment we wanted to automatically label high level activities. In the first step boundaries of high level activities were identified using the GPS data. Segmentation was performed by identifying intervals when the user was at home and when he was outside the home location. Algorithm 1 was used to identify these $outOfHome$ locations. Then for each interval corresponding to one $outOfHome$ log the activity logged by the user that overlapped it best was searched and the $outOfhome$ log was labeled with the name of this activity.

After this segmentation 113 distinct activities were found, 7 of these were assigned a unique label, these were removed because there will be no data left to split them into training and testing sets. The remaining 106 activities were used for the rest of the experiment, Table 3 shows distribution of classes in the data set.

For each $outOfHome$ entry a feature vector f was constructed. $f = \langle$*the length of the interval, the distance traveled, the time spend in movement/time without movement, the time of the day when the activity started, the time of the day when the activity ended, average speed when moving, the east, west, south and north most locations in that interval,* $w_1, ..., w_o, g_1, ..., g_p\rangle$, where w_i represented WiFi networks obtained as in the previous experiment, the same applies to the g_i Gowalla places.

Algorithm 1 Movement segmentation

```
Require: $locations$ — sequence of GPS locations
1: $filteredLocations \leftarrow$ all locations from $locations$ with accuracy better than 80 meters
2: $parts \leftarrow$ identify intervals of movement and intervals without movement from $filteredLocations$, remember the location of intervals without movement
3: $averageHomeLocation \leftarrow$ find a location that occurs most often at 3 a.m. of each day from the interval, this is considered to be the home location
4: find sequences of parts from $parts$ list that begin and end near the $averageHomeLocation$, add each such sequence to the $outOfHomeList$
5: return $outOfHomeList$ %% list with entries corresponding to intervals when the user was outside the home location
```

Results

Because our dataset was relatively small and some classes were represented by very few examples we used the leave one out cross validation. This means that we always build a classification model from $n - 1$ examples and used it to predict the n-th example. Again as in the previous experimented we tested several machine learning algorithms. The best performing was the CART decision tree, the accuracy of classification was 67.92% (1-NN 42%, Naive bayes 51%). Table 4 shows confusion matrix of the best classifier.

Class name	Instances
School visit	23
Shopping	16
Training	15
Visiting friends	12
Work day	12
Weekend trip	11
Visiting parents	10
Trip	5
Visiting doctor	2

Table 3: Class distribution of the activities

Discussion

As can be seen the classifier performed relatively well. There is a mutual confusion between *School visit* and *Work day* classes because both involve traveling throuhg the same part of the city. The *Trip* and *Shopping* classes are classified relatively bad. This can be caused by high variance inside those classes, the shopping activity involved several shops and there were several different targets of trips. The data from Gowalla database did not help in this case, inspection of the learned decision trees showed that the Gowalla places were not used.

The main problem of this approach is the assumption that each $outOfHome$ segment corresponds to only one activity class. It is often the case that the segment consists of 8 hours of *School visit* followed by 30 minutes of *Shopping* and finally 2 hours of *Visiting parents*. In the current procedure this whole segment would be labeled as a *School visit*. Finer grained segmentation remains as future work.

7 General discussion and Future work

Accuracy of low level activity recognition has to be improved to match result reported in [Lu *et al.*, 2010] where mobile phones were also used to collect data. Higher level activity recognition provides better results and it is closer to use in real lifelogging applications. Future directions of work on our system include:

- Inclusion of accelerometer data — this should increase accuracy of low level action inference.
- Connection of low level and high level activities inference — high level activity recognition performed better than low level one. This could be used to create a two level recognizer where a high level activity can be used to restrict possible lower level activities thus improving the lower level classifier's performance.
- Finer grained activity segmentation — the procedure segmenting activities based on home location can be used to provide rough bounds that can be later refined by a different segmentation technique. We want to try HMM or Conditional Random Fields for this purpose.
- Inclusion of long term time dependencies — from the collected data we know that e.g. *Training* occurs twice a week whereas *Shopping* occurs usually once a week. Explanations of activity that are in accordance with this prior knowledge could be then preferred.

		True class									
		SV	VP	WT	VF	WD	TRI	TRA	SH	VD	precision
Predicted class	School visit (SV)	18	0	0	0	2	0	0	0	0	90%
	Vis. parents (VP)	0	6	0	0	0	0	0	4	0	60%
	Weekend trip (WT)	1	0	11	0	0	1	1	0	0	79%
	Vis. friends (VF)	1	0	0	8	0	0	1	1	0	73%
	Work day (WD)	3	0	0	1	7	0	0	3	0	50%
	Trip (TRI)	0	1	0	1	0	1	0	0	0	33%
	Training (TRA)	0	0	0	1	3	1	12	1	0	67%
	Shopping (SH)	0	3	0	1	0	2	1	7	0	50%
	Vis. doctor (VD)	0	0	0	0	0	0	0	0	2	100%
	recall	78%	60%	100%	67%	58%	20%	80%	44%	100%	

Table 4: Confusion matrix for the high level activity classification.

8 Conclusion

We have presented initial work on our activity recognition system built on Android mobile phones. Performance of high level activities that were segmented using GPS data is promising, however the lower level activities inference has to be improved. Besides technical issues there are also law issues regarding collecting of WiFi and Bluetooth signals. For example in Czech Republic where the data were collected it is legal to store data about presence of mobile phone's Bluetooth if the identity of a phone's owner cannot be revealed from this data. Phone owner's written permission is required otherwise.

Acknowledgments

This work was supported by the grants GA UK 21809 and GA CR 201/09/H057 and by the research project MSM0021620838 of the Ministry of Education of the Czech Republic.

References

[Bell and Gemmell, 2009] C.G. Bell and J. Gemmell. *Total recall: how the E-memory revolution will change everything*. Dutton, 2009.

[Blaylock and Allen, 2006] N. Blaylock and J. Allen. Fast hierarchical goal schema recognition. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 796. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. Wadsworth & Brooks. *Cole, Pacific Grove, California, USA*, 1984.

[Consolvo *et al.*, 2008] S. Consolvo, D.W. McDonald, T. Toscos, M.Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, et al. Activity sensing in the wild: a field trial of ubifit garden. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1797–1806. ACM, 2008.

[Gowalla, 2011] Gowalla. Gowalla homepage. `http://gowalla.com/`, 2011.

[Hart, 1967] P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[Huynh *et al.*, 2008] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 10–19. ACM, 2008.

[Kautz *et al.*, 2003] H. Kautz, O. Etzioni, D. Fox, D. Weld, and L. Shastri. Foundations of assisted cognition systems. *University of Washington, Computer Science Department, Technical Report, Tech. Rep*, 2003.

[Kröse *et al.*, 2008] B. Kröse, T. van Kasteren, C. Gibson, and T. van den Dool. Care: Context awareness in residences for elderly. In *International Conference of the International Society for Gerontechnology, Pisa, Tuscany, Italy*, pages 101–105, 2008.

[Liao *et al.*, 2007] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

[Lu *et al.*, 2010] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84. ACM, 2010.

[Oliver and Horvitz, 2005] N. Oliver and E. Horvitz. A comparison of hmms and dynamic bayesian networks for recognizing office activities. *User Modeling 2005*, pages 199–209, 2005.

[Rabiner, 1989] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Stikic and Schiele, 2009] M. Stikic and B. Schiele. Activity recognition from sparsely labeled data using multi-instance learning. *Location and Context Awareness*, pages 156–173, 2009.

[Tapia, 2008] M. Tapia. *Using machine learning for real-time activity recognition and estimation of energy expenditure*. PhD thesis, Massachusetts Institute of Technology, 2008.

[Vemuri *et al.*, 2006] S. Vemuri, C. Schmandt, and W. Bender. iRemember: a personal, long-term memory prosthesis. In *Proceedings of the 3rd ACM workshop on Continuous archival and retrival of personal experences*, pages 65–74. ACM, 2006.

[Yin *et al.*, 2004] J. Yin, X. Chai, and Q. Yang. High-level goal recognition in a wireless LAN. In *Proceedings of the national conference on artificial intelligence*, pages 578–584. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[Zacks and Swallow, 2007] J.M. Zacks and K.M. Swallow. Event segmentation. *Current Directions in Psychological Science*, 16(2):80, 2007.

Contract-Based Cooperation for Ambient Intelligence

Fatma Başak Aydemir and **Pınar Yolum**
Department of Computer Engineering
Bogazici University
34342, Bebek, İstanbul, Turkey
aydemirfb@gmail.com, pinar.yolum@boun.edu.tr

Abstract

Ambient Intelligence (AmI) describes environments that sense and react to the humans in time to help improve their living quality. Software agents are thus important in realizing such environments. While existing work has focused on individual agent's reactions, more interesting applications will take place when agents cooperate to provide composed services to humans. When cooperation is required, the environment needs mechanisms that regulate agent's interactions but also respect their autonomy. Accordingly, this paper develops a contract-based approach for offering composed services. At runtime, agents autonomously decide whether they want to enter contracts. Agents then act to fulfill their contracts. Ontologies are used to capture domain information. We apply this multiagent system on an intelligent kitchen domain and show how commitments can be used to realize cooperation. We study our application on realistic scenarios.

Keywords: Agents, commitments, ontologies

1 Introduction

Ambient Intelligence (AmI) indicates environments that are aware of and responsive to human presence. Besides various types of sensors and nanotechnology, software agents are one of the emerging technologies for AmI. Autonomous, intelligent agents are used for a wide range of tasks from searching for information to adaptive decision making [WP12, 2007]. With this aspect of it, AmI can be realized by a multiagent system. Multiagent systems are systems where multiple intelligent agents interact [Wooldridge, 2002]. These interactions are generally given a meaning using commitments, which are contracts among agents to satisfy certain properties [Singh, 1999]. Using contracts among agents regulate the interactions and enable cooperation among them.

In this paper, we propose an AmI system which consists of autonomous agents. The system is dynamic in various ways: resources can be added or consumed, agents may enter and leave the system or they can change the services they provide. We follow a user centered design focusing on the user's needs and demands [Saffer, 2006] for this system as it is consistent with the human-centric nature of the AmI systems. One of the intelligent agents represents the user of the system and it is called User Agent (UA). Other agents cooperate with UA in order to satisfy the user's needs. One distinguishing aspect is that predefined contracts, which are generated before agent interaction, do not exist in the system. Such static structures do not apply well to the dynamism of the system described above. Instead of relying on predefined contracts, relevant contracts are created in conformity with the internal states of the parties during agent interactions. The internal states of the agents are not visible to other agents and the agents decide whether or not to take part in the contracts themselves. When a contract cannot be created, it is UA's duty to establish another one that guarantees realization of the properties needed to satisfy the user.

The rest of the paper is organized as follows: Section 2 explains the advantages of the dynamically generated contracts over statically generated ones. Section 3 describes the overall system architecture and explains contract evolutions. Section 4 demonstrates the application of the system on an example domain. Section 5 studies the system over selected scenarios and Section 6 compares the system with the related work.

2 Contracts for Ambient Intelligence

A contract between agents X and Y is represented as CC(X,Y,Q,P) and interpreted as the debtor agent X is committed to bring the proposition P to the creditor agent Y when the condition Q is realized. Contracts assure that the creditor obtains the promised properties and ease the process of tracing the source of possible exceptions. In some multiagent systems, the system is designed so that the role of the agents are set, agent capabilities do not change, the resources to realize these capabilities are determined and the agents' access to these resources are unlimited. In such static environments, the contracts can be specified during compile time and agents can follow these contracts at run time. Since the system is not going to change at run time, there is no reason to attempt to generate the contracts at run time.

Consider a multiagent AmI system with UA and two other agents, Agent 1 and Agent 2. Assume that the following contracts are generated at design time and adopted by the agents:

1. CC(Agent 1, UA , Service 1 Request, Service 1)

2. CC(Agent 2, UA , Service 2 Request, Service 2)

That is, if UA requests Service 1, then Agent 1 will always provide that service. Similarly, if UA requests Service 2, then Agent 2 will always provide that service. These two contracts work well as long as the agents of the system, their capabilities, resources and the user preferences do not change.

Scarce Resources: The scenario depicted above is far from being realistic. Any change in the environment prevents the system from satisfying the user's needs. Consider the case that the resources necessary to provide the services 1 and 2 are not available any more. For example, Agent 1 may run out of Resource 1 that is fundamental to serve Service 1. So, Agent 1 fails to serve Service 1 when requested, although it is committed to serve it. This leads to an overall system failure since UA is not served a part of the service bundle. In such cases, the statically generated contracts described above are not sufficient to realize the user's preferences. Instead, the agents should decide whether or not to take part in the contracts and also they should try to generate new contracts that may help to fulfill the former ones. For scarce Resource 1 example, Agent 1 may ask for a new contract including the following commitment: CC(Agent 1, UA , Resource 1, Service 1), which means that if UA provides Resource 1, then Agent 1 can provide Service 1. If UA accepts the new proposed contract and provides Resource 1 to Agent 1, Agent 1 provides Service 1 to UA . Service 1 would not be provided if the later contract had not been generated by Agent 1 dynamically.

Dynamic Environment: In an open environment, agents may leave the system, the agents that have left the system may come back, or new agents may enter the system. When UA tries to serve a bundle, states of the agents should be considered. It is not rational to wait for a service from an agent that has already left the system, although it is committed to serve it. So, the appropriate agents should be carefully selected before agreeing on any commitment. For example, in the above scenario, Agent 1 decides to leave the system for some reason, meanwhile, a new agent, Agent 3, which offers the same services as Agent 1 enters the system. Although there is a contract agreed on with Agent 1, in order to receive Service 1, UA should make another contract with Agent 3: CC(UA , Agent 3, Service 1 Request, Service 1). If UA can dynamically create a new contract with Agent 3, it can ensure receiving Service 1.

3 Approach

We develop a contract-based multiagent system for ambient intelligence. The agents cooperate by creating and carrying out contracts that they dynamically generate at run time.

Architecture: Main components of the system are depicted in Figure 1. Agents are shown in rectangle nodes and the ontologies are shown in ellipse nodes. Line edges describe two way interaction whereas dashed edges represent access to the ontologies.

There is one UA which interacts with all of the agents in the system. UA keeps track of the user's needs and desires and tries to provide the user her preferred set of services. Elements of this set are often served by various agents, so other agents cooperate with UA to offer their services. UA usually starts the communication, however other agents are also able to make contract requests. All agents make the decision for whether or not entering in a contract themselves.

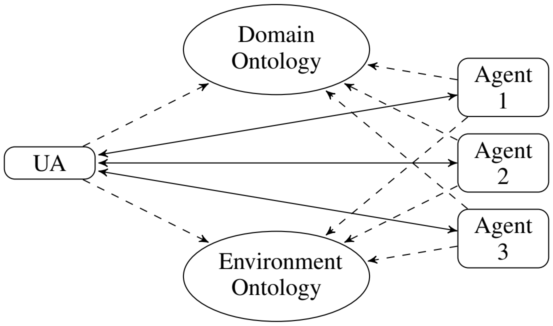


Figure 1: Architecture of the system

There are two ontologies that are accessible by all of the agents in the system. An ontology is the description of the conceptualization of a domain [Aarts, 2004]. Elements that are described in an ontology are the individuals, that are the objects of the domain; classes, that are collections of objects; attributes, that are properties of objects; relations that are the connections between objects and rules defined on these elements [Gruber, 1995].

The first ontology is the environment ontology, which describes the environment. The agent, contract and service bundle descriptions as well as additional spatial information about the environment is described in the environment ontology. Although the descriptions for the agent and contract structures are depicted in this ontology, information about individuals are not kept in here. The information not revealed in this ontology is a part of the agent's initial state and managed by the related agent itself. The second ontology is the domain ontology. In this ontology, detailed descriptions of the services and other domain dependent information are provided.

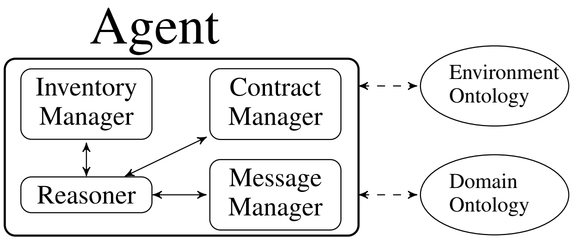


Figure 2: Architecture of an agent

Figure 2 depicts the structure of an agent. Each agent in the system has access to the environment ontology and the domain ontology. Every agent has a local inventory where it keeps the availability information on the service resources. The inventory of an agent is consulted first to decide if the necessary service resources are available. The information about the agent's inventory is private and it is not shared with the other agents of the system. The contract manager of an agent manages the contracts of the agent. It updates the contract states, traces the fulfillment of the propositions and conditions. Obviously, each agent handles its contracts itself so there is not a common contract base of the system as it is not the case in the real life. The reasoner of the agent makes the decisions, takes actions and handles messages.

Contract Lifecycle: In our system, interaction among agents is conducted via messages and it is based on contracts between two agents. Contracts are dynamic entities of the system end their states are updated by the agents after receiving or sending certain type of messages. States of contracts used in the system are:

- requested: These contracts are requested from an agent, however the reply for the request has not been received yet.
- rejected: These contracts are the ones that are requested and got a negative respond in return. They do not have any binding effect on either of the parties.
- conditional: These contracts are agreed on and created by both parties. However, their conditions and propositions remain unsatisfied.
- active: These contracts are agreed on and created by both parties. Moreover; their conditions are satisfied by the creditor.
- fulfilled: These contracts are agreed on and created by both parties. Their conditions and propositions are satisfied.

The message types used to carry these contract, their conditions and propositions are listed below:

- request: These messages are used to form a contract, thereby leading the contract to its requested state.
- reject: A reject message changes the contract state from requested to rejected.
- confirm: A confirm message updates the states of the requested contracts to conditional.
- inform: An inform message is used to fulfill the conditions of the conditional contracts (thereby, making the contract active) or the propositions of the active contracts (thereby, making the contract fulfilled).

Agent Lifecycle: Workflow diagram for UA is given in Fig. 3. When UA tries to establish contracts for a service bundle, it starts with getting addresses of the agents that provides services from the bundle. If it cannot find any agents for one or more services, bundle cannot be served (Failure). If there are agents that serve services of the bundle, UA sends them contracts requests and starts waiting for the replies. Once it receives a confirmation for a contract, it checks whether it gathers confirmation for all contracts it has requested. If there are still some contracts to be confirmed, UA continues to wait for the replies. If all of the contracts are confirmed, UA provides the conditions of the contracts. UA duty ends here as it is the other agents duty to provide the services promised and the exceptions are not in the scope of this work. If UA receives a rejection instead of a confirmation, it searches for other agents that serve the same service immediately. If there are no such agents, UA cannot provide the bundle to the user (Failure). If there are other agents serving the same service, UA repeats the process of requesting contracts. UA may also receive a contract request as a reply for its initial request.

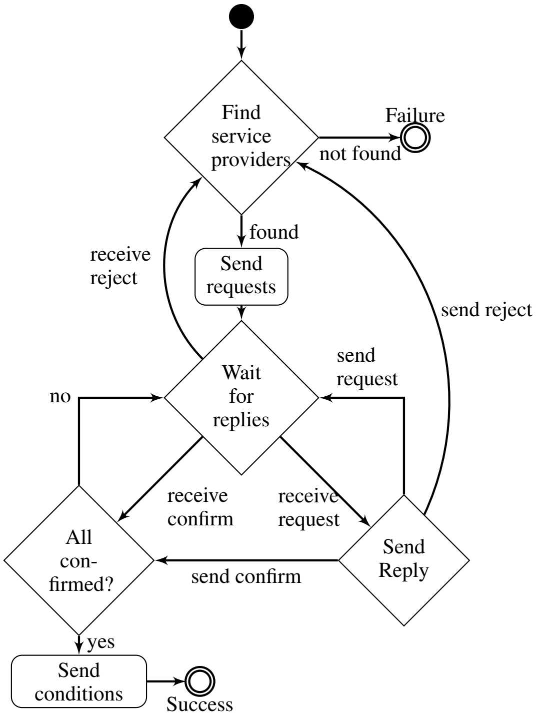


Figure 3: Workflow of User Agent

When an agent including UA receives a contract request, it should decide to create it or not. There are three possible reactions that it may take: 1) Rejecting to create the contract, 2) Creating the contract in line with the requester's desire, 3) Requesting another contract that has the same proposition as the contract requested by the requester with a different set of conditions. It is assumed that agents are willing to create contracts unless they lack necessary amount of ingredients and they do not receive any contract requests beyond their serving capabilities.

Algorithm 1 explains the behavior of an agent other than UA when it receives a request message. The message received can start a new conversation between agents, or it might carry on a previous one. So, an agent checks whether the message is part of a previous conversation or not (line 5). If the message is related to a previous contract, it retrieves the contract from its contract base and calculates the similarity between the conditions of the two contracts (line 6). If the similarity is above a threshold set by the agent itself (line 7), it confirms the contract and prepares a confirmation message to be sent to the requester via the message manager of the agent (line 8). If the similarity is below the threshold, a rejection message is prepared instead of the confirmation message (line 10). If the message is not related to any other conversation, the agent checks its inventory for the proposition (line 18). If the proposition is not ready in the inventory (line 19), for this time the agent checks the inventory for the ingredi-

Algorithm 1: Request Received

```
   Input: request:Request Message received
   Output: m:Message to send
1  String id=request.getConversationID();
2  Contract c=request.getContent();
3  boolean found=false;
4  for i ← 1 to contracts.size() do
5      if contracts(i).conversationID==id then
6          similarity=getSimilarity(c.proposition,
           contracts(i).proposition);
7          if similarity>threshold then
8              m.type ← confirm
9          else
10             m.type ← reject
11         found = true;
12         break;
13 if !found then
14     ResourceList rList=c.getProposition();
15     ResourceList missing;
16     for i ← 1 to gList.size() do
17         Resource r=rList.elementAt(i);
18         double invQ=Inventory.getResourceQuantity(r);
19         if g.RequestedQuantity > invQ then
20             missingResources(g,missing);
21             if missing.size()!=0 then
22                 m.type ← request;
23                 c.condition ← missing;
24                 m.add(c);
25                 return m
26         m.type ← confirm;
27         m.add(c);
28         return m
```

ents of the proposition. If there are some missing ingredients (line 21), the agents prepares a request message asking for the missing ingredients in return of the proposition of the contract and returns this message (lines 20-25). Otherwise, the agent prepares a confirm message (lines 26,27).

In addition to receiving a request message, an agent can also receive an inform message. If that is the case, the agent extracts the messages to get its content and finds relevant contracts through its contract manager. If it finds a contract whose condition matches the content and whose state is conditional, it updates the state to active. This means that, the agent itself is now responsible to carry out the rest of the contract by bringing about its proposition. On the other hand, if it finds a contract whose proposition matches the content and its state is active, meaning if the sending agent is fulfilling a contract, it updates its state to fulfilled.

4 Example Domain

We apply our approach on an AmI kitchen domain. An AmI kitchen consists of various autonomous agents such as Coffee Machine Agent (CMA), Tea Machine Agent (TMA), Fridge Agent (FA) and Mixer Agent (MA), which represent devices in a regular kitchen. Each of these agents provide different services. Agents use some ingredients related to their services as resources. For example, CMA, which serves coffee, has coffee beans and water in its inventory. It may also have some coffee ready in its inventory. Similarly, TMA which serves Tea is expected to have tea leaves and water. On the other hand, FA has some cake to serve. UA of the system tries to serve the user a service bundle which is a menu consisting of several beverages and dishes for this domain. Each element of a menu is usually served by a different agent of the kitchen.

The user of the system is satisfied when she gets the exact menu she prefers. Establishing contracts is a necessity in such a system for user satisfaction since the static contracts will not work for the reasons described in Section 2. Agents of the system may get broken, broken ones may be fixed or replaced, or new agents may enter the system so the assuring power of the predefined contracts established between agents is limited. The availability of the resources is limited, so the agents do not always have access to the resources they need.

The environment ontology of this system describes the agent structure, contract structure and spatial information about the kitchen such as the temperature and humidity level. The domain ontology of this environment is a food ontology, in which various types of food and beverages together with their ingredients are described. Agents use the recipes provided in the ontology for their services. In this ontology. the ingredients and types of some most popular items such as coffee and tea, are carefully classified and some similarity factor is placed between pairs that are substitutable. The similarity factor shows how well these items can substitute each other. Higher the similarity factor is, stronger the similarity relationship between the items that are compared to. These similarity factors are used to serve the demanded dish with a slightly different recipe when the original ingredients are not available in the inventory of the agent and UA cannot establish a contract that promises the missing ingredient. In such cases, the agent may try to prepare the dish using the substitude of the missing ingredient. Let's consider three types of Flour that are classified under Wheat Flour class. These types are All Purpose Flour, Cake Flour, and Bread Flour. All Purpose and Cake Flour are 0.7 similar, whereas Cake Flour and Bread Flour are 0.8 similar. When a service which requires one of these types of flour is requested, and the exact resource is not available, the resource that are similar may be substituted by one of the other types, leading to the same service served with tolerably different resources.

The detail level of a domain ontology changes from system to system. Agents of another kitchen may use a domain ontology just for the ingredients without the similarity relationship. Another one may also include the types of silverware that should be used with a specific dish.

4.1 Scenario 1

For the first scenario, user tells UA that she wants a menu consisting of two different services, coffee and cake, which should be served together. UA needs to find the agents serving the menu items, for this case they are CMA and FA . Then, UA needs to establish contracts for all of the items in

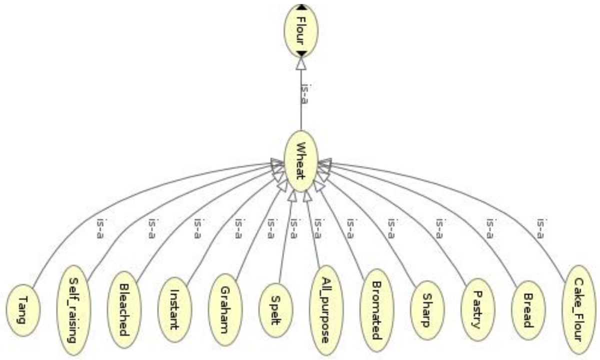


Figure 4: A caption from example domain ontology, representing flour class

the menu and receive the items. CMA needs some coffee beans to serve coffee and it manages to create a contract when it accepts to provide coffee beans to CMA . Once all contracts are established, UA fulfills the conditions of the contracts and gets served.

4.2 Scenario 2

For the second scenario, UA again tries to serve a coffee and cake menu of the user's choice. The menu item coffee is served by CMA and the cake is served by MA. UA establishes a contract with CMA. However, MA is out of cake flour which is essential for serving a cake. It requests some from UA , however UA cannot provide it and after consulting the domain ontology, UA offers bread flour, which is a replacement for the original ingredient. Once again, after all contracts are established, UA fulfills the conditions of the contracts and gets served.

4.3 Scenario 3

The third scenario begins similar to the second one. UA tries to establish contracts for the coffee and cake menu. It establishes one with CMA . MA asks for a substitute for the cake flour, which is an ingredient to make the cake. Not being able to provide the cake flour, UA offers bread flour. However, this time MA does not find the substitute similar enough to replace the original item. UA cannot establish a contract with Mixer Agent and looks for another agent that can provide cake. It discovers FA and establishes a contract with it. UA fulfills the conditions and waits for the services but CMA gets broken and does not respond.

5 Results

Jade [Bellifemine *et al.*, 1999] agent development framework is used to implement the agents, which natively provides messaging system, yellow pages and the distributed system architecture. Yellow pages service is given by Directory Facilitator (DF) Agent of each container and once the agents register their services to DF, others can find them through a query to DF. Agent implementation is separated from the underlying details of the messaging service.

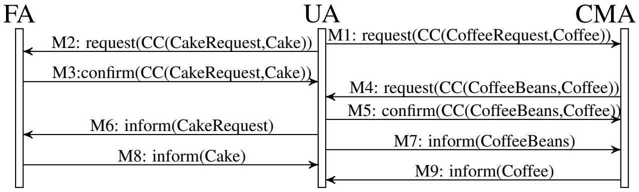

FA
UA
CMA
M1: request(CC(CoffeeRequest,Coffee))
M2: request(CC(CakeRequest,Cake))
M3:confirm(CC(CakeRequest,Cake))
M4: request(CC(CoffeeBeans,Coffee))
M5: confirm(CC(CoffeeBeans,Coffee))
M6: inform(CakeRequest)
M7: inform(CoffeeBeans)
M8: inform(Cake)
M9: inform(Coffee)

Figure 5: Sequence Diagram for Scenario 1

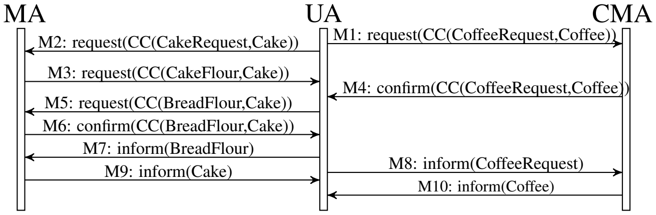

MA
UA
CMA
M1: request(CC(CoffeeRequest,Coffee))
M2: request(CC(CakeRequest,Cake))
M3: request(CC(CakeFlour,Cake))
M4: confirm(CC(CoffeeRequest,Coffee))
M5: request(CC(BreadFlour,Cake))
M6: confirm(CC(BreadFlour,Cake))
M7: inform(BreadFlour)
M8: inform(CoffeeRequest)
M9: inform(Cake)
M10: inform(Coffee)

Figure 6: Sequence Diagram for Scenario 2

5.1 Execution of Scenario 1

Figure 5 depicts the scenario described in section 4.1. For simplicity, agent names are omitted from the contracts. In order to realize the scenario, UA sends request messages to start conversation (M 1 and M 2). FA immediately sends a confirmation back (M 3). On the other hand CMA is in need of some coffee beans, so it sends a request message back (M 4). UA accepts this offer (M 5). By accepting CMA's request, UA establishes all contracts necessary to serve the menu. It sends an inform message to realize the condition of the contract with FA (M 6). It also sends an inform message to deliver the condition of the contracts with CMA (M 7). FA and CMA send the propositions of the corresponding contracts (M 8, 9).

5.2 Execution of Scenario 2

For the scenario described in Section 4.2, the flow of communication is depicted in Fig. 6. UA sends relevant request messages to start conversation (M 1 and M 2). Mixer Agent immediately makes a request for cake flour, since it does not have the necessary amount of flour to bake the cake (M 3). Unfortunately, UA cannot provide cake flour, but it consults the domain ontology for the most similar item and it finds out that it is the bread flour and luckily, it can provide bread flour, so it makes a contract request back with bread flour as condition and cake as proposition (M 5). The substitute satisfies MA and it accepts to take part in the contract (M 6). So, UA establishes all contracts that it needs to do, since CMA has already accepted the request with M 4. UA sends inform messages to both agents, satisfying the conditions of the contracts (M 7, 8). After receiving the conditions, agents serve the propositions of their contracts (M 9, 10).

5.3 Execution of Scenario 3

Communication flow between agents for the scenario described in Section 4.3 is represented in Fig. 7. The scenario starts with UA's sending contracts requests to service

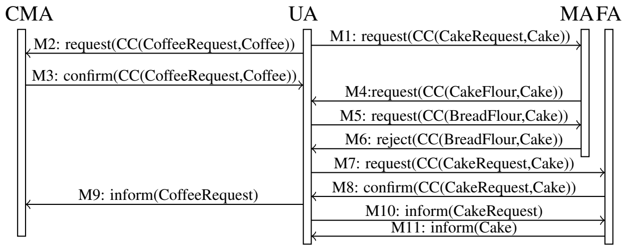

CMA
UA
MA
FA
M1: request(CC(CakeRequest,Cake))
M2: request(CC(CoffeeRequest,Coffee))
M3: confirm(CC(CoffeeRequest,Coffee))
M4:request(CC(CakeFlour,Cake))
M5: request(CC(BreadFlour,Cake))
M6: reject(CC(BreadFlour,Cake))
M7: request(CC(CakeRequest,Cake))
M8: confirm(CC(CakeRequest,Cake))
M9: inform(CoffeeRequest)
M10: inform(CakeRequest)
M11: inform(Cake)

Figure 7: Sequence Diagram for Scenario 3

providers MA and CMA (M 1, 2). CMA sends a confirmation (M 3) whereas MA requests another contract, demanding cake flour to provide cake (M 4). Unable to provide cake flour, UA requests yet another contract, offering bread flour to get some cake from MA (M 5). MA does not find bread flour similar enough to cake flour, so it rejects the contract offered by UA (M 6). UA searches for another agent that can provide cake service, so it discovers FA. It makes a request (M 7) and receives confirmation in return (M 8). With this confirmation, UA gets confirmation for all contracts to get services for the cake and coffee bundle, so it fulfills the conditions of the contracts (M 9, 10). FA provides the service it is committed to serve (M 11), however CMA gets broken and cannot provide the service. After a certain time, UA gives up hope on CMA and starts looking for a new agent to provide the same service.

6 Discussion

The main contribution of our work is to dynamically generate and use contracts to ensure that the user's needs are satisfied in a dynamic environment. Unlike Hagras *et al.*, we do not assume that any agent serving a person must always and immediately carry out any requested actions [Hagras *et al.*, 2004]. Instead, we develop a model for an open dynamic system where the continuity of the services are secured, even when some agents stop working or leave the system, without being noticed by the user.

Although it is assumed that the agents are willing to cooperate under certain conditions in Section 5, the model which is represented in this paper does not have a predefined communication protocol. The existence of less or more cooperative agents in the system does not destroy the system's ability to operate. We can also say that the agents in the system do not have designated roles, as they can change the services provided by them.

We benefit from the ontologies to achieve a high degree of interoperability; however, the contracts that are generated in the system are not kept in ontologies like in the case of Fornara and Colombetti [Fornara and Colombetti, 2010]. Since the evolution of the contracts are handled by the agents, our model deliberately lacks a central monitoring system, which has access the information on all of the transactions. Hence, contracts are also kept independently.

Unlike some AmI frameworks such as Amigo [Thomson *et al.*, 2008], our application does not offer a low level interoperation structure. In Amigo framework agents do not have any options but to provide their services when their relevant methods are called by the other agents. Also, in that framework, exact structure of the service methods of the provider agent such as the parameters and the name and so on should be known by the demanding agent. Instead of such framework, we provide a high level interaction model where agents willingly provide their services or not. It is not necessary for the demanding agent to know the details about the provider agent's methods.

Future work may include the development of a policy for exceptions. The sanctions that will be applied to an agent that does not follow a contract should be set to avoid the abuse of the system. Also, the cancellation and release policies for agents should be defined, so that the agents can inform the other party when they cannot deliver the services they are committed to.

Acknowledgement

This research has been supported by Bogazici University Research Fund under grant BAP5694. The first author is supported by TÜBİTAK National Graduate Scholarship (2210).

References

Emile Aarts. Ambient intelligence: A multimedia perspective. *IEEE Multimedia*, 11:12–19, 2004.

Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE–A FIPA-compliant agent framework. In *Proceedings of PAAM*, volume 99, pages 97–108, 1999.

Nicoletta Fornara and Marco Colombetti. Ontology and time evolution of obligations and prohibitions using semantic web technology. *Declarative Agent Languages and Technologies VII*, pages 101–118, 2010.

Thomas Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, November 1995.

Hani Hagras, Victor Callaghan, Martin Colley, Graham Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *Intelligent Systems, IEEE*, 19(6):12–20, 2004.

Dan Saffer. *Designing for Interaction:Creating Smart Applications and Clever Devices*. Peachpit Press, 2006.

Munindar P. Singh. An ontology for commitments in multiagent systems. *Artificial Intelligence and Law*, 7(1):97–113, 1999.

Graham Thomson, Daniele Sacchetti, Yérom-David Bromberg, Jorge Parra, Nikolaos Georgantas, and V. Issarny. Amigo Interoperability Framework: Dynamically Integrating Heterogeneous Devices and Services. *Constructing Ambient Intelligence*, pages 421–425, 2008.

Michael J. Wooldridge. *An introduction to multiagent systems*. Wiley, 2002.

WP12. *D12.2: Study on Emerging AmI Technologies*. Future of Identity in the Information Society Consortium, www.fidis.net., October 2007.

Using data mining approaches for sustainable elderly care

Bogdan Pogorelc, Matjaž Gams
Jožef Stefan Institute & Špica International d.o.o.
Ljubljana, Slovenia
{bogdan.pogorelc, matjaz.gams}@ijs.si

Abstract

Nowadays, the percentage of elderly people is increasing. Consequently, there is not enough younger people to take care of them. To provide sustainable elderly care we propose health care monitoring system, based on data mining approach. The aim of this study is to provide health monitoring system to allow quality and safe living of elderly at homes instead of needing them to go to nursing homes, which are also overcrowded. Moreover, their offspring would not be overwhelmed with care for the elderly. Therefore, the aim of this research is to provide sustainable elderly care. The study proposes a general and specific approach, both achieving classification accuracies over 97%.

1 Introduction

Nowadays, the percentage of elderly people is increasing [Toyne, 2003]. Elderly tend to lead an isolated life away from their offspring; however, they may fear being unable to obtain help if they are injured or ill. During the last decades, this fear has generated research attempts to find assistive technologies for making living of elderly people easier and independent. The aim of this study is to provide ambient assistive living services to allow quality and safe living of elderly at home instead of needing them to go to nursing homes, which are overcrowded. Moreover, their offspring or other relatives would not be overwhelmed with care for the elderly. Therefore, the aim of this research is to provide sustainable elderly care.

We propose two approaches to an intelligent and ubiquitous care system to recognize a few of the most common and important health problems of the elderly, which can be detected by observing and analyzing the characteristics of their movement. In the first approach we use medically defined attributes and support vector machine classification into five health states: healthy, with hemiplegia (usually the result of stroke), with Parkinson's disease, with pain in the leg and with pain in the back [Pogorelc, B. and Gams, M. 2010].

In the second approach we classify into same five health states using more general data mining approach. The movement of the user is captured with the motion capture system, which consists of the tags attached to the body, whose coordinates are acquired by the sensors situated in the apartment. Output time series of coordinates are modeled with the proposed data mining approaches in order to recognize the specific health problem. In the case that health problem is recognized, the medical center is notified.

2 Related Work

In the related work, motion capturing is usually done with inertial sensors [Strle and Kempe, 2007; Bourke et al, 2006], computer vision and also with specific sensor for measurement of angle of joint deflection [Ribarič and Rozman, 2007] or with electromyography [Trontelj et al, 2008]. For our study, the (infra-red) IR camera system with tags attached to the body [eMotion, 2009] was used.

We do not address the recognition of activities of daily living such as walking, sitting, lying, etc. and detection of falling, which has already been addressed [Confidence, 2009; Luštrek, and Kaluža, 2009], but more challenging recognition of health problems based on motion data.

Using similar motion capture system as in our approach the automatic distinguishing between health problems such as hemiplegia and diplegia is presented [Lakany, 2008]. However, much more common approach to recognition of health problems is capturing of movement which is later examined by medical experts by hand [Ribarič and Rozman, 2007; Craik and Oatis, 1995; Moore et al, 2006]. Such approach has major drawback in comparison to ours, because it needs constant observation from the medical professionals.

The paper [Miskelly, 2001] presented a review of assistive technologies for elderly care. The first technology consists of a set of alarm systems installed at person's homes. A system includes a device in the form of mobile phone, pendant or chainlet that has an alarm button. They are used to alert and communicate with the warden. When the warden is not available, the alert is sent to the control centre. However, such devices are efficient only if the person recognizes an emergency and has the physical and mental capacity to press the alarm button.

The second technology presented in [Miskelly, 2001] is video-monitoring. The audio-video communication is done in real-time over the ordinary telephone line. The video can be viewed on monitor or domestic television. The problems of the presented solution are ethical issues, since the elderly users don't want to be monitored by video [Confidence, 2009]. Moreover, such approach requires constant attention of the emergency center.

The third technology in [Miskelly, 2001] is based on health monitors. The health monitor is worn on the wrist and continuously monitors pulse, skin temperature and movement. At the beginning of the system usage, the pattern for the user is learned. Afterwards, the deviations are detected and alarms are sent to the emergency centre. Such system detects collapses, faints, blackouts etc.

Another presented technology is the group of fall detectors. They measure the accelerations of the person with the tags worn around the waist or the upper chest. If the accelerations exceed a threshold during a time period, an alarm is raised and sent to the community alarm service. Bourke et al. [Bourke et al, 2007] present the acceleration data produced during the activities of daily living and during the person falls. The data was acquired by monitoring young subjects performing simulated falls. In addition, elderly people have performed activities of daily living. By defining the appropriate threshold they can distinguish between the accelerations during the falls and the accelerations produced during normal activities of daily living. Therefore, the accelerometers with the threshold can be used for monitoring elderly people and recognizing falls. However, threshold based algorithms produce mistakes, for instance fast standing up from/sitting down on the chair could result in crossing the threshold which is erroneously recognized as a fall.

In [Rudel, 2008], architecture of a system that enables the control of the users at their homes is described. It consists of three levels. The first level represents the ill persons at their homes equipped with communication and measurement devices. The second level represents information and communication technology that enables the communication with the main server. The last level represents the telemedicine center including duty operator, doctors and technical support; the centre for the implementation of direct assistance at home; and team of experts for implementing telemedicine services. Such system does not provide any automatic detection of an unusual behavior but instead requires constant observation by the medical center.

Williams et al. [2003] have showed that the ability to perform daily activities is decreased for the people that have fallen several times and that the decrease can be detected using accelerometers. They have tested elderly people that have not fallen yet and those that have fallen several times. All of them were asked to perform a predefined scenario including sentence writing, objects picking etc. The accelerations differ significantly between the two groups of people during the test.

The aim of this paper is to realize an automatic classifier able to support autonomous living of elderly by detecting health problems recognizable through the movement. Earlier works (e.g. [Kaluza et al, 2010]) describe machine learning techniques employed to analyze activities based on the static positions and recognized postures of the users. Although that kind of approaches can leverage a wealth of machine-learning techniques, they fail to keep into account the dynamics of the movement.

3 Materials and Methods

3.1 Targeted Activities and Health Problems for Detection

The research is comparing the specific and the more general approach to recognition of health problems. It classifies walking patterns into five different health states; one healthy and four unhealthy. All the health problems we are recognizing were suggested by the collaborating medical expert on the basis of occurrence in the elderly aged 65+, the medical significance and the feasibility of their recognition from movements.

The following four health problems were chosen as the most appropriate [Craik and Oatis, 1995]:

- Parkinson's disease: a degenerative disease of the brain (central nervous system) that often impairs motor skills, speech, and other functions. The symptoms are frequently tremor, rigidity and postural instability. The rate of the tremor is approximately 4–6 Hz. The tremor is present when the involved part(s), usually the arms or neck, are at rest. It is absent, or diminished with sleep, sedation, and when performing skilled acts.
- Hemiplegia: is the paralysis of the arm, leg and torso on the same side of the body. It is usually the result of a stroke, although diseases affecting the spinal cord and the brain are also capable of producing this state. The paralysis hampers movement, especially walking, and can thus cause falls.
- Pain in the leg: resembles hemiplegia in that the step with one leg is different from the step with the other. In the elderly this usually means pain in the hip or in the knee.
- Pain in the back: There is also a similarity to hemiplegia and pain in the leg in the inequality of steps; however, the inequality is not as pronounced as in walking with pain in the leg.

Classification was done using i) medically defined attributes and SVM classifier and ii) the k-nearest neighbor machine learning algorithm and dynamic time warping for the similarity measure.

3.2 Features for Data Mining

The recordings consisted of the position coordinates for the 12 tags worn on the shoulders, the elbows, the wrists, the hips, the knees and the ankles, sampled with 10 Hz. The tag coordinates were acquired with a Smart IR motion-capture system with a 0.5-mm standard deviation of noise. From the motion capture system we get position of each tag in x-y-z coordinates.

In the first (specific) approach using medically defined attributes 13 attributes were defined with help of an medical expert. These are:

I. Absolute difference between i) the average distance between the right elbow and the right hip and ii) the average distance between the right wrist and the left hip

II. Average angle of the right elbow

III. The quotient between the maximum angle of the left knee and the maximum angle of the right knee

IV. Difference between the maximum and minimum angle of the right knee

V. Difference between the maximum and minimum height of the left shoulder

VI. Difference between the maximum and minimum height of the right shoulder

VII. Quotient between i) the difference between the maximum and minimum height of the left ankle and ii) the maximum and minimum height of the right ankle

VIII. Absolute difference between i) the difference between the maximum and minimum speeds (magnitudes of velocity) of the left ankle and ii) the difference between the maximum and minimum speeds of the right ankle

IX. Absolute difference between i) the average distance between the right shoulder and the right elbow and ii) the average distance between the left shoulder and the right wrist

X. Average speed (magnitude of velocity) of the right wrist

XI. Frequency of the angle of the right elbow passing the average angle of the right elbow

XII. Average angle between i) the vector between the right shoulder and the right hip and ii) the vector between the right shoulder and the right wrist

XIII. Difference between the average height of the right shoulder and the average height of the left shoulder

We compared the specific approach with the general approach where movements were represented with more general attributes. The advantage of latter approach is that we can add some new health state(s) to be recognized using the same algorithm and attributes.

Considering the abovementioned, in the general approach we designed attributes as the angles between adjacent body parts. The angles between body parts that rotate in more than one direction are expressed with quaternions:

- $q^t{}_{SL}$ and $q^t{}_{SL}$ ... left and right shoulder angles with respect to the upper torso at the time t
- $q^t{}_{HL}$ and $q^t{}_{HR}$ ... left and right hip angles with respect to the lower torso
- $q^t{}_{T}$ ... the angle between the lower and upper torso
- $\alpha^t{}_{EL}$, $\alpha^t{}_{ER}$, $\alpha^t{}_{KL}$ and $\alpha^t{}_{KR}$ ... left and right elbow angles, left and right knee angles.

3.3 Dynamic Time Warping

We will present dynamic time warping (DTW) as a robust technique to measure the “distance” between two time series [Keogh and Ratanamahatana, 2005]. Dynamic Time Warping aligns two time series in the way some distance measure is minimized (usually Euclidean distance is used). Optimal alignment (minimum distance warp path) is obtained by allowing assignment of multiple successive values of one time series to a single value of the other time series and therefore DTW can also be calculated on time series of different lengths. Figure 1 shows examples of two time series and value alignment between them for Euclidean distance (left) and DTW similarity measure (right).

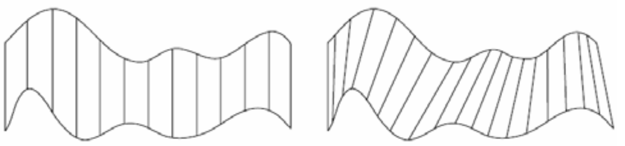


Figure 1: Example of two time series. Lines between time series show value alignment used by Euclidean distance (left) and Dynamic Time Warping similarity measure (right).

Notice that the time series have similar shapes, but are not aligned in time. While Euclidean distance measure does not align time series, DTW does address the problem of time difference. By using DTW, optimal alignment is found among several different warp paths. This can be easily represented if two time series *A = (a₁, a₂, ..., aₙ)* and *B = (b₁, b₂, ..., bₘ)*, $a_i, b_j \in R$ are arranged to form a n-by-m grid. Each grid point corresponds to an alignment between elements $a_i \in A$ and $b_j \in B$. A warp path $W = w_{1,} w_2, ..., w_k, ...w_K$ is a sequence of grid points, where each w_k corresponds to a point $(i, j)_k$ – warp path *W* maps elements of sequences *A* and *B*.

A warp path is typically subject to several constraints:

- Boundary conditions: $w_1 = (1,1)$ and $w_k = (n,m)$. This requires the warping path to start in first point of both sequences and end in last point of both sequences.
- Continuity: Let $w_k = (a,b)$ then $w_k - 1 = (a',b')$ where $a - a' \le 1$ and $b - b' \le 1$. This restricts the allowable steps in the warping path to adjacent cells.
- Monotonicity: Let $w_k = (a,b)$ then $w_k - 1 = (a',b')$ where $a - a' \ge 0$ and $b - b' \ge 0$. This forces the points in W to be monotonically spaced in time.

From all possible warp paths DTW finds the optimal one:

$$DTW(A,B) = min_W[\sum_{k=1}^{K} d(w_k)]$$

Here *d(w$_k$)* is the distance between elements of time series.

Algorithm The goal of DTW is to find minimal distance warp path between two time series. Dynamic programming can be used for this task. Instead of solving the entire problem all at once, solutions to sub problems (sub-series) are found and used to repeatedly find the solution to a slightly larger problem. Let *DTW(A, B)* be the distance of the optimal warp path between time series $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_m)$ and let $D(i, j) = DTW(A', B')$ be the distance of the optimal warp path between the prefixes of the time series *A* and *B*:

$$D(0,0) = 0$$
$$A' = (a_1, a_2, ..., a_i), B' = (b_1, b_2, ..., b_j)$$
$$0 \leq i \leq n, 0 \leq j \leq m$$

Then *DTW(A, B)* can be calculated using the following recursive equations:

$$D(0,0) = 0$$
$$D(i,j) = min(D(i-1,j), D(i,j-1), D(i-1,j-1)) + d(a_i, b_j)$$

Here $d(a_i, b_j)$ is the distance between two values of the two time series (usually Euclidean distance is used). The most common way of calculating *DTW(A, B)* is to construct a *n*m* cost matrix *M*, where each cell corresponds to the distance of the minimal distance warp path between the prefixes of the time series *A* and *B* (Figure 2):

$$M(i,j) = D(i,j)$$
$$1 \leq i \leq n$$
$$1 \leq j \leq m$$

We start by calculating all the fields with small indexes and then progressively continue to calculate fields with higher indexes:

```
  for i = 1...n
    for j = 1...m
      M(i,j)  =  min(M(i-1,j),  M(i,j-1), M(i,j)) + dst(ai,bj )
```

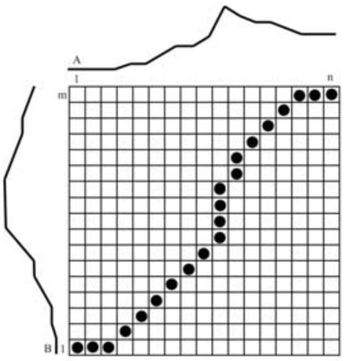


Figure 2: Minimal distance warp path between time series A and B.

The distance corresponding to the minimal distance warp path equals the value in the cell of a matrix M with the highest indexes *M(n,m)*. A minimal distance warp path can be obtained by following cells with the smallest values from *M(n,m)* to *M(1, 1)* (in Figure 2 the minimal distance warp path is marked with dots).

Many attempts to speed up the DTWs have been proposed [Salvador and Chan, 2007] which can be categorized as constraints. Constraints limit a minimum distance warp path search space by reducing allowed warp along time axis. Two most commonly used constraints are Sakoe-Chiba Band [Sakoe and Chiba, 1978] which we used and Itakura Parallelogram [Itakura, 1975] which are shown in Figure 3.

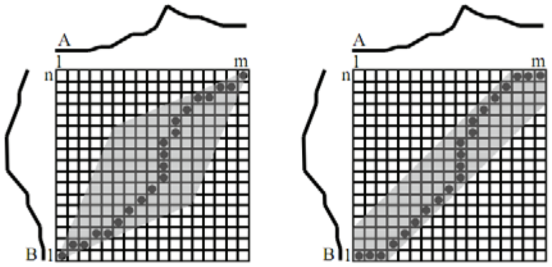


Figure 3: Itakura Parallelogram (left) and Sakoe-Chiba Band (right) constraints. Only shaded cells are used by DTW algorithm.

3.4 Modification of the Algorithm for Multidimensional Classification

The DTW algorithm commonly described in the literature is suitable to align one-dimensional time series. This work employed a modification of the DTW which makes it suitable for multidimensional classification.

First, each time point of the captured time series consisting of the positions of the 12 tags coming out of motion capture system is transformed into angle attribute space, as defined in this paper. The classification will then be performed in the transformed space.

To align an input recording with a template recording (on which the classifier was trained), we first have to compute the matrix of local distances, *d(i,j)*, in which each element *(i, j)* represents the local distance between the *i*-th time point of the template and the input at the time *j*. Let C_{js} be a generic feature vector element relative to a template recording, and Q_{is} be the feature vector element relative to a new input recording to recognize, where $1 \leq s \leq N$ is the considered feature.

For the definition of the local distance the Euclidean distance was used, defined as follows:

$$d_{Euc} = \sum_{s=1}^{N}(C_{js} - Q_{is})$$

Given the matrix of local distances a matrix of global distances *D* is built. The value of the minimum global distance for the complete alignment of DTW procedure, i.e. the final algorithm output, is found in the last column and row, $D(T_r, T_r)$. The optimal alignment can also be efficiently found by back tracing through the matrix: the alignment path starts from $D(T_r, T_r)$, then it proceeds, at each step, by selecting the cell which contains the minimum cumulative distance, between those cells consented by the alignment path constraints, until $D(1, 1)$ is reached.

4 Results

In the first (specific) approach, for each recording attributes were calculated and SVM classifier was used to classify them into five health states. Confusion matrix, which represents number of examples of a certain true class (in rows) classified in one of possible five classes (in columns), is shown in Table 1.

In the second (general) approach, the DTW algorithm was used to stretch and compress an input time series in order to minimize a suitably-chosen distance measure from a given template. We used a nearest neighbor classifier based on this distance measure to design the algorithm as a health state classifier.

The classification process is considering one input time series, comparing it with the whole set of templates, computing the minimum global distance for each alignment and assuming that the input recording is in the same class of the template with which the alignment gives the smallest minimum global distance (analogous to instance-based learning). Confusion matrix is shown in Table 2.

The 10-fold cross-validation for 5-nearest neighbor classifier resulted in classification accuracy of 97.9% and 97.6% for the specific and the general approach, respectively. Thus, the performance of both approaches is similar.

For the real world cases, we can use confusion matrices for three purposes:

- False positives (false alarms): How many can be expected using these classifiers. When in real world use the system would report false alarm, e.g., normal walking is classified as some health problem, ambulance could drive to pick up the elderly which would cause unnecessary costs.
- False negatives: How many can be expected using these classifiers. False negatives could mean potentially risky situation for the elderly, as his/her health problem would not be recognized automatically.
- Errors (misclassifications): Between which health states (classes) the errors (misclassifications) occurs. Consequently, we can add additional features to help distinguish between those particular classes. The misclassifications happened very rarely.

		classified as				
		H	**L**	**N**	**P**	**B**
true class	**H**	45	0	0	0	0
	L	1	24	0	0	0
	N	0	0	25	0	0
	P	2	0	0	23	0
	B	0	0	0	0	21

Table 1. Confusion matrix for the first (specific) approach, where H=hemiplegia, L=pain in the leg, N=normal (healthy) walking, P=Parkinson's disease and B=Pain in the back. Numbers denote quantity of the classified examples.

		classified as				
		H	**L**	**N**	**P**	**B**
true class	**H**	42	2	1	0	0
	L	0	25	0	0	0
	N	1	0	24	0	0
	P	0	0	0	25	0
	B	0	0	0	0	21

Table 2. Confusion matrix for the second (general) approach, where H=hemiplegia, L=pain in the leg, N=normal (healthy) walking, P=Parkinson's disease and B=Pain in the back. Numbers denote quantity of the classified examples.

The results show that in both proposed approaches false positives/negatives are very rare, i.e., they would not cause much unnecessary ambulance costs. Since the method accurately classified most true health problems, it represents high confidence and safety for the potential use in elderly care.

5 Conclusion

This paper proposed elderly health monitoring system providing sustainable elderly care. It presented comparison between specific and general approach to detection of health problems of the elderly. In the specific approach, medically defined attributes and SVM classifier was used. In the general approach, k-nearest neighbor algorithm with multidimensional dynamic time warping was employed. Both approaches classify movement of elderly person into five health states; one healthy and four unhealthy. Even though the first approach is more general and can be used also to classify other types of activities or health problems, it still achieves high classification accuracies, similar to the more specific approach. Since both approaches to health monitoring system are achieving high classification accuracies for recognition of health problems which are also causes for falls, they have advantage over often presented fall detection systems in sense that they recognize health problems in their early stages and also prevent falls.

Acknowledgments

Operation leading to this paper is partially financed by the European Union, European Social Fund. Authors thanks Martin Tomšič, Bojan Nemec and Leon Žlajpah for their help with data acquisition and Anton Gradišek for lending his medical expertise.

References

[Bourke *et al.*, 2007] Bourke, A. K., O'Brien, J. V., Lyons, G. M. 2007. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm, Gait&Posture 2007; 26:194-99.

[Bourke *et al.*, 2006] Bourke, A.K. et al. 2006. An optimum accelerometer configuration and simple algorithm for accurately detecting falls. In Proc. BioMed 2006 (2006), 156–160.

[Confidence Consortium 2011] Confidence Consortium 2011. Ubiquitous Care System to Support Independent Living. http://www.confidence-eu.org.

[Craik and Oatis, 1995] Craik, R. and Oatis C. 1995. Gait Analysis: Theory and Application. Mosby-Year Book (1995).

[eMotion, 2010] eMotion. 2010. Smart motion capture system. http://www.emotion3d.com/smart/smart.html.

[Itakura, 1975] Itakura, F. 1975. Minimum prediction residual principle applied to speech recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on 23(1):67–72.

[Kaluza *et al.,* 2010] Kaluza, B., Mirchevska, V., Dovgan, E., Lustrek, M., Gams, M., 2010. An Agent-based Approach to Care in Independent Living, International Joint Conference on Ambient Intelligence (AmI-10), Malaga, Spain

[Keogh and Ratanamahatana, 2005] Keogh, E. and Ratanamahatana, C. A. 2005. "Exact indexing of dynamic time warping," Knowl. Inf. Syst., vol. 7, no. 3, pp. 358–386, 2005.

[Lakany, 2008] Lakany, H. 2008. Extracting a diagnostic gait signature. Patt. recognition 41(2008), 1627–1637.

[Luštrek and Kaluža, 2009] Luštrek, M., and Kaluža, B. 2009. Fall detection and activity recognition with machine learning. Informatica 33, 2 (2009).

[Miskelly, 2001] Miskelly, F. G. 2001. Assistive technology in elderly care. Age and Ageing 2001; 30:455-58.

[Moore *et al.*, 2006] Moore, S.T., et al., 2006. Long-term monitoring of gait in Parkinson's disease, Gait Posture (2006).

[Pogorelc and Gams, 2010] Pogorelc, B. and Gams, M. 2010. Identification of Gait Patterns Related to Health Problems of Elderly. UIC 2010: 179-191.

[Ribarič and Rozman, 2007] Ribarič S. and Rozman J. 2007. "Sensors for measurement of tremor type joint movements", MIDEM 37(2007)2, pp. 98-104.

[Rudel, 2008] Rudel, D. 2008. Health at home for elderly. Infor Med Slov 2008; 13(2):19-29.

[Sakoe and Chiba, 1978] Sakoe, H., and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on 26(1):43–49.

[Salvador and Chan, 2007] Salvador, S., and Chan, P. 2007. Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. 11(5):561–580.

[Strle and Kempe, 2007] Strle D., Kempe V., 2007. "MEMS-based inertial systems", MIDEM 37(2007)4, pp. 199-209.

[Toyne, 2003] Toyne, S. 2003. Ageing: Europe's growing problem. BBC News, http://news.bbc.co.uk/2/hi/business/2248531.stm.

[Trontelj *et al.*, 2008] Trontelj, J. et al. 2008. "Safety Margin at mammalian neuromuscular junction – an example of the significance of fine time measurements in neurobiology", MIDEM 38(2008)3, 155-160.

[Williams *et al.*, 2003] Williams, M.E. et al. 2003. A new approach to assessing function in elderly people. Trans Am Clin Clim Ass 2003;114:203-16.

Semantic and architectural approach for Spatio-Temporal Reasoning in Ambient Assisted Living

Lyazid Sabri, Abdelghani Chibani,Yacine Amirat, Gian Piero Zarri
LISSI Laboratory, UPEC university
France
{lyazid.sabri,chibani,amirat,gian-piero.zarri}@u-pec.fr

Abstract

Ambient Assisted Living (AAL) systems are developed to facilitate the daily lives of elderly people, increase their autonomy and improve their safety. The symbolic reasoning about space and time is a fundamental aspect of these systems for recognizing situations and providing customized assistive services. In this paper, we propose a semantic and architectural approach for the Spatio-Temporal Reasoning in AAL systems. This approach combines first order reactive reasoning, and narrative representation and reasoning. The paper presents the narrative semantic representation foundations and the narrative inference architecture. An AAL scenario inspired from everyday life situations is presented and analyzed to show the feasibility of the proposed approach.

1 Introduction

AAL systems are developed to facilitate the daily lives of elderly people, increase their autonomy and improve their safety. Such systems must be capable for instance to detect any critical event such as a cardiovascular attack, a fall or an intruders presence, and react accordingly. AAL systems aim also at assisting the elderly anytime and anywhere by providing customized services; for instance, switching the light on in a given location when the presence of a person is detected. Situations cited above involve an immediate reaction under the form of an alarm sending or an action on a device. However, there are numerous daily activities and situations that obey to complex processes and in which multiple events must be correlated in respect to their spatial and temporal ordering to infer the right situation. Inferring deterioration in the physical and psychological status of an elderly person from changes in his habitual schedule or her physical activity is a typical case illustrating the complexity of spatio-temporal modeling and reasoning at a high semantic level about situations.

To address these challenges, the conceptual representation of the entities in AAL systems must be powerful enough to supply a general description not only of the usual classes of static objects such as a phone or a chair, but also of dynamic entities like events, actions, situations, circumstances, etc. The representation of static entities like physical objects and simple events can be kept relatively simple and based, e.g., on the traditional binary model. In this last approach, the properties or attributes that define a given concept are then expressed as binary (i.e., linking only two arguments) relationships of the property/value type, independently from the fact that these relationships are organized, e.g., into frame format as in the original Protégé model, or take the form of a set of 'property' statements defining a class in a W3C language like OWL or OWL 2. However, more advanced forms of knowledge representation should be used to represent and reason about dynamic events and situations. For instance, in [Martinez et al., 2007], the authors presented a mobile robot using the W3C reasoner RACER. However, the ontology that was used is static and very limited. It permits to handle only physical entities like corridors and living_rooms, which makes it difficult to find a way to represent and reason about dynamic events like grabbing a door handle in order to open this door. An AAL system, seen as a real world application, needs a full knowledge about its environment and a non-monotonicity reasoning support based on the Closed World Assumption (CWA). In fact, facts that are interpreted under the CWA are used to reduce the values into quantified expressions describing situations. OWL is a description logic-based ontology language, based on the Open World Assumption (OWA). It is not suitable for AAL systems because it was designed to be monotonic, and therefore cannot use the CWA to prevent large ontologies from admitting inconsistencies resulting, e.g., from multiple inhe-ritance [Bertino et al., 2003].

The symbolic reasoning about space and time has been addressed in [Stock, 1997] as a fundamental aspect of the future AAL systems. An AAL system can be considered as a Dynamic Domain (DD) in which entities can generate events anywhere and anytime. A reasoning process based on a chronological and semantic analysis, about past and ongoing events, requires therefore focusing about narratives, predictive reasoning, etc. In this paper, we propose a Semantic and architectural approach for Spatio-Temporal Reasoning in AAL systems. This approach, based on CWA, combines first order reactive reasoning, and narrative representation and reasoning. It transforms low-level events of virtual or physical entities generated by sensors, using for instance a Rete first order inference engine, into higher level abstraction of complex and structured situations through conceptual representation implying mutual relationships among events captured at a lower level. To describe entities and events, the approach uses hierarchical structures of semantic predicates and functional roles of the Narrative Knowledge Representation Language (NKRL) [Zarri, 2009]. The reasoning process analyzes the chronological semantic relationships between events. This approach benefits from the potential semantic relationships about events occurrences in both past and present time, and also about their spatial temporal dependencies. NKRL provides also a formal representation of any elementary event and determines through requests (questions) the general conceptual category of the specific event. This paper is organized as follows: Section 2 presents some related work on qualitative reasoning for AAL systems. Section 3 and section 4 describe respectively the narrative semantic representation foundations and the narrative inference architecture used in this paper. Section 5 presents the whole architecture of the proposed framework. Section 6 illustrates the feasibility of the proposed approach in an AAL scenario inspired from everyday life situations. The last section provides concluding remarks about ongoing and future works.

2 Related Work

In the last few years, a number of qualitative and quantitative reasoning approaches have been proposed for events processing, contexts and situation recognition from low-level sensors [Ye et al., 2011]. Using quantitative reasoning approaches in AAL systems is not sufficient to deal with the heterogeneity of sensors and data, the sporadic occurrence of events, and also if there is a causality chain that explains an ordered occurrence of these events. Qualitative reasoning is a good complement and sometimes an alternative to quantitative approaches for recognizing human contexts and situations. This strong paradigm is based usually on a symbolic representation of the context knowledge such as propositions, predicates, description logics ontologies and rules. The reasoning is applied using inference over statements describing the relationships between facts using a collection of rules, and determines the validity of these statements on inference rules, logic programming and case based reasoning. The most cited approaches in the state of the art concern the use of logic programming or complex events processing tools for context recognition and reaction. Pashke et al. have proposed a programming logic middleware based on ECA procedures (event, condition, action). The action corresponds to the best decision regarding the current situation (i.e. context) characterized by the couple (event, condition). [Yonezawa et al., 2009] have proposed SOEML as a smart object event modeling language to enable context-aware service in ubiquitous computing environment. It allows users to define both simple and complex event based on the threshold values of sensors. In [Bhatt et al., 2010], authors highlight the importance of reasoning in qualitative spatial representation and reasoning on the dynamic context. They used Event Calculus formalism to perform spatio-temporal abduction mechanism with an off-the-shelf logic of action and change. They demonstrate the role that abductive reasoning can play in application based on hypothetical spatial structures. From an application perspective, authors will use their approach in AAL such smart environments. Moreover, authors in [Bhatt et al., 2011] have depicted another interesting emerging application where the importance of space and time cannot be ignored. D.Riboni proposed the use of description logics ontologies in the validation of the activities knowledge inferred using low level quantitative reasoning such as statistical techniques [Riboni et al., 2009]. Other approaches propose the combination of the inference rules or logic programming with Description logics ontologies. The latter can be used to support semantic reasoning and interoperability. The reasoning concerns mainly the detection of ontology inconsistency and the inference of new individuals of context concepts or roles [Bettinia et al., 2009]; for example, transforming the collected sensor data, into a new set of concepts and roles individuals. Moreover, the consistency reasoning can be used during the ontology knowledge base construction to check the consistency in a class hierarchy and the consistency between instances. For instance, the consistency cheking allows to detect either there is classes that are subclasses of two classes, classes that are declared as disjoint; or two instances that are contradictory to each other such as an elderly person is in two spatially disjoint locations at the same time. Wang et al. proposed to combine description logics ontology called CONON with horn like inference rules, while [Yau et al., 2006] developed a situation ontology, which includes concepts to express time constraints on the logical specifications of situations. They used first order logic inference rules to support the conversion of situation specifications to FOL representations that can be used by potential FOL rule-based inference engine. The performance they obtained, concerning the transformation of situations specification, makes their model unusable for the recognition of time sensitive context captures. Chen et al. proposed an ontology called SOUPA [Wang et al., 2004] with the use of the logic programming written Flora 2 to allow different agents on the one hand, to share the same consistent interpretation of knowledge and on the other hand, to perform inferences to determine the current context in a specific place. However, for many reasons, the failure or unavailability of one or several sensors can cause the failure of all the inference process. Overcoming such an issue by adding more inference rules and new sensors will lead in the most to complex and non-maintainable rule bases. Alexandre et al. proposed an ontology model of video events, based on Video Event Representation Language, VERL and Video Event Markup Language, VEML [Alexandre et al., 2006]. This model consists of two ontologies. The first ontology provides a semantic description of the resources classes corresponding to the VERL definitions. The second ontology is an event taxonomy that provides a description of the annotation structures that appears in the VEML representations that refer to the VERL event definitions ontology. This model can be used to support tasks like context understanding in video surveillance, video browsing and content-based video indexing. The lack of suited reasoning to infer semantics of the possible n-ary relations between events definitions and events taxonomy is the main limitation of this approach.

Case-based reasoning is a logic-based qualitative reasoning that is of particular interest in the context of pervasive and AAL applications because it does not need to run with training data or a huge rule base. Indeed, it allows to learn new situations and activities incrementally as they occur. Knox et al. proposed SituRes, a case-based reasoning technique combined with ontology to perform cases base reduction by semantically linking the case solutions with the semantic features of cases and their related sensors [Knox et al., 2010]. Despite scalability issues, the approaches presented above allow to build promising added value situation aware AAL environments, with a minimalistic support of reasoning about time and lack of explicit support of the possible semantic relations and chronology between events and their contexts in the present and past time. These relations from our point of view are necessary for a better recognition of situations and the adequate decision making in a dynamic domain such AAL.

3 Narrative semantic representation foundations

NKRL consists of two main ontologies: HClass and HTemp. NKRL concepts are inserted into a generalization/specialization directed graph structure (HClass) often, but not necessarily, reduced to a tree where the data structures representing the nodes of HClass correspond, essentially, to a standard 'ontology of concepts' (according to the traditional, 'binary' meaning of these terms). HTemp is a hierarchy of templates used to represent dynamic knowledge. These templates are based on the notion of 'semantic predicate' and are organized according to an n-ary structure. They are conceived as the formal representation of generic classes of elementary events like "move a physical object", "be present in a place" and their relationships. HTemp can be seen as an ontology of events.

Let Ω be the ontology of concepts, and Ψ the ontology of events. E is the set of known static entities within the environment. Each entity is an information source, abstracted as a symbol and its semantics is grounded on Ω. The approach that we propose is based on the following statements:

Definition 1:
Ω is composed of two connected components: Concepts C_i and Individuals V_i. A concept C_i represents intentional knowledge that describes the general properties of concepts. An individual V_i is characterized by the fact of always being associated, often in an implicit way, with a spatio-temporal dimension, like DAVID_, BATHROOM_. C_i are represented in lower case while V_i are represented in upper case.

Proposition 1:
A concept C_i is a 'binary' description, composed of a set of axioms having the form H $\subset$ W where H and W are concepts. For instance, "individual_person $\subset$ human_being" denotes that "individual_person" is a specialization of "human_being".

Proposition 2:
Individuals V_i are added, when necessary, as "leaves" in C_i. Two individuals associated with the same description but having different labels will be considered as different individuals. This is not true in OWL. An individual V_i represents its own instances and all the instances are subsumed by concepts. For instance, geographical_location $\supset$ location_ $\supset$ BATHROOM, denotes that BATHROOM belongs to the concept of location.

Proposition 3:
Building a correspondence between the low-level features and the high-level conceptual descriptions requires an abstract model taking into account static (physical) and dynamic (events and situations) characteristics, roles, properties, etc. of entities. A model instance M_i of each event is an n-tuple specified as follows:

$$M_i =< U_i, F^{E_i}, S^{E_i}, A^{E_i}, L^{E_i}, I^{\Psi_i}, T^{M_i} > \qquad (1)$$

Where:

- $U_i \subset \Omega$ denotes the ID of the entity E_i producing the event;
- $F^{E_i} \subset \Omega$ denotes the function of E_i
- $S^{E_i} \subset \Omega$ denotes the set of outputs of each event generated by E_i
- $A^{E_i} \subset \Omega$ denotes the set of actions handled by E_i
- $L^{E_i} \subset \Omega$ denotes the current location of E_i
- $I^{\Psi_i} \subset \Psi$ is a structure that encapsulates the inputs of the NKRLs predicate associated to E_i
- T^{M_i} corresponds to the timestamp of the event E_i

Definition 2:
NKRL's Templates are instantiated according to a n-ary structure described formally as follows [Zarri, 2009]:

$$(L_i(P_j(R_1 a_1)(R_2 a_2).........(R_n a_n))) \qquad (2)$$

With:

- L_i, a generic symbolic label identifying a given template.
- P_j a conceptual predicate pertaining to the set (MOVE, PRODUCE, RECEIVE, EXPERIENCE, BEHAVE, OWN, EXIST).
- R_n a generic role pertaining to the set (SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC and CONTEXT). The set of roles R_n is constructed from I^{Ψ_i} and the corresponding arguments encapsulated within a_n that can consist of a simple concept such as geographical_location, individual such as LIVING_ROOM or associations of structured association of several concepts like: "working_noise HOOD". The HOOD is here an instance of the concept hood_. When it is associated with the concept working_noise, it means that the HOOD is running.

When a particular elementary event pertaining to one of these general classes (Predicates and Roles) must be represented, the corresponding template is instantiated to produce what, in the NKRL's jargon, is called a predicative occurrence.

Table 1: PARTIAL REPRESENTATION OF EXPERIENCE TEMPLATE

name : EXPERIENCE: GENERIC SITUATION
nl_description : 'a given entity is affected by a gener-ic (not value charged) situation'
PREDICATE: EXPERIENCE
SUBJ var1 : [(var2)]
OBJ var3
date-1:
date-2 :
var_1 = <artefact_>| <human_being_or_social_body> |<location_> | <pseudo_sortal_geographical>
var_2 = <location_> | <pseudo_sortal_geographical>
var_3 = <condition_> | <label/name_>|<reified_event>

Proposition 4:
Each instance of Ψ corresponds to a predicative occurrence denoted by Φ. Φ is a conjunction of the following elements: Predicate, roles R_n and their list of fillers $f_n \subset \Omega$ and the corresponding location L^{E_i} of the latter, timestamp T^{M_i} and a set of constraints { $\Re$ } defined by a variable var_i (example of constraints { $\Re$ } are given in Table 1). Formally:
Φ =PREDICATE $\oplus$ { $R_n \otimes f_i \oplus L^{E_i}$ } $\oplus\, T^{E_i} \oplus$ { $\Re$ }

In a template, see Table 1, the arguments of the predicate corresponding to the an terms in Eq. 2, are represented by variables with associated constraints. The latter are expressed as concepts or combinations of concepts, i.e., using the terms of the HClass ontology. In the reasoning process, all the "explicit variables", identified by conceptual labels in var_i will be replaced by $C_i/V_i \subset \Omega$ compatible with the original constraints { $\Re$ } imposed on variables var_i.

Definition 3:
$I^{\Psi_i} \cup T^{M_i} \equiv \Phi\ M_i$. This means that M_i includes all the semantic spatio-temporal informations of $e_i^{E_i}$.

Definition 4:
All inference rules are handled according to the following equation [Zarri ,2009] :

$$X \quad iff \quad Y_1 \quad and \quad Y_2 \ ...... \ Y_n. \qquad (3)$$

Where X is the situation/context to infer and Y_1,....., Y_n represent the reasoning steps. X, Y_1,..., Y_n are represented as instances of the template (Φ).

Proposition 5:
Let e_i=<t,s> an elementary event with s $\in T^{E_i}$ its time detection and t $\in L^{E_i}$ its space location. Thus, two disjoints events e_i and e_j are formally defined as follows:
$\forall\, e_i^{E_i}$ ($\neg \exists\, e_j^{E_j}$ (($e_j^{E_j}$ = $e_i^{E_i}$) $\wedge$ (i $\neq$ j))) with $e_i^{E_i}$ the event generated by E_i

Remark 1:
The partition between sortal_concept and non_sortal_concept constitutes the main architectural principle of Ω, and corresponds to the differentiation between "(sortal)" concepts that can have direct instances like SMITH_ and "(non-sortal)" notions, which cannot be instantiated directly into specimens, like gold_, which can admit further specializations as white_gold for example, but do not have direct instances.

Definition 5:
The two temporal attributes associated with the predicative occurrence Φ, date1 and date2, constitute the "search interval" used to limit the search to the slice of time that it is considered appropriate to explore.

Example 1.
The EXPERIENCE Templates (Table 1) are used to represent situations where a given entity, human or not, is subject to some sort of "experience" (illness, richness, economical growth). For instance, the predicative occurrence Φ given below states that the concept temperature in the location BATHROOM_1 is growing as specified in the role OBJ, from the date-1: 17/4/2011/07:45.

$e_i^{E_i} \equiv \Phi =$ EXPERENCE SUBJ (SPECIF
temperature:BATHROOM_1)
OBJ growth_
date-1: 17/4/2011/07:45
date-2:

Proposition 6:
A Context Π_τ at a given time consists in an aggregation of elementary events $e_i^{E_i}$, formally:
Π_τ = ($e_1^{E_1} \wedge e_2^{E_2} \wedge$ ..... $\wedge\, e_{n-1}^{E_{n-1}} \wedge e_n^{E_n}$)

A Situation is then an aggregation of contexts denoted by $\sum \Pi_\tau$, formally:
$\Pi_\tau \wedge \Pi_\tau \wedge \Pi_\tau \wedge$ .....$\wedge\, \Pi_\tau \wedge \models \sum \Pi_\tau$

Proposition 7:
To infer a context/situation, multiple events must be correlated with respect to their spatial and temporal ordering. For this purpose, NKRL uses the second order structure called binding occurrences such as GOAL, COORD(ination), CAUSE, etc. For instance, we would state that: the system switches on the light in the bathroom once the presence of DAVID (human being) in this location is detected. The system encodes the causality of the two events at run time as follows: $M_1^{E_1}$ ="the presence of DAVID in the bathroom is detected at 07h:30" and $M_2^{E_2}$="the system switched on the light at the same time". The full description of these events are given in Table 2, where aal1) and aal2) represent respectively the first event and the second event while aal3) links these two events.

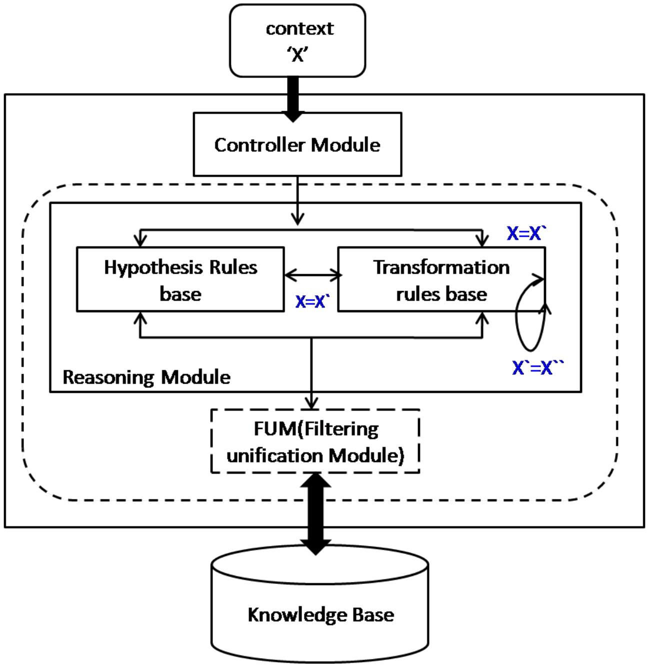
Table 2: BINDING OCCURRENCES

aal1) EXIST SUBJ DAVID_: (BATHROOM_1)
date-1: 17/4/2011/07:30
date-2:
aal2) MOVE SUBJ SYSTEM_1
OBJ (SPECIF lighting_ BATHROOM_1):
(switch_off, switch_on)
date-1: 17/4/2011/07:30
date-2:
aal3) CAUSE(aal2 aal1)

4 Narrative inference architecture

A simplified schema of the NKRL reasoner is shown in Fig 1. Each component can be described sketchily as follows [Zarri, 2009] : Context: Recognizing a context/situation is based on the concept of ”Search Patterns”. The latter are data structures that supply the general framework of information to be searched. They offer therefore the possibility of querying this base directly. Formally, a search pattern can be assimilated to specialized/partially instantiated templates where all the ”explicit variables” identified by conceptual labels in the variable var_i. In the Example 1 and Table 2, all the variables have been replaced by concepts/individuals compatible with the original constraints imposed on these variables.

Figure 1: Simplified architecture of NKRL reasoner



Filtering Unification Module(FUM):

Verifying the ”semantic congruence” between a search pattern and facts in the Knowledge Base is carried out by this module. The matching process could be better defined as a simple ”filtering” process, giving that all the variables, under the form of ”implicit variables” are only present on the search pattern side. The implicit variables (concepts) of the pattern must all find a correspondence with some of their subsumed HClass terms, concepts or individuals, within the matched occurrences. During a successful retrieval operation, any HClass concept (to be assimilated now to an implicit variable) that occurs in a search pattern can match/unify (in the corresponding predicative occurrences of the knowledge base) all the ”identical” concepts, but also all the ”subsumed” concepts (i.e. all the specifications of this concept compatible with the structure of HClass) and all the individuals representing its own instances and all the instances of the subsumed concepts. This way of operating corresponds to a sort of semantic/conceptual expansion of the original pattern; this process of search patterns unification is defined as a first level of inferencing of NKRL.

Hypothesis rules:

These rules allow to build up automatically a sort of ’causal explanation or context’ for an information (a predicative occurrence Φ) retrieved within an NKRL knowledge base using Filtering Unification Module and a context (search-pattern) in a querying-answering mode. In the context of a running hypothesis rule, the head X of Eq. 3 corresponds then to a predicative occurrence. Accordingly, the ’reasoning steps’ Y_i of Eq. 3 called ’condition schemata’ in a hypothesis context must all be satisfied (for each of them, at least one of the corresponding search patterns must find a successful unification with the predicative occurrences of the base) in order that the set of predicative occurrences retrieved in this way, can be interpreted as a context/causal explanation of the original occurrence X.

Transformations rules:

These rules try to ’adapt’, from a semantic point of view, a search pattern that ’failed’ (i.e. that was unable to find a unification within the knowledge base) to the real contents of this base making use of a sort of ’analogical reasoning’.
In a transformation context, the ’head’ X of Eq. 3 is then represented by a search pattern (p). The transformation rules try then to automatically ’transform’ pi into one or more different p_1,p_2 ....p_n that are not strictly ’equivalent’ but only ’semantically close’ to the original one : X (Fig.1).

Remark 2:

When it is impossible to find an explicit knowledge within the knowledge base using hypothesis rules, then the two inferences mechanisms mentioned above (Hypothesis rules and Transformation rules) are combined to discover all the possible implicit information associated with the original context X.

Controller module:

It is responsible for managing the whole execution of inferences procedures. The inferences procedures shown in Fig 2, by executing hypothesis rules and using eventually transformations rules, aim at finding semantic relationships between events stored into the knowledge base. The step reasoning Y_i is started once the reasoning step Y_{i-1} has succeeded. The Y_n (eq. 3) denotes the leaf in the tree-structure which symbolizes the success of reasoning process.

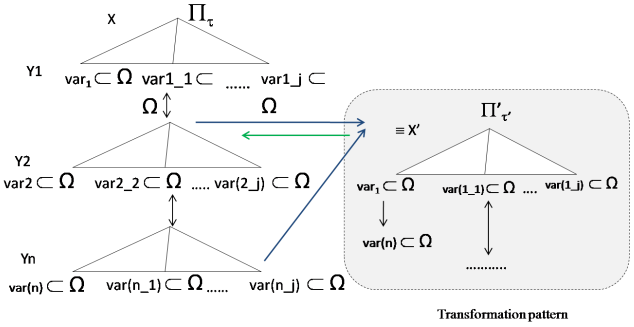
Remark 3:

Inferences procedures explore all the variables var_i at each reasoning step Y_i (Fig 2). For clarity’s sake, we explain the inference procedures by introducing the follow-ing example.

Example 2.

All the following NKRL code has been simplified for clarity’s sake. Here the aim concerns the recognition of the activity ”cooking” of a person in a kitchen, by asserting that this person has used for instance a cooking entity. The corresponding search pattern is then: ”has the person used a cooking entity?”. As it is impossible to validate this search pattern with the data that we have at our disposal, we can use a transformation rule in order to find an indirect answer to

Figure 2: Graphical representation of NKRLs Inferences procedures


X
Πτ
Y1
var1 ⊂ Ω var1_1 ⊂ ...... var1_j ⊂
Ω
Ω
Y2
var2 ⊂ Ω var2_2 ⊂ Ω ..... var(2_j) ⊂ Ω
Yn
var(n) ⊂ Ω var(n_1) ⊂ Ω...... var(n_j) ⊂ Ω
Π′τ′
≡ X′
var1 ⊂ Ω var(1_1) ⊂ Ω .... var(1_j) ⊂ Ω
var(n) ⊂ Ω
....
Transformation pattern

the original query. Let us assume that it is possible to assert that the person has powered on a coffee machine and that this coffee machine is located in the kitchen. The antecedent part, denoted by X in Table 3, of the transformation rule corresponds to say that a human_being (var_1) is "behaving" in the kitchen (var_2) as a user of some cooking entity (var_3). The first consequent scheme (denoted by Y_1 in Table 3) states that var_1 changes the state of the coffee machine from idle_ to running_. The second scheme denoted by Y_2 in Table 3) states in turn, that the coffee machine (var_3) is in the kitchen (var_2). The logic of the transformation rule is equivalent to say that moving the coffee_machine from idle_ to running_ state is equivalent to say that the person is using some cooking entity in the kitchen location given that the coffee machine is located in the kitchen. Note that coffee_machine is a specific concept of the generic concept cooking_entity, as described in the HClass ontology.

Note that all the NKRL schemata used in general in the inference rules are partial instantiations of the templates (HTemp) that are part and parcel of the definition of the language.

Table 3: TRANSFORMATION RULE

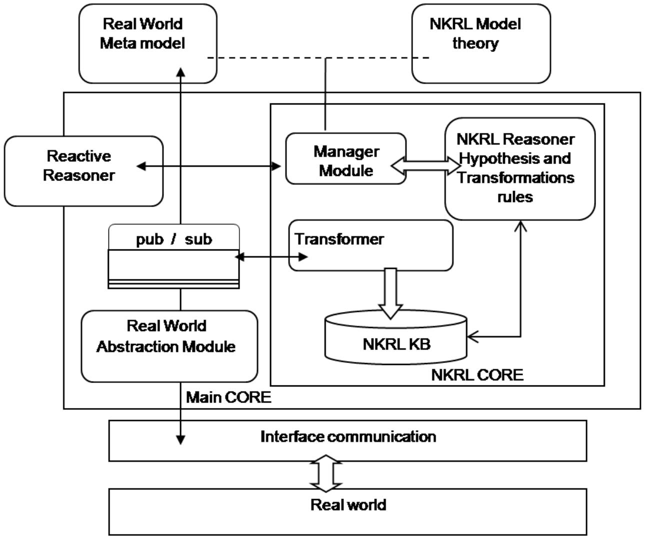
```
X)PREDICATE BEHAVE
                    SUBJ var1 : (var2)
                    MODAL user_
                    TOPIC cooking_entity
var1 =human_being
var2 =location_
Y1)PREDICATE MOVE
                    SUBJ var1: (var4)
                    OBJ var3: (idle_, running_)
var3 = coffee_machine
var4 = kitchen_
Y2)PREDICATE OWN
                    SUBJ var3
                    OBJ property_
                    TOPIC ( SPECIF var3 (SPECIF
                              located_in var4) )
```

Proposition 8:

Although the inferences procedures could infer on the context using several hierarchical steps, but in practice, the transformation and hypothesis rules have a very simple format that allow efficient reasoning from time point of view. The subsumption characteristics of Ω allows reducing considerably the number of constraints R in each node. Furthermore, given that our approach is CWA-based, two different instances of a concept cannot refer to the same entity. The name of the instance must be unique to avoid any possible contradiction.

5 Semantic architecture

Figure 3: Semantic architecture



Ensuring the homogeneity of the knowledge base and classifying each entity according to its role, allow easily aggregating spatio-temporal events into coherent facts shared between the reactive rule engine and narrative reasoning engine. The architecture of our framework, depicted in Fig 3, is composed of three main architectural blocks:

1. The Reasoning Main Core: this block keeps a coherent representation of the world situations by handling the link between the real world abstraction and the knowledge bases. This block consists of a Manager Module that handles events incoming from i) the real world sensors through the real world abstraction module; ii) the reactive rule engine; iii) the narrative rule engine. The communication between the two reasoners is handled by the Manager Module through a publish/subscribe messaging.

2. NKRL Core: it consists of a transformer module that uses narrative representation model to infer predefined contexts. This block consists also of the NKRL Reasoner Engine where all predicatives occurrences corresponding to the narrative description of the spatio-temporal events are inferred using both transformation and hypothesis rules. The proposed framework offers a high abstraction level of the underlying hardware and software sensors infrastructure. It easily collects spatio-temporal low level events and transforms them into higher level abstractions.

3. The standard communication component that represents the third block allows the communication with real world entities through heterogeneous protocols. This architecture that is flexible and extensible provides loose coupling of all components

6 Scenario

To demonstrate the feasibility sof the proposed framework, we consider an AAL scenario inspired from everyday life situations: An elderly person is looking for her/his phone in her/his home. The system tries then to recognize the corresponding context or situation by performing the following requests: when was the last time the person used his phone? Where was the person when she/he hangs up her phone? Depending on the context inferred, the system will suggest the location where the phone is. The AAL system is assumed to be instrumented with RFID tags to identify each object deployed in the environment. Pressure sensors are placed under the sofa and the chairs to detect events such as: "the person is sitting on her sofa, the person is standing up from a chair". In NKRL formalism, knowing where a given object is located is represented by the following query:
X(Query1): PREDICATE EXIST (SPECIF mobile_phone DAVID) : (geographical_location)

Detailed story's events:

The most important elementary events are represented into NKRL format in Table 4. David, an elderly person, leaves his bed at 7h:20 mn and enters in the bathroom. The system switches on the light in the bathroom at 7h:36 mn. He opens the shower tap located in the bathroom. Suddenly, his mobile_phone located in the living room rings. He takes the call and few minutes later, he returns to the bathroom, and he closes the door. The system detects that the call has ended at 7h:50 mn. David leaves his house at 8h:45mn. At 18h:30, he comes back, takes a seat on the sofa in the living room. He decides to call a friend and then he asks the system to find his phone.

Reasoning procedure:

Assuming that there is no localization system embedded in the mobile phone, the AAL system cannot get a direct response to the query, denoted by X(Query 1), and starts therefore to recognize automatically the context using the transformation rule (Table 5) to find the location of the phone.

The antecedent X' can unify the original query (Query1) at time t_0; in this way it is possible to associate to var_1 the value mobile_phone_1 and to var_2 the value DAVID_. These two values will be used all along the inferences procedures. The first consequent scheme Y_1 (Table 6) states that DAVID_ is addressing a query "touch_query" to the system concerning the location of his MOBILE_PHONE_1. The second consequent scheme goes to see if it possible to find a David's call that occurred before t_0.

The third consequent scheme (Y_3) checks that between t_{0-n} and t_0 the MOBILE_PHONE_1 is idle. This can suggests us that the location of the MOBILE_PHONE_1 is the location location of DAVID_ at time t_{0-n}, i.e the BATH_ROOM, where DAVID_ has made a call. Of course, this is a possible "indirect answer": all the inference rules used in NKRL are part of family of rules where different explanations can be verified. The Table 4 shows the most important predicatives occurrences existing in the knowledge base at a given moment; it is easy to find among them those unifying the consequent schemes given in Table 5. The system will first search in the knowledge base when the phone was used for the last time and then where the person was when the last call ended? The last location would be the most likely location where the phone is. For this purpose, the system will execute the inferences rules, especially the rule "locate an object" described in Table 5. It provide a sort of causal explanation of the triggering event by retrieving in the knowledge base in the style of: i) the elderly person has used her/his phone. The system can then suggest that the phone is in the last location where the elderly person has ended the call. Given that the event related to the last coming call is present in the knowledge base system as depicted in table 4, the system tries then to infer if the phone was used or not. Replying to this query needs to find in the knowledge base the following semantic events: 1) the person moves towards her/his phone, 2) the phone stops ringing, and 3) the call takes a certain time.

Table 4: REPRESENTATION OF THE ELEMENTARY EVENTS IN NKRL FORMAT

```
aal1) PREDICATE MOVE
          SUBJ DAVID_: LIVING_ROOM_1)
          OBJ touch_query :
                         (SYSTEM_SCREEN_1)
          TOPIC (SPECIF location_          (SPECIF
                 (MOBILE_PHONE_1 DAVID_)
          date-1: 17/4/2011/19:35
          date-2:
aal2) PREDICATE MOVE
                 SUBJ DAVID_: (BATH_ROOM_1)
                 OBJ message_
                 date-1: 17/4/2011/7:50
                 date-2
aal3) PREDICATE OWN
                 SUBJ MOBILE_PHONE_1
                 OBJ property_
                 TOPIC idle_
                 date-1: 17/4/2011/7:50
                 date-2: 17/4/2011/19:35
aal4) PREDICTE OWN
                 SUBJ DAVID_ :ROOM
                 OBJ property_
                 TOPIC up_
                 date-1:17/4/2011/7:20
                 date-2:
aal5) PREDICATE MOVE
                 SUBJ DAVID_
                 OBJ (SPECIF tap_ BATHROOM_1):
                 (trun_off, trun_on)
                 date-1:17/4/2011/7:33
                 date-2:
aal6) PREDICATE OWN
                 SUBJ PHONE_CALL_1
                 OBJ property_
                 TOPIC finish_
                 date-1:17/4/2011/7:50
                 date2:
```

Table 5: TRANSFORMATION RULE

```
X1)PREDICATE EXIST
                SUBJ (SPECIF var1 var2):
                           (geographical_location)
var1=mobile_phone
var2=human_being
Y1) PREDICATE MOVE
                SUBJ var2
                OBJ touch_query : ( system_screen)
                TOPIC (SPECIF location_
                              (SPECIF (var1 var2)
                date1: $t_0$
                date-2:
var1= human_being
var2=mobile_phone
Y2) PREDICATE MOVE
                SUBJ var1: (var3)
                OBJ message_
                date-1: $t_{0-n}$
                date-2:
var3 = geographical_location
Y3) PREDICATE OWN
                SUBJ var1
                OBJ property_
                TOPIC idle_
                date-1:$t_{0-n}$
                date-2:$t_0$
```

7 conclusion

In this paper, we presented a semantic and architectural approach for the Spatio-Temporal Reasoning in AAL systems. This approach, based on a qualitative reasoning, uses hierarchical structures of semantic predicates and functional roles of the Narrative Knowledge Representation Language (NKRL). It benefits from the potential semantic relationships about events occurrences in both past and present time, and also about their spatial temporal dependencies. Through a scenario illustrating a daily life typical situation, we have shown the feasibility of the proposed approach and also its potential to explain a causality chain between events. Our ongoing works is the real-time implementation of the proposed approach on ubiquitous platform composed of a robot compagnon and sensors/actuators dessiminated in the AAL environment.

References

[Alexandre *et al.*, 2005] R.J. Alexandre, F.R. Nevatia, J.Hobbs, R.C. Bolles. *VERL: An Ontology Framework for Representing and Annotating Video Events*. In IEEE Multimedia, vol. 12, no. 4, pp. 76-86, Oct.-Dec. 2005.

[Bhatt *et al.*, 2010] M. Bhatt, G. Flanagan. *Spatio-Temporal Abduction for Scenario and Narrative Completion (a preliminary statement)*. In International Workshop on Spatio-Temporal Dynamics, ECAI 2010, Lisbon.

[Bhatt *et al.*, 2011] M. Bhatt, H. Guesgen, H. Woelfl and S. Hazarika. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends and Future Directions. In *Special Issue of the Journal of Spatial Cognition and Computation. 11(1).London.*

[Bertino *et al.*, 2003] E. Bertino, A. Provetti and F. Salvetti Local Closed-World Assumptions for reasoning about Semantic Web data . In *In Proceedings of the APPIA-GULP-PRODE Conference on Declarative Programming*

[Bettinia *et al.*, 2009] C. Bettinia, O. Brdiczkab, K. Henricksenc, J. Indulskad, D. Nicklase, A. Ranganathanf and D. Ribonia. A survey of context modelling and reasoning techniques. In *Pervasive and Mobile Computing 6 (2) 161-180*

[Knox *et al.*, 2010] S. Knox, L. Coyle and S. Dobson. Using Ontologies in Case-Based Activity Recognition. In *23rd Florida Artificial Intelligence Research Society Conference(FLAIRS-23),Florida, AAAI Press.*

[Martinez *et al.*, 2007] O. Martinez Mozos, P. Jensfelt, H. Zender, G.J. M. Kruijff and W. Burgard. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *In: Proceedings of the ICRA-07 Workshop on Semantic Information in Robotics. Los Alamitos (CA): IEEE Computer Society Press.*

[Stock, 1997] O. Stock. Spatial and Temporal Reasoning. In *preface, edited by Oliviero Stock IRST, Italy. ISBN 0-7923-4644-0. 1997*

[Riboni *et al.*, 2009] D. Riboni and C. Bettini. Context-aware activity recognition through a combination of ontological and statistical reasoning. In *Proceedings of the international conference on Ubiquitous Intelligence and Computing, Springer, Berlin, Heidelberg, pp. 3953.*

[Yonezawa *et al.*, 2009] T. Yonezawa, J. Nakazawa, G. Kunito, T. Nagata and H. Tokuda. SOEML: Smart Object Events Modeling Language based on Temporal Intervals. In *International Journal of Multimedia and Ubiquitous EngineeringVol. 4, No. 3.*

[Wang *et al.*, 2004] X. H. Wang, D. Q. Zhang, T. Gu and H. Keng Pung. Ontology Based Context Modelling and Reasoning using OWL. In *In Workshop Proceedings of the Second IEEE Conference on Pervasive Computing and Communications (PerCom04), pp. 1822, Orlando,*

[Yau *et al.*, 2006] S. S. Yau and J. Hierarchical situation modeling and reasoning for pervasive computing. In *in: SEUS-WCCIA06: Proceedings of the 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and 2nd International Workshop on Collaborative Computing, Integration, and Assurance, IEEE Computer Society, Washington*

[Ye *et al.*, 2011] J. Ye, S. Dobson and S. McKeeve. Situation identification techniques in pervasive computing. In *A review, Elsevier, Pervasive and Mobile Computing Journal*

[Zarri, 2009] G. P. Zarri. Representation and Management of Narrative In formation: Theoretical Principles and Implementation. In *R. Series: Advanced Information and Knowledge Processing 1st Edition. Springer, 2009*

Towards a Qualitative, Region-Based Model for Air Pollution Dispersion

Jason Jingshi Li, Boi Faltings
Artificial Intelligence Laboratory, EPFL
CH-1015, Lausanne, Switzerland
jason.li, boi.faltings @epfl.ch

Abstract

Air quality has a direct impact to human health. Current advancements in sensor technology are making air pollution sensors smaller, cheaper and more mobile than ever before. This made community monitoring of air pollution a rapidly growing area in environmental sensing. In this paper, we first introduce the problem of air pollution dispersion, and survey the current mainstream models address this problem. We then identify the reasoning tasks for a model with more data available from a heterogeneous network of static and mobile air quality sensors. Lastly, we propose the framework for a data-driven, region-based model that reasons about qualitative changes in air pollution, and discuss how the forward, backward and meta-reasoning tasks would benefit form such a region-based approach.

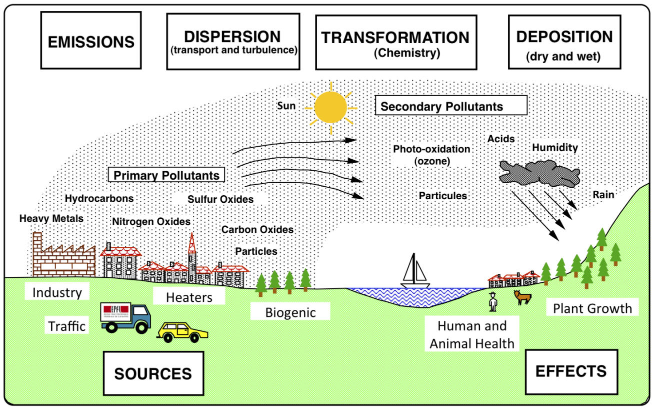


Figure 1: Air pollution processes and their effects

1 Introduction

Air pollution is a complex problem involving many variables. It has a direct impact on human health, and the World Health Organization estimated that every year it causes up to 2 million premature deaths worldwide [1]. Figure 1 illustrates the overall picture of air pollution. The pollutants are typically results of combustions due to human economic activity such as traffic, heating and industry. They are transported by wind, and some react in the presence of sunlight to form secondary pollutants. In the end, the deposition of the pollutants leads to adverse effects on human health, animal health and plant growth. In order to minimize these adverse effects, control strategies are put in place to limit the emission of pollutants at their source. Current research efforts in air pollution dispersion are focused on understanding the natural processes in order to find an effective and efficient strategy that is an optimal tradeoff between environmental impacts and economic productivity [9].

Measuring air pollution has traditionally been an expensive exercise. Equipment with high accuracy and precision are costly to both acquire and maintain, and consequently a typical city in United States or Western Europe only has a few permanent stations that continuously monitor air quality. The mainstream modeling approach relies on numerical simulations based on physical and chemical principles, and the models are typically independent to the measurements. The latter is typically used to validate the former. However, with the introduction of smaller, cheaper and more mobile air quality sensors, it is expected that data-driven, statistical approaches will play a more prominent role in the monitoring and modeling of air pollution. The deployment of such mobile sensors over a cellular network would allow air pollution information over vast regions to be updated and distributed in real time. An accurate dispersion model can then make use of such measurements build a realistic map of air pollution, and deliver advanced warnings to people who may be sensitive to elevated levels of pollution within the affected regions.

Given a set of air pollution measurements over a certain region, we would like to know three things: what are the pollution levels at places where there are no measurements; what are the expected pollution levels for certain regions of interest in the future; and where did the pollution came from. The first two correspond to a forward reasoning task, interpolating the measurements both in space and in time. The third corresponds to a backward reasoning task, where we abduce the cause of the air pollution for the given scenario, identifying possible unknown sources. In addition, there is a meta-reasoning task about where the sensors can be optimally placed to better answer the previous questions.

In this paper, we analyze the problems involved in modeling the dispersion of air pollution, and survey the current state of the art approaches in both physical and statistical models. We then identify the three type of reasoning tasks that we can perform from a set of measurements collected over a network of static and mobile air quality sensors. Lastly, we propose the framework for a qualitative, region-based model for air pollution dispersion, and outline reasons of why such a model is appropriate for this problem domain.

2 Backgrounds

2.1 The Problem Domain

Primary air pollutants such as carbon monoxide (CO), nitric oxides (NOx), fine particles (PM2.5, PM10) and volatile organic compounds (VOCs) are typically created by combustion. They are released from either stationary sources such as chimneys, or mobile sources including cars, trains, airplanes, etc. A stationary source that singularly contributes to a significant degradation of air quality is known as a point source. A number of stationary and mobile sources that individually do not significantly affect air quality, but altogether makes a significant difference in a geographical region is known as an area source. These sources are an integral part of the overall picture of air pollution, and a typical application of air quality research is to identify the most appropriate source for an economical emission control strategy.

Some primary pollutants undergo chemical reactions and produce secondary pollutants. One example is lower atmospheric ozone (O_3), an important component of photochemical smog. It is formed when NOx reacts with VOCs in the presence of sunlight. While ozone is invaluable in high atmosphere that act as a layer to absorb the majority of the sun's ultraviolet light, at lower atmosphere it is found to be toxic to living systems. At night, the ozone in lower atmosphere also reacts with NOx to form nitric acid, which in turn leads to acid rain.

The behavior of the pollutants in the atmosphere can be characterized by several processes. One process is the movement of pollutants due to horizontal wind, which is known as transport. The pollutants also gradually spread while their concentration is reduced, a process known as diffusion. The process that leads to the formation of secondary pollutants is known as chemistry. These processes, together with emissions and depositions, complete the picture for air pollution.

2.2 The Existing Models

As air pollution measurements cannot be collected everywhere at once, we need models to better inform us about the behavior of air pollution over both space and time. The models may also help us to understand what would happens in hypothetical scenarios, which allows policy makers to design better pollution control plans.

Physical Models

The physical models aim to reconstruct a complete picture of the air pollution within a given region based on physical and chemical equations that describe the behavior of air pollutants within various grid cells. Currently this approach is the most widely used and accepted methodology for both government agencies and the environmental science research community. There is a myriad of physical models that are actively deployed and used by regulatory authorities and universities, including CMAQ [5], CAMx [2], CHIMERE [4], ADMS [7], TAPOM [29], and GRAL-Sys [22]. A comprehensive review of the physical modeling approach can be found in Godish 2003 [9].

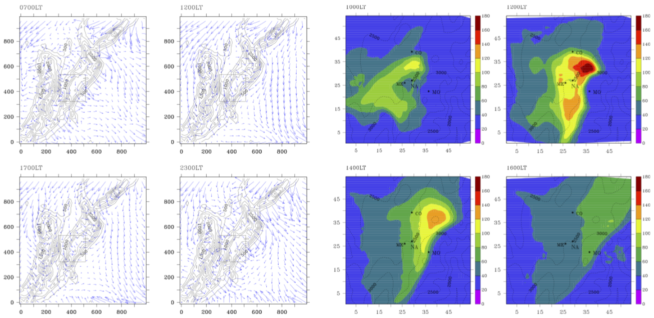


Figure 2: The windfield and the resulting O3 estimations in the city of Bogota from the physical model TAPOM [29].

A physical model does not directly use any air quality measurements, although measurements may be useful in tuning various parameters in a model. The physical modeling approach works by first collecting the emission inventory (EI), which describes all known sources and sinks of emissions. It also computes the wind fields from given meteorological and topographical parameters for every single grid cell over the modeling period. The model then uses the EI, wind fields and other meteorological information as inputs to a series of equations that model the transport, diffusion and chemistry of air pollution. The physical model can be understood as a sophisticated rule-based system, and they are considered "validated" when real-world measurements taken at various monitoring stations fit well with model projections.

The goal of these models is to determine the relations between the effects of source of emissions and ground-level pollutant concentrations [9]. A validated physical model is thought of as having understood such relations well. Ultimately, the purpose for these physical models is to design effective and efficient pollution control plans after a careful cost and benefit analysis. For example, Zarate [29] (Fig. 3) analyzed several hypothetical scenarios in validated simulation models over the city of Bogota, and concluded that most of the harmful secondary air pollution in the city are caused by the on road traffic emissions released before 9am in the city itself. This is useful in designing abatement strategies to optimize the balance between environmental impact and economic cost.

The use of such models is also a subject of some controversy, mostly involving which models should be used and the interpretation of the results. The underlying concern is about the accuracy of the predictions from such models under certain specified scenarios. In fact, Oreske $et.al.$ [23] argued that verification and validation of such models is impossible due to the fact that the subject of the model (the atmosphere) is an open system, and it is sufficient that such model give us some insight to the behavior of system in the real world. The current consensus is that there are few alternatives to their use, particularly when it involves decision-making about policies

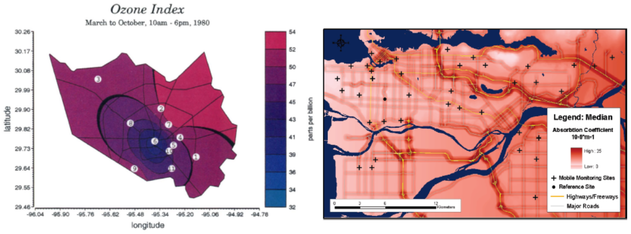


Figure 3: The estimations from statistical models for Ozone concentration in Harris County, Texas [6] (left) and black carbon concentration in the city of Vancouver [16] (right).

with both environmental and economical impacts.

Statistical Models

In contrast to physical models that simulate pollution behavior from first principles, the statistical approach constructs a model based on a dataset of measurements. A wide variety of techniques are used in different statistical models to perform spatial and temporal interpolation of measurements. For example, Caroll $et.al.$ [6] analyzed population exposure by interpolating the ozone concentration beyond the 9 to 12 measurement stations in Harris County, Texas from 1980 to 1993. The ozone model is comprised of a deterministic component that is based on time and temperature, and a non-deterministic component (a Gaussian Process) to account for all other variables. A different approach can be found in the works of Kibria $et.al.$ [13], where a Bayesian model was created to map the PM2.5 fields from eight measurement stations in the Philadelphia region between 1992 and 1993.

One feature of statistical models in the past is that there are usually a low number of measurement stations. This is understandable given the cost of the measurement stations. To compensate for this, additional information such as land-use is introduced in some models [16; 19]. However, as sensor measurements become cheaper to acquire, the data-driven, statistical approach is expected become more viable in providing a detailed snapshot of air pollution. This in turn would also give us insight into the natural processes involved in the dispersion of air pollution.

One of the most important problems in reconciling sensor measurements to models is about scale. There are many atmospheric processes that influence the air quality of a region at a given time. Each of these processes occurs at their own spatial and temporal scale. The processes that influence continuous, real-time measurements on the street level are very often due to turbulence, local traffic conditions and other micro-scale events whose precise information is difficult to ascertain. Again, this reinforces the case for the statistical approach over physical modeling for dealing with street-level pollution data and the subsequent exposure analysis.

3 Reasoning With More Data

3.1 Measurement Acquisition

Currently, air pollution information is typically collected from large, expensive measurement stations. They are mostly located on the rooftops of building or in large parks, where the sampled air is representative of the overall concentration of pollutants over a large area, and the variability is relatively low compared to streets. The advent of cheaper mobile air quality sensors has meant that measurements can now be obtained at an unprecedented scale. In the OpenSense project at EPFL [3], we are working with the public transport authority of the city of Lausanne to deploy sensing units on top of buses as well as bus stops, and the information will communicate over a wireless network in real-time. This would allow us to build up a more detailed and complete picture for the dispersion of air pollution in a real-world setting.

3.2 The Reasoning Tasks

In the setting where we have a good spatial distribution of mobile air quality sensors, there are a few things we would like to infer from the sensor measurements collected over a period of time. These include “what does the overall pollution level look like”, “where did all the pollution came from?’, “should I be outdoors in the next hour given my allergies”, etc. In this section we summarize three types of reasoning tasks involving the sensor measurements, and how the current state of the art models address these tasks.

A reasonable restriction of the problem is to limit our interests to the pollutants that are relative unreactive in the atmosphere, such as fine particles (PM2.5, PM10) and sulfur dioxide (SO2). In this case we are only interested in how they are transported and diffused, and not getting involved any non-linear chemical reactions. It is clear that before we create a full model of air pollution we must first solve this simpler subproblem.

Forward Reasoning

The first type of reasoning tasks involving these measurements is the spatial and temporal interpolation of measurements. They begin with the measurements, and by using some assumptions and inference rules, one deduce more facts from the data. They include queries for pollution levels at locations where there are no measuring devices; future pollution levels at measurement sites; likelihood of dangerous pollution levels for a given region in the next hour etc. Any model for air pollution dispersion is likely to propose a solution to these reasoning tasks, and the quality of the solution can be evaluated by taking more measurements.

A physical model can be seen as an example of a tool for forward reasoning, but it relies on emission data instead of actual pollution measurements. A physical model can be understood as a rule-based system that takes emission and meteorological information and computes the expected dispersion. Some physical models such as ADMS-Urban [7] also takes into account of pollution measurements and builds a complete and detailed map of the pollution. Similarly, statistical models can simply use the extra measurements to make more precise predictions.

Backward Reasoning

The second type of reasoning tasks involves working backwards from the sensor measurements to reach an explanation. This may involve identifying a previously unknown pollution source, understand causes to the observed level of pollution, or explain the mechanisms in the dispersion of the pollutants. Unlike forward reasoning, it appears to be a more difficult task and the models required for backward reasoning are likely to be more complicated due to uncertainties.

There are far fewer works on automated source detection in the literature on air pollution dispersion. This may be due to the fact that currently there are insufficient measurements to adequately infer about emission sources. In physical models, explanation for the formation of a secondary pollutant can be found by analyzing hypothetical emission scenarios from a validated photochemical model [26]. The problem is related to spatio-temporal event-detection in data-mining, where complex spatio-temporal patterns are extracted from a monitoring sensor network [28].

Meta-Reasoning

The third type of reasoning tasks is about how forward and backward reasoning can be better accomplished to a given utility function with the available resources. One such typical task is sensor placement: where are the best locations to place sensors in order to get the most relevant information. A similar problem in the temporal dimension is selective sampling, where one determines when measurements should be taken. A third problem that somewhat subsumes the first two is sensor selection, where constraints on battery, communication bandwidth or source reliability require us to select only a subset of sensors in a network to be sent to a central server. In the context of community sensing, the meta-reasoning tasks are important when one needs to deal with a large variety of community-sensing based applications in an efficient and sustainable manner.

Golovin $et.al.$ [10] provided a good summary of previous works that looked at the sensor selection problem, and investigated the problem in a distributed online setting. The paper is one of the latest additions to a line of works [21; 15; 14] that showed near-optimal guarantees when the sensing utility function satisfies the diminishing returns property called *submodularity*. This approach has been shown to be the current most performant method for the application of contamination detection in waterways [24].

In summary, all three types of reasoning tasks would benefit from an accurate and detailed description of the air pollution on the street level. It is also important that the model is reasonably efficient, as there is little value in predicting pollution dispersions that happened in the distant past. Therefore, an efficient spatial abstraction is essential in a model that tackles these reasoning tasks.

4 A Region-Based Model

Current numerical models of spatial phenomena have almost exclusively relied on fixed grids where all grid cells have the same size and shape. This regularity greatly simplifies modeling, as all cells can be modeled in the same way. When modeling phenomena in the environment, however, a big disadvantage is that grid cells rarely match regions where parameters behave in a homogenous way, such as streets, buildings, rivers and lakes, fields, forests, or parkland. Thus, either the grid is made extremely fine, or each cell contains a mixture of different regions.

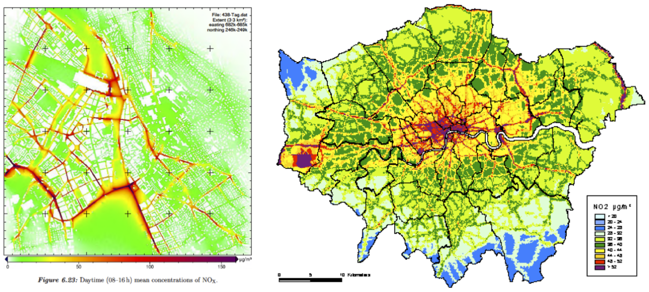


Figure 4: Outputs from microscale model GRAL-Sys for the city of Zurich (left) and ADMS-Urban for London (right)

In air pollution dispersion, evidence points to strong local variability of pollution levels; from peaks in streets, factories and buildings, pollution levels drop very rapidly to much lower values on different types of land. The cost of computation invariably increases when the grid-cells get smaller. These phenomena cannot be captured by models based on regular grids unless the grids are made unmanageably fine.

Some existing physical models do map the dispersion of air pollution with very fine grids at the microscale. Figure 4 illustrates the output from some of the current leading microscale models which look at air pollution at a street level. The left part of the figure came from a snapshot the pollution levels in the city of Zurich from the model Gral-Sys [12; 22], and the right part came from the model ADMS-Urban [7] that looks at the city of London. In both models the streets are clearly distinguishable. Gral-Sys accomplished this by creating very fine grid-cells with 3 meters mesh-width. This leads to limitations to the size of the modeled area, which in the work of Kehl [12] was the 3.3 km^2 of Zurich city centre. In contrast, ADMS uses a technique that it called "intelligent gridding", one that create extra grid points to follow the shape of the roads. The extra grid points along the street boundaries allow the pollution to be visible from the output of physical simulations.

Both model show that the concentration of pollutants is relatively homogenous within a street compared to its neighboring regions. Therefore, one intuitive way to bypass the problem of having finer grids is to adopt a region-based approach where pollutants behave homogeneously within a region. Such spatial abstraction exploits structures in the spatial information and enables more efficient reasoning. Therefore, we propose the frameworks for building a qualitative, region-based model for air pollution dispersion.

4.1 Physical Regions

In the previous example, we show that there is a case for treating the streets as special regions to its surroundings. Similarly, we can model other physical regions where the internal pollution level is likely to be similar. These regions form the basic building blocks of the model. A possible way of identify such regions is to partition the modeling region according to land use, such as streets, residential areas, parks, industrial

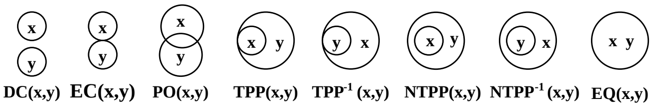


Figure 5: Topological relations between regions in the Region Connection Calculus (RCC8)

plants, airport, etc. Each region has its unique emission characteristics, and its pollution level is also influenced by the transport and diffusion of pollution to and from neighboring regions. Such regions may be obtained with vector-based GIS data, and tools such as OpenStreetMaps [11].

Apart from the physical regions denoted by land use, other regions of interest are those of certain pollution concentration. These may be defined qualitatively by a threshold concentration, such as "the region of pollution level X or higher". This then forms a naturally nested containment relation with regions of other pollution concentrations, as the region will always contain "the region of pollution level Y or higher" for any $Y > X$. Note that they are general regions in the sense of the Region Connection Calculus (RCC8) [25], and may exist in many physically separate parts. Each contiguous part may be considered a *qualitative physical field* [20].

4.2 Region-Region Relations

Given the physical regions of streets, parks and airports etc., a measurement taken on a particular street can then be interpreted as a *connection* primitive between the street region and the region of the measured level of pollution. One can then build the RCC8 relations (Fig. 5) between the street and the pollution region from existing decentralized algorithms [8]. Additional inference may be possible for some extra assumptions. For instance, if we make the reasonable assumption that the pollution regions are larger than atomic street region, then we can infer that the street is either *partially overlapped, tangential proper part, non-tangential proper part or equal* of the pollution field. The forward reasoning task that infers the relations of the pollution to neighboring regions would then be dependent on both the information on the source of emissions and the smoothing criteria on the degradation of pollution levels with respect to distance and wind direction. It would then involve reasoning with more than a singular aspect of qualitative spatial information, and techniques for combining multiple spatial calculi may be useful in this context [18; 27].

In addition to performing logical inferences with spatial relations, it is clear that many queries in the outlined reasoning tasks involve a certain degree of probabilistic reasoning. At this stage, detailed air pollution measurements are rare. However, one can learn trends of pollution dispersion between regions from numerical simulations of existing models. The knowledge can then be encoded as a graphical model, where the nodes denote the level of pollution within a region, and edges denote the influence of pollution levels between neighboring regions. The graphical model then provides a probabilistic reasoning framework for the relevant forward, backward and meta reasoning tasks.

5 Discussions

We propose to model atmospheric phenomena based on regions that mirror the actual use of the land. In such a model, the values of relevant pollution parameters are expected to be much more homogeneous than in a grid-based model. Consequently this would reduce the variance of the statistical relations used to model pollution behavior and leads to a more accurate model.

We have distinguished forward and backward reasoning tasks. Both tasks involve probabilistic inference along the adjacencies of the cells in the model. This inference is complicated by the fact that the graph that models these adjacencies contains many cycles that may make exact probabilistic inference by known methods intractable. An important open question is whether the spatial nature of the problem can be exploited for tractable probabilistic inference. Let us consider a few ideas how this might be possible.

The first property is that the adjacency graph is planar and chordal, a fact that allows polynomial-time versions of many graph-theoretic problems, in particular finding cluster trees. Another possibility is to combine inference from several region-based models with different resolutions, region structure or granularity that constrain one another. This might allow use of theories such as the region connection calculus. It is also possible to model the underlying physical phenomena in terms of regions. For examples, in a process of dispersion holes in a cloud of pollution tend to disperse, and regions tend to grow while decreasing in concentration. Wind moves a region uniformly in a certain direction. This might allow the use of theories such as process grammars [17; 20] to model the transformation of regions and reduce the uncertainty in inference.

The problem of air pollution dispersion clearly has a strong spatial component. It will be interesting to see what role existing theories of spatial reasoning can play in solving these inference problems, and what changes can be made to make some of the existing methods applicable. We expect applications in the modeling and monitoring of environmental processes to be a major driver for future research in qualitative spatial reasoning.

Acknowledgements

The work was supported by the OpenSense project (reference number 839_401) under the Nano-Tera initiative (http://www.nano-tera.ch).

References

[1] *Air Quality and Health, Fact sheet No. 313.* World Health Organization, 2008.

[2] *Comprehensive Air quality Model with extensions.* http://www.camx.com, 2011.

[3] K. Aberer, S. Sathe, D. Chakraborty, A. Martinoli, G. Barrenetxea, B. Faltings, and L. Thiele. Opensense: Open community driven sensing of environment. In *ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS)*, 2010.

[4] B. Bessagnet, L. Menut, G. Curci, A. Hodzic, B. Guillaume, C. Liousse, S. Moukhtar, B. Pun, C. Seigneur, and M. Schulz. Regional modeling of carbonaceous aerosols over europe—focus on secondary organic aerosols. *J Atmos Chem*, 61(3):175–202, 11 2008.

[5] D. Byun and K. L. Schere. Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (cmaq) modeling system. *Applied Mechanics Reviews*, 59(2):51–77, 2006.

[6] R. J. Carroll, R. Chen, E. I. George, T. H. Li, H. J. Newton, H. Schmiediche, and N. Wang. Ozone exposure and population density in harris county, texas. *Journal of the American Statistical Association*, 92(438):392–404, 06 1997.

[7] R. N. Colvile, N. K. Woodfield, D. J. Carruthers, B. E. A. Fisher, A. Rickard, S. Neville, and A. Hughes. Uncertainty in dispersion modelling and urban air quality mapping. *Environmental Science & Policy*, 5(3):207–220, 6 2002.

[8] M. Duckham, D. Nussbaum, J. Sack, and N. Santoro. Efficient, decentralized computation of the topology of spatial regions. *Computers, IEEE Transactions on*, PP(99):1, 2010.

[9] T. Godish. *Air Quality*. CRC Press, 4 edition, 2003.

[10] D. Golovin, M. Faulkner, and A. Krause. Online distributed sensor selection. In *Proc. ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.

[11] M. M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7:12–18, October 2008.

[12] P. Kehl. *GPS based dynamic monitoring of air pollutants in the city of Zurich, Switzerland*. PhD thesis, ETHZ, 2007.

[13] B. M. G. Kibria, L. Sun, J. V. Zidek, and N. D. Le. Bayesian spatial prediction of random space-time fields with application to mapping pm2.5 exposure. *Journal of the American Statistical Association.*, 97(457):112–124, March 2002.

[14] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *International Symposium on Information Processing in Sensor Networks (IPSN)*, 2006.

[15] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, 2008.

[16] T. Larson, S. B. Henderson, and M. Brauer. Mobile monitoring of particle light absorption coefficient in an urban area as a basis for land use regression. *Enviromental Science and Technology*, 43(13):4672–4678, 2009.

[17] M. Leyton. A process-grammar for shape. *Artificial Intelligence*, 34(2):213–247, March 1988.

[18] J. J. Li and J. Renz. In defense of large qualitative calculi. In M. Fox and D. Poole, editors, *AAAI*. AAAI Press, 2010.

[19] Y. Liu, H. Guo, G. Mao, and P. Yang. A bayesian hierarchical model for urban air quality prediction under uncertainty. *Atmospheric Environment*, 42(36):8464–8469, 11 2008.

[20] M. Lundell. A qualitative model of physical fields. In *AAAI/IAAI, Vol. 2*, pages 1016–1021, 1996.

[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14(1):265–294, 1978.

[22] D. Oettl, P. J. Sturm, M. Bacher, G. Pretterhofer, and R. A. Almbauer. A simple model for the dispersion of pollutants from a road tunnel portal. *Atmospheric Environment*, 36:2943–2953, 2002.

[23] N. Oreskes, S.-F. K., and B. K. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263(5147):641–646, Feb 1994.

[24] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, and et al. The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management*, 134(6):556, 2008.

[25] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Principle of Knowledge Representation and Reasoning: Proceedings of the 3^{rd} International Conference (KR'92)*, pages 165–176, 1992.

[26] S. Sillman. The relation between ozone, nox and hydrocarbons in urban and polluted rural environments. *Atmospheric Environment*, 33:1821– 1845, 1999.

[27] S. Wölfl and M. Westphal. On combinations of binary qualitative constraint calculi. In C. Boutilier, editor, *IJCAI*, pages 967–973, 2009.

[28] J. Yin, D. H. Hu, and Q. Yang. Spatio-temporal event detection using dynamic conditional random fields. In *proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1321–1326, Pasadena, CA, USA, July 2009.

[29] E. Zarate. *Understanding the origins and fate of air pollution in Bogota, Colombia*. PhD thesis, EPFL, 2007.

Dynamic signal segmentation for activity recognition

Simon Kozina, Mitja Luštrek, Matjaž Gams
Jozef Stefan Institute, Department of Intelligent Systems
Jamova cesta 39, 1000 Ljubljana, Slovenia
{simon.kozina, mitja.lustrek, matjaz.gams}@ijs.si

Abstract

Activity recognition is an essential task in many ambient assisted living applications. Activities are commonly recognized using data streams from on-body sensors such as accelerometers. An important subtask in activity recognition is signal segmentation: a procedure for dividing the data into intervals. These intervals are then used as instances for machine learning. We present a novel signal segmentation method, which utilizes a segmentation scheme based on dynamic signal partitioning. To validate the method, experimental results including 6 activities and 4 transitions between activities from 11 subjects are presented. Using a Random forest algorithm, an accuracy of 97.5% was achieved with dynamic signal segmentation method, 94.8% accuracy with non-overlapping and 95.3% with overlapping sliding window method.

1 Introduction

Activity recognition using on-body sensors is required for many ambient assisted living applications. This paper focuses on an important subtask in activity recognition: signal segmentation, the process of dividing the data into intervals. On-body sensors are collecting and continuously outputting streams of data. These streams are used to recognize the user's current activity.

The problem tackled in this paper is how to segment the data into intervals most suitable for activity recognition. Most approaches use overlapping and non-overlapping sliding windows, which means that the data is divided into intervals of fixed length. On each interval features are computed and then used as an instance for activity recognition. We present a novel method for signal segmentation, which attempts to match the intervals to the borders between different activities.

Dynamic signal segmentation method is based on searching for significant differences between consecutive data samples. A significant difference is determined by a dynamically computed threshold. It is updated whenever a new data sample is received, and adapts to changes in the data stream.

The paper is structured as follows. Section 2 gives an overview of related work on activity recognition with on body sensors. Section 3 describes two signal segmentation methods: the sliding window method and the novel dynamic signal segmentation method. Section 4 lists the attributes extracted from the input data that are fed into the machine learning algorithms. Section 5 presents the experiments in which the signal segmentation methods are compared. Finally, Section 6 concludes the paper and outlines the future work.

2 Related work

Various sensors are used for activity recognition: accelerometers and gyroscopes, real-time locating systems (RTLS) [Mircevska *et al.*, 2009], cameras [Qian *et al.*, 2004; Vishwakarma *et al.*, 2007] and environmental sensors [Zhan *et al.*, 2007]. Cameras pose a (real or perceived) threat to privacy, RTLS are expensive, and both require installation in the apartment as do environmental sensors. Because of that accelerometers and/or gyroscopes, which are inexpensive and portable, are most commonly for activity recognition, although in some situations they are not unobtrusive to the user.

Koskimaki et al. [2009] used a single wrist worn accelerometer to collect the acceleration and angular speed data. They defined four activities and one class value to denote ”other” activities. Using the overlapping sliding window method with the window size of half a second, almost 90% accuracy was achieved. Ravi et al. [2005] tried to recognize eight activities, using one accelerometer placed on the abdominal area. Two of these activities are the same as in our testing scenario, others are similar. They divided the signal into overlapping five-second windows, and achieved accuracy of 73.3%. Mannini and Sabatini [2010] tried to recognize seven activities using five accelerometers placed on the body. Four of these seven activities are identical to the ones in our scenario. They achieved 98.5% accuracy when using the 6.7 second overlapping sliding window method. None of this researches had tackled the problem of recognizing transitions between activities.

Bifet and Gavalda [2007] have presented a segmentation algorithm that is recomputing the size of the sliding window accordingly to the rate of change observed from the data. The window is growing when the data is stationary, and shrinking when change is taking place. In order to work, the algorithm has to be integrated into a machine learning algorithm. Nunez et al. [2007] also presented an incremental decision tree algorithm, which is adapting sliding window size to portions of the target concept. Each leaf of the decision tree holds a time window and a local performance measure. When the performance of a leaf decreases, the size of its local window is reduced. Some limitation may arise when dealing with large amount of data as the decision tree has to be updated when new examples are available.

3 Signal segmentation

Two methods are typically used to evaluate a stream of data for activity recognition. The first method is to use a single data point to determine the current activity. This method is not commonly used as the information gathered from a single data point is in most cases not sufficient for activity recognition. The second method involves signal segmentation. This means that consecutive sensor data are grouped. In contrast to the first method, multiple data points are used to determine the current activity. Using multiple data points allows more information to be extracted from the data, so the activities can be determined more accurately. However, the question of how exactly to group consecutive data needs to be tackled.

Some common methods for signal segmentation are overlapping and non-overlapping sliding windows, and signal spotting [Junker *et al.*, 2004; Benbasat *et al.*, 2000; Amft *et al.*, 2005]. In this section a new method for signal segmentation is proposed - dynamic signal segmentation method.

3.1 Sliding window method

The sliding window method is the most commonly used signal segmentation method for activity recognition with machine learning. The sliding window method accumulates sensor data over a fixed time window. Features are computed over one time window and are used as an instance for a learning/testing set. Two approaches are commonly used for data segmentation with sliding windows. The first approach is non-overlapping sliding windows, where consecutive time windows do not share common data samples. The second approach is overlapping sliding windows, which share common data samples between time intervals; for example, two consecutive time windows may have 50% of data samples in common.

3.2 Dynamic signal segmentation

Dynamic signal segmentation method is a novel method for signal segmentation. In principle the method can be used on any domain where a stream of sensor data has to be processed and the data has to be divided into segments. We tested the usability of the method on an acceleration-based domain for the purpose of activity recognition. We assume that, in addition to the acceleration data, the method can also be used for ECG or thermometer data, but it has not been tested yet.

The method searches for a significant change between consecutive data samples and divides the data into intervals at that point. The significant change is defined as a sequence of consecutive data samples where the values are in descending order, and the difference between the maximum and the minimum element in the sequence is larger than a threshold. The condition that the values should be in descending order is specific to our problem of accelerometer-based activity recognition, since each strong deceleration is typically quickly followed by an acceleration. Considering both would thus lead to dividing the data twice when a significant change occurs. For other types of data, both descending and ascending order should be considered.

Examples of sequences with the values in descending order are shown in Figure 1 denoted with a dotted line. When a set of descending data samples is found, the last element of this sequence is used as an ending point of one and starting point of the next interval. Therefore, the length of an interval is changing dynamically, as opposed to the sliding window, where it is set to a specific length. The features computed from each of the intervals are used as an instance for machine learning.

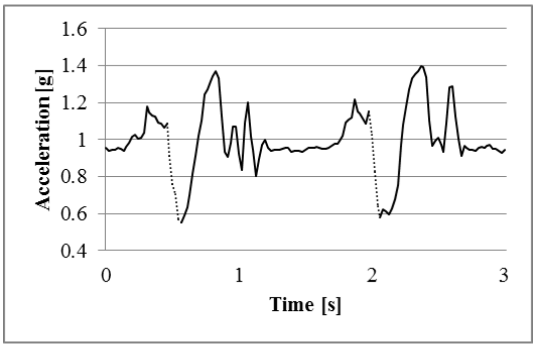


Figure 1: Descending sequence, denoted with dotted line, on a three-second time window.

The threshold, at each data sample, is computed from previous N data samples. Therefore, an initialization process of the algorithm uses N data samples to compute the first threshold. These data samples are used to compute the average minimum (avg_{min}) and average maximum (avg_{max}) values. The average minimum value is defined as the average of the first smallest ten percent of values in the last N data samples. The average maximum value is defined as the largest ten percent of values. In Figure 2, maximum values are denoted with circles and minimum values with squares. An average of each of these points is computed.

When these two values are obtained, the threshold can be computed:

$$threshold = (avg_{max} - avg_{min}) \cdot C$$

where $C \in [0, 1]$ is a constant selected prior to the start of the algorithm. This approach for setting the threshold is better than using only the minimum and the maximum values on an interval . For example, if there are some errors in the data, such as abnormal high or low peaks, these will be partially corrected with the other values for averaging minimum or maximum. The constant C can be computed from a learning dataset as follows:

$$C = \frac{\frac{1}{n} \cdot \sum_{i=1}^{n} a_i}{a_{max} - a_{min}}$$

The value n denotes the number of data samples in a learning dataset, a_{min} and a_{max} are the minimum and the maximum

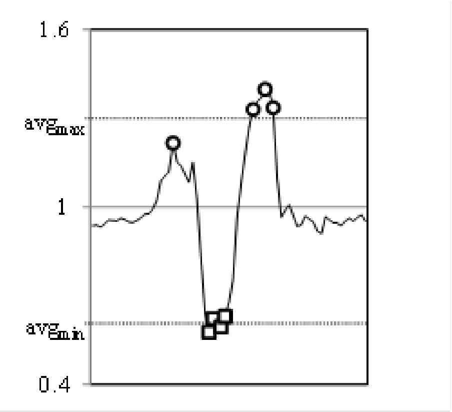


Figure 2: Four minimum and four maximum points on 40 data sample interval.

accelerations on the interval and a_i is the length of an acceleration vector at data sample i. Another way to set the constant C would be to tune it by running the dynamic signal segmentation on a separate dataset.

In addition to selecting the appropriate constant C, the developer has to determine which input signal should be used for threshold computation. This depends on a diversity of input signals and the domain. Our experiments were done using two 3-axial accelerometers attached to the left thigh and the chest. If, for example, we were driving a car, vertical acceleration would stay identical for almost all the time and same would apply for the threshold. On the other hand, if several activities, like walking, running, etc., were performed, the vertical acceleration would probably be the best choice as it would provide maximum information about activities.

A general solution when using one 3-axial accelerometer would be to use the length of the acceleration. However, when using more than one accelerometer, like in our example, the input signal for a threshold computation should be derived from multiple accelerometers. In our experiments the input signal for threshold computation was the arithmetic mean of lengths from both accelerometers and was derived as follows:

$$A = \frac{1}{2} \cdot \sqrt{a_x^2 + a_y^2 + a_z^2} \cdot \sqrt{b_x^2 + b_y^2 + b_z^2}$$

where $\vec{a} = [a_x, a_y, a_z]$ and $\vec{b} = [b_x, b_y, b_z]$ are acceleration vectors from both accelerometers.

4 Feature computation

In our experiments, once the stream of data was segmented either by the sliding window method or the dynamic signal segmentation, we used the same procedure to compute the features activity recognition by machine learning. Some additional information could be derived when using dynamic signal segmentation method, for example the time duration of an interval. However, in order to have comparable results, these additional attributes were not used.

As stated above, two accelerometers were used in our experiments. The following attributes were derived separately for the acceleration vectors from each of the accelerometers:

- The average length of the acceleration vector within the window, which could be of fixed size or computed with dynamic signal segmentation.
- The variance of the length of the acceleration vector. The variance within the window was defined as follows:

  $$\delta^2 = \frac{\sum_{i=1}^{N}(a_i - \overline{a})^2}{N}$$

  where N is the number of acceleration data within the window, is the length of the i-th acceleration vector and $\overline{a}$ is the average length of the acceleration of all previous samples.
- The average acceleration along the x, y and z axes.
- The maximum and the minimum acceleration along the x, y and z axes.
- The difference between the maximum and the minimum acceleration along the x, y and z axes.
- The angle of change in acceleration between the maximum and the minimum acceleration along the x, y and z axes. It was defined as follows:

  $$\Omega = \arctan\left(\frac{a_{max} - a_{min}}{t_{a_{max}} - t_{a_{min}}}\right)$$

  where a_{max} and a_{min} are the maximum and minimum acceleration along one axis within the window, and $t_{a_{max}}$ and $t_{a_{min}}$ are the times when they were measured. Figure 3 shows the principle of computing the angle of change in acceleration in one time window.If $t_{a_{max}} > t_{a_{min}}$ the angle is positive, otherwise the angle is negative.

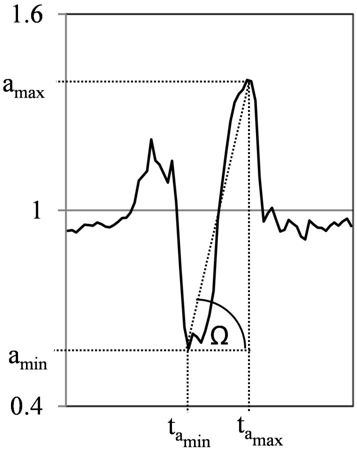


Figure 3: The angle of acceleration in a time window.

- The orientation of the accelerometer. We assumed that the acceleration vector $a = [a_x, a_y, a_z]$, which consists of the accelerations along the three axes of the accelerometer, generally points downwards (in the direction of the Earth's gravity). Let z be the axis pointing downwards when the accelerometer is in upright position. The angle ϕ between the acceleration vector and the z axis thus indicates the person's orientation, and was computed as follows:

$$\phi = \arccos\left(\frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right)$$

To sum it up, 18 attributes were computed for each accelerometer. The final attribute was the angle between accelerometer vectors. It was obtained by computing the scalar product of vectors, normalized to their length:

$$\Theta = \arccos\left(\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\cdot\|\vec{b}\|}\right)$$

Vectors $\vec{a}$ and $\vec{b}$ each represent the acceleration from both accelerometers. One instance in learning/testing set was thus represented with an attribute vector consisting of 37 attributes.

5 Experiments

We compared the performance of the signal segmentation methods on a scenario recorded by 11 healthy volunteers (7 male and 4 female), 5 times by each. Three of these recordings (2 male and 1 female) were used to create the training set and the other 8 were used to create the test set.

The scenario included 6 activities and 4 transitions. Transitions are defined as short actions between two activities. The activities and transitions are listed in Table 1.

	Activity		Transition
1.	standing	7.	falling
2.	walking	8.	sitting down
3.	on all fours	9.	standing up
4.	sitting	10.	lying down
5.	sitting on the ground		
6.	lying		

Table 1: Activities and transitions

To classify new instances we trained a classifier using the Random forest algorithm on our training set. The algorithm was implemented in Weka machine learning suite [Hall *et al.*, 2009]. The constant C, used by dynamic signal segmentation, was set to 0.4. This value was obtained by testing the dynamic signal segmentation algorithm on a different dataset than used for this paper. The same procedure was used to determine the value N for the number of data required for threshold computation. The value N was set to 100. The length of the sliding window was set to 1 second.

Each data sample in our training and test sets was labeled with an activity, whereas both the sliding window method and dynamic signal segmentation recognize the activity of a whole time interval. For training and testing purposes we thus considered the true label of an interval to be the majority if the labels of all the data samples in the interval.

5.1 Results

We compared the results of dynamic signal segmentation to overlapping and non-overlapping sliding window methods. We divided the scenario into two separate problems. The first problem was to recognize only the activities, and the second problem was to recognize both the activities and the transitions. Both problems were tested on the same dataset with one difference, in the first case the transitions were excluded from the training and test sets. The performance of the three methods was measured in terms of classification accuracy. Table 2 shows the results of all the methods.

	Methods		
	Non-overlapping sliding window	Overlapping sliding window	Dynamic signal segmentation
Activities	94.8%	95.3%	97.5%
Activities and transitions	89.0%	89.6%	92.9%

Table 2: All the methods compared using just activities and activities with transitions.

Based on these results we can conclude that there is a difference between non-overlapping and overlapping sliding windows, compared to dynamic signal segmentation method. On the other hand, the difference between the two sliding window methods and the dynamic segmentation method on the problem without transitions is 2.9 and 2.4 percentage points, and when the transitions are included it is 3.9 and 3.3 percentage points.

The quality of dynamic signal segmentation method depends on the threshold computation. For example, if the value N, which determines the number of previous samples used for threshold computation is set too high, the threshold does not update fast enough. As a consequence, transitions between activities are overlooked. On the other hand, if the number of previous samples is set too low, the threshold is over-fitted to the acceleration data. As a consequence data is fragmented and not appropriate for proper activity recognition. Figure 4 shows the threshold values in a single recording. We can notice that the threshold is changing according to activities. When a static activity, e.g. lying or sitting, is performed the algorithm updates the threshold to a small value, but while a person is walking, the threshold is updated to a higher value.

If the threshold computation was not implemented in the algorithm, the data stream would be divided using a simple, predefined threshold. Static activities like standing and sitting would not be separated using this approach. Another problem would be the determining the threshold.

As mentioned in section 3.2, the avg_{min} and avg_{max} values are used to compute the threshold. These values are obtained by computing the mean value of minimum and maximum ten percent of values. Other approaches can be used to obtain these values. Instead of the mean value, we could

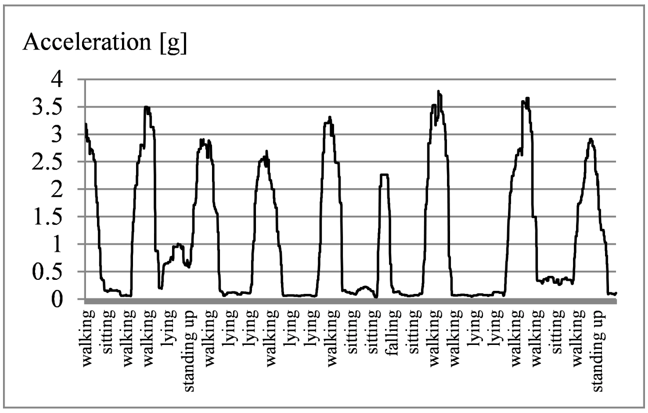


Figure 4: Changing of threshold values in a scenario.

compute a median value or use only minimum and maximum values. Results of these approaches are shown in Table 3.

	Mean	Median	Only min and max values
Accuracy	97.5%	96.9%	96.1%

Table 3: Accuracy of different approaches for computing avg_{min} and avg_{max} values.

By analyzing the confusion matrix [Kohavi and Provost, 1998] for the experiment excluding the transitions between activities (Table 4), we can conclude that the accuracy of all the activities except *on all fours* is above 90%. *On all fours* is usually confused with lying on the stomach, because the sensor orientation are the same (parallel with the ground and facing the ground). Another reason for poor performance, when recognizing *on all fours* activity, can be found in a small number of instances of this activity in the learning set (only 0.6 %). One of the solutions would be to change our scenario accordingly to extend the recording time when a person is on all fours.

	Sta	Walk	On4	Sit	SitG	Ly
Sta	95.2%	4.8%	0	0	0	0
Walk	2.0%	97.5%	0.4%	0	0	0.1%
On4	3.5%	10.6%	51.8%	0	0	34.1%
Sit	0	0.3%	0	95.9%	3.4%	0.4%
SitG	0	0.2%	0	5.2%	92.6%	2%
Ly	0	0.1%	0.1%	0.1%	0	99.7%

Table 4: Confusion matrix for activity recognition. Standing (Sta), walking (Walk), on all fours (On4), sitting (Sit), sitting on the ground (SitG), lying (Ly).

The results of activity recognition with transitions between activities are presented in Table 5. The overall accuracy of activities has decreased as there are four more classes that have to be predicted. Recognition of transitions is 62.2% accurate. This could have occurred due to the fact that the length of the transitions is much shorter than the length of activities; therefore the labels may not correspond perfectly to the data in these short intervals. This can happened because of several reasons: mislabeling, sometimes it is hard to determine the correct limit between transition and activity even by hand.

Activity/transition	Accuracy
standing	90.5%
walking	96.9%
on all fours	23.3%
sitting	96.9%
sitting on the ground	93.8%
lying	98.7%
falling	42.1%
sitting down	49.5%
standing up	69.7%
lying down	41.2%

Table 5: Accuracy of activity and posture recognition.

6 Conclusion

We have presented a novel method for signal segmentation, which is an important subtask in activity recognition. Signal segmentation is a process of dividing the stream of data into groups and is used for dividing of acceleration data, gyroscope data etc. Common methods for signal segmentation are overlapping and non-overlapping sliding windows. These methods divide the data into fixed time intervals. Our method, dynamic signal segmentation, is dividing the data based on patterns in data stream. The method is searching for significant changes in the data based on the threshold. The threshold is updated with every new data sample and is changing dynamically, according to the signal. When such a change is found in the data, it is used as a limit between consecutive intervals. An interval is then used as an input for machine learning.

We compared the performance of common signal segmentation methods with our dynamic segmentation method on a scenario recorded by 11 healthy volunteers (7 male and 4 female). Each scenario included six activities and four transitions between activities. Using the Random forest algorithm 97.5% accuracy was achieved with dynamic signal segmentation, 95.3% with overlapping and 94.8% with non-overlapping sliding window method. We have also showed that transition have negative effects on accuracy of activity recognition. All the methods had lower accuracy with transitions instances in learning/testing set.

There are several directions for future work. The first is the development of more acceleration related attributes and augment them with feature selection techniques. The second direction is automatic labeling of the data: an algorithm for semi-supervised learning which would group similar intervals together into clusters. User would only need to manually label these clusters after the experiments. The third direction is in improvement of the existing algorithm with techniques for statistical data analysis.

References

[Amft *et al.*, 2005] Oliver Amft, Holger Junker, and Gerhard Troster. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, pages 160–163, Washington, DC, USA, 2005. IEEE Computer Society.

[Benbasat *et al.*, 2000] Ari Y. Benbasat, Joseph A. Paradiso, Stephen A. Benton, Ari Yosef Benbasat, and Ari Yosef Benbasat. An inertial measurement unit for user interfaces, 2000.

[Bifet and Gavalda, 2007] Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *SIAM International Conference on Data Mining*, 2007.

[Hall *et al.*, 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[Junker *et al.*, 2004] Holger Junker, Paul Lukowicz, and Gerhard Trster. Continuous recognition of arm activities with body-worn inertial sensors. In *Proceedings of the Eighth International Symposium on Wearable Computers*, ISWC '04, pages 188–189, Washington, DC, USA, 2004. IEEE Computer Society.

[Kohavi and Provost, 1998] Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30(2-3), 1998.

[Koskimaki *et al.*, 2009] Heli Koskimaki, Ville Huikari, Pekka Siirtola, Perttu Laurinen, and Juha Roning. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. *Mediterranean Conference on Control and Automation*, 0:401–405, 2009.

[Mannini and Sabatini, 2010] Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[Mircevska *et al.*, 2009] Violeta Mircevska, Mitja Lustrek, Igone Vlez, Narciso Gonzlez Vega, and Matjaz Gams. Classifying posture based on location of radio tags. *Ambient Intelligence and Smart Environments*, 5:85–92, 2009.

[Nunez *et al.*, 2007] Marlon Nunez, Ral Fidalgo, and Rafael Morales. Learning in environments with unknown dynamics: Towards more robust concept learners, 2007.

[Qian *et al.*, 2004] Gang Qian, Feng Guo, Todd Ingalls, Loren Olson, Jody James, and Thanassis Rikakis. A gesture-driven multimodal interactive dance system. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 27–30, 2004.

[Ravi *et al.*, 2005] Nishkam Ravi, Nikhil Dandekar, Prreetham Mysore, and Michael L. Littman. Activity Recognition from Accelerometer Data. *American Association for Artificial Intelligence*, 2005.

[Vishwakarma *et al.*, 2007] Vinay Vishwakarma, Chittaranjan Mandal, and Shamik Sural. Automatic detection of human fall in video. In *Proceedings of the 2nd international conference on Pattern recognition and machine intelligence*, PReMI'07, pages 616–623, Berlin, Heidelberg, 2007. Springer-Verlag.

[Zhan *et al.*, 2007] Yi Zhan, Shun Miura, Jun Nishimura, and Tadahiro Kuroda. Human activity recognition from environmental background sounds for wireless sensor networks. In *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, pages 307–312, 2007.

Possible-World and Multiple-Context Semantics for Common-Sense Action Planning

Maria J. Santofimia,
Computer Architecture and Network Group,
School of Computing Science.
University of Castilla-La Mancha, Spain

Scott E. Fahlman,
Language Technologies Institute,
Carnegie Mellon University,
Pittsburgh, PA 15213, USA

Francisco Moya, and **Juan Carlos Lopez**
Computer Architecture and Network Group,
School of Computing Science.
University of Castilla-La Mancha, Spain

Abstract

Event management and response generation are two essential aspects of systems for Ambient Intelligence. This work proposes handling these issues by means of an approach with which to model and reason about actions and events which, under the umbrella of a philosophical and common-sense point of view, describes what actions and events are, how they are connected, and how computational systems should consider their meaning. This work uses the Scone Knowledge-Base (KB) system with which to both reason about and model the context and the related events. This approach is capable of generating ad-hoc responses, in terms of actions to be performed, supported by the knowledge about the possible-world and multiple-context semantics.

1 Introduction

It is a well known fact that intelligent systems struggle with innovation and change whereas humans seem to perform well in most cases. Why is this? or what lies beneath this human skill? The response of cognitive science to these questions points out the human ability to handle and reason about *possible worlds*. The notion of possible worlds is used here to refer to those states of affairs or "worlds" which, given an event or a premise, are true in all the worlds considered possible. For example, to state an analogy with the Sherlock Holmes stories, the true facts are provided by the clues in the case. Holmes therefore considers all the *worlds* in which the given premises are true. Note how new clues might lead Holmes to reject worlds that were previously considered to be plausible.

Closely related to the notion of possible worlds, the *context* concept is here understood as the set of facts or propositional knowledge that describe a specific state of the world, in the same way that J. Allen's refers to the *world* concept in [Allen, 1984]. This concept is represented by a description set of both the static and dynamic aspects of the world, thus modeling what is known about the past, present, and future. The J. Allen nomenclature can be used to state that the static aspects of the world are easily captured as *properties* while the dynamic aspects are captured as *occurrences* or *events*.

The notion of *multiple contexts* is connected with that of possible worlds and refers to the mechanism used to concurrently handle the possible-world semantics, at the knowledge-base level. The multiple-context mechanism provides a mean to model actions and events by describing the state of the world before, during, and after the action or event takes place. For example, a `person moving` event gives rise to a new world-state in which the person that moves changes location. If a person moves from the kitchen to the living room, the world-state, before the event takes place, is described by the person being present in the kitchen, while the world-state after the event has taken place is described by the fact that the person is then located in the living room. However, if that person, before moving, approaches an object and picks it up, where is the object after the moving event? Moreover, what will happen if it is a slippery object? The purpose of this work is to model and reason about actions and events, while considering those scenarios that involve the inference of implicit, non-deterministic or delayed effects of events. The following scenarios, extracted from [Mueller, 2006], illustrate those situations that require special attention:

1. In the kitchen, Lisa picked up the newspaper and walked into the living room.
2. Lisa put a book on a coffee table and left the living room. When she returned, the book was gone.
3. Jamie walks to the kitchen sink, puts the stopper in the drain, turns on the faucet, and leaves the kitchen.
4. Kimberly turns the fan's power switch to "on".

In the first scenario, it is easily inferred that since Lisa was intially in the kitchen, she picked up the newspaper while she was there and then took it into the living room. It is also obvious to us that if Lisa is in the kitchen she cannot be in any other room at the same time, since we are considering rooms as non-overlapping spaces in a house. With regard to the second scenario, we can easily infer that if Lisa left the living room, she is no longer there, and that if the book is not there when she returns, something must have happened because things tend to remain in the state they are unless a partiuclar event affects them. The "*frame problem*" concerns determining those things that can be assumed to stay the same from one moment to another. In the third scenario we easily conclude that, after a while, the water will start spilling onto the floor. Finally, with regard to the question of what will happen in the fourth scenario, we can assume that if everything works as it is supposed to, the fan will start up.

1.1 Action Planning in Ambient Intelligence

The objective of this work is to propose an approach for action planning with endowed capabilities to handle the non-trivial aspects of common-sense reasoning. The innovative aspect of this work lies in the heuristics provided by common-sense knowledge concerning actions and events captured in the proposed model.

This work focuses its attention on planning in Ambient Intelligence. Note that Ambient Intelligence environments are characterized by: a) the multiple sources of change affecting the context; b) the device availability aspects that cannot be determined beforehand; and c) the expection of intelligent and autonomous reactions in response to context changes. These aspects, along with the nonlinearity of the problems involved in Ambient Intelligence, are responsible for the small amount literature found in the field.

The strategy followed here consists of: a) proposing a model for actions and events that captures the common-sense knowledge involved; b) representing possible worlds by means of a context activation scheme; c) modeling actions and events in terms of the multiple contexts that describe the world before, during, and after the action or event takes place; d) and finally, rather than considering primitive and compound tasks, in an HTN-like style (Hierarchical Task Network) [Erol *et al.*, 1994], we consider actions that are provided by services and those which are not. By doing this, the proposed approach addresses the device dynamism that characterizes Ambient Intelligence environments.

The remainder of this paper is organized as follows: First, in Section 2 a model for actions and events is proposed and formalized. Section 3 describes how the proposed model is represented in Scone, emphasizing the multiple-context and context activation scheme. Section 4 demonstrates how the key issues of common-sense have been addressed. Section 5 presents an action planning strategy with common sense. A proof of the benefits derived from considering common-sense knowledge as a constituent part of an action planning approach is demonstrated with a case scenario. Finally, Section 6 shows the conclusions drawn from the work presented herein.

2 Modeling actions and events

Actions and events have commonly been treated as being equivalent, or as having the slight difference of considering actions as events which have been intentionally generated [Hommel *et al.*, 2001]. On the contrary, the theory of action for multi-agent planning [Georgeff, 1988] advocates for a distinction between actions and events, although it hints that actions are accomplished by agents in their endeavor to achieve a goal.

Davidson's theories, particularly those regarding the philosophy of action, also identify actions with events, as is argued in [Davidson, 1963]. Actions are described as a combination of two views. On the one hand, actions can be seen as causal explanations of body movements and on the other hand, actions can also be seen as the justifying reason that leads the action to take place. Davidson considers events to be equivalent to actions. The sole difference is that when an action is considered as an event, it is re-described in terms of its effects.

The model proposed here for actions and events adopts the Davidsonian view. It should be highlighted that Cyc [Lenat, 1995], through its language CycL, represents actions and events using a Davidsonian approach. Actions are described as events but are carried out by an agent. The approach implemented in Scone has been extended to include the notion of primary reasons for an action, along with its temporal and location aspects.

Apart from the concept of action and event that concern us here, some other relevant entities must also be considered in relation to actions and events so as to capture their semantics. The following definitions state the foundation of the proposed model for actions and events:

Definition 1. A Context is a set C composed of statements which, when used together, describe knowledge about the world. There may be multiple contexts describing each of the different views of the world. The meaning or truth value of a statement is a function of the context in which it is being considered.

The function $meaning : T, C \rightarrow M$, where T is the set of statements describing the world, C is the set of possible contexts, and M the set of possible meanings, $meaning(t, c)$ therefore returns the meaning or truth value of the statement t in the context c. This can be formally stated as:

$$\begin{aligned} &\forall c_i \in C \forall t_i \in T : \\ &m_i = meaning(t_i, c_i) \iff t_i \subseteq c_i \end{aligned} \tag{1}$$

The meaning or truth value of a given statement depends on the contexts in which it has been declared.

Definition 2. An Action A is causally explained from the perspective of their relation to the primary reason that rationalizes them. The function $AG : A \rightarrow G$, such that A is the actions, G is the agent, and the function AG returns the agent performing the given action. Furthermore, the function $PR : A, G \rightarrow E$ is the primary reason for an agent performing an action ton seek the effects of the event caused. Finally, the function $PA : A, O \rightarrow G$, such that O is the object, and the function returns the agent that performs the action upon the given object.

$$\begin{aligned} &\exists g \in G \exists a \in A \exists o \in O : \\ &(AG(a) \wedge PR(a, g)) \iff PA(a, o) \end{aligned} \tag{2}$$

Therefore, an action is performed upon an object, if and only if there exists an agent with a primary reason to perform the action.

Definition 3. An Event E is the individual occurrence that causes changes in the world. The criteria followed by the Davidsonian doctrine on individuation of events argues for the equality of events when the same effects occur. The Davidsonian view is here adapted to internalize the multiple contexts approach. In this paper it is therefore considered that two events are equivalent when the same effects are caused by different actions. The effects of events are captured in the *after context*, while the preconditions for an event to take place are described by the *before context*. The functions $BC : E \to C$ and $AC : E \to C$, such that $BC(e)$ and $AC(e)$ respectively return the statements of which the before and after context of a given event are composed. Furthermore, the function $effect : A, O \to S$, such that S represents the set of statements that describe the world after the event took place.

$$\forall e \in E : (BC(e) \cup effect(a, o)) \to AC(e) \tag{3}$$

Given the events e_1 and e_2, it can be said that e_1 is equivalent to e_2 when e_2 originates, at least the same effects that characterizes the *after context* of the e_1:

$$\exists e_1, e_2 \in E : e_1 = e_2 \iff AC(e_1) \subseteq AC(e_2) \tag{4}$$

Definition 3. A Service S is provided by a device D and it performs a set of actions upon an object or a set of objects. The function $PD : S \to D$, such that D is the set of available devices, and the function returns the device or devices that provide a given service.

$$\begin{gathered}\exists s \in S \exists d \in D \exists a \in A \exists o \in O : \\ (PA(a, o) \land PD(s)) \to AG(a) = d\end{gathered} \tag{5}$$

The definition of service therefore implies that the agent of an action provided by a service is a device.

Definition 4. An Object is the set O of possible environmental objects upon which actions are performed. The function $OA : A \to O$ returns the set of possible objects that can receive a given action.

$$\exists o \in O \exists a \in A \exists e \in E : OA(a) \land PA(a, o) \to e \tag{6}$$

The occurrence of an event e implies the existence of an object o upon which the action a is performed.

3 Possible worlds and multiple contexts in Scone

Automating common-sense reasoning is a task that requires a sufficiently expressive language, a knowledge base in which to store such a large amount of knowledge, and a set of mechanisms capable of manipulating this knowledge, so as to infer new information. The Scone KB project is an open-source knowledge based system, intended to represent symbolic knowledge about the world as an interconnected network made up of node units and links between them. Its principal strength lies in the way in which search and inference are implemented. Scone adopts a marker-passing algorithm[Fahlman, 2006] devised to be run in the NETL machine[Fahlman, 1979]. Despite the fact that these marker-passing algorithms cannot be compared with general theorem-provers, they are indeed faster, and most of the search and inference operations involved in common-sense reasoning are supported: inheritance of properties, roles, and relations in a multiple-inheritance type hierarchy; default reasoning with exceptions; the detection of type violations; search based on set intersection; and the maintenance of multiple, immediately overlapping world-views in the same knowledge base.

One of the main objectives with which Scone was conceived for was to emulate humans' ability to store and retrieve amounts pieces of knowledge, along with matching and adjusting existing knowledge to similar situations. To this end, the multiple-context mechanism implements an effective means to tackle this objective. The multiple-context mechanism also provides an efficient solution by which to tackle a classical problem of Artificial Intelligence, since it is frame problem.

The great potential of the multiple-context mechanism used by Scone can be better stated by using the example described in [Fahlman, 2006]. Since "Harry Potter World" is quite similar to the real world, a new context, "HPW", could be created as an instance of the real world[1]. Nevertheless, there are differences between these two contexts, such as the fact that in the "HPW" context a broom is a vehicle. This fact can be easily stated in the "HPW" without affecting real world knowledge, in the same way that knowledge of the real world could be cancelled so as to not be considered in the "HPW" context. The way in which Scone handles multiple contexts so as to avoid incongruence problems is by activating one context at a time. By doing this, only the knowledge contained in the active context is considered for the reasoning and inference task.

Unless otherwise stated, the knowledge described in a parent context is inherited by the child context. The context itself is also a node and, like the other the nodes, it stores a set of maker-bits. One of these marker-bits is the context-marker. This bit, when enabled, determines the activation of all the nodes and links that are connected to the active context.

3.1 Actions and events in Scone

Representing actions and events in Scone simply consists of defining two new contexts, one describing the world before the action or event takes place and another that represents the state of the world afterwards. The following example describes a simplified definition of the `move` event.

```
NEW-EVENT move
  :roles
     origin is a place
     destination is a place
     moving-object is a person
  :throughout
     origin differs from destination
  :before
     moving-object is located in origin
  :after
     moving-object is located in destination
```

In accordance with the aforementioned representation of the `move` event, `Lisa moves` can be defined as an individual

[1] In Scone terminology, "general" is the context node that holds knowledge about the real world, and "HPW" would be an individual node, connected by an is-a link to the "general" node.

node of the move event for the specific occurrence of Lisa moving from the kitchen to the living room.

```
NEW-EVENT-INDV Lisa moves
the origin of Lisa moves is kitchen
the destination of Lisa moves is living-room
the moving-object of Lisa moves is Lisa
IN-CONTEXT before
STATEMENT-TRUE? Lisa is in living-room
  => No
GET the location of Lisa
  => kitchen
IN-CONTEXT after
STATEMENT-TRUE? Lisa is in living-room
  => Yes
```

Note how in the before context Lisa is not yet in the living room but when the active context changes from the before context to the after context, the same question is positively answered.

4 Leveraging common sense in modeling and reasoning about actions and events

The work in [Mueller, 2006] enumerates a list of issues that should be tackled by any attempt made to automate commonsense reasoning. The following subsections analyze these issues from the viewpoint of their representation and support in performing inference and reasoning. Recall that the main focus of the proposed approach is to leverage common sense into action planning in Ambient Intelligence. Hence, the knowledge modeled has been basically restricted to aspects concerning actions and events.

4.1 Time and location

Modeling and reasoning about actions and events should be undeniably associated with a theory of time. Here, the approach proposed to model time adopts the time conceptualization of the Event Calculus[Kowalski and Sergot, 1986], augmented with the multiple-context mechanism. A *context* node can be used to capture the knowledge about the state of the world at a specific *time point* or *time interval*. Regarding space, the work in [Bhatt *et al.*, 2010] also resorts to an approach based on the Event Calculus formalism as a mean to model spatio-temporal abduction for action and change.

Considering that this work is mainly intended for action planning in Ambient Intelligence, the interest in modeling and reasoning about location is focused on providing enhanced location services. Nevertheless, the proposed approach is not exclusive to services, but can also be used to represent any aspect regarding location. Open standards have been used for interoperability purposes[2] Additionally, the work in [Bhatt, 2010] advocates the convenience of enhancing commonsensical reasoning mechanisms with qualitative representation and reasoning techniques to deal with space and location issues [Bhatt *et al.*, 2010].

4.2 Effects of events

As mentioned above, the multiple-context mechanism is the most suitable means of modeling the effects of events. In the simplest scenario, the deffinition of a new context suffices to capture the knowledge about the effects of events, or even to capture the indirect effects. Nevertheless, some other scenarios require more elaboration when describing the effects of events.

Sometimes, these effects, rather than being univocally determined by the event occurrence, are subject to the existance of certain conditions. Modeling these context-sensitive effects therefore implies considering the possible worlds that may appear as a result of the event, as determined by the given circumstances. For example, the effect of Lisa picking up an object is that of the object being held by Lisa. If we now consider the scenario of a slippery object, the effect of picking up the object does not necessarily imply that the object is being held since it might be dropped. Depending on how careful Lisa is when she picks up the object, the effect will be of the object being dropped or being held. The means of handling these sorts of effects is to define a new context for each different constraint value. Hence, in the case scenario of Lisa and the slippery object, three new context nodes hold the descriptions of the possible world. These context nodes hang from the parent after context node: one of the contexts describes the effects of picking up a slippery object without paying special attention; a second context describes the effects of picking up a slippery object while paying special attention to not dropping it; finally, the last context considers the effect of picking up a normal object.

Nevertheless, the constraints that determine the occurrence of certain effects or others cannot always be known or evaluated. For example, if the level of attention that Lisa pays to picking up the object cannot be assessed, there is no way of foreseeing whether the object will or will not be dropped. The non-determinism of those scenarios creates uncertainty which must also be captured in the action description.

The occurrence of concurrent events also requires a special treatment when coincident events involve cumulative, impossible or cancelling effects. For example, it is not possible to enter two different locations at the same time or, if a door is pulled and pushed at the same time, it remains static.

4.3 Common-sense law of inertia

The "*frame problem*" has been addressed here by means of the multiple context mechanism. Note that the after context is a copy of the before context which captures those aspects of the world that change as a result of the event occurrence. This property, which makes things continue in the same state, is known as the common-sense law of inertia.

The difficulty involved in dealing with the common-sense law of inertia is that of having to capture and model the knowledge concerning delayed effects or continuous change. As stated above, a delayed effects occurs if the kitchen sink has its plug in and someone turns on the tap: after a while the water will overflow. The common-sense law of inertia is also involved with regard to the water level since it keeps on increasing unless the tap is turned off. Nevertheless, the level does not increase endlessly but rather increases until it reaches the height of the kitchen sink. Afterwards, the water overflows until the water level equals the height of kitchen

[2] Open Geospatial Consortium (OGC). OpenGIS Location Services (OpenLS): Core Services. http://www.opengeospatial.org/standards/ols.

sink.

The event calculus notion of *fluent* is here adopted to deal with these properties that change over time, such as the water level in the open tap example. At each time instant the world must be modeled to capture the value of the changing property.

```
NEW-EVENT turn-on faucet
   :roles
      faucet-liquid is a liquid
      faucet-drain is a drain
      faucet-valve is a valve
      level-of-faucet-drain is a FLUENT
    :before
      current-time is T0
      faucet-valve is turned-off
      level-of-faucet-drain is empty
    :after
      faucet-valve is turned-on
      IN-CONTEXT time-instant T1
      level-of-faucet-drain equals (flow * (
          elapsed-time / base-area))
      IN-CONTEXT time-instant T2
      level-of-faucet-drain equals full
      faucet-liquid is dropped-off
```

4.4 Default reasoning and mental states

Default reasoning alludes to the fact that common-sense reasoning is usually performed in uncertainty. For example, the result of turning the fans power switch to on will be that the fan will start spinning around. However, what if the fan is not plugged or it is not working? Most of the time there is no complete information about all these details, so performing default reasoning with exceptions is the most appropriate way in which to handle incompleteness.

In Scone, default reasoning with exceptions is handled by means of *cancel-links*. Please, refer to the work in [Fahlman, 2006] for further information on this subject.

Reasoning about mental states has also been previously addressed. The work in [Chen and Fahlman, 2008] proposes an approach based on " *mental context*" so as to model mental states and their interactions.

5 Action planning with common sense

As has already been mentioned above, the main difficulty faced by systems for Ambient Intelligence lies in coping with innovation. Surveillance contexts typically provide an ideal scenario for unforeseen situations to take place. Furthermore, in most cases, the system will be prompted to elaborate a response in order to manage the unexpected event. A simulated intrusion in a surveyed building poses an interesting scenario in which to asses the performance, regarding action planning, of the proposed model.

First, the presence sensor installed in the servers' room detects an intruder break-in. The guards are automatically notified with the sensor detection. One of the system's goals under these circumstances is to identify and to locate the intruder.

```
IN-CONTEXT intruder-intention
GET the intention of intruder
  => Not known
IN-CONTEXT intruder-break-in
GET the location of intruder
   => servers-room
STATEMENT-TRUE? guards are notified of
    intruder-location
   => Yes
GET the identification of intruder
   => Not known
STATEMENT-TRUE? intrusion alarm status is on
   => Yes
```

The sound of the alarm makes the intruder aware that his presence has been detected. He therefore decides to run away. Meanwhile, the guards are in their way to the servers' room.

```
NEW-EVENT-INDV intruder-leaves-room
intruder is the agent
server-room is the object
IN-CONTEXT intruder intention
GET the intention of intruder
  => Too many
```

After the intruder leaves the room, his location is no longer the servers' room. On the contrary, the intruder is moving through the building in an attempt to escape without being caught. This state of affairs leads to the need for a plan to pursue the goal of locating the intruder. The location of a person is one of those properties that may need to be released from the common-sense law of inertia while the person is moving. Bearing this in mind, the trajectory of a person in movement can be inferred from the successive locations at three consecutive moments in time.

```
SET-FLUENT intruder is located in loc0 at t0
SET-FLUENT intruder is located in loc1 at t1
SET-FLUENT intruder is located in loc2 at t2
STATEMENT-HOLDS? intruder is moving
   => Yes
GET-FLUENT intruder location at t3
   => (covered-distance / elapsed-time) * t3
```

Now, at time instant t3, let us say that the intruder's presence cannot be distinguished at the expected location loc3. So what has happened? Well, in between loc2 and loc3 there is a room. What makes a person abandon the moving trajectory followed?

```
RELEASE-FLUENT intruder location
  => location fluent released
LIST-EVENTS-CAUSING moving-object abandons
    trajectory-of-move
  => enter, stop, sit, jump, lay down, ...
```

Given the plausible events, the system then becomes engaged in proving which of the actions has certainly taken place. The means of verifying this is to check whether the current context is consistent with any of the after context of the plausible actions.

```
LIST-AFTER-CONTEXT stop
   => 1. RELEASE-FLUENT moving-object from
         location
      2. the location of moving-object is
         current-location
STATEMENT-TRUE? the location of intruder is
    loc3   => Not known
LIST-EVENTS-REQUIRING the location of thing
    is place => capture, sense, notice, ...
GET service performing capture
```

```
   => video-recording, face detector,
       fingerprint reader, etc.
NEW-INDV shp3 is shape
   the center-of-shape sph3 is loc3
GET video-recording in shape shp3
   => videoRec-at-shp3
LIST-EVENTS-PRECEDING recording upon person
   => focus-person
LIST-EVENTS-PRECEDING focus-person
   => detecting-face, detecting-smile,
       detecting-temperature, etc.
GET service performing detecting-face
   => face-detector
GET face-detector in shape shp3
   => faceDet_at_server
STATEMENT-TRUE? the location of moving-object
     is shp3  => No
```

Each of the possible events causing the intruder to abandon the trajectory will be evaluated recursively[3].

The `Planning` algorithm proposed in [Santofimia *et al.*, 2010] starts with an empty plan, the Π plan, to be completed with the list of actions, provided by services. This course of actions is intended to emulate the demanded non-feasible action. The course of actions is provided as a set of actions performed on objects, A and O respectively, and the results R of accomplishing such actions. The function *resultOf* refers to the returned value obtained as result of instantiating the a_i action.

6 Conclusions and future works

This work is founded on the conviction that systems for Ambient Intelligence should consider common sense as a constituent element. This work uses action planning, enhanced with common-sense knowledge about actions and events, as the cornerstone of the decision making process.

The main contribution of this work is threefold. First, a model for actions and events in Ambient Intelligence is proposed to characterize the Ambient Intelligence domain knowledge. Second, the model is represented and enhanced to consider the key issues of common-sense reasoning. Third, the proposed strategy for action planning is grounded in multiple-context and possible-world semantics.

This work is an improvement on existing approaches for planning in Ambient Intelligence when devising ad-hoc tailored solutions, on the basis of the available devices and services. Common-sense knowledge is considered throughout the planning, so rather than constraining the planning solution to context knowledge (explicit knowledge), implicit knowledge leads to more appropriate solutions. In the aforementioned case scenario, please note how the trajectory of the intruder has been devised. Also note how the common-sense law of inertia has been used to infer that if the person is not where he was supposed to be, he must have been affected by a particular event. It has been demonstrated above that the `stop` event is not considered possible, since the current state of the world does not match the `after context` of the `stop` action.

References

[Allen, 1984] James F. Allen. Towards a general theory of action and time. *Artif. Intell.*, 23:123–154, July 1984.

[Bhatt *et al.*, 2010] Mehul Bhatt, Hans Guesgen, and Shyamanta Hazarika, editors. *Spatio-Temporal Dynamics (STeDy 10)*. ECAI Workshop Proceedings., and SFB/TR 8 Spatial Cognition Report Series, August 2010.

[Bhatt, 2010] Mehul Bhatt. Reasoning about space, actions and change: A paradigm for applications of spatial reasoning. In *Qualitative Spatial Representation and Reasoning: Trends and Future Directions*. IGI Global, USA, 2010.

[Chen and Fahlman, 2008] Wei Chen and Scott E. Fahlman. Modeling mental contexts and their interactions. In *AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures, Washington*. 2008.

[Davidson, 1963] Donald Davidson. Actions, reasons, and causes. *The Journal of Philosophy*, 60(23):685–700, 1963.

[Erol *et al.*, 1994] Kutluhan Erol, James Hendler, and Dana S. Nau. HTN planning: Complexity and expressivity. In *In AAAI-94*, 1994.

[Fahlman, 1979] Scott E. Fahlman. *NETL: A System for Representing and Using Real-World Knowledge*. MIT Press, Cambridge, MA, 1979.

[Fahlman, 2006] Scott E. Fahlman. Marker-passing inference in the scone knowledge-base system. In *First International Conference on Knowledge Science, Engineering and Management (KSEM'06)*. Springer-Verlag (Lecture Notes in AI), 2006.

[Georgeff, 1988] Michael P. Georgeff. A theory of action for multiagent planning. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 205–209. Kaufmann, San Mateo, CA, 1988.

[Hommel *et al.*, 2001] Bernhard Hommel, Jochen Musseler, Gisa Aschersleben, and Wolfgang Prinz. The theory of event coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24:849–878, 2001.

[Kowalski and Sergot, 1986] Robert A. Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.

[Lenat, 1995] Douglas Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38:33–38, 1995.

[Mueller, 2006] Erik T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, 2006.

[Santofimia *et al.*, 2010] Maria J. Santofimia, Scott E. Fahlman, Francisco Moya, and Juan Carlos López. A common-sense planning strategy for ambient intelligence. In Rossitza Setchi, Ivan Jordanov, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES (2)*, volume 6277 of *Lecture Notes in Computer Science*, pages 193–202. Springer, 2010.

[3] http://sites.google.com/site/csrijcai11/

Proceedings

Space, Time and Ambient Intelligence (STAMI 2011). Editors: Mehul Bhatt, Hans Guesgen, and Juan Carlos Augusto. Workshop Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Spain, 2011.

Also published online as part of Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, Universität Bremen / Universität Freiburg. SFB/TR 8 Reports, Bremen, Germany.